

# **HOPEX IT Architecture**

## **Guide d'utilisation**

HOPEX Aquila 6.1

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière constituer un engagement de la société MEGA International.

Aucune partie de la présente publication ne peut être reproduite, enregistrée, traduite ou transmise, sous quelque forme et par quelque moyen que ce soit, sans un accord préalable écrit de MEGA International.

© MEGA International, Paris, 1996 - 2025

Tous droits réservés.

HOPEX IT Architecture et HOPEX sont des marques réservées de MEGA International.

Windows est une marque réservée de Microsoft.

Les autres marques citées appartiennent à leurs propriétaires respectifs.

# SOMMAIRE



---

<b>Sommaire</b> . . . . .	<b>3</b>
---------------------------	----------

---

<b>Introduction</b> . . . . .	<b>17</b>
-------------------------------	-----------

---

<b>Présentation de HOPEX IT Architecture</b> . . . . .	<b>18</b>
Le périmètre couvert par HOPEX IT Architecture . . . . .	18
Synthèse des activités et livrables de HOPEX IT Architecture . . . . .	19
Structure et positionnement de la solution HOPEX IT Architecture . . . . .	19
Les profils de HOPEX IT Architecture . . . . .	20
Les rôles métier de HOPEX IT Architecture . . . . .	22
<b>La méthode de HOPEX IT Architecture</b> . . . . .	<b>23</b>
Décrire l'architecture applicative . . . . .	23
<i>Décrire un environnement de système applicatif</i> . . . . .	23
<i>Décrire les systèmes applicatifs</i> . . . . .	25
Décrire les applications . . . . .	26
<i>Décrire les scénario de flux</i> . . . . .	26
<i>Décrire la structure de l'application et ses services</i> . . . . .	27
Définir l'architecture de déploiement d'une application . . . . .	28
Construire l'architecture logique . . . . .	29
<i>Diagramme de structure de système applicatif logique</i> . . . . .	31
<i>Diagramme d'environnement de système applicatif logique</i> . . . . .	32
Analyser la couverture fonctionnelle de l'architecture mise en œuvre . . . . .	33
<i>Identifier les capacités métier</i> . . . . .	33
<i>Identifier les capacités technologiques associées aux capacités métier</i> . . . . .	34
<i>Identifier les applications associées aux fonctionnalités</i> . . . . .	35
Définir l'infrastructure technique . . . . .	36
<i>Diagramme d'environnement d'architecture de ressources</i> . . . . .	36
<i>Décrire une architecture de ressources</i> . . . . .	36
<i>Diagramme de structure d'assemblage d'infrastructure informatique</i> . . . . .	37
<i>Diagramme d'assemblage d'ordinateurs</i> . . . . .	39
Concevoir et spécifier les applications . . . . .	39
<i>Utiliser le formalisme UML</i> . . . . .	39
<i>Décrire les traitements par lot</i> . . . . .	40

<i>Décrire la liste des services et des IHM</i> . . . . .	40
<i>Décrire les processus applicatifs</i> . . . . .	41
Gérer les catalogues de service . . . . .	41
<b>Présentation du bureau HOPEX IT Architecture</b> . . . . .	<b>43</b>
Se connecter à la solution . . . . .	43
Présentation du bureau HOPEX IT Architecture . . . . .	43
<i>Présentation de l'espace de l'Architecte solution</i> . . . . .	44
<i>Présentation des menus de l'espace de l'Administrateur fonctionnel d'architecture de solution</i>	51
<i>Présentation de l'espace du concepteur d'applications</i> . . . . .	51
<i>Présentation de l'espace de l'observateur d'applications</i> . . . . .	51
Changer de profil . . . . .	52
<b>Avant de commencer avec HOPEX IT Architecture</b> . . . . .	<b>53</b>
Définir l'environnement de travail . . . . .	53
<i>Accéder à la liste des bibliothèques avec HOPEX IT Architecture</i> . . . . .	53
<i>Accéder à la liste des entreprises avec HOPEX IT Architecture</i> . . . . .	53
Utiliser les acteurs . . . . .	54
<i>Créer un acteur</i> . . . . .	54
<i>Acteur interne / externe</i> . . . . .	54
Utiliser les diagrammes d'architecture informatique . . . . .	54
<i>Créer un diagramme de structure</i> . . . . .	55
<i>Les commandes des diagrammes de HOPEX IT Architecture</i> . . . . .	56
<i>Positionnement automatique dans les diagrammes d'architecture</i> . . . . .	56
<i>Initialisation d'un diagramme d'environnement</i> . . . . .	57
<i>Créer un diagramme d'ébauche avec HOPEX IT Architecture</i> . . . . .	58
<i>Créer un diagramme ArchiMate® avec HOPEX IT Architecture</i> . . . . .	59
<i>Utiliser la comparaison de diagramme</i> . . . . .	59
Contenu d'une page de propriétés HOPEX IT Architecture . . . . .	59
Utiliser la duplication avec HOPEX IT Architecture . . . . .	61
<i>Utiliser la duplication de HOPEX IT Architecture en batch</i> . . . . .	62
Utiliser les catalogues de services . . . . .	63
<i>La mise œuvre d'un catalogue de service</i> . . . . .	63
<i>Renseigner un catalogue de services</i> . . . . .	64
<i>Créer un catalogue de services technologiques</i> . . . . .	64
<i>Ajouter un élément de catalogue de services</i> . . . . .	65
<i>Spécifier la mise en œuvre d'un élément de catalogue de services</i> . . . . .	65
<i>Les rapports de catalogues de service</i> . . . . .	65
Utiliser les workflows . . . . .	66
Définir un cadre de politique d'entreprise avec HOPEX IT Architecture . . . . .	66
<i>Définir une politique d'entreprise avec HOPEX IT Architecture</i> . . . . .	67
<i>Définir les principes d'architecture</i> . . . . .	68
Définir les catégories de données . . . . .	69
Définir les domaines méthodologiques . . . . .	69
Importer des composants HOPEX IT Architecture . . . . .	70
<i>Structure des modèles Excel d'import/export de HOPEX IT Architecture</i> . . . . .	70
<i>Importer des ordinateurs ou des technologies avec Excel</i> . . . . .	71
Utiliser les outils de conversion vers HOPEX Aquila . . . . .	73
<b>A propos de ce guide</b> . . . . .	<b>75</b>
Structure du guide . . . . .	75
Ressources complémentaires . . . . .	75
Conventions utilisées dans le guide . . . . .	76



---

**Modéliser les applications et les architectures système ..... 79**

<b>Introduction des concepts de HOPEX IT Architecture .....</b>	<b>80</b>
<i>Application .....</i>	<i>80</i>
<i>Système applicatif .....</i>	<i>80</i>
<b>Décrire une application avec HOPEX IT Architecture .....</b>	<b>82</b>
Créer une application avec HOPEX IT Architecture .....	82
Les propriétés d'une application avec HOPEX IT Architecture .....	83
Définir le périmètre fonctionnel de l'application .....	84
Décrire la structure et les services d'une application .....	85
Décrire un environnement d'application avec HOPEX IT Architecture .....	86
<i>Décrire un environnement d'application .....</i>	<i>86</i>
<i>Accéder à la liste des environnements d'application .....</i>	<i>86</i>
<i>Créer un environnement d'application .....</i>	<i>86</i>
<i>Les propriétés d'un environnement d'application .....</i>	<i>87</i>
<i>Présentation d'un diagramme d'environnement d'application .....</i>	<i>87</i>
Spécifier les risques associés à une application .....	88
<b>Décrire l'architecture système .....</b>	<b>89</b>
Décrire un système applicatif .....	89
<i>Créer un système applicatif .....</i>	<i>89</i>
<i>Les propriétés d'un système applicatif .....</i>	<i>90</i>
<i>Créer un diagramme de structure de système applicatif .....</i>	<i>91</i>
<i>Utiliser un scénario de flux de système applicatif .....</i>	<i>93</i>
Décrire un environnement de système applicatif avec HOPEX IT Architecture .....	95
<i>Accéder à la liste des environnements d'un système applicatif .....</i>	<i>95</i>
<i>Créer un environnement de système applicatif .....</i>	<i>95</i>
<i>Les propriétés d'un environnement de système applicatif .....</i>	<i>95</i>
<i>Les diagrammes d'environnement de système applicatif .....</i>	<i>96</i>

---

**Modéliser les architectures applicatives ..... 99**

<b>Décrire les flux de données .....</b>	<b>100</b>
Utiliser un diagramme de scénario de flux d'application .....	100
<i>Créer un diagramme de scénario de flux d'application .....</i>	<i>101</i>
<i>Ajouter un service applicatif au scénario de flux d'application .....</i>	<i>101</i>
<i>Gérer les flux applicatifs dans un scénario de flux d'application .....</i>	<i>102</i>
<i>Créer un événement déclencheur système .....</i>	<i>104</i>
<i>Ajouter un dépôt de données applicatives au scénario de flux d'application .....</i>	<i>104</i>
<i>Créer un canal de données applicatives .....</i>	<i>105</i>
Utiliser les systèmes de communication .....	105
<i>Accéder à la liste des systèmes de communication .....</i>	<i>106</i>
<i>Les propriétés d'un système de communication .....</i>	<i>106</i>
<i>Créer un diagramme de chaîne de communication applicative .....</i>	<i>106</i>
<i>Relier une chaîne de communication applicative à partir d'un scénario de flux .....</i>	<i>108</i>
Utiliser un diagramme de scénario de séquence de flux .....	108
<i>Créer un diagramme de scénario de séquence de flux .....</i>	<i>109</i>
<i>Instances d'application, de service applicatif ou d'IHM .....</i>	<i>110</i>
<i>Instances de messages .....</i>	<i>110</i>

<b>Décrire la structure et les services d'une application . . . . .</b>	<b>111</b>
Le diagramme de structure d'application . . . . .	111
<i>Créer un diagramme de structure d'application . . . . .</i>	<i>111</i>
<i>Les composants d'un diagramme de structure d'application . . . . .</i>	<i>112</i>
<i>Ajouter un service applicatif dans un diagramme de structure d'application . . . . .</i>	<i>112</i>
Décrire un service applicatif avec HOPEX IT Architecture . . . . .	113
<i>Les diagrammes d'un service applicatif . . . . .</i>	<i>113</i>
<i>Accéder à la liste des services applicatifs . . . . .</i>	<i>113</i>
<i>Les propriétés d'un service applicatif . . . . .</i>	<i>113</i>
<i>Utiliser le diagramme de structure de service applicatif . . . . .</i>	<i>114</i>
Décrire un microservice avec HOPEX IT Architecture . . . . .	115
<i>Les diagrammes d'un microservice . . . . .</i>	<i>115</i>
<i>Accéder à la liste des microservices . . . . .</i>	<i>115</i>
<i>Les propriétés d'un microservice avec HOPEX IT Architecture . . . . .</i>	<i>116</i>
<i>Utiliser le diagramme de structure de microservice . . . . .</i>	<i>117</i>
Créer un diagramme de cas d'utilisation d'une application . . . . .	117
<b>Décrire les processus applicatifs . . . . .</b>	<b>119</b>
Gérer les processus applicatifs avec HOPEX IT Architecture . . . . .	120
<i>Accéder aux processus applicatifs . . . . .</i>	<i>120</i>
<i>Créer un diagramme de processus applicatif . . . . .</i>	<i>120</i>
Spécifier le comportement d'une tâche d'un processus applicatif . . . . .	122
<i>Les comportements . . . . .</i>	<i>123</i>
<i>Type de tâche . . . . .</i>	<i>123</i>
Modéliser les tâches d'un processus applicatif . . . . .	124
<i>Exemple de modélisation fonctionnelle . . . . .</i>	<i>124</i>
<i>Afficher le diagramme d'une étape de processus applicatif . . . . .</i>	<i>124</i>
Modéliser les tâches d'un service applicatif . . . . .	125
<b>Gérer les données . . . . .</b>	<b>127</b>
Utiliser les dépôts de données . . . . .	127
<i>Introduction au concept de dépôt de données . . . . .</i>	<i>127</i>
<i>Les contextes d'utilisation . . . . .</i>	<i>128</i>
<i>Créer un dépôt de données local . . . . .</i>	<i>129</i>
<i>Créer un dépôt de données externe . . . . .</i>	<i>129</i>
<i>Décrire l'accès à un dépôt de données . . . . .</i>	<i>129</i>
Accéder aux supports des dépôts de données . . . . .	130
<i>Accéder aux domaines de données avec HOPEX IT Architecture . . . . .</i>	<i>130</i>
<i>Accéder à la liste des structures de fichier avec HOPEX IT Architecture . . . . .</i>	<i>130</i>
<i>Accéder aux domaines physiques (NoSQL) avec HOPEX IT Architecture . . . . .</i>	<i>131</i>
<i>Accéder à la liste des schémas relationnels avec HOPEX IT Architecture . . . . .</i>	<i>131</i>
 <b>Modéliser les architectures de déploiement . . . . .</b>	 <b>133</b>
 <b>Décrire une architecture de déploiement d'une application . . . . .</b>	 <b>134</b>
Accéder à la liste des architectures de déploiement d'une application . . . . .	134
Décrire une architecture de déploiement d'une application et son diagramme . . . . .	134
<i>Créer une architecture de déploiement d'une application . . . . .</i>	<i>136</i>
Utiliser un diagramme d'architecture de déploiement d'une application . . . . .	136
<i>Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application . . . . .</i>	<i>136</i>

Ajouter des ports techniques. . . . .	137
Décrire les connexions de module . . . . .	138
Décrire un module de déploiement d'une application . . . . .	138
<b>Décrire un environnement de déploiement d'une application . . . . .</b>	<b>140</b>
Accéder à la liste des environnements de déploiement d'une application. . . . .	140
Décrire un environnement de déploiement d'une application . . . . .	140
Créer un environnement de déploiement d'une application . . . . .	141
Utiliser un diagramme d'environnement de déploiement d'une application . . . . .	141
<b>Décrire une architecture de déploiement d'un système applicatif. . . . .</b>	<b>143</b>
Accéder à la liste des architectures de déploiement d'un système applicatif. . . . .	143
Décrire une architecture de déploiement d'un système applicatif . . . . .	143
Les propriétés d'une architecture de déploiement d'un système applicatif . . . . .	145
<b>Les modèles d'architecture de déploiement. . . . .</b>	<b>147</b>
Accéder à la liste des modèles d'architecture de déploiement . . . . .	147
Décrire un modèle de déploiement d'application . . . . .	147
Les composants d'un modèle de déploiement d'application . . . . .	147
Créer un modèle de déploiement d'une application . . . . .	148
Présentation des modèles d'architecture de déploiement standard . . . . .	149
Le modèle de déploiement d'application "Architecture 3 tiers (RDBMS)". . . . .	149
Le modèle de déploiement d'application "Architecture d'application mobile" . . . . .	149
Le modèle de déploiement d'application "Architecture d'application Internet" . . . . .	150
Utiliser un modèle de déploiement d'application . . . . .	150
<b>Décrire les technologies logicielles . . . . .</b>	<b>152</b>
Décrire une technologie logicielle . . . . .	152
Accéder à la liste des technologies logicielles . . . . .	152
Les propriétés d'une technologie logicielle. . . . .	152
Décrire une pile de technologies. . . . .	153
Accéder à la liste des piles de technologie. . . . .	153
Les propriétés d'une pile de technologies logicielles . . . . .	153
<b>Utiliser les services cloud . . . . .</b>	<b>154</b>
Accéder à la liste des services cloud. . . . .	154
Les propriétés d'un service Cloud. . . . .	155

---

## **Aligner l'informatique et les métiers . . . . . 157**

<b>Décrire l'architecture logicielle logique . . . . .</b>	<b>158</b>
Décrire un système applicatif logique avec HOPEX IT Architecture . . . . .	158
Accéder à la liste des systèmes applicatifs logiques avec HOPEX IT Architecture . . . . .	158
Créer un système applicatif logique . . . . .	158
Les propriétés d'un système applicatif logique. . . . .	159
Décrire la structure d'un système applicatif logique . . . . .	159
Décrire les applications logiques avec HOPEX IT Architecture. . . . .	162
Accéder à la liste des applications logiques avec HOPEX IT Architecture . . . . .	162
Créer une application logique . . . . .	162
Les propriétés d'une application logique . . . . .	162
Décrire un environnement de système applicatif logique . . . . .	163
Exemple d'environnement de système applicatif logique. . . . .	163
Accéder à la liste des environnements d'un système applicatif logique . . . . .	164
Créer un environnement de système applicatif logique. . . . .	164

<i>Les propriétés d'un environnement de système applicatif logique . . . . .</i>	165
<i>Utiliser le diagramme d'environnement de système applicatif logique . . . . .</i>	165
<b>Décrire les capacités métier avec HOPEX IT Architecture . . . . .</b>	<b>166</b>
Exemples de capacités métier avec HOPEX IT Architecture. . . . .	166
Utiliser les cartes de capacités métier avec HOPEX IT Architecture . . . . .	167
<i>Accéder à la liste des cartes de capacité métier . . . . .</i>	167
<i>Créer une carte de capacités métier . . . . .</i>	168
<i>Les propriétés d'une carte de capacités métier. . . . .</i>	168
<i>Créer un diagramme de carte de capacités métier . . . . .</i>	168
Utiliser les capacités métier avec HOPEX IT Architecture . . . . .	169
<i>Accéder à la liste des capacités métier avec HOPEX IT Architecture . . . . .</i>	169
<i>Créer une capacités métier . . . . .</i>	169
<i>Décrire une capacité métier. . . . .</i>	170
<i>Définir les fonctionnalités associées à une capacité métier. . . . .</i>	170
<b>Utiliser les fonctionnalités avec HOPEX IT Architecture. . . . .</b>	<b>172</b>
Décrire une carte de fonctionnalités avec HOPEX IT Architecture . . . . .	172
<i>Accéder à la liste des cartes de fonctionnalités avec HOPEX IT Architecture. . . . .</i>	173
<i>Créer une carte de fonctionnalités . . . . .</i>	173
<i>Créer un diagramme de carte de fonctionnalités. . . . .</i>	173
<i>Les propriétés d'une carte de fonctionnalités . . . . .</i>	173
Décrire les fonctionnalités avec HOPEX IT Architecture . . . . .	174
<i>Créer un diagramme de fonctionnalités avec HOPEX IT Architecture. . . . .</i>	174
Décrire une carte de capacités technologiques avec HOPEX IT Architecture . . . . .	175
<i>Accéder à la liste des cartes de capacités technologiques avec HOPEX IT Architecture . . . . .</i>	175
<i>Décrire une capacité technologique. . . . .</i>	175
<i>Décrire une capacité matérielle . . . . .</i>	175
<b>Utiliser les mécanismes de mise en œuvre. . . . .</b>	<b>176</b>
Décrire la mise en œuvre d'une capacités métier . . . . .	176
<i>Créer une mise en oeuvre de capacité métier . . . . .</i>	176
<i>Analyser la mise en oeuvre des capacités de l'entreprise. . . . .</i>	177
Décrire la mise en œuvre d'une fonctionnalité . . . . .	177
<i>Créer une mise en oeuvre de fonctionnalité. . . . .</i>	177
<i>Identifier les applications associées aux fonctionnalités. . . . .</i>	178
Accéder aux mises en œuvre à partir d'un point de service . . . . .	178
 <b>Modéliser les infrastructures informatiques . . . . .</b>	 <b>179</b>
<b>Décrire les architectures de ressources . . . . .</b>	<b>180</b>
Décrire une architecture de ressources . . . . .	180
<i>Créer un diagramme d'assemblage de ressources . . . . .</i>	180
<i>Utiliser un diagramme d'assemblage de ressources . . . . .</i>	181
Décrire un environnement d'architecture de ressources. . . . .	183
<i>Créer un environnement d'architecture de ressources. . . . .</i>	184
<i>Les propriétés d'un environnement d'architecture de ressources . . . . .</i>	184
<i>Créer un diagramme d'environnement d'architecture de ressources . . . . .</i>	184
<i>Décrire un diagramme d'environnement d'architecture de ressources. . . . .</i>	184
Décrire une configuration de ressource . . . . .	186
<i>Créer une configuration de ressource . . . . .</i>	186
<i>Créer un diagramme de configuration de ressource . . . . .</i>	186

<i>Utiliser un diagramme de configuration de ressource</i> . . . . .	187
Décrire un matériel . . . . .	188
<i>Créer un matériel</i> . . . . .	188
<i>Créer un diagramme de structure d'éléments matériels</i> . . . . .	188
<i>Utiliser un diagramme de structure d'élément matériel.</i> . . . . .	188
<b>Décrire les infrastructures informatiques</b> . . . . .	<b>190</b>
Décrire une infrastructure informatique . . . . .	190
<i>Créer une infrastructure informatique</i> . . . . .	190
<i>Créer un diagramme de structure d'assemblage d'infrastructure</i> . . . . .	190
<i>Utiliser un diagramme de structure d'assemblage d'infrastructure</i> . . . . .	190
Décrire un réseau informatique . . . . .	191
<i>Créer un réseau informatique</i> . . . . .	192
<i>Créer un diagramme de réseau informatique.</i> . . . . .	192
Décrire une installation . . . . .	192
<i>Créer une installation.</i> . . . . .	192
<i>Créer un diagramme de configuration de ressources pour une installation</i> . . . . .	192
<b>Décrire les ordinateurs</b> . . . . .	<b>194</b>
Décrire un ordinateur . . . . .	194
<i>Accéder à la liste des ordinateurs</i> . . . . .	194
<i>Créer un terminal informatique</i> . . . . .	194
<i>Créer un diagramme d'assemblage d'ordinateurs.</i> . . . . .	195
Décrire un périphérique réseau . . . . .	196
<i>Accéder à la liste des périphériques réseau.</i> . . . . .	196
<i>Créer un périphérique réseau</i> . . . . .	196
<b>Décrire les communications dans une infrastructure informatique</b> . . . . .	<b>197</b>
Décrire les communications liées aux services . . . . .	197
<i>Les interactions de service</i> . . . . .	197
<i>Les points de service</i> . . . . .	198
<i>Les points de requête.</i> . . . . .	198
Décrire les communications techniques . . . . .	199
<i>Les ports de communication</i> . . . . .	199
<i>Les canaux réseau</i> . . . . .	199
<i>Les protocoles de communication réseau</i> . . . . .	199
Relier une interaction de service à un canal réseau . . . . .	200
<hr/>	
<b>Accéder à la conception de logicielle</b> . . . . .	<b>201</b>
<b>Modélisation UML des données.</b> . . . . .	<b>202</b>
Paquetage (UML) . . . . .	202
Modèles de données . . . . .	203
Les domaines de données . . . . .	204
<b>Décrire les traitements par lots</b> . . . . .	<b>206</b>
Définition d'un traitement par lots . . . . .	206
Construire un diagramme de structure de chaîne de traitement de lot . . . . .	206
<i>Créer un diagramme de structure de chaîne de traitement de lot.</i> . . . . .	207
<i>Ajouter un appel de traitement de batch dans le diagramme.</i> . . . . .	207
<i>Définir les séquences des batchs</i> . . . . .	208
Créer un diagramme de structure de programme batch . . . . .	208
<i>Créer un diagramme de structure de programme batch</i> . . . . .	208

Ajouter un appel de programme dans le diagramme . . . . .	208
Utiliser les réalisations batch de processus applicatif . . . . .	209
<b>Décrire les interfaces utilisateur . . . . .</b>	<b>210</b>
Créer une interface homme-machine . . . . .	210
Construire un diagramme d'Interface Homme Machine . . . . .	210
Dessiner l'IHM . . . . .	211
<i>Elément D'IHM . . . . .</i>	<i>211</i>
<i>Evénement d'IHM . . . . .</i>	<i>212</i>
<hr/>	
<b>Décrire les échanges d'informations . . . . .</b>	<b>215</b>
<b>Gérer les interactions de service . . . . .</b>	<b>216</b>
Créer une interaction de service . . . . .	217
Décrire les points de service et de requête . . . . .	217
<i>Les points de service . . . . .</i>	<i>217</i>
<i>Les points de requête . . . . .</i>	<i>218</i>
<i>Créer un point de service ou un point de requête . . . . .</i>	<i>219</i>
<b>Décrire une interface de service . . . . .</b>	<b>220</b>
Exemples de diagrammes d'interface de service (BPMN) . . . . .	220
<i>Exemple de diagramme d'interface de service (BPMN) . . . . .</i>	<i>221</i>
<i>Exemple d'interface de service de communication avancé . . . . .</i>	<i>222</i>
Accéder à la liste des interfaces de service . . . . .	222
Créer une interface de service . . . . .	223
<i>Créer une interface de service en mode standard à partir d'un diagramme . . . . .</i>	<i>223</i>
Construire un diagramme d'interface de service (BPMN) . . . . .	223
<i>Créer un diagramme d'interface de service (BPMN) . . . . .</i>	<i>223</i>
<i>Définir un opération de service ou une interface de service utilisée . . . . .</i>	<i>224</i>
<b>Décrire les opérations de service . . . . .</b>	<b>225</b>
Accéder à la liste des opérations de service . . . . .	226
Créer une opération de service . . . . .	226
Décrire une opération de service . . . . .	226
<i>Créer un diagramme d'opération de service (BPMN) . . . . .</i>	<i>226</i>
<i>Créer un flux avec contenu . . . . .</i>	<i>227</i>
<i>Gérer les événements, les branchements et les enchaînements . . . . .</i>	<i>227</i>
<b>Utiliser un modèle d'interface de service . . . . .</b>	<b>228</b>
Présentation des modèles d'interface de service livrés en standard . . . . .	228
<i>Le modèle d'interface de service "Communication simple" . . . . .</i>	<i>228</i>
<i>Le modèle d'interface de service "Question - Réponse" . . . . .</i>	<i>229</i>
<i>Le modèle d'interface de service "Souscription - Publication" . . . . .</i>	<i>230</i>
Accéder à la liste des modèles d'interface de service . . . . .	230
Créer une interface de service à partir d'un modèle d'interface de service . . . . .	230
Créer un modèle d'interface de service . . . . .	231
Créer un modèle d'opération de service . . . . .	232

---

**Les rapports HOPEX IT Architecture . . . . . 233**
**Rapports sur les architectures applicatives . . . . . 234**

Matrice d'architecture technique . . . . .	234
Densité des échanges entre applications . . . . .	235
Cohérence des échanges (Structure / Scénarios) . . . . .	236
Cohérence du contenu (Structure) . . . . .	236
Cohérence du contenu (Scénario) . . . . .	237
Matrice des contenus externes (Structure) . . . . .	238
Matrice des contenus externes (Scénario) . . . . .	239
Matrice des interfaces de service externes . . . . .	240
Graphe des flux entre des agents . . . . .	241
Graphe des flux d'un agent . . . . .	242
Rationalisation des flux traités . . . . .	244
Graphe des interactions de service entre des agents . . . . .	244
Graphe des interactions de service d'un agent . . . . .	245

**Rapports sur la couverture fonctionnelle de l'architecture . . . . . 247**

Rapport de décomposition des briques d'entreprise . . . . .	247
Chevauchement d'applications . . . . .	250
Rapport de décomposition de capacité métier . . . . .	251

**Rapports sur les infrastructures. . . . . 254**

Rapport de description d'infrastructure . . . . .	254
Matrice des exigences technologiques applicatives x technologies de l'infrastructure informatique . . . . .	255
Canaux réseau x Interactions de service . . . . .	256
Canaux réseau x connexions de module . . . . .	256

**Rapports sur les architectures de déploiement . . . . . 258**

Rapport d'architecture de déploiement . . . . .	258
Matrice d'architecture de déploiement . . . . .	258
Matrice des connexions de module x Interactions de service . . . . .	259
Matrice des connexions de module x Flux applicatifs. . . . .	260

---

**A propos de l'implémentation UML. . . . . 265**
**Présentation . . . . . 266**

<i>Analyser les cas d'utilisation . . . . .</i>	266
<i>Identifier les objets . . . . .</i>	266
<i>Décrire les comportements . . . . .</i>	266
<i>Représenter les interactions entre les objets . . . . .</i>	266
<i>Répartir les classes entre les paquetages . . . . .</i>	267
<i>Définir les interfaces . . . . .</i>	267
<i>Spécifier la mise en oeuvre . . . . .</i>	267

**Organisation des diagrammes d'UML. . . . . 268**

<i>Organisation générale . . . . .</i>	268
<i>Spécification détaillée. . . . .</i>	268
<i>Spécification technique et mise en oeuvre . . . . .</i>	269
<i>Points d'entrée des diagrammes d'UML. . . . .</i>	269

## **Le diagramme de cas d'utilisation . . . . . 271**

### **Créer un diagramme de cas d'utilisation . . . . . 272**

Créer un paquetage . . . . . 272

Créer le diagramme de cas d'utilisation du paquetage . . . . . 273

### **Les éléments du diagramme de cas d'utilisation . . . . . 274**

Les acteurs . . . . . 274

Les cas d'utilisation . . . . . 275

*Faire un zoom sur un cas d'utilisation . . . . . 275*

Les paquetages . . . . . 275

Les participations . . . . . 277

*Exemple de participation . . . . . 277*

*Créer une participation . . . . . 277*

*Multiplicités d'une participation . . . . . 278*

Les associations entre cas d'utilisation : extension et inclusion . . . . . 278

*Relation d'inclusion . . . . . 278*

*Relation d'extension . . . . . 279*

Les généralisations . . . . . 281

Les interfaces . . . . . 282

*Créer une interface . . . . . 283*

*Relier une interface à un cas d'utilisation . . . . . 283*

## **Le diagramme de classes . . . . . 285**

### **Présentation du diagramme de classes . . . . . 286**

*Le diagramme de classes : synthèse . . . . . 287*

Créer un diagramme de classes . . . . . 287

### **Les classes . . . . . 288**

Définition d'une classe . . . . . 288

Créer une classe . . . . . 289

*Retrouver une classe existante . . . . . 289*

Propriétés d'une classe . . . . . 290

*Page Caractéristiques . . . . . 290*

*Autres pages de propriétés . . . . . 291*

Stéréotype d'une classe . . . . . 291

*Option d'affichage des stéréotypes . . . . . 293*

### **Les attributs . . . . . 294**

Définition d'un attribut . . . . . 294

Spécifier les attributs d'une classe . . . . . 295

*Créer un attribut standard . . . . . 295*

*Créer un attribut calculé . . . . . 295*

*Attributs hérités . . . . . 296*

Propriétés des attributs . . . . . 296

*Type d'un attribut . . . . . 296*

### **Les opérations . . . . . 298**

Définition d'une opération . . . . . 298

Spécifier les opérations d'une classe . . . . . 298

*Opérations héritées . . . . . 298*



Propriétés d'une opération . . . . .	299
Signature d'une opération ou d'un signal . . . . .	299
<i>Syntaxe des signatures</i> . . . . .	300
Paramètres d'une opération . . . . .	301
Méthodes d'une opération (comportement opaque) . . . . .	301
Conditions d'une opération . . . . .	301
<i>Exceptions d'une opération</i> . . . . .	302
Afficher les attributs et les opérations d'une classe . . . . .	302
<b>Les signaux . . . . .</b>	<b>304</b>
Définition d'un signal . . . . .	304
Spécifier les signaux d'une classe . . . . .	304
<i>Créer un signal émis ou reçu</i> . . . . .	304
<i>Propriétés d'un signal</i> . . . . .	304
<i>Paramètres d'un signal</i> . . . . .	305
<b>Les associations . . . . .</b>	<b>306</b>
Créer une association . . . . .	307
Les rôles des associations . . . . .	307
Multiplicité d'un rôle . . . . .	308
<i>Préciser la multiplicité d'un rôle</i> . . . . .	310
Navigabilité d'un rôle . . . . .	311
<i>Préciser la navigabilité d'un rôle</i> . . . . .	312
Agrégation d'un rôle . . . . .	312
<i>Préciser l'agrégation d'un rôle</i> . . . . .	312
Composition d'un rôle . . . . .	312
Rôle modifiable . . . . .	313
Ordre d'un rôle . . . . .	313
Propriété statique d'un rôle . . . . .	314
Qualificatif d'un rôle . . . . .	314
Surcharger un rôle . . . . .	315
Les classes d'association . . . . .	315
Définir une association "plus que binaire" . . . . .	316
Les associations réflexives . . . . .	316
<i>Créer une association réflexive</i> . . . . .	317
<b>Les Parties . . . . .</b>	<b>318</b>
Créer une partie entre deux classes . . . . .	318
Définir l'identifiant d'une classe à travers une partie . . . . .	318
Multiplicités des classes associées . . . . .	319
<i>Multiplicité de la classe référencée par la partie</i> . . . . .	319
<i>Multiplicité de la classe détentrice de la partie</i> . . . . .	320
Relations d'agrégation et de composition . . . . .	320
<i>Multiplicités associées</i> . . . . .	321
<b>Les généralisations . . . . .</b>	<b>322</b>
Qu'est-ce qu'une généralisation . . . . .	322
<i>Exemple</i> . . . . .	323
Cas de plusieurs sous-classes . . . . .	324
Intérêt des sous-classes . . . . .	324
Héritage Multiple . . . . .	325
Créer une généralisation . . . . .	325
Discriminant . . . . .	326
<b>Spécifier les interfaces . . . . .</b>	<b>327</b>
Créer une interface . . . . .	327
<i>Relier une interface à une classe</i> . . . . .	327

<b>Spécifier les dépendances</b> . . . . .	<b>329</b>
<b>Spécifier des classes paramétrées</b> . . . . .	<b>330</b>
<b>Les contraintes</b> . . . . .	<b>331</b>
<b>Le diagramme d'objets</b> . . . . .	<b>332</b>
Les objets . . . . .	332
<i>Créer un objet (une instance)</i> . . . . .	333
<i>Propriétés d'une instance</i> . . . . .	333
<i>Valeur d'un attribut</i> . . . . .	334
Les liens . . . . .	334
<i>Créer un lien</i> . . . . .	335
<i>Propriétés d'un lien</i> . . . . .	335
<i>Propriétés d'un rôle</i> . . . . .	335

---

## **Les diagrammes de structure et de déploiement. . . . . 337**

<b>Le diagramme de paquetages</b> . . . . .	<b>338</b>
Créer un diagramme de paquetages . . . . .	338
Définir les paquetages . . . . .	339
Définir les classes . . . . .	339
Spécifier les dépendances dans un diagramme de paquetage . . . . .	340
<b>Le diagramme de composants</b> . . . . .	<b>341</b>
Créer un diagramme de composants . . . . .	341
Les composants . . . . .	342
Les interfaces . . . . .	342
<i>Créer les interfaces des composants</i> . . . . .	342
<i>Relier les interfaces aux autres objets</i> . . . . .	342
<i>Relier des interfaces</i> . . . . .	343
Les ports . . . . .	343
Les connecteurs . . . . .	344
<i>Connecteur de délégation</i> . . . . .	344
<i>Connecteur d'assemblage</i> . . . . .	344
<b>Le diagramme de structure composite</b> . . . . .	<b>346</b>
Créer un diagramme de structure composite . . . . .	346
Les parties . . . . .	347
Les collaborations . . . . .	347
<i>Utilisation de collaboration</i> . . . . .	348
<i>Exemple d'utilisation de collaboration</i> . . . . .	348
Les liens de dépendance . . . . .	348

---

## **Le diagramme de machine à états . . . . . 351**

<b>Présentation du diagramme de machine à états</b> . . . . .	<b>352</b>
Créer un diagramme de machine à états . . . . .	352
<b>Les états</b> . . . . .	<b>354</b>
Créer un état . . . . .	354
<i>Les types d'état</i> . . . . .	354

<i>Pseudo-états</i> . . . . .	355
Précision comportementale d'un état . . . . .	356
Propriétés d'un état . . . . .	357
<b>Les transitions entre états</b> . . . . .	<b>358</b>
Créer une transition . . . . .	358
Les types de transition . . . . .	358
<i>Transition externe</i> . . . . .	358
<i>Transition interne</i> . . . . .	358
<i>Transition locale</i> . . . . .	359
Effet d'une transition . . . . .	359
<i>Affichage des effets d'une transition</i> . . . . .	359
Événement déclencheur d'une transition . . . . .	359

---

## **Le diagramme d'activités . . . . . 361**

<b>Présentation du diagramme d'activités</b> . . . . .	<b>362</b>
Créer un diagramme d'activités . . . . .	362
<b>Les partitions</b> . . . . .	<b>363</b>
Créer une partition . . . . .	363
Propriétés d'une partition . . . . .	363
<b>Les noeuds</b> . . . . .	<b>365</b>
Les noeuds d'actions . . . . .	365
<i>Créer une action</i> . . . . .	365
<i>Modifier le type d'action</i> . . . . .	365
Les noeuds de paramétrage . . . . .	365
Les noeuds de contrôle . . . . .	366
<i>Types de noeud de contrôle</i> . . . . .	366
Les noeuds d'objets : pins d'entrée, de sortie et d'échange . . . . .	367
<i>Pin d'entrée</i> . . . . .	367
<i>Pin de sortie</i> . . . . .	367
<i>Pin d'échange</i> . . . . .	367
Les Flux . . . . .	367
<i>Flux de contrôle</i> . . . . .	367
<i>Flux d'objets</i> . . . . .	367

---

## **Les diagrammes d'interaction . . . . . 369**

<b>Les interactions</b> . . . . .	<b>370</b>
Créer une interaction . . . . .	370
Créer un diagramme d'interaction . . . . .	370
<b>Le diagramme de séquence</b> . . . . .	<b>371</b>
Créer un diagramme de séquence . . . . .	372
Les lignes de vie . . . . .	372
<i>Créer une ligne de vie</i> . . . . .	372
<i>Propriétés d'une ligne de vie</i> . . . . .	372
Les messages . . . . .	372

<i>Exemples de messages échangés</i> . . . . .	373
<i>Créer un message</i> . . . . .	373
<i>Types de messages</i> . . . . .	374
Occurrence d'exécution . . . . .	374
<i>Créer une occurrence d'exécution</i> . . . . .	374
Occurrence d'événement . . . . .	375
<i>Calcul des numéros de séquence</i> . . . . .	375
Fragment combiné . . . . .	376
<i>Créer un fragment combiné</i> . . . . .	377
<i>Type d'opérateur d'interaction</i> . . . . .	378
<i>Opérande d'interaction</i> . . . . .	380
Utilisation d'interaction . . . . .	380
Porte . . . . .	381
Continuation . . . . .	382
<b>Le diagramme de communication</b> . . . . .	<b>383</b>
<i>Exemple</i> . . . . .	383
<i>Objets du diagramme</i> . . . . .	384
<b>Le diagramme de vue générale d'interaction</b> . . . . .	<b>385</b>

---

## **Le diagramme de déploiement. . . . . 387**

<b>Présentation du diagramme de déploiement</b> . . . . .	<b>388</b>
Créer un diagramme de déploiement . . . . .	388
Objets du diagramme de déploiement . . . . .	389
<i>Noeud</i> . . . . .	389
<i>Chemin de communication</i> . . . . .	389
<i>Composant</i> . . . . .	389
<i>Artefact</i> . . . . .	389
<i>Manifestation</i> . . . . .	389
<i>Spécification de déploiement</i> . . . . .	390
<i>Configuration</i> . . . . .	390

---

## **Annexe : Type des attributs. . . . . 391**

<b>Types élémentaires</b> . . . . .	<b>392</b>
Conditions préalables : importer les types élémentaires . . . . .	392
Définir un type élémentaire . . . . .	392
<b>Paquetages et types élémentaires</b> . . . . .	<b>394</b>
Paquetages . . . . .	394
<b>Définir de nouveaux types élémentaires.</b> . . . . .	<b>397</b>
Type élémentaire composé. . . . .	398

# INTRODUCTION



**HOPEX IT Architecture** permet aux responsables informatique de formaliser les besoins métier afin de définir l'architecture du système d'information qui y répond depuis l'architecture logique jusqu'à l'infrastructure technique.

**HOPEX IT Architecture** offre des facilités selon différents axes d'analyse :

- ✓ **La gestion du système d'information** : description des architectures de service et l'urbanisation du SI sont deux approches qui facilitent l'évolution du SI en fournissant un cadre de référence pour la planification de vos systèmes.
- ✓ **La cartographie applicative** : description de l'architecture applicative qui offre une vue détaillée des échanges d'information entre les applications, les services, les bases de données et les unités organisationnelles.
- ✓ **Le déploiement des applications** : description de l'infrastructure technique du système d'information afin de suivre le déploiement des applications sur les différents sites de l'entreprise. L'infrastructure technique prend en compte les principaux matériels informatiques de l'entreprise (réseaux, serveurs, postes de travail, imprimantes, coupe-feu, concentrateurs, etc).
- ✓ **La représentation des architectures de ressources** : description de systèmes complexes qui font intervenir différents types de ressources informatiques ou pas.

En complément à **HOPEX IT Architecture**, **HOPEX IT Business Management** permet aux entreprises de gérer la transformation de leur système d'information en offrant des possibilités de définir des étapes et des gérer des évaluations à chacune de ces étapes.

**HOPEX IT Architecture** propose également un outil qui permet d'importer des éléments de configuration issus de CMDB (Configuration Management DataBase) et de les mettre en correspondance avec des objets de modélisation décrits dans **HOPEX IT Architecture**. Pour plus d'informations, voir la documentation "CMDB Import".

L'objectif de ce guide est de présenter comment exploiter ces fonctionnalités pour mener à bien l'évolution de votre système d'information.

Voici les points abordés dans **HOPEX IT Architecture** :

- ✓ [Modéliser les applications et les architectures système ;](#)
- ✓ [Modéliser les architectures applicatives ;](#)
- ✓ [Modéliser les architectures de déploiement ;](#)
- ✓ [Aligner l'informatique et les métiers ;](#)
- ✓ [Modéliser les infrastructures informatiques ;](#)
- ✓ [Accéder à la conception de logicielle ;](#)
- ✓ [Décrire les échanges d'informations ;](#)
- ✓ [A propos de l'implémentation UML.](#)

# PRÉSENTATION DE HOPEX IT ARCHITECTURE

Associé à l'ensemble des produits de la suite **HOPEX**, **HOPEX IT Architecture** supporte une méthodologie et les outils qui permettent de décrire, analyser et planifier la transformation de votre système d'information.

---

## Le périmètre couvert par HOPEX IT Architecture

Les modules proposées en standard permettent de respecter une approche top-down en commençant par une réflexion sur les capacités métier de l'entreprise et sa stratégie pour terminer sur une définition précise des composants du système d'information existant ou futur.

Chacun des modules s'adresse à des profils d'utilisateurs précis. Des rapports standards sont proposés pour faciliter l'analyse des sujets traités.

La méthode décrite dans ce guide est représentée par les modules décrits ci-dessous.

➤ *L'ordre d'utilisation de ces modules est donné à titre indicatif.*

**Décrire et analyser les flux :** cette étape s'appuie sur les diagrammes de scénario qui permettent de représenter la circulation des flux entre les composants de votre système d'information.

➤ *Pour plus de détails, voir [Décrire les scénario de flux](#).*

**Analyser la couverture fonctionnelle de l'architecture technique :** cette étape utilise les rapports proposés par **HOPEX IT Architecture** pour analyser les liens entre les composants de l'architecture applicative décrite et les fonctionnalités attendues.

➤ *Pour plus de détails sur la modélisation des applications et des services, voir [Décrire l'architecture applicative](#) et [Décrire les applications](#).*

**Décrire la stratégie d'évolution du système informatique et de l'architecture logique :** cette étape consiste à décrire ce que le système d'information est en mesure de délivrer, à travers les capacités métier, et comment il le délivre à travers les architectures d'applications logiques.

➤ *Pour plus de détails, voir [Analyser la couverture fonctionnelle de l'architecture mise en œuvre](#) et [Construire l'architecture logique](#).*

**Décrire l'environnement applicatif :** cette étape consiste à décrire l'architecture de déploiement et l'ensemble des éléments qui la compose.

➤ *Pour plus de détails, voir [Définir l'architecture de déploiement d'une application](#).*

**Décrire l'infrastructure technique :** ce module permet de gérer les contraintes de déploiement et d'y associer des solutions adaptées.

➤ *Pour plus de détails, voir [Définir l'infrastructure technique](#).*

**Utiliser le formalisme UML :** enfin, vous pouvez adapter le langage de modélisation UML (Unified Modeling Language) pour modéliser votre SI.

➡ Pour plus de détails, voir [A propos de l'implémentation UML](#).

## Synthèse des activités et livrables de HOPEX IT Architecture

Des activités sont associées à chacun des modules de la méthode que nous recommandons pour la gestion de l'évolution de votre système d'information.

La solution **HOPEX IT Architecture** vous offre des facilités pour mener à bien ces activités qui sont matérialisées par des livrables.

Activités	Principaux livrables
Définir l'architecture logique	Diagrammes de structure d'architecture logique, voir <a href="#">Décrire l'architecture logicielle logique</a> .
Construire l'architecture applicative	Diagrammes de structure d'architecture applicative et les diagrammes de scénario de flux, voir <a href="#">Décrire une application avec HOPEX IT Architecture</a> .
Analyser la couverture fonctionnelle de l'architecture applicative	Évaluation de la couverture fonctionnelle par les ressources logicielles, voir <a href="#">Décrire la mise en œuvre d'une fonctionnalité</a> . Évaluation de la couverture des capacités technologiques par les ressources techniques, voir <a href="#">Décrire la mise en œuvre d'une fonctionnalité</a> .
Définir l'architecture de déploiement	Description des exigences techniques pour le déploiement des applications, voir <a href="#">Modéliser les architectures de déploiement</a> .
Définir l'infrastructure	Description des types d'infrastructure techniques pour le déploiement des applications, voir <a href="#">Modéliser les infrastructures informatiques</a> .
Gérer les catalogues de service	Description des catalogues de service et des solutions recommandées, voir <a href="#">Utiliser les catalogues de services</a> .

Présentation des livrables de **HOPEX IT Architecture**

## Structure et positionnement de la solution HOPEX IT Architecture

**HOPEX IT Architecture** peut être utilisé avec les autres produits de la suite **HOPEX**.

### **HOPEX IT Business Management**

La solution **HOPEX IT Business Management** apporte à **HOPEX IT Architecture** une méthodologie et des outils qui permettent de planifier votre transformation

métier. Les deux solutions partagent la fonctionnalité de cartographie des capacités métier.

### ***HOPEX Business Process Analysis***

En plus de **HOPEX IT Business Management**, la solution **HOPEX Business Process Analysis** vous apporte la possibilité de décrire les organisations et les processus qui mettent en œuvre les capacités métier identifiées dans **HOPEX IT Architecture**;

---

## **Les profils de HOPEX IT Architecture**

Dans **HOPEX IT Architecture**, il existe, par défaut, des profils auxquels sont associées des activités spécifiques.



La présentation de l'interface de la solution dépend du profil sélectionné par l'utilisateur lors de sa connexion à l'application ; l'arborescence des menus et les fonctionnalités sont différentes d'un profil à l'autre.

☛ Pour plus de détails sur les bureaux associés à chacun des profils, voir [Présentation du bureau HOPEX IT Architecture](#).

Profils	Tâches
Administrateur fonctionnel d'architecture de solution	<p>En plus des droits fonctionnels de l'Architecte Solution, l'administrateur fonctionnel d'architecture de solution possède des droits sur tous les objets, les méthodes, les projets et les work-flows.</p> <p>Il prépare l'environnement de travail et crée les éléments nécessaires à la gestion des éléments modélisés.</p> <p>Il gère tous les objets de l'environnement (environnements applicatifs, infrastructures, rapports, etc...).</p> <p>Pour plus de détails, voir <a href="#">Présentation des menus de l'espace de l'Administrateur fonctionnel d'architecture de solution</a>.</p>
Architecte Solution	<p>L'Architecte Solution possède des droits sur tous les objets, les méthodes, les projets et les évaluations.</p> <p>Il est chargé de construire les modèles des architectures applicatives, techniques IT et d'infrastructures IT qui lui sont confiées. Il peut gérer des projets de transformation.</p> <p>L'Architecte Solution il est chargé des spécifications détaillées du système et de la construction des diagrammes UML.</p> <p>Pour plus de détails, voir <a href="#">Présentation de l'espace de l'Architecte solution</a>.</p>
Contributeur d'applications	<p>Le Contributeur d'applications est chargé de la validation de la conception des applications qui leur sont confiées.</p> <p>Pour plus de détails, voir <a href="#">Présentation de l'espace du concepteur d'applications</a>.</p>
Observateur d'applications	<p>L'Observateur d'applications n'a que des droits de lecture sur les objets du référentiel.</p> <p>Pour plus de détails, voir <a href="#">Présentation de l'espace de l'observateur d'applications</a>.</p>

---

## Les rôles métier de HOPEX IT Architecture

Dans **HOPEX IT Architecture**, il existe, par défaut, des rôles métier qui peuvent être assignés à certains utilisateurs. Ces rôles sont :

- **Concepteur de logiciels** qui permet d'assigner un utilisateur à des éléments logiciels. Le Concepteur logiciel est responsable de la conception des logiciels qui lui sont assignés.
- **Concepteur de données** qui est responsable de la gestion des données.
- **Propriétaire local d'application** qui permet d'assigner un utilisateur à des applications. Le Propriétaire local d'application est responsable des tâches suivantes :
  - Identifier les risques
  - Répondre aux questionnaires
  - Définir et mettre en œuvre les plans d'action
  - Valider les modifications effectuées par l'architecte dans le cadre des workflows de revue d'objets).
- **Propriétaire métier** spécifie les caractéristiques des installations logicielles et des applications dont il a la charge sur le plan métier.
- **Responsable informatique** spécifie les caractéristiques des ressources informatiques dont il est responsable.

# LA MÉTHODE DE HOPEX IT ARCHITECTURE

La méthode présentée dans ce paragraphe est donnée à titre indicatif. En fonction de votre contexte de travail, vous pouvez enchaîner les étapes décrites de manière différente.

## Décrire l'architecture applicative

**HOPEX IT Architecture** offre des moyens de représenter différents niveaux d'architectures d'applications : depuis la description de l'environnement applicatif jusqu'aux composants techniques à mettre en œuvre.

Ces représentations permettent de définir les composants logiciels et matériels et d'identifier de manière cohérente les données échangées entre eux.

☛ Pour plus de détails sur l'utilisation d'une architecture applicative, voir [Modéliser les applications et les architectures système](#).

La description des **systèmes applicatifs** peut se faire selon une approche descendante, en commençant par décrire les systèmes applicatifs principaux de l'entreprise, ou bien selon une approche unitaire en décrivant seulement certains systèmes applicatifs.

📖 Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.

## Décrire un environnement de système applicatif

Si vous utilisez une approche unitaire, vous devrez décrire l'**environnement de système applicatif** pour décrire le contexte d'utilisation du système applicatif décrit et ses interactions de service avec les composants externes.

📖 Un environnement de système applicatif permet de présenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels ce système applicatif est susceptible d'interagir.

☛ Pour plus de détails sur les environnements de systèmes applicatifs, voir [Décrire un environnement de système applicatif avec HOPEX IT Architecture](#).

Outre une description précise de l'architecture applicative à mettre en œuvre, cette étape permet de couvrir les points suivants :

- Identifier précisément les échanges entre les différents composants logiciels et matériels et les formaliser par des **interfaces de service**.

📖 Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations

déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

☛ Pour plus de détails sur les interfaces de service, voir [Décrire les échanges d'informations](#).

- Vérifier que l'architecture applicative couvre les besoins fonctionnels recensés dans les cartes de capacités métier.

☛ Pour plus de détails sur l'analyse fonctionnelle, voir [Analyser la couverture fonctionnelle de l'architecture mise en œuvre](#).

## Décrire les systèmes applicatifs

Dans une approche descendante, le diagramme de structure du système applicatif principal est le point d'entrée de la description d'un système applicatif existant ou prévu.

➡ Pour plus de détails sur les systèmes applicatifs, voir [Décrire un système applicatif](#).

Le diagramme suivant décrit le système applicatif du traitement des demandes d'achat.

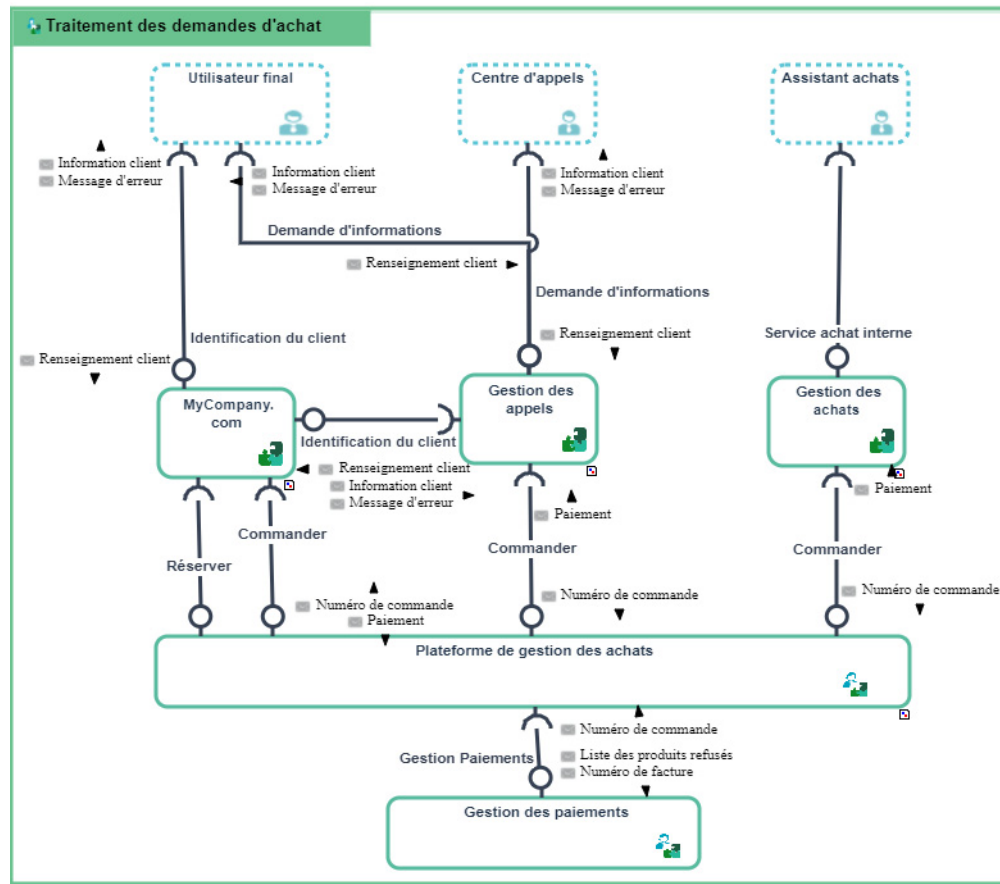


Diagramme de structure de système applicatif "Traitement des demandes d'achat"

Les demandes d'achat sont formulées par des utilisateurs externes via une application de eCommerce ou un centre d'appels. Les utilisateurs internes passent par un "Assistant achats" qui utilise une application de "Gestion des demandes d'achat".

Les sous-systèmes applicatifs peuvent ensuite être décrits hiérarchiquement en faisant apparaître à chaque niveau les points d'échange avec l'extérieur.

Les *dépôts de données* permettent de représenter les données qui seront stockées dans des base de données.



*Un dépôt de données fournit un mécanisme pour mettre à jour ou consulter des données qui vont persister au-delà de la portée du processus en cours. Il permet de stocker les données des flux en entrée et de les retransmettre par l'intermédiaire d'un ou plusieurs flux en sortie.*

☛ Pour plus de détails sur les dépôts de données, voir [Gérer les données](#).

---

## Décrire les applications

Avec **HOPEX IT Architecture**, une application est décrite par :

- les flux d'information qu'elle traite et transporte, voir [Décrire les scénario de flux](#),
- les éléments qui assurent les services associés aux fonctionnalités qu'elle couvre, voir [Décrire la structure de l'application et ses services](#).

Ces deux approches complémentaires permettent de dresser une liste exhaustive des composants d'une application (services et API) et des composants de son environnement avec lesquels elle interagit.

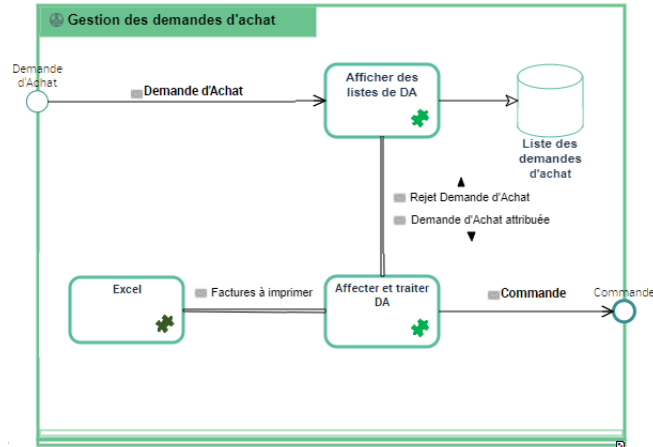
### Décrire les scénario de flux

**HOPEX IT Architecture** offre les notions de scénario de flux pour décrire précisément les données échangées.

A chaque niveau de l'architecture applicative, il est possible de définir des scénario qui représentent les flux échangés entre les composants du système dans des contextes spécifiques.

L'objectif des scénario de flux est de vérifier que les contenus sont correctement véhiculés entre les composants.

Le diagramme de scénario de flux ci-après décrit l'application "Gestion des demandes d'Achat".



Exemple de scénario de flux d'application de "Gestion des demandes d'achat"

➤ Pour plus de détails sur les scénario de flux, voir [Créer un diagramme de scénario de séquence de flux](#).

## Décrire la structure de l'application et ses services

Les diagrammes de structure utilisent les interactions de service pour décrire les échanges de données entre les composants.

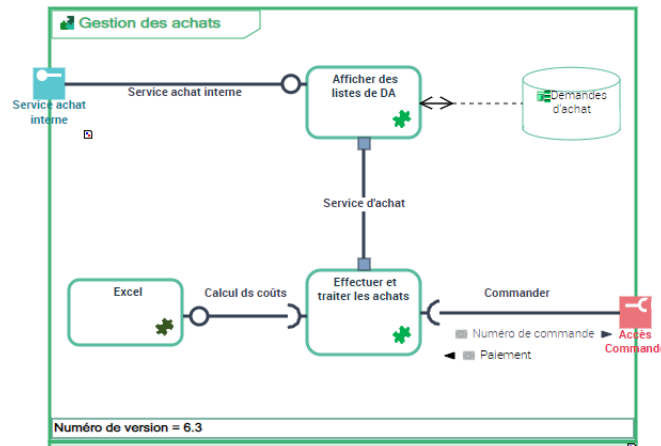


Diagramme de structure de l'application "Gestion des demandes d'achat"


L'application de gestion des demandes d'achats s'appuie sur deux services applicatifs spécialisés : "Afficher des listes de DA" et "Affecter et traiter DA". Le service


applicatif "Affecter et traiter DA" utilise le microservice Excel.


---

## Définir l'architecture de déploiement d'une application

Une architecture de déploiement d'une application permet de représenter les *modules de déploiement d'une application* et les *modules de déploiement des données* qui seront déployées ainsi que les *connexions de modules* utilisées pour leurs échanges.

 *Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/ Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/ PaaS cloud service ou un modèle de serveur).*

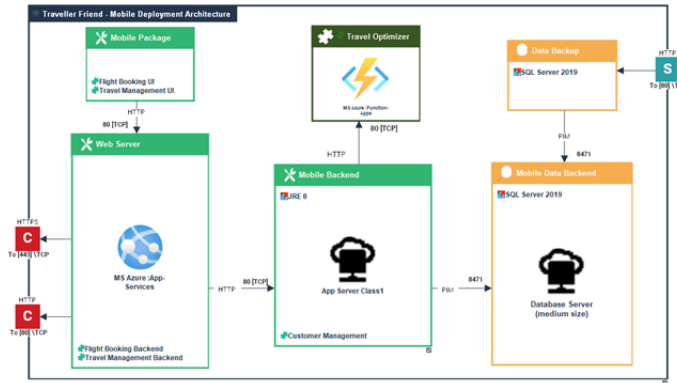
 *Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).*

 *Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.*

Plusieurs angles de vue sont proposés avec **HOPEX IT Architecture** :

- L'*environnement de déploiement d'une application* qui permet de représenter les déploiements d'applications partenaires ainsi que les microservices requis autour de l'application sujet, voir [Décrire un environnement de déploiement d'une application](#).
- L'*architecture de déploiement du système d'application* qui permet de représenter l'ensemble des architectures de déploiement d'application qui doivent être coordonnées pour couvrir les dépendances requises entre elles, voir [Décrire une architecture de déploiement d'un système applicatif](#).
- L'*architecture de déploiement d'une application* qui présente la liste des modules de déploiement et des connexions de module, voir [Décrire une architecture de déploiement d'une application](#).





### Diagramme d'architecture de déploiement d'une application

Afin de faciliter la création de vos diagrammes d'architecture de déploiement d'une application, **HOPEX IT Architecture** propose des modèles d'architecture de déploiement.


➡ Pour plus de détails, voir [Les modèles d'architecture de déploiement](#).

## Construire l'architecture logique


**HOPEX IT Architecture** offre des moyens de définir des architectures d'applications logiques qui représentent des architectures idéales. Ces représentations permettent de concevoir les structures logiques pour les architectures d'applications, de rationaliser les échanges entre ces structures et d'identifier les données utilisées. Les architectures d'applications logiques peuvent ensuite être comparées aux architectures mises en œuvre pour détecter les écarts entre le réel et l'idéal.

➡ Pour plus de détails sur l'utilisation d'un système applicatif logique, voir [Décrire l'architecture logicielle logique](#).

La description des *systèmes applicatifs logiques* peut se faire selon une approche descendante, en commençant par décrire les systèmes applicatifs principaux de l'entreprise, ou bien selon une approche unitaire en décrivant seulement certains systèmes applicatifs logiques.

 *Un système applicatif logique est un assemblage d'autres architectures, d'applications logiques et d'utilisateurs finaux interagissant avec les composants applicatifs, dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

Si vous utilisez un approche unitaire, vous devrez décrire l'*environnement de système applicatif logique* pour décrire le contexte d'utilisation du système logique décrit et ses interactions de service avec les composants logiques externes.

 Un environnement de système applicatif logique présente un contexte d'utilisation de systèmes applicatifs logiques. Il décrit les interactions de service, entre le système applicatif logique et ses partenaires externes, qui lui permettent de remplir sa mission et d'assurer les fonctionnalités attendues.

A ce niveau de la méthode, cette étape, qui n'est pas obligatoire, permet néanmoins de couvrir les points suivants :

- Identifier les échanges entre les différents composants logiques et les formaliser par des *interfaces de service*.



*Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).*



*Pour plus de détails sur les interfaces de service, voir [Décrire les échanges d'informations](#).*

- Vérifier que l'architecture logique couvre les besoins fonctionnels recensés dans les cartes de capacités métier.

## Diagramme de structure de système applicatif logique

Les composants du *système applicatif logique* sont décrits dans un diagramme qui présente :

- les services offerts ou requis ;
- les processus pris en charge, les composants et leurs interactions de service ;
- les utilisateurs finaux en interaction avec les composants applicatifs.

Le diagramme suivant décrit la structure du système applicatif logique "Traitement des demandes d'achat" proposé aux clients.

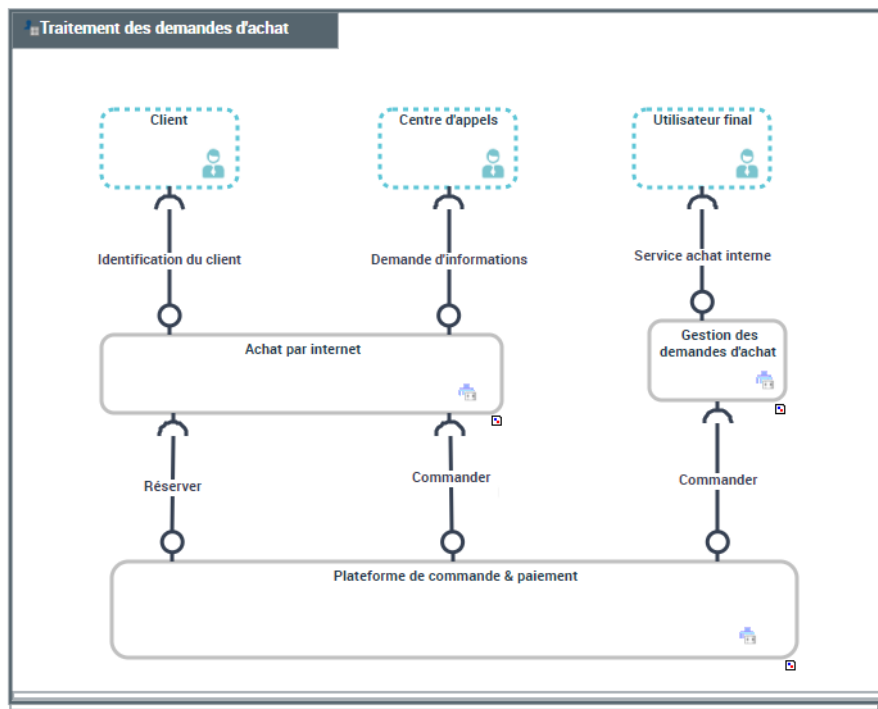


Diagramme de structure de système applicatif logique "Traitement des demandes d'achat"

Les "Demandes d'achats par Internet" sont proposés à des clients soit directement soit par l'intermédiaire d'un "Centre d'appels".

Les demandes formulées par les clients sont traitées par un système applicatif logique de "Demandes d'achats par Internet".

Le diagramme de structure de système applicatif logique, chargé de gérer les "Gestion des demandes d'achats" présente différentes applications logiques, un accès à une

base de donnée logique ainsi que des points de service et de requête pour "Service achat interne" ou "Commander".

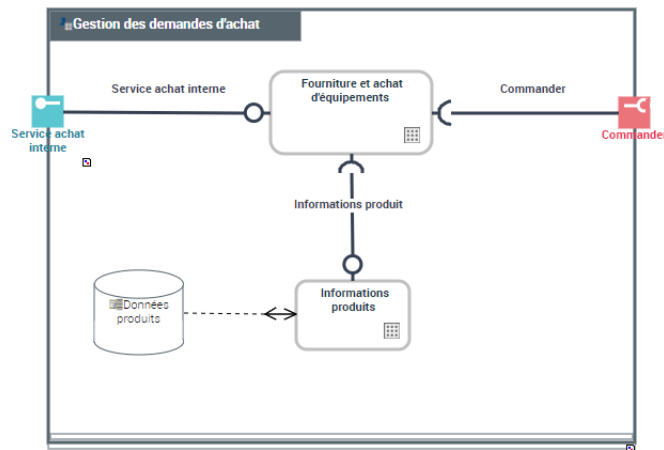


Diagramme de structure de système applicatif logique "Gestion des demandes d'achat"

➤ Pour plus de détails sur l'utilisation d'un système applicatif logique, voir [Décrire un système applicatif logique avec HOPEX IT Architecture](#).

## Diagramme d'environnement de système applicatif logique

📖 Un environnement de système applicatif logique présente un contexte d'utilisation de systèmes applicatifs logiques. Il décrit les interactions de service, entre le système applicatif logique et ses partenaires externes, qui lui permettent de remplir sa mission et d'assurer les fonctionnalités attendues.

Les composants d'un **environnement de système applicatif logique** sont présentés dans un diagramme d'environnement de système applicatif logique qui présente les

systèmes applicatifs logiques internes ainsi que les systèmes applicatifs logiques partenaires.

s

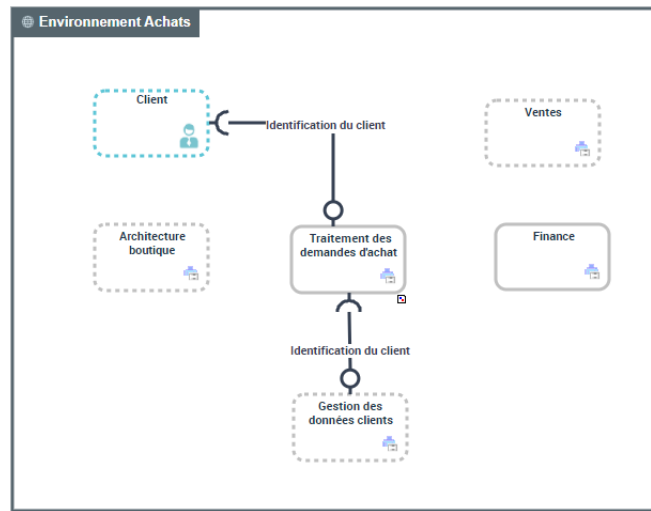


Diagramme d'environnement de système applicatif logique

Les demandes d'achat sont formulées par des clients dans des conditions définies par le service Ventes et le service "Finance" qui est externe à l'environnement décrit.

✎ Pour plus de détails sur l'utilisation d'un environnement de système applicatif logique, voir [Décrire un environnement de système applicatif logique](#).

## Analyser la couverture fonctionnelle de l'architecture mise en œuvre

Cette étape, de niveau stratégique, a pour objectif de vérifier l'adéquation entre les capacités métier de l'entreprise, les fonctionnalités attendues et les applications qui les réalisent.

### Identifier les capacités métier

✎ Pour plus de détails sur la gestion des capacités métier avec **HOPEX IT Architecture**, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).

Une **capacité métier** définit une aptitude escomptée.

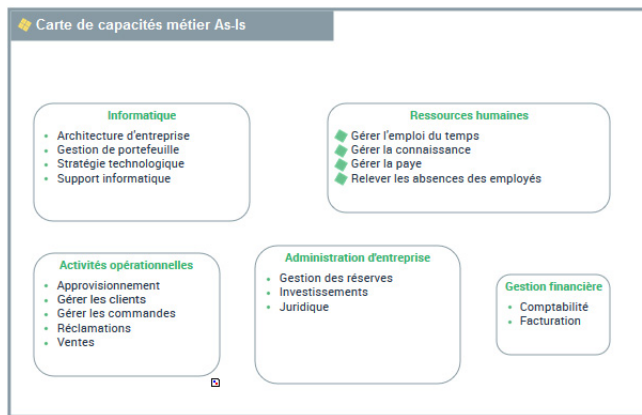
📖 Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).

Une **carte de capacités métier** permet de décrire ce que l'entreprise est capable de produire pour ses besoins internes ou pour répondre aux attentes de ses clients. Elle s'appuie donc sur les capacités métier principales de son activité à un instant donné.



*Une carte de capacités métier est un assemblage de capacités métier avec leurs dépendances qui, conjointement, définissent un cadre pour une phase d'entreprise.*

Par exemple, la capacité standard à gérer les "Activités opérationnelles" s'appuie sur les capacités métier de traiter les "Approvisionnements", les "Ventes" et les "Réclamations", de "Gérer les commandes" et de "Gérer les clients".



Exemple de carte de capacités métier



*Pour plus de détails sur la gestion d'une carte de capacités métier, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).*

## Identifier les capacités technologiques associées aux capacités métier

Il s'agit ici de relier les **capacités technologiques**, qui correspondent à ce que l'on attend pour atteindre les objectifs, aux moyens de réalisation qui sont représentés par des **applications** ou des **Systèmes applicatifs** à un niveau conceptuel



*Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

La construction de la **carte de fonctionnalités** d'une part et de l'**environnement de système applicatif** d'autre part permet de vérifier que les fonctionnalités sont bien mises en œuvre par des composants applicatifs.




*Pour plus de détails sur les applications logiques associées à des capacités métier, voir [Décrire la mise en œuvre d'une fonctionnalité](#).*

**HOPEX IT Architecture** met à votre disposition un rapport qui vous présente le résultat de la mise en œuvre des capacités métier par des **applications** ou des **Systèmes applicatifs** concrets ou logiques.



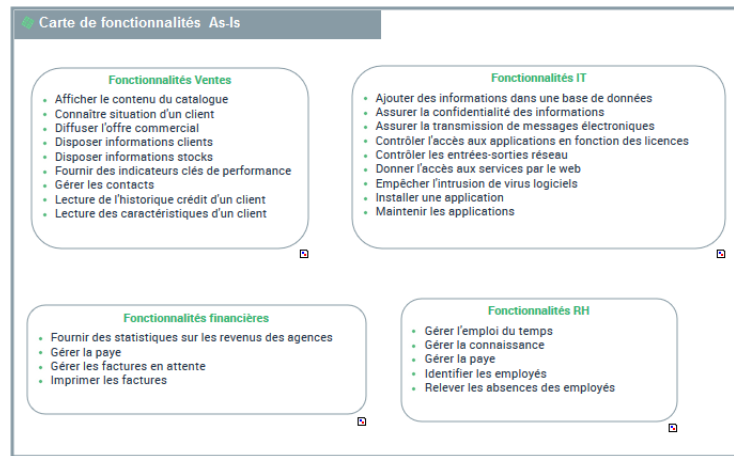
*Pour plus de détails sur la mise en œuvre des capacités métier, voir [Rapport de décomposition de capacité métier](#).*

Une **capacité technologique** définit une aptitude attendue d'un équipement.

 Une **capacité technologique** est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.

Une **carte de fonctionnalités** permet de décrire l'ensemble des fonctionnalités que l'entreprise est capable de couvrir pour ses besoins internes ou pour répondre aux attentes de ses clients.


 Une **carte de capacités technologiques** est un assemblage de **capacités technologiques** avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture.



Exemple de carte de fonctionnalités

 Pour plus de détails sur la gestion d'une carte de fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#).

La description des **capacités métier** et des **fonctionnalités** est particulièrement intéressante si les capacités métiers sont associées aux fonctionnalités qui permettent de les satisfaire.

 Pour plus de détails sur cette analyse, voir [Décrire la mise en œuvre d'une capacités métier](#).

## Identifier les applications associées aux fonctionnalités

Les **applications** couvrent des **fonctionnalités** associées à des **capacités métier**. Un rapport **HOPEX IT Architecture** permet de vérifier la couverture fonctionnelle du parc applicatif.

 Pour plus de détails sur cette couverture fonctionnelle, voir [Décrire la mise en œuvre d'une capacités métier](#).


## Définir l'infrastructure technique

Décrire l'infrastructure technique permet de concevoir des applications et les modules de déploiement des données afin de préparer le déploiement.

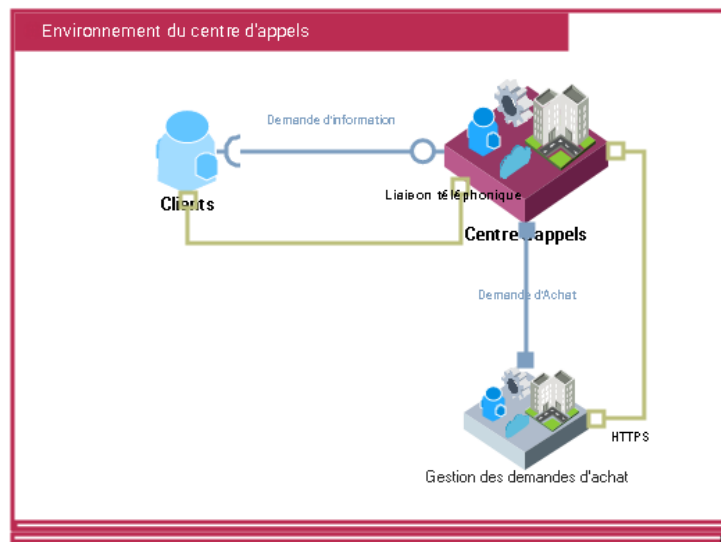
Les éléments d'infrastructure technique sont identifiés et caractérisés par des technologies de soutien et de services informatiques hébergés.

Avec **HOPEX IT Architecture**, l'infrastructure peut être décrite selon une approche ascendante, du plus détaillé vers le plus conceptuel, ou descendante, du plus conceptuel vers le plus détaillé. La présentation de ces fonctionnalités s'appuie sur l'exemple d'un centre d'appels.

### Diagramme d'environnement d'architecture de ressources

 *Un environnement d'architecture métier représente les relations d'un domaine fonctionnel métier avec ses partenaires.*

Le diagramme suivant décrit l'environnement d'un centre d'appels.



Le centre d'appels répond aux demandes des clients. Il s'appuie sur un service extérieur pour effectuer d'éventuelles demandes d'achats.

➡ Pour plus de détails, voir [Décrire les infrastructures informatiques](#).

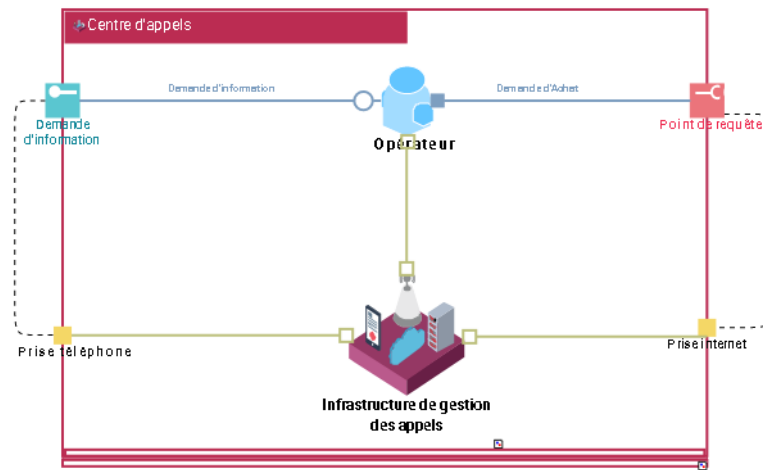
### Décrire une architecture de ressources

Le diagramme d'assemblage de ressources décrit les ressources matérielles et organisationnelles nécessaires à la prise en charge des demandes de services.

➡ Pour plus de détails, voir [Décrire les architectures de ressources](#).




Dans l'exemple du centre d'appels, nous considérons seulement l'opérateur et l'infrastructure informatique qui représente son matériel.



Une équipe d'opérateurs prend en charge toutes les demandes, quelle que soit leur nature, par téléphone ou par mail.

L'opérateur identifie le client, enregistre la demande, effectue un premier filtre (en cas d'erreur) et enregistre éventuellement une demande d'achat via des points de requête.

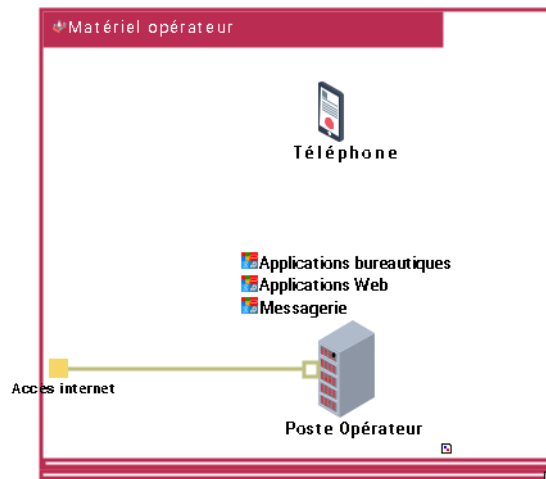
Ce diagramme contient un **point de requête** à partir desquels les opérateurs effectuent des demandes d'achats.

 *Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.*

## Diagramme de structure d'assemblage d'infrastructure informatique


Ce diagramme présente une infrastructure informatique. Il contient des composants informatiques de l'infrastructure tels que des serveurs ou des équipements techniques.

 *Pour plus de détails, voir [Décrire les infrastructures informatiques](#).*



L'architecture matérielle de base d'un centre d'appels comprend deux points de liaison avec l'extérieur : un lien téléphonique et un lien vers un réseau privé qui permet la liaison HTTP pour la demande d'achat.

Notez que les *protocoles de communication* utilisés sont précisés sur les canaux réseau.

 *Un protocole de communication est un ensemble de règles normalisées pour la transmission d'informations (voix, données, images) sur un canal de communication. Les différentes couches de protocoles peuvent prendre en charge la détection et le traitement des erreurs, l'authentification des correspondants, la gestion du routage.*

## Diagramme d'assemblage d'ordinateurs

Le diagramme d'assemblage d'ordinateurs présenté ci-dessous décrit les technologies logicielles, les modules de déploiement d'une application et les périphériques informatiques installés sur un PC standard.

➡ Pour plus de détails, voir [Décrire les ordinateurs](#).

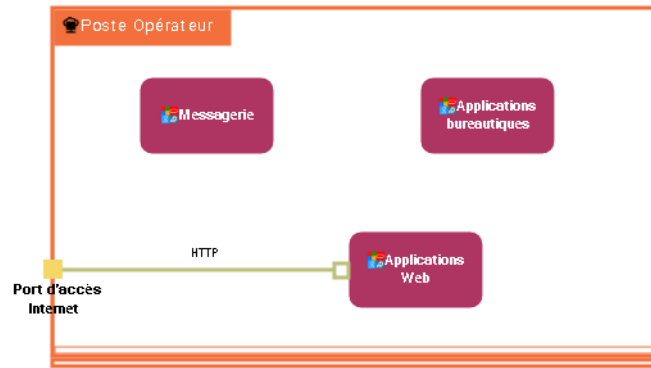


Diagramme d'assemblage d'ordinateur d'un PC standard

Un PC standard est équipé d'applications bureautiques et d'une messagerie.

Un PC standard dispose également d'une connexion HTTP pour accéder aux applications Web de gestion des demandes d'achat.

## Concevoir et spécifier les applications

**HOPEX IT Architecture** offre des facilités pour assister les architectes dans la spécification des évolutions de leur système d'information.

➡ Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Concepteur d'application informatique** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.

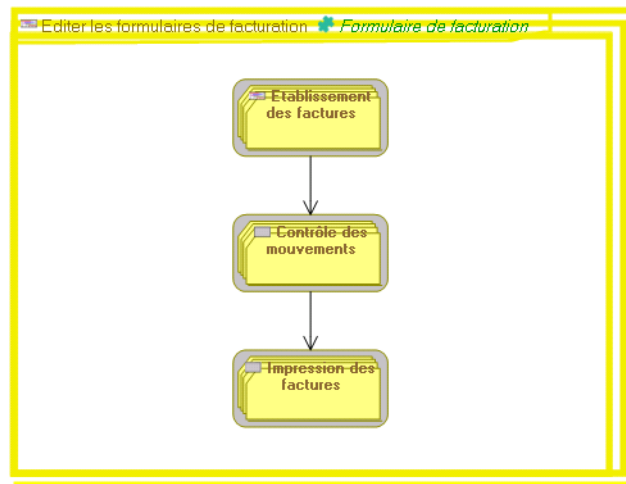
### Utiliser le formalisme UML

**HOPEX IT Architecture** fournit les outils nécessaires à la modélisation des données logiques via les diagrammes de classes et les modèles de données.

➡ Pour plus de détails sur la description des principaux concepts UML, voir [Modélisation UML des données](#).

## Décrire les traitements par lot

L'enchaînement de traitements automatisés peut être décrit dans un **diagramme de structure de chaîne de traitement de lot**.

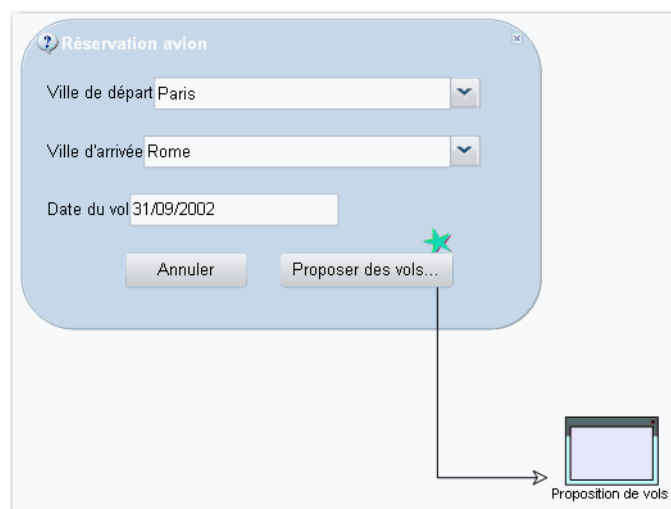


☛ Pour plus de détails sur la description des traitements par lot, voir [Décrire les traitements par lots](#).

## Décrire la liste des services et des IHM

Il est possible de décrire les interfaces reliant les services ou les opérations avec l'extérieur. Cette description s'effectue dans un diagramme d'IHM.


Une diagramme d'IHM permet de décrire les IHM prévues.




☛ Pour plus de détails sur la description des IHM, voir [Décrire les interfaces utilisateur](#).

## Décrire les processus applicatifs

Enfin, **HOPEX IT Architecture** offre la possibilité de vérifier que les processus réalisés par le système applicatif sont correctement couverts en décrivant des *processus applicatifs*.

 Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.

 Pour plus de détails sur les processus applicatifs, voir [Décrire les processus applicatifs](#).

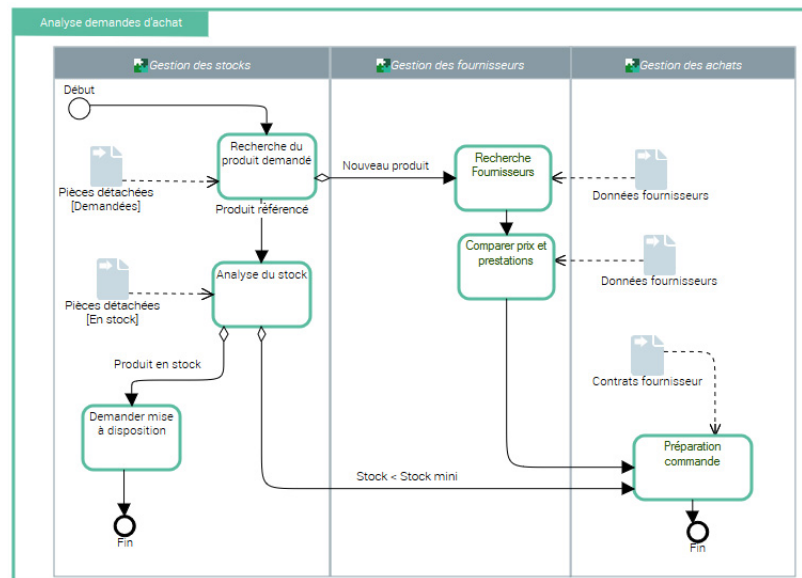



Diagramme du processus applicatif


## Gérer les catalogues de service

Un catalogue de service décrit la liste des fonctionnalités couvertes par une solution ainsi que les éléments techniques ou fonctionnels qui mettent en œuvre ces fonctionnalités.

 Un catalogue de services contient une liste d'offres de services clés pour lesquelles des solutions sont recommandées.

**HOPEX IT Architecture** propose les catalogues de services suivants :

- les *catalogues de services métier*,

 Un catalogue de services métier fournit une source d'information centralisée sur les services matériels proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services associés aux capacités métier, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services métier présente la liste des

fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services Cloud*,



Un catalogue des services d'information fournit une source d'information centralisée sur les services d'information proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services d'information utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services d'information présente la liste des fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services technologiques*,



Un catalogue de services technologiques fournit une source d'information centralisée sur les services technologiques proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services technologiques utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services technologiques présente la liste des fonctionnalités IT référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services d'information*,



Un catalogue des services d'information fournit une source d'information centralisée sur les services d'information proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services d'information utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services d'information présente la liste des fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services matériels*.



Un catalogue de services matériels fournit une source d'information centralisée sur les services matériels proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services matériels utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services matériels présente la liste des fonctionnalités matérielles référencées ainsi que les recommandations de mise en œuvre.

➡ Pour plus de détails sur l'utilisation des catalogues de services, voir [Utiliser les catalogues de services](#).

# PRÉSENTATION DU BUREAU HOPEX IT ARCHITECTURE

☛ **HOPEX IT Architecture** s'adresse principalement à des utilisateurs web. Les bureaux décrits dans ce guide sont accessibles exclusivement aux utilisateurs de bureaux web.

---

## Se connecter à la solution

Pour se connecter à **HOPEX IT Architecture**, voir **HOPEX Common Features**, "Le bureau HOPEX".

---

## Présentation du bureau HOPEX IT Architecture

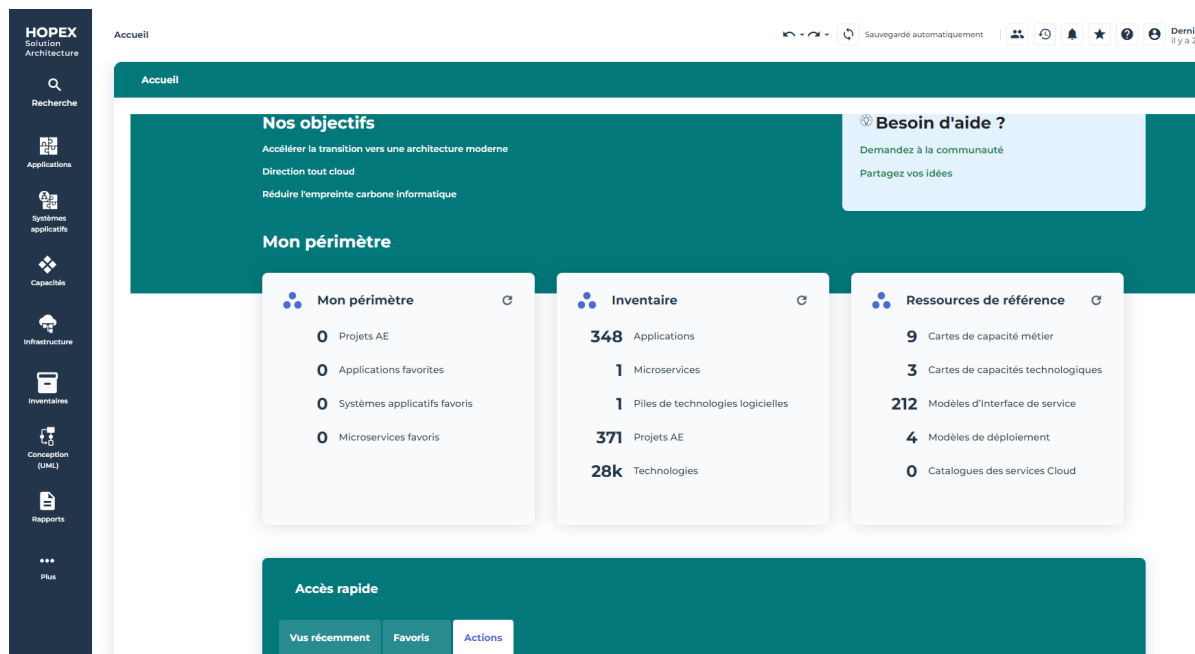
Les menus et commandes disponibles dans **HOPEX IT Architecture** dépendent de la licence dont vous disposez et du profil avec lequel vous êtes connecté.

☛ Pour plus de détails sur l'utilisation de l'interface des solutions HOPEX, voir le chapitre « Présentation de l'interface » du guide **HOPEX Common Features**.

L'ensemble des utilisateurs connectés au produit **HOPEX IT Architecture** disposent d'un même bureau et de l'accès à différents menus de navigation.

## Présentation de l'espace de l'Architecte solution

Les menus de navigation proposés aux utilisateurs connectés avec le profil **Architecte de solution** permettent d'accéder à l'ensemble des fonctionnalités offerte par **HOPEX IT Architecture**.



### Le menu Applications

Le menu **Applications** donne accès à l'ensemble des applications du référentiel.

➡ Pour plus de détails sur la description des applications, voir [Décrire une application avec HOPEX IT Architecture](#).

### Le menu Systèmes applicatifs



Le menu **Systèmes applicatifs** donne accès à l'ensemble des systèmes applicatifs du référentiel.

➡ Pour plus de détails sur la description des applications, voir [Décrire l'architecture système](#).



## Le menu Capacités

Le menu **Capacités** donne accès aux thèmes suivants.

- Thème **Capacité métier**, pour décrire les capacités métier ainsi que les cartes de capacités métier.  
 Pour plus de détails sur les architectures logiques, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).
- Thème **Fonctionnalités**, pour décrire les cartes de fonctionnalités du système d'information.  
 Pour plus de détails sur les fonctionnalités, voir [Utiliser les fonctionnalités avec HOPEX IT Architecture](#).


## Le menu Infrastructure

Le menu **Infrastructure** donne accès aux thèmes relatifs aux infrastructures.

-  Pour plus de détails sur les infrastructures, voir [Modéliser les infrastructures informatiques](#).
- Thème **Infrastructure informatique**, pour décrire les éléments d'une infrastructure informatique.  
 Pour plus de détails sur la description de ces infrastructures, voir [Décrire les infrastructures informatiques](#).
- Thème **Architecture de ressources**, pour décrire les éléments qui constituent une architecture de ressources.  
 Pour plus de détails sur les architectures de ressources, voir [Décrire les architectures de ressources](#).
- Thème **Configuration de ressource**, pour décrire les éléments techniques serveur et réseau.  
 Pour plus de détails sur les configurations de ressource, voir [Décrire une configuration de ressource](#).

## Le menu Inventaires

Le menu **Inventaires** donne accès aux thèmes suivants répartis sur plusieurs thèmes.

- Thème **Logiciel**, qui donne accès aux éléments suivants :
  - **Services applicatifs**,  
 Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.  
 Pour plus de détails sur les services applicatifs, voir [Décrire un service applicatif avec HOPEX IT Architecture](#).
  - **Microservices**,  
 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel

déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

☛ Pour plus de détails sur les microservices, voir [Décrire un microservice avec HOPEX IT Architecture](#).

- **Processus applicatifs,**

📖 Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.

☛ Pour plus de détails sur les processus applicatifs, voir [Décrire les processus applicatifs](#).

- **Catalogues de services métier,** pour décrire les services proposés par les métiers aux utilisateurs.

☛ Pour plus de détails sur les catalogues de services, voir [Utiliser les catalogues de services](#).

- **Systèmes de communication,**

📖 Un système de communication consiste à identifier et décrire les principaux systèmes d'intégration à partir des chaînes de communication applicative et des services de communication utilisés.

☛ Pour plus de détails sur les systèmes de communication, voir [Utiliser les systèmes de communication](#).

- **Architecture logicielle logique,** pour décrire les éléments de l'architecture logique du système d'information.

☛ Pour plus de détails sur l'architecture logique, voir [Décrire l'architecture logicielle logique](#).

- Thème **Déploiement** pour décrire les éléments liés au déploiement d'un système d'information.

- **Capacités Technologiques,**

📖 Une capacité technologique est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.

☛ Pour plus de détails sur l'utilisation des capacités technologiques, voir [Décrire une capacité technologique](#).

- **Technologies logicielles,** pour décrire les éléments techniques du système d'information.








☛ Pour plus de détails sur les technologies logicielles, voir [Décrire les technologies logicielles](#).

- **Catalogues de services Cloud.**

☛ Pour plus de détails sur les catalogues de services Cloud, voir [Utiliser les services cloud](#).

- **Catalogues de services technologiques.**

☛ Pour plus de détails sur les catalogues de services technologiques, voir [Utiliser les services cloud](#).

- Thème **Infrastructure** qui donne accès aux éléments suivants :
  - **Installations,**
    -  Une installation est un modèle de site d'intérêt pour l'entreprise. Par exemple : un centre de données, une usine ou une agence.
    -  Pour plus de détails sur les installations, voir [Décrire une installation](#).
  - **Réseaux informatiques,**
    -  Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur informatique). Un réseau informatique peut se décomposer en sous-réseaux.
    -  Pour plus de détails sur les réseaux informatiques, voir [Décrire un réseau informatique](#).
  - **Ordinateurs,**
    -  Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur informatique). Un réseau informatique peut se décomposer en sous-réseaux.
    -  Pour plus de détails sur les ordinateurs, voir [Décrire un ordinateur](#).
  - **Périphériques réseau,**
    -  Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur

informatique). Un réseau informatique peut se décomposer en sous-réseaux.

☛ Pour plus de détails sur les réseaux informatiques, voir [Décrire un périphérique réseau](#).

- Thème **Données**, pour décrire vos données métier.

☛ Pour plus de détails sur la gestion des données métier, voir [Gérer les données](#).

- **Dictionnaires métier**,

📖 Un dictionnaire métier rassemble et structure un ensemble de concepts qui expriment les connaissances d'un domaine particulier..

☛ Pour plus de détails sur les dictionnaires métier, voir le guide **HOPEX Data Governance**.

- **Dictionnaires de données**,

📖 Un dictionnaire de données vous permet de décrire tous les éléments qui définissent votre architecture de données logiques.

☛ Pour plus de détails sur les dictionnaires de données, voir le guide **HOPEX Data Governance**.

- **Catalogues de services d'information**, pour décrire les services proposés par le système d'information aux utilisateurs.

☛ Pour plus de détails sur les catalogues de services, voir [Utiliser les catalogues de services](#).

- Thème **Matériel** pour décrire les éléments matériels du système d'information.

- **Capacités matérielles**, pour décrire les fonctionnalités liées à des éléments matériels.

📖 Une capacité matérielle est la capacité à fournir un résultat attendu par une ressource matérielle afin de réaliser un travail. Cette fonctionnalité matérielle est généralement nécessaire à un système informatique pour exécuter une opération spécifique.

☛ Pour plus de détails sur l'utilisation des fonctionnalités, voir [Décrire une capacité matérielle](#).

- **Matériels**,

📖 Un matériel non-informatique peut embarquer des ordinateurs. De façon conjointe avec ses ordinateurs, il fournit les services d'information et les données. Exemples : camion connecté avec une application de calendrier de livraison, drone connecté avec application de paiement en ligne. Un matériel fournit des services matériels. Par exemple : un réfrigérateur connecté fournit des fonctionnalités de commandes et bien entendu une fonctionnalité matérielle de refroidissement, les drones connectés volent et fournissent une fonctionnalité de paiement en ligne.

☛ Pour plus de détails sur les réseaux informatiques, voir [Décrire un matériel](#).

- **Catalogues des services matériels**.

☛ Pour plus de détails sur les catalogues des services matériels, voir [Utiliser les services cloud](#).

- Thème **Toutes les ébauches**, pour accéder à l'ensemble des ébauches de votre référentiel.

☛ Pour plus de détails sur l'utilisation des diagrammes d'ébauche avec **HOPEX IT Architecture**, voir [Créer un diagramme d'ébauche avec HOPEX IT Architecture](#).

## Le menu Conception (UML)

Le menu **Conception (UML)** donne accès aux thèmes suivants :

- Thème **Implémentation OO (UML)**, pour modéliser votre SI en adoptant le formalisme UML.

☛ *Plus plus de détails sur l'utilisation des concepts UML, voir [A propos de l'implémentation UML](#).*

En fonction des options sélectionnées, deux thèmes sont également accessibles :

☛ *Pour visualiser ces deux thèmes, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Fonctionnalité de modélisation d'IHM et de traitements par lots (ADES)** est activée.*

- Thème **Mise en œuvre de traitements par lots et de programmes**,

☛ *Plus plus de détails, voir [Décrire les traitements par lots](#).*

- Thème **Interfaces utilisateur**.

☛ *Plus plus de détails sur la description des IHM, voir [Décrire les interfaces utilisateur](#).*

## Le menu Rapports

Le menu **Rapports** donne accès à l'ensemble des rapports proposés pour chacune des solutions.

☛ *Pour plus de détails sur l'utilisation de ces rapports, voir le chapitre "Générer des rapports d'analyse" dans le guide **HOPEX Common Features**.*

☛ *Pour plus de détails sur les rapports de **HOPEX IT Architecture**, voir [Les rapports HOPEX IT Architecture](#).*

## Le menu Gouvernance

Le menu **Gouvernance** donne accès aux thèmes suivants.

- Thème **Cadre de politique d'entreprise** donne accès aux cadres qui déterminent la politique de l'entreprise.

☛ *Pour plus de détails sur les cadres de politique d'entreprise, voir [Définir un cadre de politique d'entreprise avec HOPEX IT Architecture](#).*

☛ *Pour plus de détails, voir le guide **HOPEX Data Governance**.*

- Thème **Projets AE**, pour accéder aux fonctionnalités de gestion des projets.

☛ *Pour plus de détails sur la gestion des projets, voir la partie «Les projets d'architecture d'entreprise (AE) dans HOPEX» dans le guide **HOPEX Common Features**.*

- Thème **Plans d'action**, pour décrire et gérer les plans d'action liés à la transformation de votre SI.

☛ *Pour plus de détails sur la gestion des plans d'actions, voir «Utiliser les plans d'action» dans le guide **HOPEX Common Features**.*

## Le menu Environnement

Le menu **Environnement** donne accès aux thèmes suivants.

- Thème **Conteneurs**, pour accéder aux fonctionnalités de gestion des bibliothèques et des entreprises.

☛ Pour plus de détails sur les **Conteneurs** et l'**Organisation**, voir [Définir l'environnement de travail](#).

- Thème **Organisation**, pour accéder aux principaux objets manipulés avec la solution **HOPEX IT Architecture**.

- **Lignes métier,**

📖 Une ligne métier est un haut niveau de classification des principales activités de l'entreprise. Elle correspond, par exemple, à des grands segments produits ou à des canaux de distribution. Elle permet de classer les processus de l'entreprise, des unités organisationnelles ou des applications qui servent un produit spécifique et/ou un marché spécifique.

- **Catégories de processus,**

📖 Une catégorie de processus définit un ensemble de processus. Elle est reliée à une carte de processus ou à une catégorie de processus de niveau supérieur. Elle regroupe plusieurs processus et/ou éléments catégorisés (chaines de valeur, applications). Elle sert de niveau intermédiaire pour hiérarchiser les processus, afin d'accéder progressivement jusqu'au niveau le plus fin de granularité des processus.

- **Processus,**

📖 Un processus est un ensemble d'opérations réalisées par des acteurs d'une entreprise ou d'une organisation en vue de produire un résultat. Il est décrit comme une séquence d'opérations, contrôlée par des événements et des conditions. Dans la notation BPMN, un processus représente un sous-processus du point de vue de l'organisation.

- **Sites,**

📖 Un site est un lieu géographique où est implantée l'entreprise. Les sites peuvent être des sites-types tels que le siège, l'agence, l'usine, ou des lieux géographiques précis comme l'agence de Marseille, l'usine de Poissy, etc.

- **Acteurs.**

📖 Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur

externe représente un organisme qui échange des flux avec l'entreprise.  
Ex : Client, Fournisseur, Administration.

☛ Pour plus de détails sur l'utilisation des **Acteur**, voir [Utiliser les acteurs](#).

- Thème **Mes RFC**, pour traiter vos demandes de changement.
- Thème **Idées**, pour accéder aux fonctionnalités de gestion des bibliothèques et des environnements.  
Si vous avez la licence associée, **Idées** vous permet d'accéder aux facilités de gestion des idées.

☛ Pour plus de détails sur la gestion des idées de projets, voir la partie «Soumettre et évaluer des idées» dans le guide **HOPEX Common Features**.



- Thème **Commun**, pour accéder aux objets suivants :

- **Tags**

📖 Un tag est une description typologique qui permet de caractériser les objets.

☛ Pour plus d'informations sur l'utilisation des tags, voir PLATEFORME - Fonctionnalités communes > Outils collaboratifs > Communiquer dans HOPEX > Tags.

- **Report DataSets**

📖 Un Report DataSet est un ensemble de données extraites du référentiel HOPEX et utilisé comme source de données dans les rapports.

☛ Pour plus d'informations, voir PLATEFORME - Fonctionnalités communes > Documentation > Générer la documentation > Gérer les Report DataSets

- **Toutes les ébauches**, pour accéder à l'ensemble des ébauches de votre référentiel

📖 Une ébauche est un dessin qui vous permet d'échanger avec vos collaborateurs sans préoccupation méthodologique ou souci de respect d'un formalisme.

☛ Pour plus de détails sur l'utilisation des diagrammes d'ébauche avec **HOPEX IT Architecture**, voir [Créer un diagramme d'ébauche avec HOPEX IT Architecture](#).

## Le menu Administration

Le menu **Administration** donne accès aux thèmes suivants :

- Thème **Modèles**. qui donne accès aux modèles associés aux composants suivants :
  - **Interface de service**, voir [Utiliser un modèle d'interface de service](#).
  - **Opérations de service**, voir [Utiliser un modèle d'interface de service](#)
  - **Contenus**, voir [Utiliser un modèle d'interface de service](#)
  - **Architecture de déploiement**, voir [Les modèles d'architecture de déploiement](#).
- Thème **Graphes de Catégorisation**, voir [Définir les catégories de données](#).
- Thème **Domaines méthodologiques**, voir [Définir les domaines méthodologiques](#).

## Présentation des menus de l'espace de l'Administrateur fonctionnel d'architecture de solution

Les activités proposées uniquement aux utilisateurs connectés avec le profil **Administrateur fonctionnel d'architecture de solution** concernent :

- l'administration via le menu **Administration**,
- la création d'objets à partir du menu **Environnement**, voir [Le menu Environnement](#).

## Présentation de l'espace du concepteur d'applications

Les menus de navigation proposés aux utilisateurs connectés avec le profil **Concepteur d'applications** sont identiques aux menus de navigation proposés utilisateurs connectés avec le profil **Architecte de solution**.

Toutefois, les utilisateurs connectés avec le profil **Concepteur d'applications** peuvent modifier les propriétés des objets mais ne peuvent pas modifier les diagrammes.

## Présentation de l'espace de l'observateur d'applications

Les menus de navigation proposés aux utilisateurs connectés avec le profil **Observateur d'applications** sont identiques aux menus de navigation proposés utilisateurs connectés avec le profil **Architecte de solution**.

Toutefois, les utilisateurs connectés avec le profil **Observateur d'applications** n'ont accès aux objets qu'en consultation.

---

## Changer de profil

A partir du bureau de **HOPEX IT Architecture**, vous pouvez passer à un bureau spécifique de l'une des solutions **HOPEX** dont vous disposez, sans vous déconnecter, en changeant de profil.

Par exemple, pour accéder à un profil spécifique :

1. Cliquez sur le **Menu principal > Changer de profil**.
2. Sélectionnez le profil avec lequel vous voulez vous connecter.
3. (Si vous avez effectué des modifications dans votre espace de travail privé) Cliquez sur :
  - **Oui**, pour enregistrer vos modifications dans le référentiel.
  - **Non**, si vous ne voulez pas enregistrer dans le référentiel les modifications que vous avez effectuées depuis votre dernière publication. Vos modifications apportées à votre bureau sont aussi perdues.

Le bureau du profil concerné s'affiche.


➡ Cliquez sur **Annuler** pour revenir à votre travail.




# AVANT DE COMMENCER AVEC HOPEX IT ARCHITECTURE


## Définir l'environnement de travail

Dans le contexte de la solution **HOPEX IT Architecture**, une **bibliothèque** peut détenir l'ensemble des éléments de votre projet : des processus et des acteurs, par exemple.

 Une bibliothèque est un regroupement d'objets qui permet de découper le contenu d'un référentiel en plusieurs parties indépendantes. Les bibliothèques permettent de réaliser des partitions virtuelles du référentiel. En particulier, deux objets appartenant à des bibliothèques différentes peuvent avoir le même nom.

L'utilisation d'une **entreprise** permet de préparer un projet de transformation.

 Une entreprise a pour finalité la mise à disposition de biens et de services en adéquation avec la mission que lui ont assignée ses fondateurs et principales parties prenantes (entrepreneurs, actionnaires, dirigeants, responsables opérationnels, architectes). Au cours de son développement, l'entreprise doit s'adapter à son environnement et établit des buts et des objectifs de transformation à atteindre, ainsi que les plans d'action stratégiques visant à l'accomplissement de ces objectifs. La conception et la mise en place des différentes phases d'adaptation et de transformation peut conduire à modifier les frontières des entités juridiques concernées. Ceci nécessite la mise en place d'une équipe intégrée, sous la responsabilité d'un organe de gouvernance, afin d'impliquer les principales parties prenantes de la transformation.

 Pour plus de détails sur l'utilisation des conteneurs, voir le chapitre "Entreprises et bibliothèques" du guide **HOPEX Common Features**.

## Accéder à la liste des bibliothèques avec HOPEX IT Architecture

Pour accéder à la liste des bibliothèques à partir du menu de navigation **Environnement** :

- Sélectionnez **Conteneurs > Bibliothèques**.  
L'arbre des bibliothèques s'affiche.

## Accéder à la liste des entreprises avec HOPEX IT Architecture

Pour accéder à la liste des entreprises à partir du menu de navigation **Environnement** :

- Sélectionnez **Conteneurs > Entreprises**.  
La liste des entreprises s'affiche.

Dans le contexte de la solution **HOPEX IT Architecture**, les caractéristiques des entreprises sont simplifiées par rapport à la solution **HOPEX IT Business Management**.

 Pour plus de détails sur l'utilisation des entreprises dans le cadre d'une transformation, voir le chapitre "Les éléments stratégiques de la phase de transformation" du guide **HOPEX IT Business Management**.



---

## Utiliser les acteurs

La notion d'**acteur** est utilisée dans plusieurs diagrammes. Les principaux points concernant ce type d'objet sont rappelés dans cette section.



*Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.*

### Créer un acteur

Pour créer un acteur à partir du menu de navigation **Environnement** :

1. Sélectionnez **Organisations > Acteurs**.  
La liste des acteurs s'affiche en ordre alphabétique.
2. Sélectionnez le dossier **Acteurs** et cliquez sur le bouton **Nouveau**.
3. Dans la fenêtre de choix, sélectionnez **Acteurs**.
4. Dans la fenêtre de **Création d'un Acteur**, saisissez le nom de l'acteur que vous souhaitez créer.
5. Cliquez sur **OK**.  
L'acteur apparaît dans la liste.

### Acteur interne / externe

Lors de leur création, par défaut les acteurs sont considérés comme des éléments internes à l'entreprise.

Pour spécifier qu'un acteur ne fait pas partie de l'entreprise, il faut modifier les propriétés de cet acteur et lui attribuer le statut "Acteur Externe".

Pour attribuer à l'acteur la caractéristique "Externe" :

1. Faites un clic droit sur l'acteur et sélectionnez **Propriétés**.
2. Ouvrez la page de propriétés **Caractéristiques** d'un acteur.
3. Dans la section **Identification**, sélectionnez dans le champ **Acteur externe**.

Cette caractéristique est représentée graphiquement et s'applique automatiquement dans les diagrammes.

---

## Utiliser les diagrammes d'architecture informatique

➡ Pour plus de détails sur l'utilisation des diagrammes, voir «Utiliser les diagrammes» dans le guide **HOPEX Common Features**.

Avec **HOPEX IT Architecture**, un objet applicatif peut être décrit par différents diagrammes. Chaque type de diagramme présente un aspect spécifique de l'objet : architecture interne, architecture de déploiement, flux échangés à l'intérieur de l'objet et flux échangés à l'extérieur en fonction des contextes d'utilisation de

l'objet. En fonction de l'objet décrit, chacune de ces représentation est associée à type de diagramme.

- les **flux de données externes** sont représentés dans un *diagramme d'environnement*. Ce type de diagramme contient l'objet applicatif concerné et les flux applicatifs échangés avec des partenaires (autres systèmes applicatifs, applications, dépôts de données, acteurs ou types de postes).
- l'**architecture interne** est décrite par un *diagramme de structure* qui présente les composants de l'objet et leurs échanges. Un *diagramme de structure* peut être construit pour une application, un environnement d'application, un système applicatif, un environnement de système applicatif, un système applicatif logique, un service applicatif ou un microservice.
- L'**architecture de déploiement** est décrite par un diagramme qui présente les *modules de déploiement d'une application*, les *microservices* et les *modules de déploiement des données* utilisés ainsi que les techniques de communication nécessaires.
- Les **flux de données internes** sont représentés dans un *diagramme de scénario de flux* qui décrit les messages échangés entre les composants de l'objet. Avec **HOPEX IT Architecture**, deux types de *diagramme de scénario de flux* sont proposés :
  - les *diagrammes de scénario de flux* qui décrivent les flux échangés dans différents scénario d'utilisation de l'objet décrit.
  - les *diagrammes de scénario de séquences de flux* qui décrivent la chronologie des flux échangés dans différents scénarios d'utilisation de l'objet décrit.

☛ Pour visualiser les diagrammes de scénario de séquence de flux, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Activer les diagrammes de scénario de séquence de flux** est activée.

- Les **interactions externes** sont représentés dans un *diagramme de scénario de flux* qui décrit les interactions de service externes d'un objet applicatif dans un environnement donné. Ce diagramme contient l'objet décrit et ses interactions de service avec des partenaires (autres systèmes).

Représentation	Liste des diagrammes concernés
Architecture interne	<ul style="list-style-type: none"> <li>• Diagramme de structure d'application</li> <li>• Architecture</li> <li>• Diagramme de structure de système applicatif</li> <li>• Diagramme de structure de service applicatif</li> <li>• Diagramme de structure de microservice</li> <li>• Diagramme de module de déploiement d'une application</li> <li>• Diagramme de structure de système applicatif</li> <li>• Diagramme de structure d'application logique</li> <li>• Diagramme de structure de système applicatif logique</li> <li>• Diagramme de module de déploiement d'une application</li> <li>• Diagramme d'assemblage de ressources</li> <li>• Diagramme d'architecture d'un matériel</li> <li>• Diagramme d'architecture d'un objet connecté</li> </ul>
Flux internes de données	<ul style="list-style-type: none"> <li>• Diagramme de scénario de flux d'application</li> <li>• Diagramme de scénario de flux de système applicatif</li> <li>• Diagramme de scénario de flux de service applicatif</li> <li>• Diagramme de scénario de flux de microservice</li> <li>• Diagramme de scénario de flux de système applicatif logique</li> <li>• Diagramme de scénario de séquence d'application (UML)</li> <li>• Diagramme de scénario de séquence de flux de système applicatif (UML)</li> </ul>
Flux externes de données	<ul style="list-style-type: none"> <li>• Diagramme de scénario de flux d'environnement d'application</li> <li>• Diagramme de scénario de flux d'environnement de système applicatif</li> <li>• Diagramme de scénario de flux d'environnement de système applicatif logique</li> <li>• Diagramme de scénario de séquence d'environnement d'application (UML)</li> <li>• Diagramme de scénario de séquence de flux de système applicatif (UML)</li> </ul>
Interactions externes	<ul style="list-style-type: none"> <li>• Diagramme d'environnement d'application</li> <li>• Diagramme d'environnement de système applicatif</li> <li>• Diagramme d'environnement de système applicatif logique</li> <li>• Diagramme d'environnement d'architecture de ressource</li> </ul>
Architecture de déploiement	<ul style="list-style-type: none"> <li>• Diagramme d'architecture de déploiement d'une application</li> <li>• Diagramme d'architecture de déploiement d'un système applicatif</li> <li>• Diagramme d'architecture de déploiement d'un microservice</li> </ul>

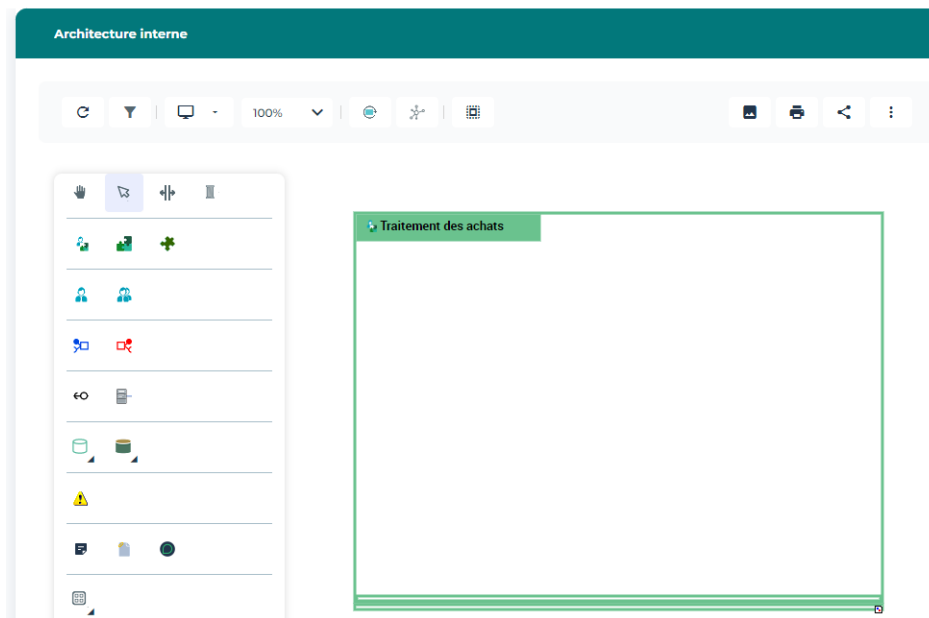
## Créer un diagramme de structure

Pour créer un *diagramme de structure de système applicatif*, par exemple :

1. Ouvrez la page **Diagrammes** du système applicatif qui vous intéresse et cliquez sur le bouton **Créer un diagramme**.
2. Dans la fenêtre de choix, sélectionnez l'un des types de **Diagramme Structuré > Architecture interne**.

Le diagramme s'ouvre dans la zone d'édition. Vous êtes dans l'éditeur graphique de **HOPEX**. Le cadre de l'objet décrit apparaît dans le diagramme.

### Exemple d'un nouveau diagramme de structure



Par défaut, le diagramme est initialisé avec l'objet décrit, représenté par un cadre.




*➡ Dans le diagramme ci-dessus, l'objet décrit est un système applicatif.*

Les composants de l'objet décrit sont positionnés en haut du diagramme.

Lorsque l'objet décrit est déjà utilisé dans un diagramme de niveau supérieur, le nouveau diagramme est initialisé en prenant en compte des points de requête et de service associés aux flux présents dans les diagrammes existants.

## Les commandes des diagrammes de HOPEX IT Architecture

En fonction de leur type, les diagrammes de **HOPEX IT Architecture** disposent de commandes spécifiques.

Icône	Description
	<b>Rafraîchir les canaux</b> Permet de mettre à jour le contenu des canaux décrits dans un scénario de flux. Voir <a href="#">Créer un canal de flux applicatifs</a> .
	<b>Réinitialiser les composants</b> Ajoute, dans le diagramme, l'ensemble des composants de premier niveau de l'objet décrit.
	<b>Positionnement automatique</b> Permet de disposer automatiquement les composants ajoutés au diagramme. Voir <a href="#">Positionnement automatique dans les diagrammes d'architecture</a> .
	<b>Ajouter des éléments</b> Permet de compléter le diagramme courant avec des éléments déjà définis par ailleurs. Voir <a href="#">Initialisation d'un diagramme d'environnement</a> . Disponible uniquement dans les diagrammes d'environnement d'application et de système applicatif.

## Positionnement automatique dans les diagrammes d'architecture

Lorsque l'environnement détient des composants avec des interactions de service entre eux, chaque nouveau diagramme est initialisé en prenant en compte l'ensemble de ces éléments qui apparaissent en haut et à gauche du cadre de l'environnement décrit.

Le bouton **Positionnement automatique** vous permet de réorganiser les éléments du diagramme en prenant en compte les échanges de flux.

Le **coefficient de compression/dilatation** vous permet de spécifier la distance à prévoir entre les éléments.


Si vous avez coché la case **Positionnement automatique** vous perdez la présentation existante de votre diagramme.


Le positionnement automatique est proposé pour les diagrammes suivants :

- Environnement d'application
  - Diagramme d'environnement d'application
  - Diagramme de scénario de flux d'environnement d'application
- Application
  - Diagramme de scénario de flux d'application
  - Diagramme de structure d'application
  - Diagramme d'environnement de déploiement d'une application
- Environnement de système applicatif
  - Scénario de flux d'environnement de système applicatif
  - Diagramme d'environnement de système applicatif
- Système applicatif
  - Scénario de flux de système applicatif
  - Diagramme de structure de système applicatif
- Service applicatif
  - Diagramme de scénario de flux de service applicatif
  - Diagramme de structure de service applicatif
- Application logique
  - Diagramme de scénario de flux d'application logique
  - Diagramme de structure d'application logique
  - Diagramme d'environnement de déploiement d'une application logique
- Environnement de Système applicatif logique
  - Diagramme d'environnement de système applicatif logique
  - Scénario d'environnement de système applicatif logique
- Système applicatif logique
  - Diagramme de structure de système applicatif logique
  - Scénario de système applicatif logique
- Environnement d'architecture de ressources
  - Diagramme d'environnement d'architecture de ressources

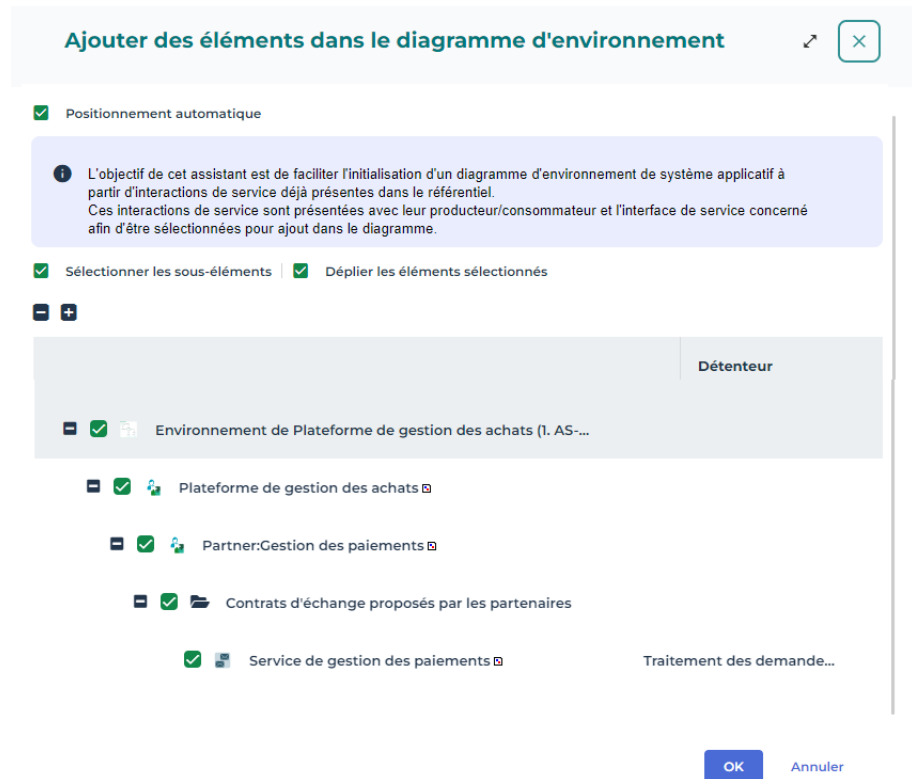
## Initialisation d'un diagramme d'environnement

Par définition, un diagramme d'environnement représente un contexte d'utilisation d'une application ou d'un système applicatif.

 *Un environnement d'application présente un contexte d'utilisation d'une application. Il permet de représenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels cette application est susceptible d'interagir.*

 *Un environnement de système applicatif permet de présenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels ce système applicatif est susceptible d'interagir.*

Afin de faciliter la description d'un contexte spécifique d'utilisation d'un système applicatif, par exemple, le bouton **Ajouter des éléments** vous permet de dresser la liste des composants avec lesquels le système applicatif interagit et de sélectionner ceux que vous souhaitez représenter dans votre diagramme d'environnement.



Les boutons **Sélection des sous-éléments** et **Déplier les éléments sélectionnés** vous assistent dans votre sélection.

## Créer un diagramme d'ébauche avec HOPEX IT Architecture

Une ébauche est un dessin qui vous permet d'échanger avec vos collaborateurs sans préoccupation méthodologique ou souci de respect d'un formalisme.

Les diagrammes d'ébauche pourront ensuite être retravaillés et transformés pour aboutir à des diagrammes reconnus par une solution **HOPEX**.

➡ Pour plus de détails sur l'utilisation des ébauches, voir «Ebauches» dans le guide **HOPEX Common Features**.

Pour créer un **diagramme d'ébauche** pour une application, par exemple, avec **HOPEX IT Architecture** :

1. A partir du menu de navigation **Applications**, sélectionnez l'application qui vous intéresse et cliquez sur le bouton **Créer diagramme**.
2. Dans la fenêtre de choix, sélectionnez **Diagramme d'ébauche**. Le diagramme s'ouvre dans la fenêtre d'édition. Vous êtes dans l'éditeur graphique de **HOPEX**.



## Créer un diagramme ArchiMate® avec HOPEX IT Architecture

Le produit **HOPEX for the ArchiMate® Framework** offre la possibilité d'utiliser l'ensemble des concepts standards définis par l'Open Group pour ArchiMate®. Afin de gérer une compatibilité et une continuité des modèles, les concepts ArchiMate® sont mis en correspondance avec les briques dédiées à l'Architecture d'Entreprise disponibles dans de **HOPEX**.

Vous pouvez ainsi associer à une application, par exemple, un diagramme basé sur le formalisme ArchiMate.

☛ Pour plus de détails, voir «Utiliser les diagrammes ArchiMate dans une solution d'Architecture d'Entreprise» dans le guide **HOPEX IT Business Management**.

## Utiliser la comparaison de diagramme

La comparaison des diagrammes de système applicatif ou d'architecture de déploiement de système applicatif ou d'application permet de comparer différentes versions d'un même objet.

☛ Pour plus de détails sur l'utilisation générale des fonctionnalités de comparaison de diagramme, voir «Comparer les diagrammes» dans le guide **HOPEX Common Features**.



---

## Contenu d'une page de propriétés HOPEX IT Architecture

**HOPEX IT Architecture** présente des pages de propriétés spécifiques pour chaque type d'objet.

☛ Pour plus de détails sur l'accès aux pages de propriétés, voir le chapitre "Gérer les priorités des objets" dans le guide **HOPEX Common Features**.

☛ En utilisant les fonctionnalités décrites dans le guide **HOPEX Power Studio**, vous pouvez personnaliser les pages de propriétés de votre solution.

Les pages ci-dessous sont communes aux principaux objets de **HOPEX IT Architecture**.

- la page **Composants** permet d'accéder à la liste des composants de l'objet décrit définis dans les différents types de diagramme.
  - la section **Dépôts de données** permet d'accéder zones de stockage de données spécifiques à l'objet décrit.
    - ☛ Pour plus de détails sur les dépôts de données, voir [Gérer les données](#).
  - la section **Composants internes** permet d'accéder à la liste des composants de l'objet décrit définis dans ses scénario de flux.
  - la section **Composants frontières** permet d'accéder à la liste des composants de l'objet décrit définis dans ses diagrammes de structure.
    - ☛ Pour plus de détails sur les composants d'un objets applicatif, voir [Le diagramme de structure d'application](#).
  - la page **Scénarios de flux** permet d'accéder à l'ensemble des représentations des flux internes de l'objet décrit.
    - ☛ Pour plus de détails sur les scénario de flux, voir [Décrire les flux de données](#).
  - la page **Environnements** permet d'accéder à l'ensemble des contextes d'utilisation de l'objet décrit.
    - ☛ Pour plus de détails sur les environnements, voir [Décrire un environnement d'application avec HOPEX IT Architecture](#).
  - la page **Architecture de déploiement** permet d'accéder à la liste des composants nécessaire au déploiement de l'objet applicatif.
    - ☛ Pour plus de détails sur les composants d'un diagramme d'architecture de déploiement, voir [Utiliser un diagramme d'architecture de déploiement d'une application](#).
  - la page **Usage** qui permet d'accéder aux informations relatives aux contextes d'utilisation de l'objet décrit dans différents types de diagrammes.
    - la section **Scénario** permet d'accéder à la liste des éléments qui utilisent l'objet décrit dans leurs scénarios de flux.
      - ☛ Pour plus de détails sur les composants d'un scénario de flux, voir [Utiliser un diagramme de scénario de flux d'application](#).
    - la section **Structure** permet d'accéder à la liste des éléments qui utilisent l'objet décrit dans leur environnement.
      - ☛ Pour plus de détails sur les composants d'un diagramme de structure d'application, voir [Le diagramme de structure d'application](#).
  - la page **Processus exécuté** permet d'accéder aux processus applicatifs exécutés par l'objet décrit.
    - ☛ Pour plus de détails sur les processus applicatifs, voir [Décrire les processus applicatifs](#).
  - La page **Gouvernance** permet de saisir les **Décisions** d'architecture relatives à l'application.
    - ☛ Pour plus d'informations sur les décisions, voir le chapitre [Dresser l'inventaire des applications > Enregistrer les décisions d'architecture dans le guide HOPEX IT Portfolio Management](#).
- Elle donne également accès à la liste des **Politiques d'entreprise** auxquelles l'application doit se conformer.

➤ Pour plus de détails sur la prise en compte des politiques d'entreprise avec HOPEX IT Architecture, voir [Définir un cadre de politique d'entreprise avec HOPEX IT Architecture](#).

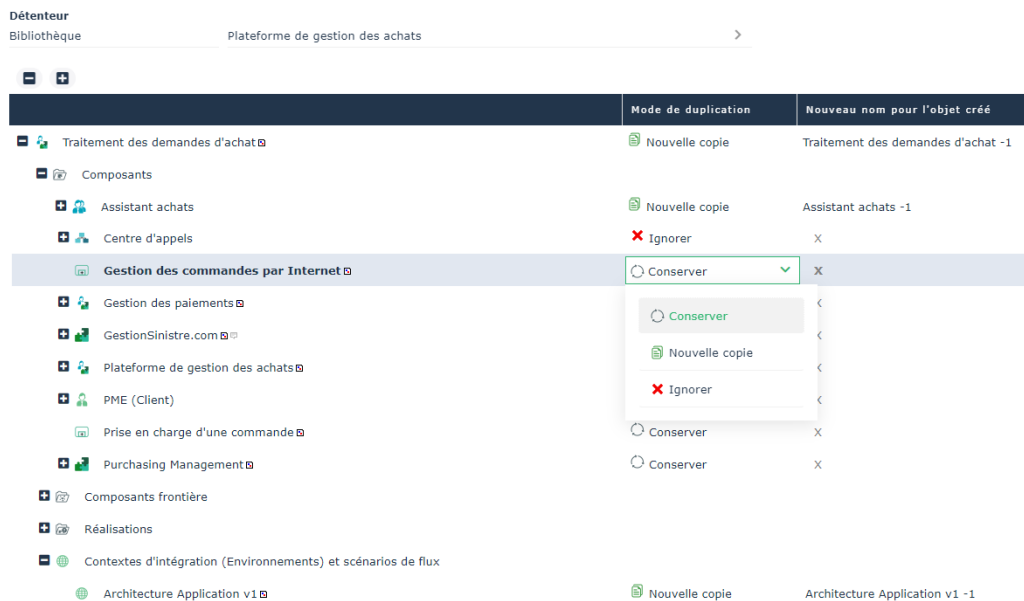
- ➡ Pour plus de détails sur les rapports disponibles avec **HOPEX IT Architecture**, voir [Les rapports HOPEX IT Architecture](#).

## Utiliser la duplication avec HOPEX IT Architecture

La solution **HOPEX IT Architecture** intègre des facilités de duplication de briques spécifiques telles que les applications, services applicatifs ou architecture de déploiement.

Pour dupliquer un système applicatif, par exemple :

1. Faites un clic droit sur le système applicatif que vous souhaitez dupliquer pour faire apparaître son menu contextuel.
2. Sélectionnez **Administration > Dupliquer**.  
Une fenêtre s'ouvre pour vous assister dans cette duplication en vous présentant la liste des composants de l'objet dupliqué.



Fenêtre de duplication avec **HOPEX IT Architecture**

3. Sélectionnez le **Détenteur** de la nouvelle brique.

4. Pour chaque composant, la colonne **Mode de duplication** vous propose les choix suivants :
  - Créer une **Nouvelle copie** du composant sélectionné. Dans ce cas, le nom du nouveau composant apparaît dans la colonne **Nouveau nom pour l'objet créé**.
  - **Conserver** le composant, qui sera ainsi partagé entre les briques sources et cibles.
  - **Ignorer** le composant qui ne sera ni dupliqué, ni référencé.
5. Pour valider vos choix, cliquez sur **Lancer la duplication**.  
La brique créée est accessible à partir de son détenteur.

## Utiliser la duplication de HOPEX IT Architecture en batch

Il est possible de dupliquer en batch une brique ainsi que les objets qui lui sont connectés tels que définis dans le MetaModel.

La fonction **SmartDuplicate** renvoie la référence vers la nouvelle brique créée.

Les paramètres de la fonction **SmartDuplicate** sont définis dans la chaîne **sOptions** de la manière suivante :

- Fonction **SmartDuplicate**(ByVal **sOptions** As String) As MegaObject
- **sOptions** est défini avec le format "K1=V1,K2=V2, ...", les valeurs proposées sont :
  - **Root**=[NewCopy] - Par défaut : NewCopy  
**Root** se rapporte à la brique dupliquée
  - **Components**=[Keep|NewCopy|Ignore] - Par défaut : Keep  
**Components** concernent les composants de la brique qui est dupliquée.
  - **Boundaries**=[Keep|Ignore] - Par défaut : Keep  
**Boundaries** concernent les éléments à la frontière de la brique qui est dupliquée.
  - **Scenarios**=[Keep|Ignore] - Par défaut : Keep  
**Scenarios** concernent les scénarios de flux de la brique qui est dupliquée.
  - **Environments**=[Keep|Ignore] - Par défaut : Keep  
**Environments** concernent les environnements de la brique qui est dupliquée.


### Exemple d'utilisation :


```
Set newObject=  
myApplication.SmartDuplicate("Root=NewCopy,Components=Keep,  
Environments=Ignore")
```

## Utiliser les catalogues de services


### La mise œuvre d'un catalogue de service

Dans **HOPEX IT Architecture**, un catalogue de services est composé d'éléments de catalogues de service. Par exemple un *catalogues de services matériels* est composés de plusieurs *éléments de catalogue de services matériels*.


 Un catalogue de services matériels fournit une source d'information centralisée sur les services matériels proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services matériels utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services matériels présente la liste des fonctionnalités matérielles référencées ainsi que les recommandations de mise en œuvre.

 Un élément de catalogue de services matériels définit quelle capacité matérielle est au catalogue et quels sont les artefacts matériels qui fournissent la capacité matérielle.

Un *élément de catalogues de services* est relié à une ou plusieurs *fonctionnalités*.

 Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.

Avec **HOPEX IT Architecture**, la mise en œuvre d'une fonctionnalité est représentée par une *mise en œuvre*.

 Une mise en œuvre décrit une relation entre une entité logique et l'entité physique qui la met en œuvre. L'entité physique donne la liste des entités logiques qu'elle implémente.

Le tableau ci-dessous dresse la synthèse des objets qui mettent en œuvre les catalogues de service en fonction de leur catégorie.

Type de catalogue de services	Type de fonctionnalité	Types d'objets qui assurent la mise en œuvre du service
métier	Fonctionnalités	Capacité métier.
Information	Fonctionnalités	Tous les types d'objets techniques et fonctionnels qui mettent en œuvre une fonctionnalité avec <b>HOPEX IT Architecture</b> . Pour plus de détails, voir <a href="#">Décrire une carte de fonctionnalités avec HOPEX IT Architecture</a> .
Cloud	Capacités technologiques	Acteur.
Technique	Capacités technologiques	Technologies
matériel	Capacités matérielles	Matériel et objet connecté.

## Renseigner un catalogue de services

Le principe de gestion d'un catalogue de services est identique pour tous les types de catalogue. Les types de catalogue de service proposés sont :

- les *catalogues de services métier*,



Un catalogue de services métier fournit une source d'information centralisée sur les services matériels proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services associés aux capacités métier, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services métier présente la liste des fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services Cloud*,



Un catalogue des services d'information fournit une source d'information centralisée sur les services d'information proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services d'information utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services d'information présente la liste des fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services technologiques*,



Un catalogue de services technologiques fournit une source d'information centralisée sur les services technologiques proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services technologiques utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services technologiques présente la liste des fonctionnalités IT référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services d'information*.



Un catalogue des services d'information fournit une source d'information centralisée sur les services d'information proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services d'information utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services d'information présente la liste des fonctionnalités référencées ainsi que les recommandations de mise en œuvre.

- les *catalogues de services matériels*.




Un catalogue de services matériels fournit une source d'information centralisée sur les services matériels proposés par l'organisation fournisseur de services. Il contient une vue orientée client des services matériels utilisés, de la manière dont ils sont supposés être utilisés, des processus qu'ils supportent ainsi que le niveau de qualité de service attendu. Le catalogue de services matériels présente la liste des fonctionnalités matérielles référencées ainsi que les recommandations de mise en œuvre.

Ce chapitre s'appuie sur l'exemple d'un *catalogue de services d'information*.

## Créer un catalogue de services technologiques

Pour créer un *catalogue de services technologiques* :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Déploiements > Catalogues de Services Technologiques**.  
La listes des catalogues des services technologiques apparaît.

2. Cliquez sur **Nouveau**.  
Un nouveau catalogue des services d'information apparaît dans la liste.
3. Saisissez le **Nom** de votre catalogue et son **Détenteur**.  
 De manière identique, vous pouvez créer un *catalogue de services technologiques* ou un *catalogue de services matériels*.

## Ajouter un élément de catalogue de services

La page **Caractéristiques** des propriétés d'un catalogue de services permet d'accéder à :


- son **Détenteur**, par défaut lors de la création du catalogue de services, il s'agit de la bibliothèque courante.
- son **Nom**,
- la liste des éléments de catalogue de services détenus.

Pour ajouter un *élément de catalogues de services technologiques* :

1. Ouvrez la page **Caractéristiques** de propriétés d'un catalogue de services technologiques.
2. Dans la section **Éléments de catalogue technique détenu**, cliquez sur **Nouveau**.  
Une fenêtre vous permet de sélectionner une capacité technologique existante ou d'en créer une nouvelle.
3. Cliquez sur **OK**.  
L'élément de catalogue apparaît dans la liste avec le nom de la capacité technologique associée.


## Spécifier la mise en oeuvre d'un élément de catalogue de services

Pour décrire comment la fonctionnalité (ou la capacité) associée à un élément de catalogue de services est mise en oeuvre, vous allez définir une *Réalisation*.

 Une mise en oeuvre décrit une relation entre une entité logique et l'entité physique qui la met en oeuvre. L'entité physique donne la liste des entités logiques qu'elle implémente.

Pour décrire qu'un élément de catalogue de services est mis en oeuvre :

1. Ouvrez la page de propriétés **Caractéristiques** de l'élément de catalogue de services.
2. Dans la section **Élément de catalogue des services technologiques**, cliquez sur **Nouveau**.  
Une fenêtre de création de mise en oeuvre s'ouvre.
3. Sélectionnez le type d'objet et l'équipement qui vous intéressent.
4. Cliquez sur **Nouveau**.  
La mise en oeuvre de la capacité technologique apparaît dans la page de propriétés.

 Pour plus de détails la création d'une mise en oeuvre, voir *Utiliser les mécanismes de mise en oeuvre*.

## Les rapports de catalogues de service

La **Matrice de couverture des services** est un rapport permet de connaître, pour une liste de catalogues de services, la liste des fonctionnalités couvertes et, pour chacune d'elles, les moyens mis en oeuvre.

## Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Liste des catalogues de services	Tous les types de catalogues de services : <ul style="list-style-type: none"><li>- information,</li><li>- technique,</li><li>- matériel.</li></ul>	Un catalogue obligatoire.

## Accéder à un rapport de catalogue services



Pour accéder à un rapport de catalogue de services :

1. Ouvrez la page de propriétés **Rapports** d'un catalogue de services d'information.
2. Dans la section **Paramètres**, sélectionnez les catalogues de services que vous voulez présenter dans votre rapport.
3. Cliquez sur **Rafraîchir le rapport**.  
Le nouveau rapport apparaît.

---

## Utiliser les workflows

Avec **HOPEX IT Architecture** vous pouvez utiliser les workflows standards pour gérer :

- Les demandes de validation ;  
 Pour plus de détails sur le workflow de demande de validation, voir le chapitre "Utiliser les demandes de validation" du guide **HOPEX Common Features**.
- Les demandes de revue.  
 Pour plus de détails sur le workflow de demande de revue, voir [Workflow de validation de processus](#).

---

## Définir un cadre de politique d'entreprise avec HOPEX IT Architecture



Une politique d'entreprise est une directive dont le but est de gouverner ou de guider l'entreprise.



Un cadre de politique d'entreprise définit un ensemble de politiques d'entreprise. Il est composé de sections et de sous-sections qui représentent des catégories de politique entreprise. Sous ces sections vous pouvez définir les politiques d'entreprise, les actifs contraints par les politiques en question ainsi que leur mise en œuvre.



Pour plus de détails sur les cadres de politique d'entreprise, voir le guide **HOPEX Data Governance**.



Il est possible d'importer dans **HOPEX** des *cadres de politique d'entreprise*.

☛ Ces cadres de politique d'entreprise sont mis à disposition par votre administrateur dans un module. Pour plus de détails, voir le chapitre "Importer un module dans **HOPEX**" du guide **HOPEX Administration**.

## Définir une politique d'entreprise avec HOPEX IT Architecture

📖 Une politique d'entreprise est une directive dont le but est de gouverner ou de guider l'entreprise.

### Accéder aux politiques d'entreprise avec HOPEX IT Architecture

Pour accéder à la liste des *Politiques d'entreprise* :

1. A partir du menu de navigation **Gouvernance**, sélectionnez **Cadre de politique d'entreprise**.  
L'arbre des *Cadres de politiques d'entreprise* apparaît.
2. Dépliez les dossiers pour voir les *Politiques d'entreprise*.

### Créer une politique d'entreprise

Pour créer une *Politique d'entreprise* :

1. A partir du menu de navigation **Gouvernance**, sélectionnez **Cadre de politique d'entreprise**.  
L'arbre des *Cadres de politique d'entreprise* apparaît.
2. Dépliez le dossier du *Cadre de politique d'entreprise* qui vous intéresse.
3. Sélectionnez la *Section de cadre de politique d'entreprise* qui vous intéresse et cliquez sur **Nouveau > Politique d'entreprise**.  
La nouvelle *Politique d'entreprise* apparaît dans l'arbre.

### Relier une politique d'entreprise à une application

Pour identifier les *politiques d'entreprise* auxquelles une application doit se conformer :

1. Ouvrez la page de propriétés **Gouvernance**, de l'application qui vous intéresse.
2. Dans la section **Politiques à respecter**, cliquez sur **Relier**.  
Une fenêtre présente sous forme d'arbre, la liste des cadres de politiques d'entreprises ainsi que les politiques d'entreprise associées.
3. Sélectionnez les politiques d'entreprise qui vous intéressent et cliquez sur **Relier**.  
Les politiques d'entreprise sélectionnées apparaissent dans la section **Politiques à respecter**.

### Evaluer la conformité d'une application à une politique d'entreprise

Le résultat de la dernière évaluation du conformité d'une application est représenté par une vignette en haut de la page de propriétés **Gouvernance** de l'application.

Les résultats possibles sont les suivants :

- **Conforme** : si toutes les politiques d'entreprise reliées à l'application sont conformes.
- **Non conforme** : si toutes les politiques d'entreprise reliées à l'application sont non conformes.
- **Conformité partielle** : si toutes les politiques d'entreprise reliées à l'application ne sont pas évaluées au même niveau de conformité.

Pour évaluer le niveau de conformité d'une application à une *politique d'entreprise* :

1. Ouvrez la page de propriétés **Gouvernance**, de l'application qui vous intéresse.
2. Dans la section **Politiques à respecter**, sélectionnez les politiques d'entreprise que vous souhaitez évaluer.
3. Dans la section **Evaluation de la conformité aux politiques**, cliquez sur **Evaluer**.  
Une fenêtre **Evaluation de la gouvernance** présente sous forme d'arbre, la liste des cadres de politiques d'entreprises et des politiques d'entreprise à évaluer.
4. Renseignez la **Date de l'évaluation**.
5. Dans la colonne **Conformité**, sélectionnez le niveau de conformité que vous souhaitez attribuer à l'application.
6. Cliquez sur **OK**.  
Le niveau de conformité apparaît en haut de la page de propriétés.

## Définir les principes d'architecture



*Un principe d'architecture est une directive générale qui informe, soutient et contraint les organisations dans la manière de concevoir et construire des architectures.*

### Accéder aux principes d'architecture

Pour accéder à la liste des *Principes d'architecture* :

1. A partir du menu de navigation **Gouvernance**, sélectionnez **Cadre de politique d'entreprise**.  
L'arbre des *Cadres de politiques d'entreprise* apparaît.
2. Dépliez les dossiers pour voir les *Catégories de principe d'architecture*.



*Une catégorie de principe d'architecture permet de définir des groupes cohérents de principes d'architecture.*

### Créer un principe d'architecture

Un *Principe d'architecture* est détenu par une *Catégorie de principe d'architecture*

Pour créer une *Principe d'architecture* :

1. A partir du menu de navigation **Gouvernance**, sélectionnez **Cadre de politique d'entreprise**.  
L'arbre des *Cadres de politique d'entreprise* apparaît.
2. Dépliez le dossier du *Cadre de politique d'entreprise* qui vous intéresse.
3. Sélectionnez la *Catégorie de principe d'architecture* qui vous intéresse et cliquez sur **Nouveau > Principe d'architecture**.  
Le nouveau *Principe d'architecture* apparaît dans l'arbre.

## Définir le périmètre d'un principe d'architecture

Pour définir le périmètre d'un *principe d'architecture* :

1. Ouvrez la page de propriétés **Caractéristiques**, du principe d'architecture qui vous intéresse.
2. Dans la section **Périmètre**, cliquez sur **Nouveau**.  
Une fenêtre de création d'**Exigence réglementaire** s'ouvre.
3. Sélectionnez le **Type d'objet** concerné, par exemple une **Architecture de déploiement d'un application**.
4. Sélectionnez l'**Elément soumis à la réglementation** et cliquez sur **Ajouter**.  
Un nouvel élément de mise en œuvre de principe d'architecture est créé.

---

## Définir les catégories de données

La solution **HOPEX IT Architecture** permet de classifier les données en utilisant des *catégories de données*.

☛ Pour plus de détails sur les catégories de données, voir le guide **HOPEX Data Governance**.

Pour accéder à la liste des *catégories de données* à partir du menu de navigation **Administration** :

- ☐ Sélectionnez **Graphes de catégorisation**.  
La liste des graphes de catégorisation s'affiche.

☛ Pour plus de détails sur les catégories de données, voir le chapitre "Définir les graphes de catégorisation" du guide **HOPEX IT Business Management**.

---

## Définir les domaines méthodologiques

Chaque solution **HOPEX** propose ses propres domaines méthodologiques qui définissent les objectifs communs visés par les utilisateurs de la solution.

Les domaines méthodologiques proposés par **HOPEX IT Architecture** concernent :

- L'utilisation des Services Cloud,
- La transition vers une architecture moderne,
- La réduction de l'empreinte carbone liés à l'informatique.

---

## Importer des composants HOPEX IT Architecture

**HOPEX IT Architecture** utilise les assistants d'échange de données avec Excel, pour exporter et importer des éléments d'architecture existants.

☛ Pour plus de détails sur les assistants d'échange avec Excel, voir le chapitre "Échanger des données avec Excel" du guide **HOPEX Common Features**.




Deux modèles de fichiers sont proposés :


- **Hardware\_Functionalities\_Template.xlsx** pour importer/exporter des éléments d'infrastructure : ordinateurs reliés à la décomposition des fonctionnalités matérielles couvertes.
- **Technical\_Functionalities\_Template.xlsx** pour importer/exporter des éléments d'architecture technique : technologies reliées à des vendeurs d'une part et à la décomposition des fonctionnalités techniques couvertes.

## Structure des modèles Excel d'import/export de HOPEX IT Architecture


Les modèles Excel de **HOPEX IT Architecture** qui permettent d'importer des éléments matériels ou techniques sont structurés de manière identique.


- Au niveau matériel, les éléments sont les suivants : *Matériel*, *Terminal informatique*, *Serveur informatique* et *Objet connecté*.

 *Un objet connecté est à la fois un équipement matériel et un ordinateur, qui fournit des services matériels et informatiques combinés à ses utilisateurs. En tant qu'équipement matériel, il embarque des capteurs (accéléromètre, par exemple) qui fournissent des données aux ordinateurs embarqués. En tant qu'ordinateur, il peut héberger des bases de données ou exécuter des applications. Exemples d'objets connectés : montre connectée avec traqueur GPS, caméra de vidéo surveillance en ligne avec flux vidéo IP, pèse personne connectés avec gestion de l'historique du poids.*


 *Pour plus de détails sur les éléments matériels, voir [Décrire un ordinateur](#).*

- Au niveau des technologies, les éléments sont les suivants : *Vendeur (Acteur)* et *Technologie logicielle*.



 *Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.*

 *Pour plus de détails sur les technologies, voir [Décrire une technologie logicielle](#).*

- Au niveau des fonctionnalités (techniques ou matérielles), les éléments sont les suivants :
  - les *Capacités (matérielles ou technologiques)*,

 *Une capacité matérielle est la capacité à fournir un résultat attendu par une ressource matérielle afin de réaliser un travail. Cette*

fonctionnalité matérielle est généralement nécessaire à un système informatique pour exécuter une opération spécifique.

- les **Cartes de capacités (matérielles ou technologiques)**,  
 Une carte de capacités technologiques est un assemblage de capacités technologiques avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture.  
 Pour plus de détails sur les fonctionnalités, voir [Utiliser les fonctionnalités avec HOPEX IT Architecture](#).
- les **Sous-fonctionnalités**, qui permettent de définir le lien entre une fonctionnalité et la carte de fonctionnalités (ou la fonctionnalité) dans laquelle elle est référencée.
- les **Mises en œuvre de fonctionnalité** qui permettent de définir le lien entre une fonctionnalité et l'objet matériel ou technique qui la met en œuvre.

La liste des informations prévues dans le modèle Excel livré avec **HOPEX IT Architecture** sont attendues dans l'ordre suivants :

- Pour les éléments de type : **Carte de capacités (matérielle ou technologique)**, **Ordinateur (Terminal informatique, Serveur, objet connecté)** :
  - **Short Name** : nom de l'objet du type concerné.
  - **Comment** : commentaire concernant l'objet du type concerné.
- Pour les éléments de type **Technologie** :
  - **Short Name** : nom de l'objet du type concerné.
  - **Technology Code**.
  - **Comment** : commentaire concernant l'objet du type concerné.
  - **Vendor**.
- Pour les éléments de type **Acteur** :
  - **Short Name** : nom de l'objet du type concerné.
  - **Internal/External**.
  - **Org-Unit Type**.
  - **Comment** : commentaire concernant l'objet du type concerné.
- Pour chaque élément de type **Composition de fonctionnalité** :
  - Nom de l'objet composé : carte de fonctionnalités ou fonctionnalité détentrice,
  - Nom de la fonctionnalité détenue.
- Pour chaque élément de type **Mise en œuvre de fonctionnalité** :
  - **Fulfilled Hardware/Technical Functionality** : nom de la fonctionnalité mise en œuvre.
  - Nom de l'objet (ordinateur ou technologie) qui assure la mise en œuvre de la fonctionnalité.

## Importer des ordinateurs ou des technologies avec Excel

 Pour plus de détails sur la structure du modèle Excel, voir [Importer des composants HOPEX IT Architecture](#).

Si vous souhaitez exporter des ordinateurs ou des technologies ou des cartes de fonctionnalités qui existent dans un autre référentiel que votre référentiel courant, par exemple, vous pouvez utiliser les modèles Excel de **HOPEX IT Architecture**.

Plusieurs étapes doivent être respectées pour que l'import Excel d'une décomposition de capacités métier puisse être réalisé correctement :

1. (optionnel) [Spécifier la bibliothèque courante](#),
2. [Exporter les données de votre référentiel avec HOPEX IT Architecture](#),
3. [Compléter votre fichier d'import pour HOPEX IT Architecture](#),
4. Importer votre nouveau fichier dans votre référentiel.

☛ Pour plus de détails sur l'assistant d'import Excel, voir le chapitre "Import de données via Excel" du guide **HOPEX Common Features**.



### **Spécifier la bibliothèque courante**

Cette étape optionnelle permet de relier automatiquement les objets importés à la bibliothèque courante.

Une *bibliothèque* et une *entreprise* permettent de représenter un contexte de travail unique.



Une bibliothèque est un regroupement d'objets qui permet de découper le contenu d'un référentiel en plusieurs parties indépendantes. Les bibliothèques permettent de réaliser des partitions virtuelles du référentiel. En particulier, deux objets appartenant à des bibliothèques différentes peuvent avoir le même nom.

Afin que les données que vous importez avec Excel soient reliées à un conteneur précis, vous devez spécifier la bibliothèque courante.

☛ Pour plus de détails, voir [Définir l'environnement de travail](#).

### **Exporter les données de votre référentiel avec HOPEX IT Architecture**

Si vous souhaitez exporter des cartes de fonctionnalités (matérielles ou technologiques) qui existent dans un autre référentiel que votre référentiel courant, par exemple, vous pouvez utiliser le modèle Excel de **HOPEX IT Architecture**.

☛ Pour plus de détails sur l'assistant d'export Excel, voir le chapitre "Export de données via Excel" du guide **HOPEX Common Features**.



Quand le fichier Excel est renseigné avec le nom des objets que vous souhaitez exporter, vous devez compléter les informations nécessaires à l'import dans **HOPEX IT Architecture**.

☛ Pour plus de détails sur les informations complémentaires, voir [Compléter votre fichier d'import pour HOPEX IT Architecture](#).

### **Compléter votre fichier d'import pour HOPEX IT Architecture**

Pour que votre fichier d'import soit correct, vous devez avoir renseigné les éléments suivants :

- Pour chaque élément de type *Carte de capacité (matérielle ou technologique)*, *Vendeur*, *Technologie*, *Ordinateur* (*Terminal*)

*informatique, Serveur, objet connecté*), vous devez indiquer le nom de chaque objet.

- Pour chaque décomposition (Feuille Excel **Technical Function\_Composition** ou **Functionality Composition**), vous devez indiquer :
  - le nom de l'objet composé : carte de fonctionnalités ou fonctionnalité détentrice,
  - le nom de la fonctionnalité détenue.
- Pour spécifier qu'une technologie met en œuvre une fonctionnalité par exemple, vous devez indiquer dans la feuille Excel **Technical Function\_Fulfillment** :
  - le nom de la capacité métier mise en œuvre dans la colonne **Fulfilled Hardware/Technical Functionality**,
  - Nom de l'objet (ordinateur ou technologie) qui assure la mise en œuvre de la fonctionnalité.

🔊 **Les deux premières lignes de chaque feuille Excel sont réservées au paramétrage du fichier ; veillez à ce que les deux premières lignes du fichier importé restent identiques à celles obtenues à l'issue d'un export.**

🔗 Pour plus de détails sur la structure du modèle Excel, voir [Structure des modèles Excel d'import/export de HOPEX IT Architecture](#).

## Utiliser les outils de conversion vers HOPEX Aquila

Si vous utilisez **HOPEX Aquila** et que votre administrateur a mis en œuvre la conversion des instances de **Personne métier** de vos référentiels antérieurs à **HOPEX V5** en instances de **Personne (Système)**, vous devez vous assurer que des assignations ont été créées entre ces nouvelles instances de **Personne (Système)** et les objets qui étaient sous la responsabilité des instances de **Personne métier**.

🔗 Pour plus d'information sur la conversion d'un référentiel, consultez la note technique **How to migrate to HOPEX**.

L'outil spécifique de conversion "Mega Repository - Convert Person and ARC Responsibility into Person (System) and Assignment" doit être activé par votre administrateur en utilisant l'**Environment automatic update**.

Le principe de cet outil de conversion spécifique à **HOPEX IT Architecture** est le suivant :

1. Une nouvelle instance de **Personne (Système)** doit être créée pour chaque instance de **Personne métier** ayant un lien **Responsabilité** vis à vis d'une **Application**, d'une **Base de données**, d'un **Réseau**, d'un **Nœud**, d'un **Serveur**, d'un **Service applicatif** ou d'une **Station de travail**.
2. Pour chaque lien **Responsabilité** entre un actif et une ancienne instance de **Personne métier**, une assignation est créée entre l'actif et l'instance de **Personne (Système)** pour les cas prévus dans le tableau des responsabilités à remplacer par des assignations.

🔗 Le Tableau des rôles métier utilisés pour les assignations (ci-dessous) permet de préciser si une assignation doit être créée entre une nouvelle instance de **Personne (Système)** et un actif en fonction du type de

lien de **Responsabilité** de l'instance de **Personne métier** convertie et le type d'actif concerné. Si aucun rôle métier n'est précisé, aucune assignation n'est créée et l'ancienne instance de **Personne métier** n'est pas convertie.

Type d'Actif	Correspondant informatique	Correspondant métier	Gestionnaire Qualité	Gestionnaire de risques ou Fournisseur
Application	Propriétaire local d'application	Responsable métier	-	-
Base de données	Propriétaire de données	-	Responsable qualité des données	-
Réseau	-	-	-	-
Nœud	-	-	-	-
Serveur	-	-	-	-
Service Applicatif	Propriétaire local d'application	-	-	-
Station de travail	-	-	-	-

Tableau des rôles métier utilisés pour les assignations créées



## A PROPOS DE CE GUIDE

Ce guide vous présente comment tirer parti de **HOPEX IT Architecture** pour assurer une gestion efficace de vos projets d'architecture informatique.

---

### Structure du guide

La première partie du guide **HOPEX IT Architecture** est composé des chapitres suivants :


- [Modéliser les applications et les architectures système](#) ; présente les fonctionnalités offertes par **HOPEX IT Architecture** pour décrire les éléments informatiques de votre entreprise.
- [Aligner l'informatique et les métiers](#) ; explique comment **HOPEX IT Architecture** vous assiste dans l'analyse de votre architecture logique.
- [Modéliser les architectures de déploiement](#) ; explique comment préparer le déploiement des composants de votre SI.
- [Modéliser les infrastructures informatiques](#) ; décrit les fonctionnalités offertes par **HOPEX IT Architecture** pour des systèmes faisant intervenir des ressources autres qu'informatiques.

La seconde partie du guide **HOPEX IT Architecture** est composé des chapitres consacrés à la modélisation UML.

---

### Ressources complémentaires

Ce guide est complété par :

- le guide **HOPEX Common Features**, qui décrit l'interface Web et les outils spécifiques aux solutions **HOPEX**.  
 *Il peut être utile de consulter ce guide pour une présentation générale de l'interface.*
- le guide **HOPEX Business Process Analysis**, qui décrit les fonctionnalités proposées pour gérer les processus ;
- le guide **HOPEX IT Portfolio Management**, qui décrit les fonctionnalités proposées pour gérer votre parc applicatif ;
- le guide **HOPEX IT Business Management** qui décrit les fonctionnalités proposées pour gérer vos projets de transformation d'architecture ;
- le guide d'administration **HOPEX Power Supervisor**.

---

## Conventions utilisées dans le guide

- 👉 *Remarque sur les points qui précèdent.*
- 📖 *Définition des termes employés.*
- 😊 *Astuce qui peut faciliter la vie de l'utilisateur.*
- 🦖 *Compatibilité avec les versions précédentes.*
- 💣 **Ce qu'il faut éviter de faire.**



**Remarque très importante à prendre en compte pour ne pas commettre d'erreurs durant une manipulation.**

Les commandes sont présentées ainsi : **Fichier > Ouvrir.**

Les noms de produits et de modules techniques sont présentés ainsi : **HOPEX.**

# **Spécification d'architecture**





# MODÉLISER LES APPLICATIONS ET LES ARCHITECTURES SYSTÈME



**HOPEX IT Architecture** vous permet de représenter et documenter vos architectures IT selon une approche orientée service.

Modéliser une architecture selon une approche orientée service facilite l'analyse des communications entre les architectures. Ainsi, la description des architectures s'appuie sur des concepts qui permettent une utilisation plus générique du produit.

Les points traités ici sont les suivants :

- ✓ [Introduction des concepts de HOPEX IT Architecture](#) ;
- ✓ [Décrire une application avec HOPEX IT Architecture](#) ;
- ✓ [Décrire l'architecture système.](#)

# INTRODUCTION DES CONCEPTS DE HOPEX IT

## ARCHITECTURE

Le système d'information peut être décomposé selon deux niveaux de granularité : l'application et le système applicatif.

### Application

Une application est un ensemble de composants logiciels qui constituent un tout cohérent au regard de leur déploiement, de leur couverture fonctionnelle et des techniques informatiques utilisées.

L'application est l'unité de gestion et de déploiement d'un ensemble de composants logiciels. Elle est susceptible d'être déployée sur une ou plusieurs machines. Une application répond à :

- des besoins métiers  
Ex. : facturation, comptabilité, gestion de matériel, calcul de charge/capacité.
- des besoins techniques  
Ex. : interface de communication spécifique, contrôle d'accès.
- des besoins transverses  
Ex. : messagerie, annuaire, application bureautique.

Pour la création d'une application, voir [Décrire une application avec HOPEX IT Architecture](#).

### Système applicatif

Un système applicatif est un assemblage d'applications répondant à un ensemble cohérent de fonctionnalités, mises en œuvre grâce à la coopération des applications qui composent ce système.

Un système applicatif peut être constitué d'une suite d'applications regroupées pour des raisons commerciales (les progiciels de gestion intégrés ou PGI tels que SAP, Oracle Applications, Siebel).

Un système applicatif peut aussi correspondre à un regroupement d'applications qui ont les mêmes objectifs fonctionnels (système de gestion comptable et financière intégrant toutes les applications de comptabilité : générale, fournisseurs, analytique ainsi que les modules d'analyse financière et budgétaire, systèmes de gestion des ressources humaines intégrant paye, gestion du temps, gestion des carrières,...).

Un système applicatif, comme une application, peut faire l'objet de développements spécifiques (réalisés en interne ou bien achetés / sous-traités), ou être des produits du marché (proiciels).

L'organisation logique et la structure des systèmes applicatifs et des applications, ainsi que la description de leurs échanges, constituent les fondations de l'architecture applicative. Ainsi, la représentation des flux dans un système applicatif permet d'identifier l'impact du retrait d'une application sur l'ensemble du système.

Pour la création d'un système applicatif, voir [Décrire l'architecture système](#).

# DÉCRIRE UNE APPLICATION AVEC HOPEX IT

## ARCHITECTURE

Un projet de description de l'architecture fonctionnelle d'un système d'information permet de recenser les *applications* existantes et leurs échanges.



*Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.*

---

### Créer une application avec HOPEX IT Architecture

Pour créer une *application* :

1. Cliquez sur le menu de navigation **Applications**.  
La liste des applications apparaît dans la zone d'édition.
2. Cliquez sur **Nouveau**.
3. La fenêtre **Création d'une application** apparaît.
4. Saisissez le **Nom** de votre application.
5. (Optionnel) Sélectionnez son **Détenteur**.
6. Cliquez sur **OK**.  
La nouvelle application apparaît dans la liste.



## Les propriétés d'une application avec HOPEX IT Architecture

La page **Caractéristiques** des propriétés d'une application permet d'accéder à différentes sections.

- la section **Identification** permet d'accéder aux informations suivantes :
  - le **Nom**
  - son **Détenteur**, par défaut lors de la création de l'application, il s'agit de la bibliothèque courante.
  - le texte de sa **Description**.
  - le **Code** interne.
  - le **Numéro de version**.
  - le **Type d'application**.
  - le **Cloud Computing**.
  - une **Description**.
- la section **Service Level Agreement** permet d'obtenir le **Temps d'interruption maximum tolérable** ainsi que le **Niveau de SLA** de l'application, à partir de la qualification des grandeurs suivantes :
  - **Objectif de point de reprise (RPO)**,
  - **Objectif de temps de reprise (RTO)**,
  - **Temps de récupération du travail (WRT)**.
- la section **Périmètre fonctionnel** de l'application, voir [Définir le périmètre fonctionnel de l'application](#).
- la section **Spécifications**, voir [Créer un diagramme de cas d'utilisation d'une application](#).
- la **Responsabilité** : il s'agit de la (ou des) personnes responsable(s) de l'application :
  - Propriétaire local d'application,
  - Responsable informatique,
  - Propriétaire métier.

☛ Pour plus de détails sur ces rôles, voir [Les rôles métier de HOPEX IT Architecture](#).

- la section **Technologies** permet d'accéder à la liste des **Technologies logicielles** et à la liste des **Piles de technologie logicielles** utilisées par l'application.



Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la

présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.



Une pile de technologies logicielles est un ensemble de technologies logicielles.

☛ Pour plus de détails sur les technologies logicielles, voir [Décrire les technologies logicielles](#).

- la section **Échanges** décrits les flux applicatifs émis et reçus par l'application. Voir [Utiliser un diagramme de scénario de flux d'application](#).
- pour plus de détails sur la section **Données**, voir [Gérer les données](#).
- la section **Risques** présente les risques associés à l'application, voir [Décrire un environnement d'application avec HOPEX IT Architecture](#).
- les **Pièces jointes** associées.




☛ Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

Avec **HOPEX IT Architecture** une application est décrite par d'autres pages de propriétés, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

---

## Définir le périmètre fonctionnel de l'application

Pour indiquer les objets qui définissent la couverture fonctionnelle d'une application :

1. Ouvrez la page de propriétés **Caractéristiques** de l'application.
2. Dépliez la section **Périmètre fonctionnel**.  
Les types de données qui définissent la couverture fonctionnelle de l'application sont :
  - Les **Catégories de processus** outillées par l'application  
 Une catégorie de processus définit un ensemble de processus. Elle est reliée à une carte de processus ou à une catégorie de processus de niveau supérieur. Elle regroupe plusieurs processus et/ou éléments catégorisés (chaînes de valeur, applications). Elle sert de niveau intermédiaire pour hiérarchiser les processus, afin d'accéder progressivement jusqu'au niveau le plus fin de granularité des processus.
  - Les **capacités métier** couvertes par l'application  
 Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).  
☛ Pour plus de détails sur les capacités métier, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).  
☛ Un rapport permet de visualiser la répartition des applications dans les capacités métier, voir [Rapports sur la couverture fonctionnelle de l'architecture](#).
  - Les **fonctionnalités implémentées** sont mises en œuvre par l'application,  
 Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération

particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.

☛ Pour plus de détails sur les fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#).

☛ Pour plus de détails sur les mises en œuvre, voir [Utiliser les mécanismes de mise en œuvre](#).

- Les **réalisations logiques** de l'application sont les applications ou les systèmes applicatifs logiques réalisés par l'application.

☛ Pour plus de détails sur les mises en œuvre, voir [Utiliser les mécanismes de mise en œuvre](#).

---

## Décrire la structure et les services d'une application

Dans un premier temps, une application peut être décrite d'un point de vue logique, voir [Décrire l'architecture logicielle logique](#).

Toutefois, et d'un point de vue concret, une **application** est décrite par plusieurs types de diagramme :

- un **diagramme de structure d'application** permet de représenter les interactions de service entre les différents composants de l'application sous forme d'interface de service.

☛ Pour plus de détails, voir [Le diagramme de structure d'application](#).

- un **scénario de flux d'application** présente les flux échangés entre les services applicatifs ou les microservices utilisés par cette application. Un scénario peut représenter un cas d'utilisation particulier de l'application ou plus globalement l'ensemble des flux échangés à l'intérieur de cette application.

☛ Pour plus de détails, voir [Utiliser un diagramme de scénario de flux d'application](#).

- un **scénario de séquences de flux** présente les agents nécessaires au scénario (services applicatifs, microservices, dépôts de données) et les séquences de flux applicatifs échangés avec l'application.

☛ Pour plus de détails, voir [Créer un diagramme de scénario de séquence de flux](#).

- une **architecture de déploiement d'une application** qui permet de représenter les éléments techniques qui supportent l'application.


☛ Pour plus de détails, voir [Décrire une architecture de déploiement d'une application](#).

- un **diagramme de cas d'utilisation** qui permet de représenter, selon une approche UML, les échanges entre l'application et les acteurs de l'organisation.


☛ Pour plus de détails, voir [Créer un diagramme de cas d'utilisation d'une application](#).

Pour plus de détails sur la modélisation de vos applications avec **HOPEX IT Architecture**, voir [Modéliser les architectures applicatives](#).




## Décrire un environnement d'application avec HOPEX IT Architecture

 Un environnement d'application présente un contexte d'utilisation d'une application. Il permet de représenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels cette application est susceptible d'interagir.

### Décrire un environnement d'application

 Un environnement d'application présente un contexte d'utilisation d'une application. Il permet de représenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels cette application est susceptible d'interagir.

Un **environnement d'application** est décrit par plusieurs types de diagramme :

- un **diagramme d'environnement d'application** permet de représenter les échanges entre l'application et les éléments de son environnement dans un contexte donné.  
 Pour plus de détails, voir [Présentation d'un diagramme d'environnement d'application](#).
- un **scénario de flux d'environnement d'application** présente les flux échangés entre l'application décrite et ses partenaires : applications, systèmes applicatifs, services applicatifs ou les microservices utilisés par l'application décrite dans un contexte donné.  
 Pour plus de détails, voir [Utiliser un diagramme de scénario de flux d'application](#).
- un **scénario de séquences de flux** présente les agents nécessaires au scénario (applications, services applicatifs, microservices, dépôts de données) et les séquences de flux applicatifs échangés.  
 Pour plus de détails, voir [Utiliser un diagramme de scénario de séquence de flux](#).

### Accéder à la liste des environnements d'application

Pour accéder à la liste des environnement d'application à partir du menu de navigation **Applications** :

1. Ouvrez la page **Environnements** de l'application qui vous intéresse.  
La liste des environnements d'application apparaît dans la zone d'édition.

### Créer un environnement d'application

Pour créer un **environnement d'application** :

1. Ouvrez la page **Environnements** de l'application qui vous intéresse.  
La listes des environnements d'application apparaît dans la zone d'édition.
2. Cliquez sur **Nouveau**.  
Le nouvel environnement d'application apparaît dans la liste, il porte le nom de l'application précédé de « Environnement ».

## Les propriétés d'un environnement d'application

La page **Caractéristiques** des propriétés d'un environnement d'application permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de l'environnement de système applicatif, il s'agit de la bibliothèque courante.
- son **Nom**,
- le texte de sa **Description**.

☞ Pour plus de détails sur les autres pages d'un environnement d'application, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

La page de propriétés **Composants** d'un environnement d'application permet d'accéder aux éléments partenaires :

- Applications,
- Microservice,
- Services applicatifs,
- Utilisateurs du système.

☞ Pour plus de détails sur les composants d'un diagramme environnement d'application, voir [Présentation d'un diagramme d'environnement d'application](#).

## Présentation d'un diagramme d'environnement d'application

Avec **HOPEX IT Architecture**, un *environnement d'application* est totalement décrit par un diagramme d'environnement d'application qui permet de décrire les interactions de service entre les applications de l'environnement décrit, ses utilisateurs et les applications externes.

Un diagramme d'environnement d'application est composé :

- d'*applications* qui représentent l'environnement décrit.

Dans l'exemple, il s'agit des applications utilisées pour l'achat des pièces détachées.



*Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.*

- d'*applications*, de *services applicatifs* ou de *microservices partenaires* qui représentent les éléments externes utilisés dans l'environnement décrit.

Dans l'exemple, il s'agit des services web automatisés.



*Un service applicatif est l'élément de découpage d'une application qui est mis à la disposition de l'utilisateur final de cette application dans le cadre de son travail.*

- d'*acteurs* ou de *positions type* qui représentent les utilisateurs ou les fournisseurs de l'environnement décrit.

Dans l'exemple, il s'agit des intervenants locaux.



*Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur*

externe représente un organisme qui échange des flux avec l'entreprise.  
Ex : Client, Fournisseur, Administration.

- **interactions de service** entre les composants.



Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

- points d'accès, de requête et de service.

---

## Spécifier les risques associés à une application

**HOPEX IT Architecture** vous permet d'identifier les risques liés à une application, et de récupérer les évaluations définies dans la solution **HOPEX Enterprise Risk Management**. Vous pouvez définir un nouveau risque à partir de l'application ou relier un risque préalablement défini.

Pour relier un risque à une application :

1. Ouvrez les pages de propriétés **Caractéristiques** de l'application.
2. Dépliez la section **Risque**.
3. Cliquez sur **Relier**.  
La fenêtre de recherche apparaît.
4. Recherchez et sélectionnez le risque voulu et cliquez sur **OK**.

Pour plus de détails sur les risques et leur évaluation, voir **HOPEX Enterprise Risk Management**.

# DÉCRIRE L'ARCHITECTURE SYSTÈME





## Décrire un système applicatif

Un projet de description de l'architecture fonctionnelle d'un système d'information permet aussi de recenser les *systèmes applicatifs* existants et leurs échanges.



*Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

Un *système applicatif* est décrit par plusieurs types de diagramme :

- un *diagramme de structure de système applicatif* qui permet de représenter les interactions de service entre les différents composants du système applicatif sous forme d'interfaces de service.  
 Pour plus de détails, voir [Créer un diagramme de structure de système applicatif](#).
- une *architecture de déploiement d'un système applicatif*, qui permet de représenter l'architecture technique choisie pour le déploiement de chacune des composants qui supportent le système applicatif ainsi que les techniques de communications nécessaires.  
 Pour plus de détails, voir [Décrire une architecture de déploiement d'un système applicatif](#).
- un *scénario de flux de système applicatif* présente les flux échangés entre les autres systèmes applicatifs, les applications ou les microservices utilisés par ce système applicatif. Un scénario peut représenter un cas d'utilisation particulier du système applicatif ou plus globalement l'ensemble des flux échangés à l'intérieur de ce système applicatif.  
 Pour plus de détails, voir [Utiliser un scénario de flux de système applicatif](#).
- un *scénario de séquences de flux* présente les agents nécessaires au scénario (services applicatifs, microservices, dépôts de données) et les séquences de flux applicatifs échangés avec l'application.  
 Pour plus de détails, voir [Utiliser un diagramme de scénario de séquence de flux](#).

## Créer un système applicatif

Pour créer un *système applicatif* :

1. Cliquez sur le menu de navigation **Système applicatif**.  
La liste des systèmes applicatifs apparaît.
2. Cliquez sur **Nouveau**.  
La fenêtre **Création d'un système applicatif** apparaît.
3. Saisissez le **Nom** de votre système applicatif et cliquez sur **OK**.  
Le nouveau système applicatif apparaît dans la liste.

## Les propriétés d'un système applicatif

La page **Caractéristiques** des propriétés d'un système applicatif permet d'accéder à différentes sections.

- la section **Identification** permet d'accéder aux informations suivantes :
  - le **Nom**,
  - son **Détenteur**, par défaut lors de la création du système applicatif, il s'agit de la bibliothèque courante,
  - le texte de sa **Description**,
  - le **Code** interne,
  - le **Numéro de version**,
  - une **Description**.
- la section du **Périmètre fonctionnel** du système applicatif, voir [Définir le périmètre fonctionnel de l'application](#).
- la section **Cas d'utilisation**, voir [Créer un diagramme de cas d'utilisation d'une application](#).
- la section des **Responsabilités** : il s'agit de la ou des personnes responsable(s) de du système applicatif :
  - Concepteur de logiciel
  - Propriétaire local d'application

➡ Pour plus de détails sur ces rôles, voir [Les rôles métier de HOPEX IT Architecture](#).

- la section des **Pièces jointes** associées.

➡ Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

Avec **HOPEX IT Architecture** un système applicatif est décrit par d'autres pages de propriétés, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).



## Créer un diagramme de structure de système applicatif

Ce diagramme décrit la structure interne d'un système applicatif :

- les services offerts ou requis
- les composants applicatifs et leurs interactions de service, il s'agit de systèmes applicatifs, d'applications et de microservices,
- les utilisateurs finaux en interaction avec les composants applicatifs.

Le diagramme suivant décrit le système applicatif du traitement des demandes d'achat.

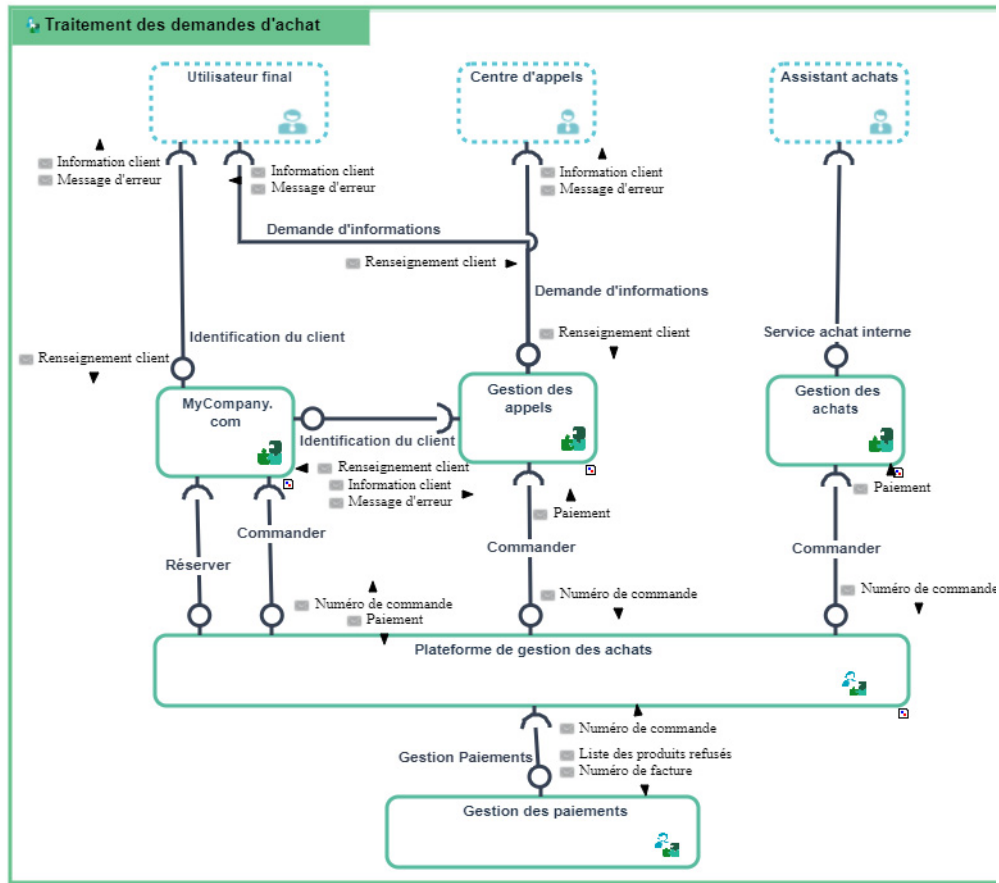


Diagramme de structure de système applicatif "Traitement des demandes d'achat"

Pour créer un diagramme de structure de système applicatif :


1. Ouvrez la page **Diagrammes** du système applicatif qui vous intéresse et cliquez sur **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré > Architecture interne**.  
Un **Diagramme de structure de système applicatif** s'ouvre dans la zone d'édition.

## Ajouter un système applicatif dans un diagramme de structure de système applicatif

Pour décrire qu'un système applicatif met en œuvre un autre système applicatif, vous allez ajouter un **système applicatif** du diagramme de structure de système applicatif.


Par exemple, le système de traitement des demandes d'achat utilise les services des systèmes applicatifs de la "Plateforme de gestion des achats" et de "Gestion des paiements".

Pour ajouter un **Système applicatif** :


1. Dans la barre d'objets du diagramme de structure de système applicatif, cliquez sur le bouton  **Système applicatif**.
2. Cliquez dans le cadre du système applicatif décrit.  
Une fenêtre d'ajout vous demande de choisir le **système applicatif** mis en œuvre (par exemple "Gestion des paiements").
3. Sélectionnez un système applicatif.
4. Cliquez sur **OK**.  
Le système applicatif apparaît dans le diagramme.


## Ajouter un utilisateur final dans un diagramme de structure de système applicatif


Pour décrire qu'un système applicatif, tel que le traitement des demandes d'achat, est activé par des acteurs internes et externes, vous allez ajouter un **utilisateur final** associer.

 *L'utilisateur final représente une unité organisationnelle interagissant aux frontières d'un système applicatif ou d'un système applicatif logique.*

Pour ajouter un **utilisateur final** :

1. Dans la barre d'objets du diagramme de structure de système applicatif, cliquez sur le bouton  **Utilisateur final** et cliquez dans le cadre du diagramme.  
Une fenêtre d'ajout vous demande de choisir le **Type d'objet** que vous souhaitez utiliser : **Acteur** ou **Type de Position**.
2. Sélectionnez, par exemple, le type d'objet **Acteur**.

 *Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.*

 *Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.*

3. Sélectionnez l'acteur qui vous intéresse et cliquez sur **OK**.  
L'acteur apparaît dans le diagramme.

## Utiliser un scénario de flux de système applicatif

☛ Pour plus de détails sur l'utilisation des scénarios de flux, voir [Utiliser un diagramme de scénario de flux d'application](#).

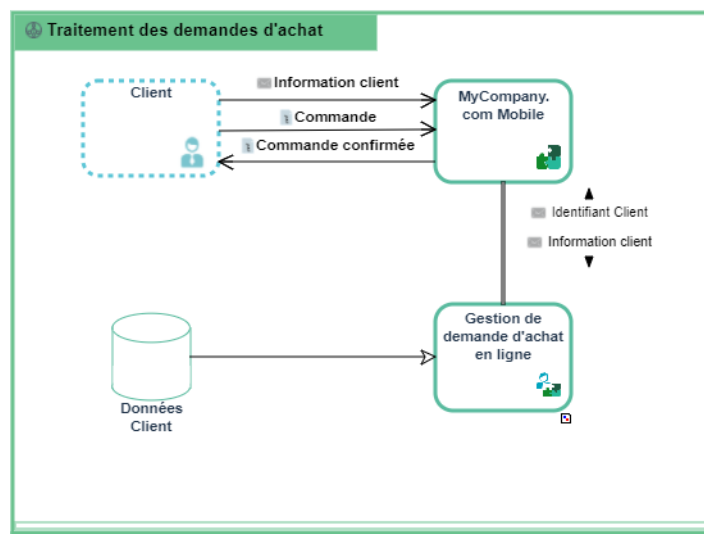
Un scénario de flux de système applicatif permet de représenter les flux échangés entre certains éléments du système applicatif dans un contexte donné. Les éléments représentés sont :

- des systèmes applicatifs,
- des applications,
- des microservices,
- des acteurs de l'organisation,
- des dépôts de données applicatives locales ou externes,
- des ports applicatifs en entrée ou en sortie.

Les échanges proposés entre ces éléments :

- des flux applicatifs qui portent un contenu,
- des canaux de flux applicatifs qui regroupent plusieurs flux applicatifs sur un même lien,
- des canaux de données applicatives qui représentent les échanges effectués avec les dépôts de données applicatives.

Le scénario de flux de système applicatif ci-dessous décrit les échanges entre un client et l'application de eCommerce.




Exemple de scénario de flux de système applicatif de "Traitement des demandes d'achat"


Pour créer un diagramme de scénario de flux de système applicatif :

1. Ouvrez la page **Diagrammes** du système applicatif qui vous intéresse et cliquez sur **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré > Flux externes de données**.  
Un **Diagramme de scénario de flux de système applicatif** s'ouvre dans la zone d'édition.

## Ajouter un acteur de l'organisation au scénario de flux de système applicatif

Un acteur de l'organisation est représenté soit par un **Acteur** soit par un **Type de position**.

 Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.


 Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.

Pour ajouter un acteur de l'organisation :

1. Dans la barre d'objets du scénario de flux de système applicatif, cliquez sur le bouton **Acteur**.
2. Cliquez dans le cadre du système applicatif décrit.  
Une fenêtre d'ajout vous demande de choisir le nom de l'acteur que vous souhaitez utiliser.
3. Sélectionnez l'acteur qui vous intéresse et cliquez sur **OK**.  
L'acteur apparaît dans le diagramme.

➡ Pour créer un nouvel acteur, saisissez son nom et cliquez sur **OK**.

## Ajouter une application au scénario de flux de système applicatif


 Une application est un composant logiciel déployable qui fournit un ensemble de fonctionnalités à des utilisateurs.

Pour ajouter une **application** :


1. Dans la barre d'objets du scénario de flux de système applicatif, cliquez sur le bouton **Application**.
2. Cliquez dans le cadre du système applicatif décrit.  
Une fenêtre d'ajout vous demande de choisir l'**application** que vous souhaitez utiliser (par exemple "Achat eCommerce").
3. Sélectionnez l'application et cliquez sur **OK**.  
L'application apparaît dans le diagramme.

De la même manière vous pouvez ajouter :

- un système applicatif

 Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.

- un microservice.

 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

Vous noterez que si vous ouvrez la page **Caractéristiques** du composant que vous venez d'ajouter au scénario de flux de système applicatif, une nouvelle section a été créée si le composant est déjà décrit par un scénario.

Pour plus de détails, voir [Réinitialiser les composants d'un scénario de flux](#).

---

## Décrire un environnement de système applicatif avec HOPEX IT Architecture



*Un environnement de système applicatif permet de présenter les autres systèmes applicatifs, applications, microservices ou acteurs avec lesquels ce système applicatif est susceptible d'interagir.*

### Accéder à la liste des environnements d'un système applicatif

Pour accéder à la liste des environnements d'un système applicatif à partir du menu de navigation **Système applicatif** :

- 1. Ouvrez la page **Environnements** du système applicatif qui vous intéresse.  
La liste des environnements de système applicatif apparaît.

### Créer un environnement de système applicatif

Pour créer un *environnement de système applicatif* à partir du menu de navigation **Système applicatif** :

1. Ouvrez la page **Environnements** du système applicatif qui vous intéresse.  
La liste des environnements de système applicatif apparaît.
2. Cliquez sur le bouton **Nouveau**.  
Le nouvel environnement de système applicatif apparaît dans la liste, il porte le nom du système applicatif suivi de "Environment".

### Les propriétés d'un environnement de système applicatif

La page **Caractéristiques** des propriétés d'un environnement de système applicatif permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de l'environnement de système applicatif, il s'agit de la bibliothèque courante.
- son **Nom**,
- le texte de sa **Description**.

Avec **HOPEX IT Architecture** un environnement de système applicatif est décrit par d'autres pages de propriétés, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Les diagrammes d'environnement de système applicatif

Un *environnement de système applicatif* est décrit par plusieurs types de diagramme :

- un *diagramme d'environnement de système applicatif* permet de représenter les échanges entre le système applicatif et les éléments de son environnement dans un contexte donné.

➡ Pour plus de détails, voir [Décrire un diagramme d'environnement de système applicatif](#).

- un *scénario de flux d'environnement de système applicatif* présente les flux échangés entre les services applicatifs ou les microservices utilisés par le système applicatif décrit dans un contexte donné.

➡ Pour plus de détails, voir [Décrire un scénario de flux d'environnement de système applicatif](#).

### Décrire un diagramme d'environnement de système applicatif

Un environnement de système applicatif est décrit par un **diagramme d'environnement de système applicatif** qui décrit les interactions de service entre les systèmes applicatifs internes, ses utilisateurs et les systèmes applicatifs partenaires.

➡ Pour plus de détails sur l'utilisation d'un diagramme de structure, voir [Le diagramme de structure d'application](#).

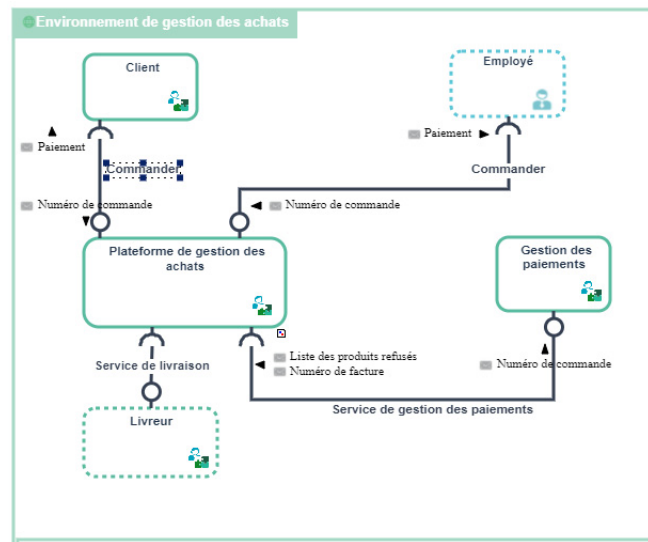


Diagramme d'environnement du système applicatif de gestion des achats

Les demandes d'achat sont formulées par des clients ou des employés via une "Plateforme de gestion des achats".

Le système applicatif "Plateforme de gestion des achats" fait appel à un système applicatif interne pour la "Gestion des paiements" et à un système applicatif partenaire pour la livraison des achats.

Les éléments d'un diagramme d'environnement de système applicatif sont :

- le **système applicatif** principal décrit par l'environnement.



*Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

- des **systèmes applicatifs partenaires** qui représentent les autres systèmes applicatifs avec lesquels le système applicatif principal décrit par l'environnement interagit.

Dans l'exemple, il s'agit de deux services de prêts proposés à des particuliers ou à des entreprises.



*Un système applicatif partenaire est un système applicatif externe à l'environnement de service applicatif décrit. Le système applicatif partenaire peut être fournisseur ou consommateur de service vis à vis des utilisateurs de système applicatif.*

- les catégories d'utilisateurs de services proposés par l'environnement sont représentés soit par un **Acteur** soit par un **Type de position**.



*Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.*



*Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.*

Il s'agit dans l'exemple de deux catégories de clients : les particuliers et les entreprises.

- des **interactions de service** entre les composants



*Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*

## **Décrire un scénario de flux d'environnement de système applicatif**

Un scénario d'environnement de système applicatif décrit les flux échangés entre les éléments de l'environnement de système applicatif.



*Pour plus de détails sur l'utilisation d'un scénario de flux, voir [Utiliser un diagramme de scénario de flux d'application](#).*

Les éléments d'un scénario d'environnement de système applicatif sont :

- le **système applicatif** principal décrit par l'environnement.



*Un système applicatif est un assemblage d'autres systèmes applicatifs, d'applications et d'utilisateurs finaux interagissant avec les*

composants applicatifs dans le but de mettre en œuvre une ou plusieurs fonctionnalités.

- des **systèmes applicatifs partenaires** qui représentent les autres systèmes applicatifs avec lesquels le système applicatif principal décrit par l'environnement interagit.



*Un système applicatif partenaire est un système applicatif externe à l'environnement de service applicatif décrit. Le système applicatif partenaire peut être fournisseur ou consommateur de service vis à vis des utilisateurs de système applicatif.*

- des **utilisateurs finals participants** qui représentent les catégories d'utilisateurs du système applicatif de l'environnement.
- les catégories d'utilisateurs de services proposés par l'environnement sont représentés soit par un **Acteur** soit par un **Type de position**.



*Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.*



*Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.*

- des **interactions de service** entre les composants



*Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*



# MODÉLISER LES ARCHITECTURES APPLICATIVES



Avec **HOPEX IT Architecture**, une application est décrite par les flux d'information qu'elle traite et transporte, les composants (services et API) qui assurent les services associés aux fonctionnalités qu'elle couvre ainsi que les composants de son environnement avec lesquels elle interagit.

Après avoir décrit les fonctionnalités attendues d'une application pour répondre aux besoins fonctionnels, ce chapitre décrit comment décrire les flux et la structure des applications.

Les points traités ici sont les suivants :

- ✓ Décrire les flux de données ;
- ✓ Décrire la structure et les services d'une application ;
- ✓ Décrire les processus applicatifs ;
- ✓ Gérer les données.

# DÉCRIRE LES FLUX DE DONNÉES

Un diagramme de scénario de flux décrit les messages échangés entre les éléments du système représenté.

Deux types de diagramme sont proposés :

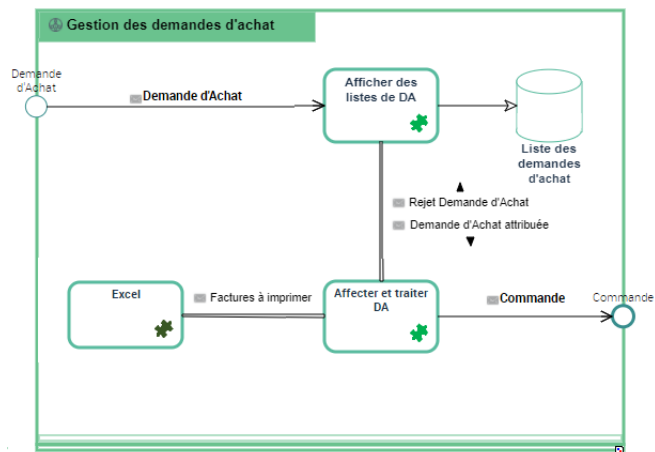
- les *diagrammes de scénario de flux* qui décrivent les flux échangés dans différents scénarios d'utilisation de l'objet décrit.
- les *diagrammes de scénario de séquences de flux* qui décrivent la chronologie des flux échangés dans différents scénarios d'utilisation de l'objet décrit.

➡ Pour visualiser les diagrammes de scénario de séquence de flux, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Activer les diagrammes de scénario de séquence de flux** est activée.

## Utiliser un diagramme de scénario de flux d'application

Un diagramme de scénario de flux peut être construit pour un environnement d'application, une application, un système applicatif, un service applicatif ou un microservice. Ce diagramme permet de décrire les échanges réalisés à l'intérieur de l'objet décrit dans un contexte particulier.

Le diagramme de scénario de flux ci-après décrit l'application "Gestion des demandes d'Achat".



Exemple de scénario de flux d'application de "Gestion des demandes d'achat"

Dans un diagramme de scénario de flux d'application, les éléments représentés sont :

- des services applicatifs, voir [Décrire un service applicatif avec HOPEX IT Architecture](#),
- des microservices, voir [Décrire un microservice avec HOPEX IT Architecture](#),
- des dépôts de données applicatives internes ou externes, voir [Utiliser les dépôts de données](#),
- des *événements déclencheurs système* et des *événements déclenchés système*, voir [Créer un événement déclencheur système](#).

Les échanges proposés entre ces éléments sont :

- des flux applicatifs qui portent un contenu,
- des canaux de flux applicatifs qui regroupent plusieurs flux applicatifs sur un même lien,
- des canaux de données applicatives qui représentent les échanges effectués avec les dépôts de données applicatives.

## Créer un diagramme de scénario de flux d'application

Pour créer un scénario de flux d'application à partir du menu de navigation **Applications** :

1. Faites un clic droit sur l'application qui vous intéresse et cliquez sur **Créer diagramme**.
2. Dans la fenêtre de choix du type de diagramme, sélectionnez **Diagramme structuré > Flux internes**.

Le diagramme de scénario de flux d'application apparaît.

## Ajouter un service applicatif au scénario de flux d'application



*Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.*

Pour ajouter un **service applicatif** :

1. Dans la barre d'objets du diagramme de scénario de flux d'application, cliquez sur le bouton **Service applicatif**.
2. Cliquez dans le cadre de l'application décrite.  
Une fenêtre d'ajout vous demande de choisir le **Service applicatif** que vous souhaitez utiliser (par exemple "Gestion des clients").
3. Sélectionnez le service applicatif et cliquez sur **OK**.  
Le service applicatif apparaît dans le diagramme.

De la même manière vous pouvez ajouter un microservice.



*Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.*

## Gérer les flux applicatifs dans un scénario de flux d'application

### Créer un flux applicatif avec un contenu



Un flux applicatif représente la circulation d'informations entre des applications ou à l'intérieur d'une application. Un flux applicatif peut transporter un contenu.

Les flux applicatifs échangés entre les services applicatifs, les microservices ou les ports applicatifs d'un scénario de flux d'application sont associés à un **contenu**.



Le contenu désigne le contenu d'un message ou d'un événement indépendamment de sa structure. Cette dernière peut être représentée par un schéma XML relié au contenu. Un contenu peut être utilisé par plusieurs messages puisqu'il n'est pas associé à un émetteur et à un destinataire. Il ne peut y avoir qu'un contenu par message ou par événement, mais un même contenu peut être utilisé par plusieurs messages ou événements.

Vous devez préciser le **contenu** d'un **flux applicatif** directement lors de sa création.

Pour créer le **flux applicatif** :

1. Dans la barre d'objets du diagramme de scénario de flux d'application, cliquez sur le bouton **flux applicatif** et sélectionnez le **Type de flux applicatif** qui vous intéresse :
  - **Résultat/Fourniture** pour représenter le résultat d'un service,
  - **Appel/Requête** pour représenter une demande de service,
  - **Signal** pour représenter un échange d'information.

☛ Pour plus de détails, voir [Associer une interface de service utilisée à un flux applicatif](#).

2. Cliquez sur le premier objet qui représente l'émetteur du flux et, en maintenant le bouton de la souris enfoncé, tirez un trait jusqu'à l'objet récepteur du flux.  
La fenêtre de **Création de Flux applicatif** s'ouvre.
3. Sélectionnez le contenu que vous souhaitez associer au flux à partir de la liste déroulante du champ **Contenu**.  
Un arbre des **mesures de flux** utilisées par d'autres flux applicatifs ayant le même contenu s'affiche.

☛ Pour plus de détails sur la notion de **mesures de flux applicatif**, voir [Qualifier un flux applicatif](#).

4. Sélectionnez les mesures de flux qui vous intéressent et cliquez sur **Ajouter**.  
Le flux applicatif s'affiche avec son contenu dans le diagramme.

### Qualifier un flux applicatif

Les **catégories de flux applicatif** permettent de définir les paramètres qui caractérisent les flux applicatifs décrits dans les scénario de flux.

Pour qualifier un flux applicatif à partir de **mesures de flux** :

1. Ouvrez la page de propriétés **Qualification** du flux applicatif qui vous intéresse.

2. Dans la section **Qualification de flux**, cliquez sur le bouton **Relier**. Une fenêtre de sélection s'ouvre et présente l'arbre des catégories de flux qui vous sont proposées.

☛ Pour accéder à la liste des catégories de flux applicatifs : à partir du menu de navigation **Administration**, sélectionnez **Catégorisation** et déployez le dossier **Catégorisation de schéma de mesure**. La liste des catégories de flux applicatif du référentiel s'affiche.

☛ Pour plus de détails sur la notion de **catégorie de flux applicatif**, voir [Définir les catégories de données](#).

### Associer une interface de service utilisée à un flux applicatif

📖 Une interface de service utilisée est associée à une interface de service. Elle permet de représenter des échanges complexes.

☛ Pour plus de détails sur la notion d'**interface de service**, voir [Décrire une interface de service](#).

Associer une interface de service à un flux applicatif permet d'établir une correspondance entre un diagramme de scénario de flux et le diagramme de structure d'une application ou d'un système applicatif.

L'utilisation peut être de type interface invoquée si le flux applicatif est de type **Appel/Requête** ou interface fournisseur si le flux applicatif est de type **Résultat/Fournisseur**.

Pour associer une **interface de service** à un **flux applicatif** :

1. Ouvrez la page **Interface de service utilisée** du flux applicatif qui vous intéresse.
2. Cliquez sur le bouton **Nouveau**. Une fenêtre s'ouvre pour vous proposer une liste d'**interfaces de service utilisées** dont le contenu est un point d'entrée ou un résultat.

### Créer un canal de flux applicatifs

📖 Un canal de flux applicatif permet de regrouper graphiquement plusieurs flux applicatifs en un seul flux.

Pour créer un canal de flux applicatif, vous devez d'abord créer le canal et relier ensuite les flux applicatifs qu'il regroupe.

Pour créer un **canal de flux applicatifs** :

1. Dans la barre d'objets du scénario de flux d'application, cliquez sur le bouton **canal de flux applicatifs**.
2. Cliquez sur le premier objet en communication et, en maintenant le bouton de la souris enfoncé, tirez un trait jusqu'à l'autre objet. Le canal de flux applicatif apparaît dans le diagramme.

Pour relier les flux applicatifs au **canal de flux applicatifs** :

1. Ouvrez la page de propriétés **Caractéristiques** du canal de flux applicatif.
2. Dans la section **Flux Groupés**, cliquez sur le bouton **Relier**. Une fenêtre de sélection s'ouvre et présente la liste des flux applicatifs non groupés qui existent entre l'émetteur et le récepteur.
3. Sélectionnez les flux que vous souhaitez regrouper et cliquez sur **OK**. Les contenus des flux applicatifs s'affichent dans la liste des **Flux Groupés**.

4. Cliquez sur le bouton **Rafraîchir les canaux** du diagramme.  
Les flux applicatifs regroupés dans le canal disparaissent et leur contenu s'affiche près du canaux.

☛ *Si vous détruisez le canal, seuls les flux applicatifs créés à partir de la liste des **Flux Groupés** sont détruits. Les flux applicatifs reliés sont affichés à nouveau si vous cliquez sur le bouton **Rafraîchir les canaux** du diagramme.*

## Créer un événement déclencheur système

Le processus de création d'un **événement déclencheur système** et d'un **événement déclenché système** est identique.

Pour créer un **événement déclencheur système** :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Événement déclencheur système**.
2. Positionnez l'objet sur le bord du cadre de l'objet décrit.  
Une fenêtre de création s'ouvre.
3. Cliquez sur la flèche à droite du champ **Contenu référencé** pour définir le contenu qui permet l'activation de ce déclencheur.
4. Cliquez sur **Ajouter**.

L'**événement déclencheur système** apparaît dans le diagramme.

Tout flux applicatif dont l'origine est l'**événement déclencheur système** sera associé au même contenu.

Pour créer un flux applicatif à partir d'un **événement déclencheur système** :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Participation à l'événement**.
2. Cliquez sur l'événement et, en maintenant le bouton de la souris enfoncé, tirez un trait jusqu'à l'objet récepteur du flux.  
Le flux applicatif s'affiche avec le contenu associé à l'événement.

## Réinitialiser les composants d'un scénario de flux

Si vous ajoutez à un scénario de flux un composant qui est lui même décrit par un scénario de flux, vous noterez qu'une nouvelle section est créée dans la page de propriétés **Caractéristiques** du composant que vous venez d'ajouter. Cette section vous permet de préciser le scénario de flux du composant qui a un sens dans le contexte du scénario de flux du système applicatif décrit.

Le bouton **Réinitialiser les composants** du scénario de flux du composant ajouté vous permet d'intégrer dans le diagramme les composants issus du scénario de flux de plus haut niveau.

## Ajouter un dépôt de données applicatives au scénario de flux d'application



*Un dépôt de données applicatives matérialise l'utilisation de données dans le contexte d'un composant logiciel (par exemple une application). Un dépôt de données applicatives fournit un mécanisme pour mettre à jour ou consulter des données stockées à l'intérieur du composant logiciel en cours.*

Un dépôt de données peut être local ou externe à l'application.

Pour ajouter, par exemple, un dépôt de données applicatives locales dans un scénario de flux d'application :

1. Dans la barre d'objets du scénario, cliquez sur le bouton **Dépôt de données applicatives local**.
2. Cliquez dans le cadre de l'application décrite.  
Une fenêtre d'ajout vous demande de choisir le **Type d'objet** qui représente la structure physique qui va supporter concrètement le dépôt de données applicatives.

☞ Pour plus de détails sur la gestion des dépôts de données, voir [Gérer les données](#).

3. En fonction du **Type d'objet**, sélectionnez ensuite l'objet qui vous intéresse.
4. Cliquez sur **OK**.  
Le dépôt de données applicatives local apparaît dans le diagramme, il porte le nom du domaine de données physique sélectionné.

## Créer un canal de données applicatives

Les applications, les systèmes applicatifs et les microservices peuvent accéder en lecture ou en écriture à un dépôt de données applicatives local ou externe.

Pour créer un canal de données applicatives qui représente un accès en lecture :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Canal de données applicatives**.
2. Tirez un lien entre le dépôt de données applicatives et l'objet qui lit les données.

Un canal de données applicatives apparaît dans le diagramme.

☞ Pour créer un lien en écriture, vous devez tirer un lien entre l'objet qui écrit et le dépôt de données applicatives.

## Utiliser les systèmes de communication

Un *système de communication* permet de décrire un processus d'intégration en utilisant la notation de l'**Enterprise Integration Pattern** (<http://www.enterpriseintegrationpatterns.com>).

📖 Un système de communication consiste à identifier et décrire les principaux systèmes d'intégration à partir des chaînes de communication applicative et des services de communication utilisés.

Cette représentation permet de modéliser le processus d'intégration des flux applicatifs utilisés dans les scénarios de flux de **HOPEX**.

☞ Pour plus de détails sur les scénarios de flux, voir [Utiliser un diagramme de scénario de flux d'application](#).

☞ Pour plus de détails sur la manière de relier un système de communication à un scénario de flux, voir [Relier une chaîne de communication applicative à partir d'un scénario de flux](#).

Un rapport vous permet d'obtenir la répartition des contenus multi-chaînes de communication applicative.

☞ Pour plus de détails, voir [Rationalisation des flux traités](#).

## Accéder à la liste des systèmes de communication

Pour accéder à la liste des systèmes de communication à partir du menu de navigation **Inventaires** :


- Sélectionnez **Logiciels > Systèmes de communication**.  
La liste des systèmes de communication du référentiel s'affiche.

## Les propriétés d'un système de communication

La description complète d'un système de communication est accessible à partir de sa pages de propriétés.


La page **Caractéristiques** des propriétés d'un système de communication permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de l'application spécifiée lors de sa création.
- le texte de sa description.
- la section **Chaîne de communication applicative** qui permet d'accéder à la liste des composants du système de communication décrit.


 Une chaîne de communication applicative décrit les mécanismes de routage d'un contenu entre un système émetteur et un système récepteur. Cette description inclut le routage, les canaux et la traduction des messages.

- la section **Service de communication** qui permet d'accéder à la liste des éléments de la chaîne de communication logicielle.  
Trois types de services peuvent être utilisés :


- **Canal de message**,

 Un canal de message est l'endroit où un message peut être écrit ou lu par d'autres composants dédiés à la communication. Il s'agit par exemple d'une file d'attente, d'un dossier, d'une base de données,...

- **Routeur de message**,


 Un routeur de message est une étape de communication qui permet d'identifier le chemin que le message doit prendre pour atteindre l'étape suivante.

- **Traducteur de message**,

 Un traducteur de message est une étape de communication qui traduit un message d'un format à un autre. Il peut être utilisé, par exemple, pour de la diversification ou de la conversion de types de données.

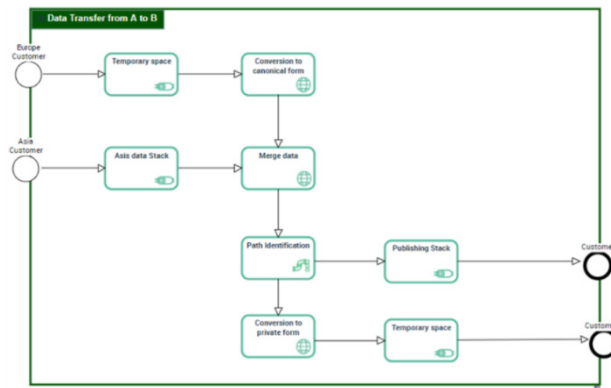
➡ Pour plus de détails sur les composants d'un système de communication, voir [Créer un diagramme de chaîne de communication applicative](#).

## Créer un diagramme de chaîne de communication applicative

 Un diagramme de chaîne de communication applicative décrit les mécanismes de routage d'un contenu entre un système émetteur et un



système récepteur. Cette description inclut le routage, les canaux et la traduction des messages.



Un diagramme de chaîne de communication applicative est composé de :

- Un **événement de début de communication** et un **événement de fin de communication** qui désignent les points de départ et de fin du processus d'intégration décrit.
- **Séquences de communication** qui décrivent l'enchaînement des étapes.  
**Canal de message** qui désigne l'endroit où une application peut lire ou écrire des informations.



Un canal de message est l'endroit où un message peut être écrit ou lu par d'autres composants dédiés à la communication. Il s'agit par exemple d'une file d'attente, d'un dossier, d'une base de données,...

- **Routeur de message** qui détermine le canal de destination à utiliser pour l'étape de transport suivante.



Un routeur de message est une étape de communication qui permet d'identifier le chemin que le message doit prendre pour atteindre l'étape suivante.

- **Traducteur de message** pour convertir un message d'un format à un autre.



Un traducteur de message est une étape de communication qui traduit un message d'un format à un autre. Il peut être utilisé, par exemple, pour de la diversification ou de la conversion de types de données.

### Créer une chaîne de communication applicative à partir du menu de navigation Déploiement

Pour créer une chaîne de communication applicative à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Logiciels > Systèmes de communication**.  
L'arbre des systèmes de communication du référentiel s'affiche.
2. Faites un clic droit sur le système de communication et cliquez sur **Nouveau > Chaîne de communication logicielle**.
3. Dans la fenêtre de création, spécifiez le **Nom** de la chaîne de communication logicielle créée et sélectionnez le **Contenu transféré**.

4. Cliquez sur **OK**.

La chaîne de communication applicative apparaît dans la liste.

☛ *Plusieurs chaînes de communication applicative peuvent être reliées à un même message.*

### **Créer un diagramme de chaîne de communication applicative à partir du menu de navigation Déploiement**

Pour créer un diagramme de chaîne de communication applicative à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Logiciels > Systèmes de communication**.  
L'arbre des systèmes de communication du référentiel s'affiche.
2. Dépliez le dossier **Chaîne de communication applicative**.
3. Sélectionnez la chaîne de communication applicative qui vous intéresse et cliquez sur **Créer diagramme**.
4. Dans la fenêtre de choix du type de diagramme, sélectionnez **Diagramme de chaîne de communication applicative**.
5. Cliquez sur **OK**.  
Le diagramme s'ouvre dans la zone d'édition.

### **Relier une chaîne de communication applicative à partir d'un scénario de flux**

A partir d'un scénario de flux, vous pouvez relier une nouvelle chaîne de communication applicative à partir d'un contenu.

☛ *Pour plus de détails sur les scénarios de flux, voir [Utiliser un diagramme de scénario de flux d'application](#).*

Pour relier une chaîne de communication applicative à partir d'un scénario de flux :

1. Ouvrez le diagramme de scénario de flux qui contient le contenu qui vous intéresse.
2. Ouvrez la page de propriétés **Intégration** du contenu qui vous intéresse.
3. Cliquez sur **Nouveau**.  
La fenêtre de création de la chaîne de communication applicative s'ouvre.
4. Spécifiez le **Nom** de la chaîne de communication applicative créée.
5. Dans la champs **Détenteur**, sélectionnez le système de communication auquel appartient la chaîne de communication applicative créée.
6. Cliquez sur **OK**.  
Le diagramme s'ouvre dans la zone d'édition.

---

## **Utiliser un diagramme de scénario de séquence de flux**

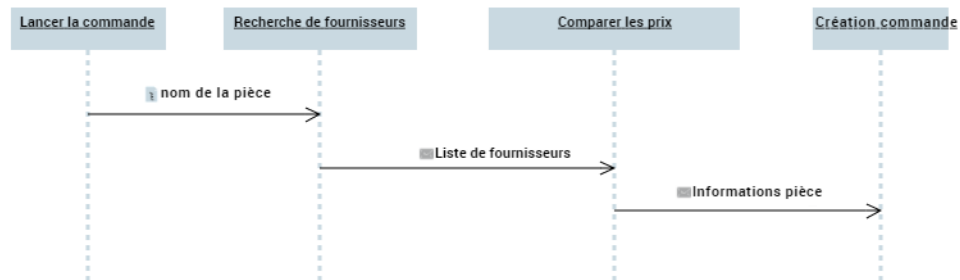
☛ *Pour visualiser les diagrammes de scénario de séquence de flux, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Activer les diagrammes de scénario de séquence de flux** est activée.*

Ce type de diagramme peut être construit pour un système applicatif, un environnement d'application, une application, un service applicatif ou un microservice.

Pour chaque contexte d'utilisation, vous pouvez créer des diagrammes de scénario de séquence de flux. Un diagramme de scénario de séquence de flux présente les mêmes échanges entre les éléments d'un système, en mettant en évidence leur chronologie. Les éléments du scénario sont représentés dans le diagramme par des lignes.

Un diagramme de scénario de séquence de flux contient :

- Des lignes qui définissent les participants de l'interaction de service tels que les instances d'applications, de services ou d'IHM.
- Différents types de messages échangés entre les participants.
- Des fonctionnalités avancées qui permettent de décrire de manière concise plusieurs séquences d'exécution.



Ce diagramme décrit le fonctionnement du cas d'utilisation "Commander des pièces non référencées" :

Quand une demande d'achat est saisie dans l'interface utilisateur, le nom de la pièce est reçu par le service "Recherche de fournisseurs", qui dresse la liste des fournisseurs proposant la pièce demandée.

Le service "Comparer les prix" recherche le produit le moins cher et envoie les informations à un service qui calcule le montant de la commande.

Une fois le montant établi, un dernier service passe la commande auprès de l'interface.


## Créer un diagramme de scénario de séquence de flux


Pour créer un scénario de séquence de flux d'environnement d'application, par exemple :

1. Ouvrez la page **Diagramme** de l'environnement d'application qui vous intéresse et cliquez sur **Nouveau**.
2. Dans la fenêtre de choix du type de diagramme, sélectionnez **Scénario de flux d'environnement d'application- Diagramme de scénario de séquence d'environnement d'application**.

## Instances d'application, de service applicatif ou d'IHM

Selon qu'il porte sur une IHM, une application ou un service applicatif, le diagramme de scénario d'interaction de service décrit les messages échangés entre des instances d'application, des instances de *service applicatif* ou des instances d'*IHM*.

 Une interface homme-machine permet de définir un écran d'une application ou d'un service applicatif.

 Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.

Pour créer une instance de service applicatif, par exemple :

1. Cliquez sur le bouton **Service applicatif** de la barre d'outils.
2. Cliquez dans le diagramme.  
Une boîte d'**Ajout de service applicatif** s'affiche.
3. Cliquez sur la flèche à droite du champ **Nom** et sélectionnez **Relier Service applicatif** dans la liste déroulante.  
La liste des services applicatifs accessibles à partir de la bibliothèque courante apparaît.
4. Sélectionnez le service applicatif de votre choix.
5. Cliquez sur le bouton **OK**.  
L'instance de service applicatif apparaît dans le diagramme.

## Instances de messages

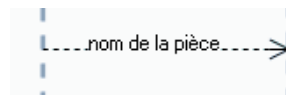
Les instances de messages définissent les informations qui circulent entre les instances d'applications, de services applicatifs ou d'IHM. La séquence décrite dans le diagramme de scénario de séquence de flux indique l'ordre d'envoi des messages.

Les instances de messages représentées dans le diagramme de scénario de séquence correspondent aux messages détenus par l'application et définis précédemment dans un autre diagramme.

Pour créer une instance de message :


1. Cliquez sur la bobine de la barre d'outils.
2. Cliquez sur la ligne pointillée sous le premier objet et, en maintenant le bouton gauche de la souris enfoncé, tracez une ligne jusqu'au deuxième objet.
3. Relâchez le bouton gauche de la souris.

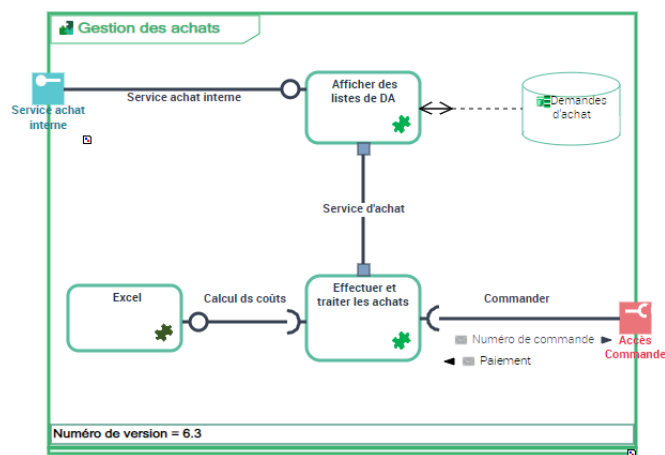
L'instance de message échangée entre les deux objets se dessine.



# DÉCRIRE LA STRUCTURE ET LES SERVICES D'UNE APPLICATION

## Le diagramme de structure d'application

 Un diagramme de structure d'application montre graphiquement les composants de premier niveau d'une application, les points d'accès (point de service et de requête) et les connexions entre les composants.



L'application de gestion des achats s'appuie sur deux services applicatifs spécialisés : "Afficher des listes de DA" et "Affecter et traiter DA". Le service applicatif "Affecter et traiter DA" utilise le microservice Excel.

## Créer un diagramme de structure d'application

Pour créer un diagramme de structure d'application, par exemple :

1. Sélectionnez l'application qui vous intéresse et cliquez sur **Créer diagramme**.
2. Dans la fenêtre de choix, sélectionnez **Diagramme structuré > Architecture interne**.  
Le diagramme de structure d'application s'ouvre dans la zone d'édition.  
Le cadre de l'application décrite apparaît dans le diagramme.

## Les composants d'un diagramme de structure d'application

Un diagramme de structure d'application est composé de :

- **services applicatifs** qui représentent les services applicatifs utilisés et déployés avec l'application.

Dans l'exemple, il s'agit des services "Afficher des listes de DA" et "Affecter et traiter DA".



*Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.*

- **microservices** qui représentent les services utilisés indépendants de l'application.

Dans l'exemple, il s'agit de l'application Excel.



*Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.*

- points d'accès, de requête et de service.



*Pour plus de détails, voir [Décrire les points de service et de requête](#).*

- **interactions de service** entre les composants.



*Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*

- **dépôts de données physiques** utilisés par l'application.




*Pour plus de détails, voir [Gérer les données](#).*

## Ajouter un service applicatif dans un diagramme de structure d'application

Pour décrire qu'une application utilise un service applicatif, vous allez :



1. Dans la barre d'objets du diagramme de structure d'application, cliquez sur le bouton **Service applicatif** et cliquez dans le cadre de l'application décrite.  
Une fenêtre d'ajout vous demande de choisir le **Nom** d'un service applicatif existant.
2. Sélectionnez un service applicatif existant.
3. Cliquez sur **OK**.  
Le service applicatif apparaît dans le diagramme.

## Décrire un service applicatif avec HOPEX IT Architecture

 Un service applicatif est un composant logiciel d'une application, qui ne peut être déployé seul, et qui réalise un sous-ensemble des fonctionnalités de l'application pour des utilisateurs de cette application ou à l'intérieur de cette application (ou d'une autre application). Ceci inclut les programmes batch.

### Les diagrammes d'un service applicatif

Un *service applicatif* est décrit par plusieurs types de diagramme :

- un *diagramme de structure de service applicatif* permet de représenter les interactions de service entre les différents composants du service applicatif sous forme d'interfaces de service.  
 Pour plus de détails, voir [Utiliser le diagramme de structure de service applicatif](#).
- un *scénario de flux de service applicatif* présente les flux échangés entre les services applicatifs ou les microservices utilisés par ce service applicatif. Un scénario peut représenter un cas d'utilisation particulier du service applicatif ou plus globalement l'ensemble des flux échangés à l'intérieur de ce service applicatif.  
 Pour plus de détails, voir [Utiliser un diagramme de scénario de flux d'application](#).

### Accéder à la liste des services applicatifs









Pour accéder à la liste des services applicatifs à partir du menu de navigation **Inventaires** :

- Sélectionnez **Logiciel > Services applicatifs**.  
La liste des services applicatifs apparaît dans la zone d'édition.

### Les propriétés d'un service applicatif

La description complète d'un service applicatif est accessible à partir de ses pages de propriétés.

La page **Caractéristiques** des propriétés d'un service applicatif permet d'accéder à :

- son **Détenteur**, par défaut lors de la création du service applicatif, il s'agit de la bibliothèque courante.
- son **Nom**,
- le **Type**,
- la **Visibilité**,
- le **Statut de revue**,
- le texte de sa **Description**,
- la section **Périmètre fonctionnel** permet de décrire :
  - Les **capacités métier** couvertes par le service applicatif,  
 Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).  
 Pour plus de détails sur les capacités métier, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).
  - Les **fonctionnalités implémentées** sont mises en œuvre par le service applicatif,  
 Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.  
 Pour plus de détails sur les fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#).  
 Pour plus de détails sur les mises en œuvre, voir [Utiliser les mécanismes de mise en œuvre](#).
- la section **Cas d'utilisation**, voir [Créer un diagramme de cas d'utilisation d'une application](#).
- la section **Responsabilité** permet d'identifier la (ou des) personnes responsable(s) du service applicatif :
  - Concepteur de logiciel
  - Propriétaire local d'application  
 Pour plus de détails sur ces rôles, voir [Les rôles métier de HOPEX IT Architecture](#).
- la section **Technologies** permet d'accéder à la liste des **technologies logicielles** utilisées par le service applicatif.  
 Une technologie est une définition ou un format qui a été approuvé par une organisation de standardisation ou qui est accepté comme standard de fait par l'industrie.  
 Pour plus de détails sur les technologies logicielles, voir [Décrire une technologie logicielle](#).
- la section **Risques** présente les risques associés à l'application, voir [Décrire un environnement d'application avec HOPEX IT Architecture](#).

Avec **HOPEX IT Architecture** un service applicatif est décrit par d'autres pages de propriétés, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Utiliser le diagramme de structure de service applicatif


 Pour plus de détails sur l'utilisation d'un diagramme de structure, voir [Le diagramme de structure d'application](#).



Avec **HOPEX IT Architecture**, les composants d'un service applicatif peuvent être décrits par un diagramme de structure de service applicatif.


Un diagramme de structure de service applicatif est composé de :

- des services applicatifs,
- des microservices,
- des dépôts de données physiques, voir [Gérer les données](#).
- points d'accès, de requête et de service, voir [Créer un point de service ou un point de requête](#).
- *interactions de service* entre les composants

 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.




---

## Décrire un microservice avec HOPEX IT Architecture

 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

### Les diagrammes d'un microservice

Un *microservice* est décrite par plusieurs types de diagramme :

- un *diagramme de structure du microservice* permet de représenter les interactions de service entre les différents composants du microservice sous forme d'interfaces de service.  
 Pour plus de détails, voir [Utiliser le diagramme de structure de microservice](#).
- un *scénario de flux de microservice* présente les flux échangés entre les éléments du microservices dans un contexte donné.  
 Pour plus de détails, voir [Utiliser un diagramme de scénario de flux d'application](#).
- une *architecture de déploiement de microservice* qui permet de représenter les éléments techniques qui supportent le microservice.  
 Pour plus de détails, voir [Décrire une architecture de déploiement d'une application](#).

### Accéder à la liste des microservices

Pour accéder à la liste des microservices à partir du menu de navigation **Inventaires** :

- Sélectionnez **Logiciel > Microservices**.  
La liste des microservices apparaît dans la zone d'édition.

## Les propriétés d'un microservice avec HOPEX IT Architecture

La description complète d'un microservice est accessible à partir de ses pages de propriétés.

La page **Caractéristiques** des propriétés d'un microservice permet d'accéder à :

- son **Détenteur**, par défaut lors de la création du microservice, il s'agit de la bibliothèque courante.
- son **Nom**,
- le **Statut de revue**,
- le texte de sa **Description**,
- la section **Périmètre fonctionnel** permet de décrire :
  - Les **capacités métier** couvertes par le microservice,



*Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).*

➡ Pour plus de détails sur les capacités métier, voir [Décrire les capacités métier avec HOPEX IT Architecture](#).

- Les **fonctionnalités implémentées** sont mises en œuvre par le microservice,



*Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.*

➡ Pour plus de détails sur les fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#).

➡ Pour plus de détails sur les mises en œuvre, voir [Utiliser les mécanismes de mise en œuvre](#).

- la section **Cas d'utilisation**, voir [Créer un diagramme de cas d'utilisation d'une application](#).
- la section **Responsabilité** permet d'identifier la (ou des) personnes responsable(s) du service applicatif :

- Concepteur de logiciel
- Propriétaire local d'application

➡ Pour plus de détails sur ces rôles, voir [Les rôles métier de HOPEX IT Architecture](#).

- la section **Technologies** permet d'accéder à la liste des **technologies logicielles** utilisées par le microservice.



*Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.*

➡ Pour plus de détails sur les technologies logicielles, voir [Décrire une technologie logicielle](#).

- la section **Risques** présente les risques associés à l'application, voir [Décrire un environnement d'application avec HOPEX IT Architecture](#).

Avec **HOPEX IT Architecture** un microservice est décrit par d'autres pages de propriétés, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Utiliser le diagramme de structure de microservice

☛ Pour plus de détails sur l'utilisation d'un diagramme de structure, voir [Le diagramme de structure d'application](#).

Avec **HOPEX IT Architecture**, les composants d'un microservice peuvent être décrits par un diagramme de structure de microservice.

Un diagramme de structure de microservice est composés de :

- des services applicatifs,
- 
- des dépôts de données physiques, voir [Gérer les données](#).
- points d'accès, de requête et de service, voir [Créer un point de service ou un point de requête](#).
- *interactions de service* entre les composants

📖 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

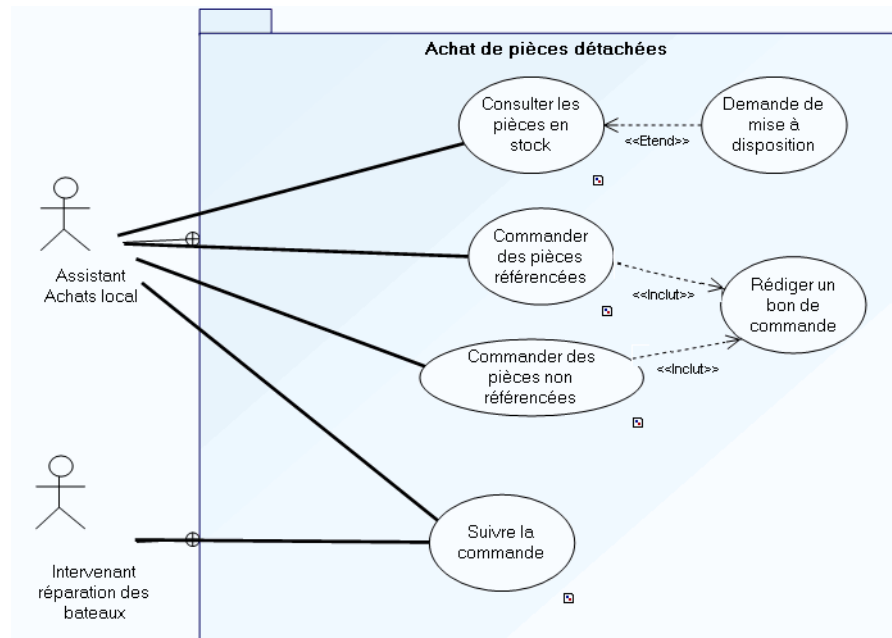
---

## Créer un diagramme de cas d'utilisation d'une application

☛ Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte de Solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.

Un diagramme de *cas d'utilisation* permet de décrire les interactions de service entre un système et les acteurs de l'organisation.

📖 Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.



Le système est utilisé pour la consultation des pièces en stock et la commande de nouvelles pièces détachées.

La consultation des pièces en stock est effectuée par l'assistant achat local. Suite à la consultation, l'assistant peut faire une demande de mise à disposition.

Deux types de commandes sont possibles; une commande de pièces déjà référencées ou une commande de pièces non référencées. Dans les deux cas, un bon de commande doit être rédigé.

Le suivi de la commande est assuré à la fois par l'assistant achat local et l'intervenant chargé de la réparation des bateaux.

➡ Pour plus de détail sur les diagrammes de cas d'utilisation, voir [Le diagramme de cas d'utilisation](#).

Pour créer un diagramme de cas d'utilisation à partir d'une application, par exemple :

1. Ouvrez la page **Caractéristiques** de l'objet concerné et dépliez la section **Cas d'utilisation**.
2. Ouvrez la page **Diagramme** du cas d'utilisation qui vous intéresse.
3. Cliquez sur **Créer un diagramme** et sélectionnez **Diagramme de cas d'utilisation**.

Le diagramme créé s'ouvre dans la zone d'édition.

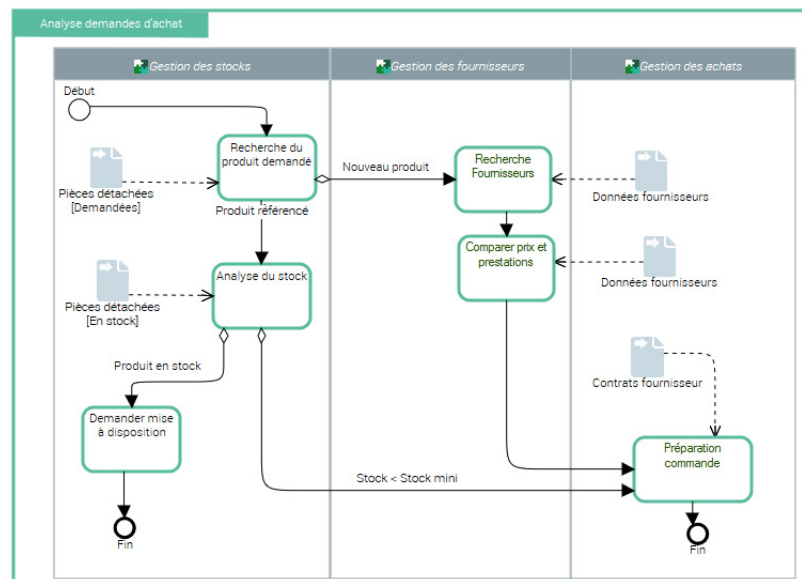
## DÉCRIRE LES PROCESSUS APPLICATIFS

En phase de spécification détaillée, le déroulement des tâches mises en œuvre dans un service applicatif peut également être modélisé par un processus applicatif. Plus généralement, le fonctionnement d'un élément d'architecture peut être décrit par un processus applicatif qui modélise, par exemple, l'enchaînement des écrans présentés à l'utilisateur.

### Exemple de diagramme de processus applicatif

Le diagramme ci-après représente le traitement des demandes d'achat.

- Une recherche du produit est effectuée à partir de la base des produits Référencés.
- Si le produit est nouveau, une recherche de fournisseur et une étude comparative des prix est menée. Puis une commande est émise et le processus prend fin.
- Si le produit est référencé, l'étude de l'état des stocks est réalisée.
- Si le stock est suffisant, une demande de mise à disposition est activée et le processus prend fin.
- Si le stock est inférieur au stock minimum, une commande est émise vers le fournisseur et le processus prend fin.



## Gérer les processus applicatifs avec HOPEX IT Architecture



Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.



Pour plus de détails sur la construction d'un diagramme de processus applicatif, voir dans le guide **HOPEX Business Process Analysis** le paragraphe "Gérer un processus applicatif".



### Accéder aux processus applicatifs

Pour accéder à la liste des processus applicatifs à partir du menu de navigation

#### **Inventaires :**

- Sélectionnez **Logiciel > Processus applicatifs**.  
La liste des processus applicatifs apparaît.

### Créer un diagramme de processus applicatif

Le diagramme de processus applicatif utilise la notation proposée par la norme BPMN. L'algorithme du processus applicatif peut être exprimé par une succession de tâches et de décisions.

Un diagramme de processus applicatif peut être créé et mis à jour aussi en mode saisie tabulaire.



Pour plus de détails sur l'utilisation de la saisie tabulaire, voir le chapitre "Saisie de diagramme en mode tabulaire" du guide **HOPEX Common Features**.

Pour créer un diagramme de processus applicatif :

1. Sélectionnez le processus applicatif qui vous intéresse et cliquez sur le bouton **Créer diagramme**.  
La fenêtre de choix du type de diagramme s'ouvre.
2. Cliquez sur **Diagramme de processus applicatif**.  
Le diagramme s'ouvre dans la zone d'édition. Le cadre de processus applicatif décrit apparaît dans le diagramme.

### Créer une tâche

Les **tâches** correspondent aux étapes du traitement.




Une tâche est une étape élémentaire à l'intérieur d'un processus applicatif. Une tâche est utilisée quand une étape d'un processus n'est plus décomposée en un niveau de description plus fin. Généralement un utilisateur ou un service applicatif exécutent cette tâche.

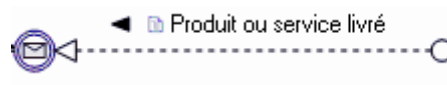
Pour créer une tâche :


1. Dans la barre d'insertion du diagramme, cliquez sur l'icône **Tâche** puis cliquez dans le diagramme.
2. Saisissez le nom de la tâche et cliquez sur **OK**.  
La tâche apparaît dans le diagramme.

## Les flux

Les **flux** représentent les échanges entre le processus applicatif et l'extérieur.


 Un flux représente la circulation d'information à l'intérieur d'une interface de service. Un flux transporte son contenu.




 Un flux peut être lié à un événement de type message.

## Les enchaînements

L'organisation des tâches qui composent un processus applicatif est représentée par des **enchaînements** entre les tâches.


 Un enchaînement est utilisé pour montrer l'ordre d'exécution des étapes d'une interface de service. Un enchaînement n'a qu'une source et une cible.



 Pour plus d'informations sur la gestion des enchaînements, voir le chapitre "Décrire l'enchaînement des opérations" du guide HOPEX Business Process Analysis.

## Les événements

Les **événements** représentent des faits qui se produisent durant l'exécution du processus.

 Un événement représente un fait ou une action se produisant dans le système, par exemple - modification de l'adresse client. Il est géré par un broker. Pour signaler qu'elle peut produire un événement, une application déclare qu'elle le publie. Si elle est intéressée par un événement, une application déclare qu'elle y souscrit.

Ce peut être, par exemple, le début ou la fin du processus applicatif.



Début




Fin

L'événement peut également être l'envoi ou la réception d'un flux.



### Les branchements

 Un branchement est un élément de modélisation utilisé pour contrôler des enchaînements qui convergent ou qui divergent à l'intérieur d'un processus.

➡ Pour plus d'informations sur la gestion des branchements, voir le chapitre "Utiliser les branchements" du guide HOPEX Business Process Analysis.

### Créer un participant dans un diagramme de processus applicatif

Dans un diagramme de processus applicatif, un participant permet de regrouper les tâches affectées à une application ou un service.

Pour créer un participant

1. Dans la barre d'insertion du diagramme, cliquez sur la flèche à droite de l'icône **Participant (Application)**.
2. Dans la liste des choix proposés sélectionnez, par exemple, **Participant (Application)** et cliquez dans le diagramme. La fenêtre de création d'un participant apparaît.
3. Cliquez sur la flèche descendante du champ **Application** et sélectionnez les applications qui vous intéressent.
4. Cliquez sur **OK**.

Le participant créé apparaît dans le diagramme avec une entête portant le nom d'une application assignée.

➡ Pour poser un participant dont l'assignation n'est pas encore connue, sélectionnez l'icône **Participant**.

Pour affecter une tâche à un participant :

- ☐ Posez la tâche dans le cadre du participant.

---

## Spécifier le comportement d'une tâche d'un processus applicatif

Conformément à la norme BPMN, un processus peut avoir des comportements différents. Avec **HOPEX Business Process Analysis**, ces comportements sont disponibles pour les processus, les opérations, les processus applicatifs et les tâches.



## Les comportements

Les comportements proposés sont :

- **Transaction** : une transaction est un ensemble d'activités coordonnées entre elles en vue d'obtenir un résultat cohérent et vérifiable.
- **Boucle** : une boucle est une étape d'un processus qui est répétée tant qu'une condition est vérifiée.
  - "Faire tant que" : la condition est évaluée avant la première exécution.
  - "Faire jusqu'à ce que" : la condition est évaluée après la première exécution. Dans ce cas, le processus est toujours réalisé au moins une fois.

Le prédicat permet de spécifier la condition d'exécution de la boucle.

- **Ad hoc** : les étapes d'un processus de ce type ne sont pas contrôlées, ni ordonnées. L'ordre de réalisation est déterminé par les participants qui exécutent le processus.
- **Multiple** : le processus est répété un nombre prédéfini de fois qui est évalué avant sa première exécution. Il est possible de spécifier le type d'exécution :
  - "Parallèle" : toutes les exécutions sont réalisées en même temps
  - "Séquentielle" : les exécutions sont réalisées les unes après les autres.
- **Compensation** : une compensation définit l'ensemble des activités qui sont réalisées après l'annulation d'une transaction pour compenser les activités réalisées durant le déroulement normal du processus.

Pour décrire, par exemple, qu'un processus est exécuté en boucle :

1. Ouvrez la page de propriétés **Caractéristiques** du processus.
2. Dans la section **Détails**, à droite du champ **Boucle**, sélectionnez le type de boucle correspondant au comportement du processus et ajoutez le texte de la condition.  
La forme du processus est modifiée pour faire apparaître le symbole de la boucle.



## Type de tâche


Pour préciser le type d'une tâche :

1. Ouvrez la page de propriétés **Caractéristiques** du processus.

2. Dans la section **Détails**, cliquez sur la flèche à droite du champ **Type de tâche**.

La liste des types de tâche s'affiche.

- **Réception** : tâche élémentaire qui attend l'arrivée d'un message en provenance d'un participant externe au processus. Quand le message a été reçu, la tâche est terminée.
- **Emission** : tâche qui envoie un message à un participant externe au processus. Quand le message a été envoyé, la tâche est terminée.
- **Manuelle** : tâche réalisée sans l'aide d'un moteur d'exécution automatique d'un processus ou d'une application informatique.
- **Règle métier** : tâche d'exécution d'une règle métier qui dispose d'un moteur de règles qui traite les données en entrée et retourne en sortie les résultats du calcul effectué.
- **Script** : tâche réalisée par un moteur d'exécution de processus. Le concepteur définit un script dans un langage que le moteur est capable d'interpréter. Quand la tâche est prête à démarrer, le moteur exécute le script. La tâche est terminée quand l'exécution du script est terminée.

 La forme du processus est modifiée pour faire apparaître le symbole associé au type de tâche.

---

## Modéliser les tâches d'un processus applicatif


La phase d'analyse fonctionnelle décrit les processus applicatifs mis en œuvre dans les différents cas d'utilisation d'une application ou d'un service.

Un diagramme de processus applicatif permet de spécifier l'enchaînement des tâches à réaliser de manière à ce que l'utilisateur puisse vérifier que l'application décrite réponde à son besoin.

### Exemple de modélisation fonctionnelle

Les *processus applicatifs* utilisés pour l'analyse fonctionnelle d'un projet sont rangés dans un paquetage.

Dans l'exemple du projet d'automatisation du traitement des demandes d'achat les processus applicatifs sont rangés dans le paquetage "Demandes d'achat urgentes".

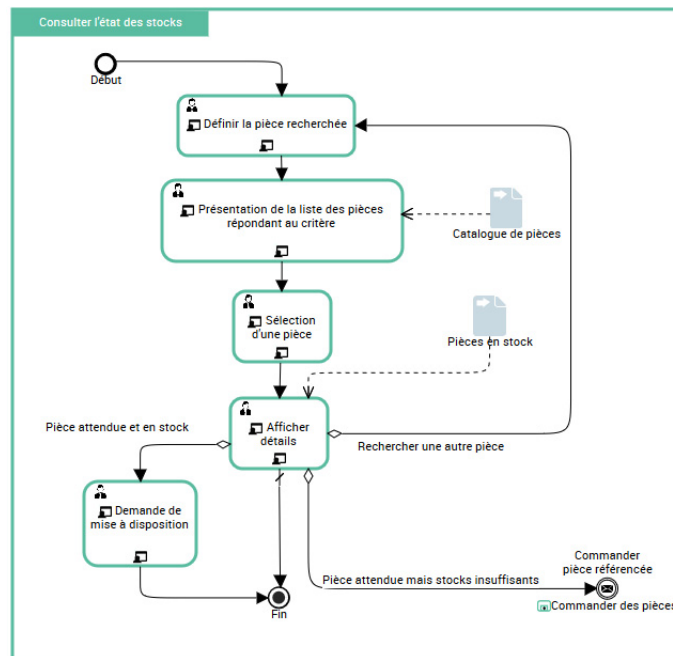
 Un processus applicatif est la représentation exécutable d'un processus. Les éléments formalisant un processus applicatif sont les suivants : les événements du workflow, les tâches à accomplir durant le traitement, les éléments algorithmiques d'enchaînement des tâches, les flux d'information échangés avec les participants.

### Afficher le diagramme d'une étape de processus applicatif

Pour afficher le diagramme décrivant en détail une étape du processus applicatif :

1. Cliquez sur le processus applicatif, par exemple "Consulter l'état des stocks" pour faire apparaître son menu contextuel.

2. Sélectionnez **Diagramme de processus applicatif**.  
Le diagramme associé au processus s'ouvre.



La consultation de l'état des stocks commence par l'affichage d'un écran permettant l'identification de la pièce recherchée. La liste des pièces trouvées dans le catalogue est présentée dans l'écran suivant.

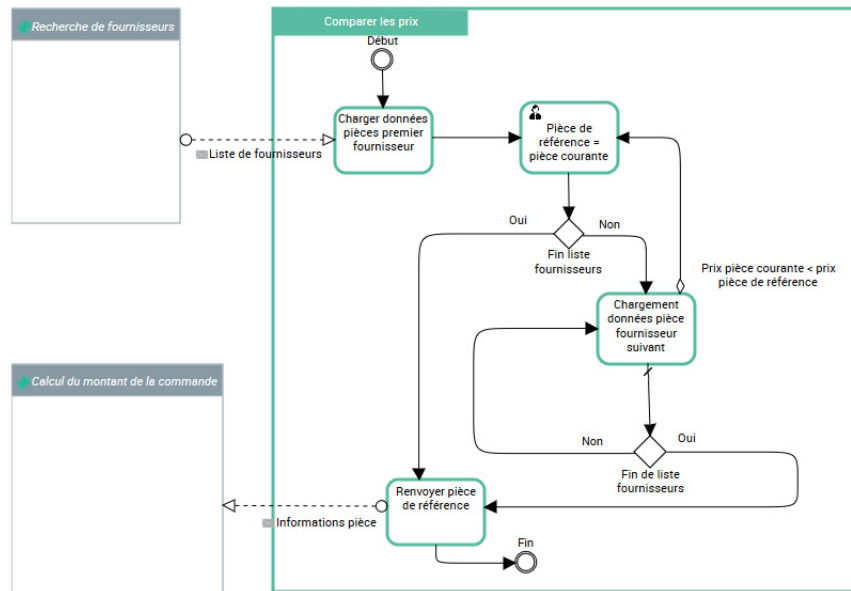
Quand l'utilisateur a sélectionné la pièce qui l'intéresse, les informations de détails sont affichées. A partir de cet écran, il est possible de rechercher des informations sur une autre pièce, ou bien de demander la mise à disposition de la pièce, ou bien encore de commander cette pièce.

## Modéliser les tâches d'un service applicatif

La phase d'analyse détaillée des composants système impactés par projet consiste à modéliser finement le fonctionnement des services applicatifs.

Dans le contexte de l'exemple d'automatisation du traitement des demandes d'achat urgentes, le service de

comparaison des prix est représenté par un processus applicatif.




Ce diagramme décrit l'algorithme du service "Comparer les prix" qui doit retourner la référence de la pièce dont le prix est le plus bas.

La liste des fournisseurs de la pièce recherchée est donnée en entrée. La pièce proposée par le premier fournisseur de cette liste devient la pièce de référence. Ensuite, et tant que la liste des fournisseurs n'est pas vide, les données concernant la pièce recherchée sont analysées. Si le prix de la pièce courante est moins élevé que le prix de la pièce de référence, la pièce de référence devient la pièce courante. Quand la liste complète des fournisseurs a été analysée, les informations concernant la pièce de référence sont envoyées vers le service "Calcul du montant de la commande".


## GÉRER LES DONNÉES

Les *dépôts de données* sont utilisés dans les diagrammes d'architecture pour représenter les données qui doivent être stockées pour être partagées entre différents composants.


 Un dépôt de données fournit un mécanisme pour mettre à jour ou consulter des données qui vont persister au-delà de la portée du processus en cours. Il permet de stocker les données des flux en entrée et de les retransmettre par l'intermédiaire d'un ou plusieurs flux en sortie.

Un *dépôt de données* peut être supporté par différents types d'objet :


- un *domaine de données*

 Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.


- une *structure de fichier*

 Une structure de fichier représente un dossier de fichiers ou un fichier utilisé dans l'architecture technique d'une application.

- un *domaine physique (NoSQL)*

 Un domaine physique (NoSQL) représente un ensemble de données stockées dans un système de gestion de base de données NoSQL et utilisé dans l'architecture technique d'une application.


- un *schéma relationnel*

 Un schéma relationnel représente un ensemble de données stockées dans un système de gestion de base de données relationnel et utilisé dans l'architecture technique d'une application.

---


### Utiliser les dépôts de données

Un dépôt de données permet de référencer dans un processus ou un système applicatif des données persistantes, définies dans un domaine de données.


 Un dépôt de données fournit un mécanisme pour mettre à jour ou consulter des données qui vont persister au-delà de la portée du processus en cours. Il permet de stocker les données des flux en entrée et de les retransmettre par l'intermédiaire d'un ou plusieurs flux en sortie.

### Introduction au concept de dépôt de données


Si vous décrivez un système applicatif logique, seuls des *dépôts de données logiques* peuvent être utilisés.


 Un dépôt de données logique matérialise l'utilisation des données par les systèmes applicatifs sans considérer comment leur accès sera concrètement mis en œuvre.


Si vous décrivez un système applicatif, seuls des *dépôts de données physiques* peuvent être utilisés.

 *Un dépôt de données physique représente l'implémentation d'un dépôt de données logique.*


Si vous décrivez un scénario de séquences ou un scénario de flux, vous ne pouvez utiliser que les *dépôts de données applicatives*.


 *Un dépôt de données applicatives matérialise l'utilisation de données dans le contexte d'un composant logiciel (par exemple une application). Un dépôt de données applicatives fournit un mécanisme pour mettre à jour ou consulter des données stockées à l'intérieur du composant logiciel en cours.*

 *Les diagrammes de scénario de flux qui décrivent les flux échangés dans différents scénarios d'utilisation de l'objet décrit.*

 *Les diagrammes de scénario de séquences de flux qui décrivent la chronologie des flux échangés dans différents scénarios d'utilisation de l'objet décrit.*

Enfin, vous pouvez également distinguer les dépôts de données locaux à un système des dépôts de données externes qui sont positionnés à la frontière des diagrammes.

 *Un dépôt de données local représente un dépôt de données utilisé uniquement à l'intérieur du système décrit.*

 *Un dépôt de données externe représente un dépôt de données utilisé à l'intérieur et à l'extérieur du système décrit.*

## Les contextes d'utilisation

Le tableau ci-dessous présente la liste des diagrammes qui utilisent les différents types de dépôts de données.

Type de dépôt de données	Diagrammes
Dépôt de données logique	Diagrammes de structure de système applicatif logique
Dépôt de données physique	Diagrammes de structure <ul style="list-style-type: none"><li>- d'application,</li><li>- de système applicatif,</li><li>- de services applicatifs,</li><li>- de microservices.</li></ul>
Dépôt de données applicatives	Diagrammes de scénario de séquence <ul style="list-style-type: none"><li>- d'application,</li><li>- de système applicatif,</li><li>- de services applicatifs,</li><li>- de microservices,</li></ul> Diagrammes de scénario de flux <ul style="list-style-type: none"><li>- d'application,</li><li>- de système applicatif,</li><li>- de service applicatif,</li><li>- de microservice.</li></ul>

## Créer un dépôt de données local



Un dépôt de données local représente un dépôt de données utilisé uniquement à l'intérieur du système décrit.

Pour créer, par exemple, un **dépôt de données physique local** à partir d'un diagramme de structure de système applicatif :

1. Ouvrez le diagramme qui vous intéresse.
2. Dans la barre d'objets du diagramme, cliquez sur le bouton **Dépôt de données physique local** et sélectionnez le **Type d'objet** qui va supporter concrètement le dépôt de données que vous décrivez.

☛ Pour plus de détails sur les types d'objets proposés voir [Accéder aux supports des dépôts de données](#).

3. Cliquez dans le cadre du système applicatif décrit.
4. En fonction du **Type d'objet**, sélectionnez ensuite l'objet qui vous intéresse.
5. Cliquez sur **OK**.  
Le dépôt de données physique local apparaît dans le diagramme, il porte le nom de l'objet associé.

## Créer un dépôt de données externe



Un dépôt de données externe représente un dépôt de données utilisé à l'intérieur et à l'extérieur du système décrit.

Pour créer, par exemple, un dépôt de données physique externe à partir d'un diagramme de structure de système applicatif :

1. Ouvrez le diagramme qui vous intéresse.
2. Dans la barre d'objets du diagramme, cliquez sur le bouton **Dépôt de données physique externe** et sélectionnez le **Type d'objet** qui va supporter concrètement le dépôt de données que vous décrivez.

☛ Pour plus de détails sur les types d'objets proposés voir [Accéder aux supports des dépôts de données](#).

3. Cliquez sur le bord du cadre du système applicatif décrit.
4. En fonction du **Type d'objet**, sélectionnez ensuite l'objet qui vous intéresse.
5. Cliquez sur **OK**.  
Le dépôt de données physique local apparaît dans le diagramme, il porte le nom de l'objet associé.

## Décrire l'accès à un dépôt de données

Pour créer un accès en lecture au dépôt de données :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Lien**
2. Tirez un lien entre le dépôt de données et l'entité qui lit les données (composant ou utilisation de système applicatif).

Un **Accès en lecture au stockage de données** est automatiquement créé avec le lien orienté du dépôt de données vers l'entité est créé.

☛ Pour créer un lien en écriture, vous devez tirer un lien entre l'entité qui écrit et le dépôt de données. Un **Accès en écriture au stockage de données** est alors automatiquement créé.

## Accéder aux supports des dépôts de données

Un *dépôt de données* peut être supporté par différents types d'objet :

- un *domaine de données*



Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.

- une *structure de fichier*



Une structure de fichier représente un dossier de fichiers ou un fichier utilisé dans l'architecture technique d'une application.

- un *domaine physique (NoSQL)*



Un domaine physique (NoSQL) représente un ensemble de données stockées dans un système de gestion de base de données NoSQL et utilisé dans l'architecture technique d'une application.

- un *schéma relationnel*



Un schéma relationnel représente un ensemble de données stockées dans un système de gestion de base de données relationnel et utilisé dans l'architecture technique d'une application.

## Accéder aux domaines de données avec HOPEX IT Architecture



Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles. Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.

✎ Pour plus de détails sur les domaines de données, voir le guide **HOPEX Data Architecture**.

Pour accéder à la liste des *domaines de données* à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Données > Dictionnaires de données**.  
L'arbre des dictionnaires de données apparaît.
2. Dépliez le dossier **Paquetage**.
3. Dépliez le dossier du paquetage qui vous intéresse.
4. Dépliez le dossier **Briques de données logiques**.  
La listes des domaines de données du paquetage apparaît.

## Accéder à la liste des structures de fichier avec HOPEX IT Architecture




Une structure de fichier représente un dossier de fichiers ou un fichier utilisé dans l'architecture technique d'une application.

Pour accéder à la liste des *structures de fichier* à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Données > Dictionnaires de données**.  
L'arbre des dictionnaires de données apparaît.
2. Dépliez le dossier **Catalogue des briques NoSQL**.  
La liste des *structures de fichier* apparaît.




## Accéder aux domaines physiques (NoSQL) avec HOPEX IT Architecture

 *Un domaine physique (NoSQL) représente un ensemble de données stockées dans un système de gestion de base de données NoSQL et utilisé dans l'architecture technique d'une application.*

Pour accéder à la liste des *domaines physiques (NoSQL)* à partir du menu de navigation **Inventaires** :

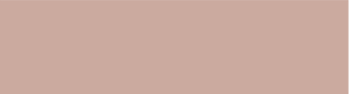
1. Sélectionnez **Données > Dictionnaires de données**.  
L'arbre des dictionnaires de données apparaît.
2. Dépliez le dossier **Catalogue des briques NoSQL**.  
La liste des *domaines physiques (NoSQL)* apparaît.

## Accéder à la liste des schémas relationnels avec HOPEX IT Architecture

 *Un schéma relationnel représente un ensemble de données stockées dans un système de gestion de base de données relationnel et utilisé dans l'architecture technique d'une application.*

Pour accéder à la liste des *schémas relationnels* à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Données > Dictionnaires de données**.  
L'arbre des dictionnaires de données apparaît.
2. Dépliez le dossier **Base de données**.
3. Dépliez le dossier de la base de données qui vous intéresse.
4. Dépliez le dossier **Structure physique BDR**.  
La listes des *schémas relationnels* de la base de données apparaît.



# MODÉLISER LES ARCHITECTURES DE DÉPLOIEMENT



Une architecture de déploiement permet de décrire l'ensemble des éléments qui doivent être déployés pour mettre en œuvre une architecture applicative : *modules de déploiement d'une application*, *modules de déploiement des données* ainsi que les *connexions de module* utilisées pour les échanges de données.


Plusieurs angles de vue sont proposés avec **HOPEX IT Architecture** :

- ✓ L'*environnement de déploiement d'une application* qui permet de représenter les déploiements d'applications partenaires ainsi que les microservices requis autour de l'application sujet.
- ✓ L'*architecture de déploiement du système d'application* qui permet de représenter l'ensemble des architectures de déploiement d'application qui doivent être coordonnées pour couvrir les dépendances requises entre elles.
- ✓ L'*architecture de déploiement d'une application* qui présente la liste des modules de déploiement.

Les points traités ici sont les suivants :

- ✓ Décrire une architecture de déploiement d'une application ;
- ✓ Décrire un environnement de déploiement d'une application ;
- ✓ Décrire une architecture de déploiement d'un système applicatif ;
- ✓ Les modèles d'architecture de déploiement
- ✓ Décrire les technologies logicielles ;
- ✓ Utiliser les services cloud.

# DÉCRIRE UNE ARCHITECTURE DE DÉPLOIEMENT D'UNE APPLICATION

 Une architecture de déploiement d'une application décrit une configuration possible pour déployer une application. Elle présente les modules de déploiement à héberger, préconise un type d'hébergement et les connexions techniques requises pour qu'ils puissent communiquer entre eux (protocoles de communication, numéros de port,...) . Une application peut avoir plusieurs architectures de déploiement possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)

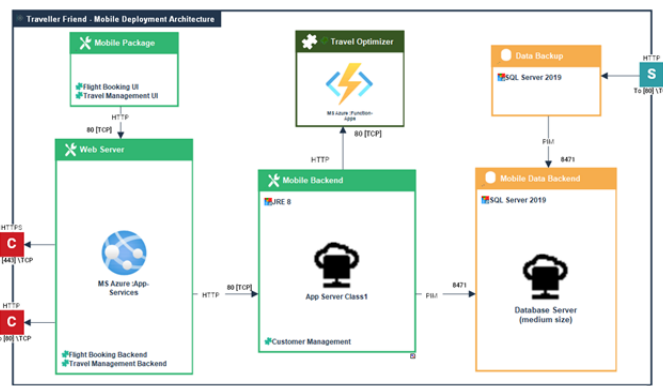
## Accéder à la liste des architectures de déploiement d'une application

Pour accéder à la liste des architectures de déploiement d'une application à partir du menu de navigation **Applications** :











- 1 Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.  
La liste des architectures de déploiement de l'application apparaît.

## Décrire une architecture de déploiement d'une application et son diagramme

Une architecture de déploiement d'une application permet de décrire l'ensemble des éléments qui doivent être déployés pour mettre en œuvre une architecture applicative : *modules de déploiement d'une application*, *modules de déploiement des données* ainsi que les *connexions de module* utilisées pour les échanges de données.



Un diagramme d'architecture de déploiement composé des éléments suivants :

- des **modules de déploiement d'une application**,  
 Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/ Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).
- des **modules de déploiement des données**,  
 Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).  
 Pour plus de détails sur les modules de déploiement des données, voir [Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application](#).
- des **microservices**,  
 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.
- des **ports serveur techniques** et **ports client techniques**,  
 Un port serveur technique est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).  
 Un port client technique est un point permettant de demander l'ouverture d'une communication depuis une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).  
 Pour plus de détails sur les ports techniques, voir [Ajouter des ports techniques](#).
- des **connexions de module**.  
 Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.  
 Vous pouvez créer une architecture de déploiement d'une application en créant un diagramme d'architecture de déploiement directement à partir de l'application qui vous intéresse.  
 Pour plus de détails sur les connexions de modules, voir [Décrire les connexions de module](#).

## Créer une architecture de déploiement d'une application

Pour créer une architecture de déploiement d'une application à partir du menu de navigation **Applications** :

1. Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.  
La liste des architectures de déploiement de l'application apparaît .
2. Cliquez sur **Nouveau**.  
Une fenêtre de sélection d'un modèle d'architecture de déploiement s'ouvre.

☛ Pour plus de détails sur les modèles de déploiement d'une application, voir [Les modèles d'architecture de déploiement](#).

3. Sélectionnez **Modèle d'architecture de déploiement vide** et cliquez sur **OK**.

☛ Pour plus de détails sur l'utilisation des modèles de déploiement d'une application, voir [Utiliser un diagramme d'environnement de déploiement d'une application](#).

La nouvelle architecture de déploiement d'une application est créée avec son diagramme.

☛ Vous pouvez aussi créer une architecture de déploiement d'une application en créant un diagramme d'architecture de déploiement directement à partir de l'application qui vous intéresse.

---

## Utiliser un diagramme d'architecture de déploiement d'une application

Pour créer un diagramme d'architecture de déploiement d'une application à partir d'un environnement de déploiement d'application existant :

1. Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.
2. Faites un clic droit sur l'architecture de déploiement de l'application qui vous intéresse et sélectionnez **Créer diagramme**.
3. Dans la fenêtre de création, sélectionnez **Architecture de déploiement**.

Le diagramme s'ouvre dans la fenêtre d'édition. Les composants de l'architecture de déploiement sont positionnés dans le diagramme.

☛ Lors de la création d'une architecture de déploiement d'une application, un diagramme d'architecture de déploiement d'une application est automatiquement créé.

## Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application

### **Ajouter un module de déploiement d'une application**

📖 Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son

exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).

Pour ajouter un **module de déploiement d'une application** :

1. Dans la barre d'objets de l'architecture de déploiement d'une application, cliquez sur le bouton **Module de déploiement d'une application**.
2. Cliquez dans le cadre de l'application décrite.  
Une fenêtre d'ajout vous demande de choisir le **Module de déploiement d'une application** que vous souhaitez utiliser.
3. Sélectionnez le module de déploiement d'une application et cliquez sur **OK**.

Le module de déploiement d'une application apparaît dans le diagramme.

☛ Pour plus de détails sur la description d'un module de déploiement d'une application, voir [Décrire un module de déploiement d'une application](#).

### Ajouter un module de déploiement des données

De la même manière que vous ajoutez un module de déploiement d'une application, vous pouvez ajouter des modules de déploiement des données.

📖 Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).

### Ajouter des ports techniques

Les **ports techniques** assurent le transfert physique des informations échangées entre les différents composants de l'architecture de déploiement.

📖 Un port serveur technique est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

📖 Un port client technique est un point permettant de demander l'ouverture d'une communication depuis une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

Les ports techniques se conforment à des protocoles d'application réseau.


☛ Les protocoles d'application réseau supportés par un port de communication doivent être compatibles avec les protocoles supportés par le port de communication auquel il est connecté.

Pour créer un **port client technique** :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Port client technique**.
2. Cliquez sur le cadre de l'architecture de déploiement décrite.
3. Dans la fenêtre de création du port technique, sélectionnez le **Protocole d'application réseau** puis la **Connexion réseau technique**.
4. Cliquez sur le bouton **Ajouter**.  
Le port technique apparaît dans le diagramme. Le nom du protocole apparaît au dessus.

## Décrire les connexions de module

Les connexions entre les modules de déploiement d'une application et de données sont décrites par des connexions de module. Une connexion de module supporte les *protocoles d'application réseau* définis pour la établir communication.

 Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.

Pour créer une connexion de module, vous devez créer le lien et spécifier les *protocole d'application réseau* qui sont utilisés.


Pour créer une *connexion de module* :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Connexion de module**.
2. Tirez un lien entre les deux objets en communication.
3. Dans la fenêtre de création de la connexion de module, sélectionnez le *Protocole d'application réseau* puis le *Type de communication*.
4. Cliquez sur le bouton **Nouveau**.  
La connexion de module apparaît dans l'architecture. Le nom du protocole apparaît à coté de la ligne.


 Dans la page de propriétés **Caractéristiques** d'une connexion de module, le champ **Format de communication utilisé** permet de spécifier un **Format de communication** spécifique. Le format sélectionné apparaît dans le diagramme en complément du nom du protocole retenu.

---

## Décrire un module de déploiement d'une application

 Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/ Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/ PaaS cloud service ou un modèle de serveur).

### Définir les technologies logicielles utilisées par un module de déploiement d'une application

 Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.

Pour spécifier les **technologies logicielles requises** pour un **module de déploiement d'une application** :

1. Ouvrez la page de propriétés **Caractéristiques** du module de déploiement d'une application qui vous intéresse.



2. Dans la section **Technologie logicielle requise**, cliquez sur **Relier**.  
Une fenêtre de sélection vous permet de sélectionner la **Technologie logicielle** que vous souhaitez utiliser.  
Les technologies logicielles sélectionnées apparaissent sur l'icône du module de déploiement d'une application.

### ***Définir les composants d'un module de déploiement d'une application***

Pour spécifier par exemple qu'un service Cloud est utilisé par un **module de déploiement d'une application** :

1. Ouvrez la page de propriétés **Caractéristiques** du module de déploiement d'une application qui vous intéresse.
2. Dépliez la section **Composant de déploiement d'une application**.
3. Dans le champs **Unité de traitement recommandée**, cliquez sur **Relier**.  
Une fenêtre de sélection vous permet de sélectionner le **Service Cloud** que vous souhaitez utiliser.

☛ Pour plus de détails sur les Services Cloud, voir [Utiliser les services cloud](#).

# DÉCRIRE UN ENVIRONNEMENT DE DÉPLOIEMENT D'UNE APPLICATION

L'environnement de déploiement d'une application est considéré comme le centre de l'intégration et tous les déploiements requis d'applications partenaires ou de microservices.



*Un diagramme d'environnement de déploiement d'une application représente l'architecture de déploiement d'une application sujet, les architectures de déploiement d'une application et les microservices partenaires ainsi que les techniques de communication nécessaires.*

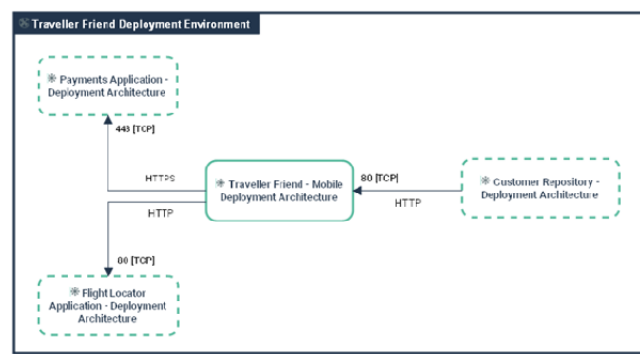
## Accéder à la liste des environnements de déploiement d'une application

Pour accéder à la liste des environnements de déploiement d'une application:\*

1. Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.  
La liste des architectures de déploiement de l'application apparaît.
2. Dépliez le dossier **Environnement** d'une architecture de déploiement de l'application.  
La liste des environnements de déploiement d'une application apparaît.


## Décrire un environnement de déploiement d'une application

Un environnement de déploiement d'une application est décrit par un diagramme d'environnement de déploiement d'une application.




Un diagramme d'environnement de déploiement d'une application est composé des éléments suivants :


- des **architectures de déploiement d'application sujet** et des **architectures de déploiement d'application partenaire**,

 Une architecture de déploiement d'une application décrit une configuration possible pour déployer une application. Elle présente les modules de déploiement à héberger, préconise un type d'hébergement et les connexions techniques requises pour qu'ils puissent communiquer entre eux (protocoles de communication, numéros de port,...) . Une application peut avoir plusieurs architectures de déploiement possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)

- des **microservices partenaires**,

 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

- des **connexions de module**.

 Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.

## Créer un environnement de déploiement d'une application

Pour créer un environnement de déploiement d'une application à partir du menu de navigation **Application** :

- Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.  
La liste des architectures de déploiement de l'application apparaît.
- Ouvrez la page **Environnement** d'une architecture de déploiement de l'application.
- Cliquez sur **Nouveau**.  
L'environnement de déploiement d'une application créé apparaît dans la liste.

---

## Utiliser un diagramme d'environnement de déploiement d'une application

Pour créer un diagramme d'environnement de déploiement d'une application à partir du menu de navigation **Application** :


- Ouvrez la page **Architectures de déploiement** de l'application qui vous intéresse.  
La liste des architectures de déploiement de l'application apparaît.
- Ouvrez la page **Environnement** d'une architecture de déploiement de l'application.
- Sélectionnez un environnement de déploiement d'une application et cliquez sur **Créer Diagramme**.

4. Dans la fenêtre de création, sélectionnez **Diagramme d'environnement de déploiement d'une application**.

Le diagramme d'environnement de déploiement d'une application s'ouvre dans la zone d'édition. L'*architectures de déploiement d'application sujet* est placée au centre du cadre.

## DÉCRIRE UNE ARCHITECTURE DE DÉPLOIEMENT D'UN SYSTÈME APPLICATIF

L'architecture de déploiement du système applicatif consiste en un ensemble d'architectures de déploiement d'application qui doivent être coordonnées pour couvrir les dépendances requises entre elles.

 Une architecture de déploiement d'un système applicatif décrit une configuration possible pour déployer un système applicatif. Elle présente les architectures de déploiement des composants applicatifs retenus et spécifie les protocoles de communication utilisés entre eux ainsi que les numéros de port.

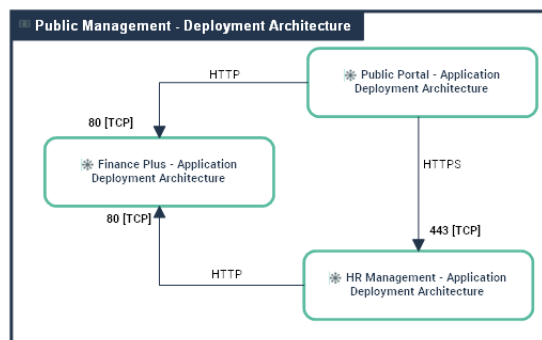
### Accéder à la liste des architectures de déploiement d'un système applicatif

Pour accéder à la liste des architectures de déploiement d'un système applicatif à partir du menu de navigation **Systèmes applicatifs** :

- 1 Ouvrez la page **Architectures de déploiement** du système applicatif qui vous intéresse.  
La liste des architectures de déploiement du système applicatif apparaît.

### Décrire une architecture de déploiement d'un système applicatif

Une architecture de déploiement d'un système applicatif est décrite par un diagramme d'architecture de déploiement d'un système applicatif.



Un diagramme d'architecture de déploiement d'un système applicatif est composé des éléments suivants :

- des **architectures de déploiement d'une application**,



Une architecture de déploiement d'une application décrit une configuration possible pour déployer une application. Elle présente les modules de déploiement à héberger, préconise un type d'hébergement et les connexions techniques requises pour qu'ils puissent communiquer entre eux (protocoles de communication, numéros de port,...). Une application peut avoir plusieurs architectures de déploiement possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)



Pour plus de détails sur les architecture de déploiement d'une application, voir [Décrire une architecture de déploiement d'une application](#).

- des **architectures de déploiement d'un système applicatif**,



Une architecture de déploiement d'un système applicatif décrit une configuration possible pour déployer un système applicatif. Elle présente les architectures de déploiement des composants applicatifs retenus et spécifie les protocoles de communication utilisés entre eux ainsi que les numéros de port.



Pour plus de détails sur les architecture de déploiement d'un système applicatif, voir [Décrire une architecture de déploiement d'un système applicatif](#).

- des **microservices**,



Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

- des **modules de déploiement des données**,



Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).



Pour plus de détails sur les modules de déploiement des données, voir [Décrire une architecture de déploiement d'un système applicatif](#).

- des **connexions de module**.



Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.

- des **ports serveur techniques** et **ports client techniques**,



Un port serveur technique est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).



Un port client technique est un point permettant de demander l'ouverture d'une communication depuis une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).



La procédure de création d'un diagramme d'architecture de déploiement d'un système applicatif est identique à la procédure de création d'un diagramme d'architecture de déploiement d'une

application. Pour plus de détails, voir [Utiliser un diagramme d'architecture de déploiement d'une application](#).

## Les propriétés d'une architecture de déploiement d'un système applicatif

La description complète d'une architecture de déploiement d'un système applicatif est accessible à partir de ses pages de propriétés.

La page **Caractéristiques** des propriétés d'une architecture de déploiement d'un système applicatif permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de l'application spécifiée lors de sa création.
- le texte de sa description.

Avec **HOPEX IT Architecture** une architecture de déploiement d'un système applicatif est décrite par d'autres pages de propriétés.

La page **Composants** qui permet d'accéder à la liste des composants de l'architecture décrite.

✎ Pour plus de détails sur les composants d'une d'architecture de déploiement d'un système applicatif, voir [Décrire une architecture de déploiement d'un système applicatif](#).

- La section **Architectures de déploiement** donne accès aux onglets suivants :

- **Architectures de déploiement d'un système applicatif**,

📖 Une architecture de déploiement d'un système applicatif décrit une configuration possible pour déployer un système applicatif. Elle présente les architectures de déploiement des composants applicatifs retenus et spécifie les protocoles de communication utilisés entre eux ainsi que les numéros de port.

- **Architectures de déploiement**, pour accéder à la liste des architectures de déploiement d'une application,

📖 Une architecture de déploiement d'une application décrit une configuration possible pour déployer une application. Elle présente les modules de déploiement à héberger, préconise un type d'hébergement et les connexions techniques requises pour qu'ils puissent communiquer entre eux (protocoles de communication, numéros de port,...). Une application peut avoir plusieurs architectures de déploiement possibles (Ex: installation autonome, déploiement horizontal ou vertical, etc.)

✎ Pour plus de détails sur les architecture de déploiement d'une application, voir [Décrire une architecture de déploiement d'une application](#).

- **Déploiement de microservice détenu**, pour accéder à la liste des microservices,

📖 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel

déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

- La section **Module de données** donne accès à la liste des *modules de déploiement des données*,



Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).



Pour plus de détails sur les modules de déploiement des données, voir [Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application](#).

- La section **Connexion du déploiement** donne accès à la liste des *connexions de module*.



Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.



Pour plus de détails sur les connexions de module, voir [Décrire les connexions de module](#).

- La section **Ports techniques** donne accès aux onglets :

- **Port serveur**



Un port serveur technique est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

- **Port client**



Un port client technique est un point permettant de demander l'ouverture d'une communication depuis une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).



Pour plus de détails sur les ports techniques, voir [Ajouter des ports techniques](#).

La page **Rapports** qui permet d'accéder aux différents rapports associés l'architecture de déploiement d'un système applicatif décrite.



## LES MODÈLES D'ARCHITECTURE DE DÉPLOIEMENT

Les *modèles d'architecture de déploiement* facilitent la création des architectures de déploiement d'application. Les composants de la nouvelle architecture de déploiement d'une application sont automatiquement créés à partir des composants du modèle d'architecture de déploiement.

La nouvelle architecture de déploiement peut ensuite être mise à jour et complétée si nécessaire.

Des exemples de *modèles d'architecture de déploiement* sont livrés avec la solution.

---

### Accéder à la liste des modèles d'architecture de déploiement

Pour accéder à la liste des *modèles d'architecture de déploiement* de votre référentiel :




- 1 A partir du menu de navigation **Administration**, sélectionnez **Modèles > Architectures de déploiement**.  
La liste des modèles d'architecture de déploiement apparaît.

---

### Décrire un modèle de déploiement d'application

#### Les composants d'un modèle de déploiement d'application

Un modèle de déploiement d'application est décrit par un diagramme de modèle de déploiement d'application composé des éléments suivants :

- des *modèles de déploiement d'application*, qui servent à créer les modules de déploiement d'une application d'une nouvelle architecture de déploiement d'une application.  
 *Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).*  
 *Pour plus de détails sur les modules de déploiement des applications, voir [Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application](#).*
- des *modèles de déploiement des données*, qui servent à créer les modules de déploiement des données d'une nouvelle architecture de déploiement d'une application.  
 *Un module de déploiement des données représente la partie "Données" d'un déploiement d'application qui doit être hébergée pour être accessible lors de l'exécution de services applicatifs (code). Chaque*

module de déploiement des données est associé à des technologies requises (pour l'hébergement et l'accès) et peut héberger plusieurs structures de données. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).

➡ Pour plus de détails sur les modules de déploiement des données, voir [Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application](#).

- des **microservices**,

📖 Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.

➡ Pour plus de détails sur les microservices, voir [Décrire un microservice avec HOPEX IT Architecture](#),

- des **ports serveur techniques** et **ports client techniques**,

📖 Un port serveur technique est un point permettant l'ouverture d'une communication vers une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

📖 Un port client technique est un point permettant de demander l'ouverture d'une communication depuis une architecture technique ou une zone technique d'application conformément à un protocole de communication particulier (SMTP, HTTP, etc.).

➡ Pour plus de détails sur les ports techniques, voir [Ajouter des ports techniques](#).

- des **connexions de module**.

📖 Une connexion de module définit une exigence de connexion entre deux modules de déploiement. Un module client demande la connexion à un module serveur. La connexion est définie par un type et peut être caractérisée par des mesures de flux techniques.

➡ Vous pouvez créer une architecture de déploiement d'une application en créant un diagramme d'architecture de déploiement directement à partir de l'application qui vous intéresse.

➡ Pour plus de détails sur les connexions de modules, voir [Décrire les connexions de module](#).

## Créer un modèle de déploiement d'une application

Pour accéder à la liste des **modèles de déploiement d'application** de votre référentiel :

1. A partir du menu de navigation **Administration**, sélectionnez **Modèles > Architectures de déploiement**.  
La liste des modèles de déploiement d'application apparaît.
2. Cliquez sur **Nouveau**.  
Le modèle de déploiement d'une application apparaît dans la liste.

➡ Pour plus de détails sur la création du diagramme de déploiement d'application, voir [Utiliser un diagramme d'architecture de déploiement d'une application](#).

## Présentation des modèles d'architecture de déploiement standard

Des modèles d'architecture de déploiement sont livrés en standard pour simplifier la création de vos architectures de déploiement.

### Le modèle de déploiement d'application "Architecture 3 tiers (RDBMS)"

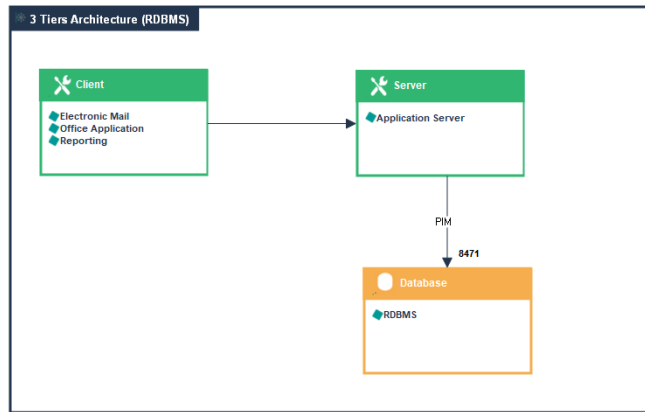


Diagramme du modèle de déploiement d'application "Architecture 3 tiers (RDBMS)"

### Le modèle de déploiement d'application "Architecture d'application mobile"

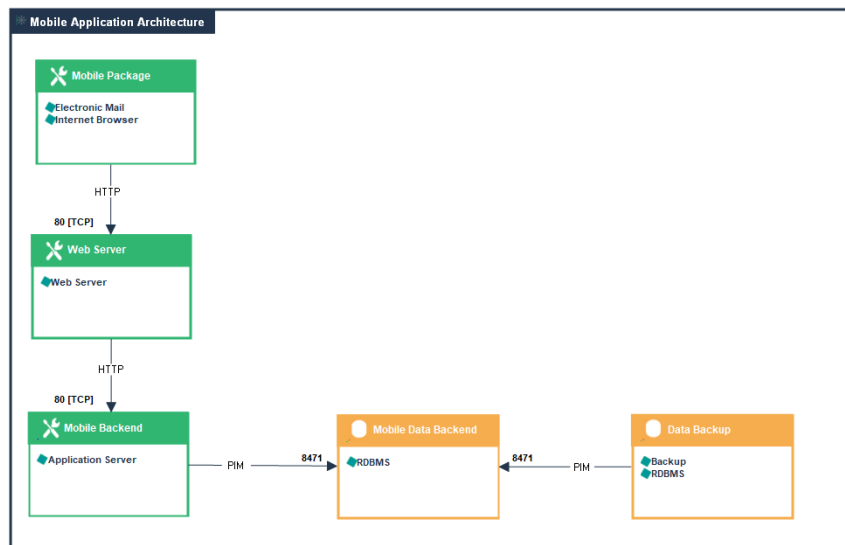


Diagramme du modèle de déploiement d'application "Architecture d'application mobile"

## Le modèle de déploiement d'application "Architecture d'application Internet"

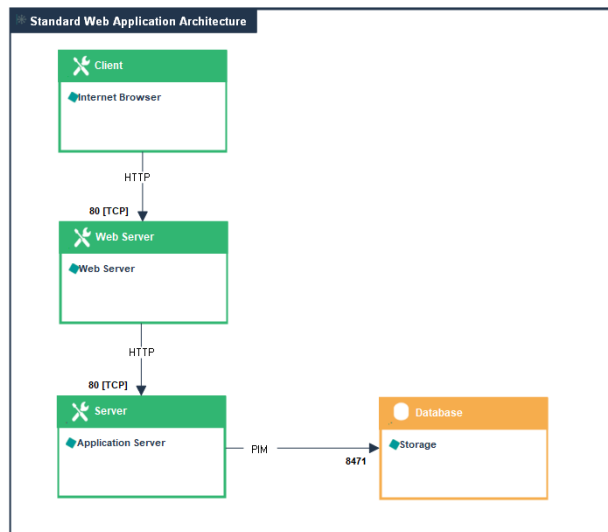


Diagramme du modèle de déploiement d'application "Architecture d'application Internet"

---

## Utiliser un modèle de déploiement d'application

Pour créer une architecture de déploiement d'une application à partir d'un *modèle de déploiement d'application* :

1. Ouvrez la page **Architecture de déploiement** de l'application qui vous intéresse.
2. Sélectionnez le **Modèle de déploiement d'une application**.

✎ Pour plus de détails sur les modèles de déploiement d'une application, voir [Les modèles d'architecture de déploiement](#).

3. Cliquez sur le bouton **Suivant**.  
Une fenêtre vous présente la liste des composants de la nouvelle architecture.

Création d'un(e) Architecture de déploiement d'une application - Modules de déploiement

La nouvelle architecture de déploiement basée sur le modèle peut réutiliser des modules de déploiements existants de l'application ou en créer de nouveaux.  
Si de nouveaux modules sont créés il est possible de préciser quelles technologies sont requises : Hopex propose les technologies identifiées directement dans le modèle ou celles permettant de couvrir les fonctionnalités techniques requises par le modèle.

Afficher le diagramme du modèle

	Module Création\Réutilisation	Nom du module	Technologies requises
Modules d'application			
Client	Nouveau module	Client	
Electronic Mail (EN)			...
Office Application (EN)			...
Reporting (EN)			...
Serveur	Nouveau module	Serveur	
Application Server (EN)			...
Modules de données			
Database (EN)	Nouveau module	Base de données	

Précédent Suivant **OK** Annuler

4. (Optionnel) Dans la colonne **Module Création/Réutilisation**, sélectionnez les composants que vous souhaitez réutiliser.  
*Vous ne pouvez réutiliser que des composants qui existent dans une autre architecture de déploiement de la même application.*
5. Cliquez sur le bouton **OK**.  
 Le diagramme s'ouvre et vous pouvez ensuite modifier le contenu de votre nouvelle architecture de déploiement d'une application.  
*Pour plus de détails sur la mise à jour d'un diagramme d'architecture de déploiement d'une application, voir [Utiliser un diagramme d'architecture de déploiement d'une application](#).*

# DÉCRIRE LES TECHNOLOGIES LOGICIELLES

Cette description s'appuie sur des *technologies logicielles* et sur des *piles de technologies logicielles*.

---

## Décrire une technologie logicielle



Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.

### Accéder à la liste des technologies logicielles

Pour accéder à la liste des technologies logicielles à partir du menu de navigation **Inventaires** :

- 1 Sélectionnez **Déploiement > Technologies logicielles**.  
La liste des technologies logicielles apparaît sous l'onglet **Technologies logicielles**.

### Les propriétés d'une technologie logicielle

La description complète d'une technologie logicielle est accessible à partir de ses pages de propriétés.

La page **Caractéristiques** des propriétés d'une technologie logicielle permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création de la technologie, il s'agit de la bibliothèque courante.
- son **Code**,
- son **Fournisseur**,
- un texte de **Commentaire**.

La page **Caractéristiques** permet également d'accéder aux sections suivantes :


- **Types de technologies** qui définissent la technologie logicielle concernée,
- **Responsabilité**,
- **Réalisation détenue** qui représente la liste des capacités technologiques couvertes par cette technologie logicielle.

☛ Pour plus de détails sur les fonctionnalités, voir [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

☛ Pour plus de détails sur les réalisations, voir [Décrire la mise en œuvre d'une fonctionnalité](#).

## Décrire une pile de technologies

 Une pile de technologies logicielles est un ensemble de technologies logicielles.

 Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.

### Accéder à la liste des piles de technologie

Pour accéder à la liste des piles de technologies logicielles à partir du menu de navigation **Inventaires** :

- 1 Sélectionnez **Déploiement > Technologies logicielles**.  
La liste des piles de technologies apparaît dans l'onglet **Piles de technologies**.

### Les propriétés d'une pile de technologies logicielles

La description complète d'une pile de technologies logicielles est accessible à partir de ses pages de propriétés.


La page de propriété **Caractéristiques** d'une pile de technologies logicielles permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création de la pile de technologies logicielles, il s'agit de la bibliothèque courante.
- son **Code**,
- un texte de **Commentaire**.

Cette page permet également d'accéder aux sections suivantes :

- **Composants** qui permet de décrire la liste des technologies logicielles concernées,
- **Responsabilité**,
- **Réalisation détenue** qui représente la liste des capacités technologiques couvertes par cette pile de technologies logicielles.

 Pour plus de détails sur les fonctionnalités, voir [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

 Pour plus de détails sur les réalisations, voir [Décrire la mise en œuvre d'une fonctionnalité](#).

# UTILISER LES SERVICES CLOUD



*Le service Cloud (considéré comme un objet connecté) peut être utilisé dans des architectures de déploiement. L'image du service cloud apparaît alors dans la forme du module de déploiement.*

Il est possible d'importer dans **HOPEX** des *catalogues des services cloud* tels que : Amazon (AWS), Microsoft (Azure) et Google (GCS). Les fichiers importés contiennent le noms et les images des services cloud de l'éditeur concerné.

☛ Ces services Cloud sont mis à disposition par votre administrateur en utilisant l'import de module. Pour plus de détails, voir le chapitre "Importer un module dans **HOPEX**" du guide **HOPEX Administration**.

## Accéder à la liste des services cloud

Pour accéder à la liste des informations concernant les Services Cloud à partir du menu de navigation **Déploiement** :

1. Sélectionnez **Catalogues des services du Cloud**.
2. Dépliez le dossier **Catalogue Editeur**.  
La liste des catalogues des services Cloud apparaît dans la zone d'édition.
3. Déplier le dossier d'un catalogue.  
Les dossiers suivants permettent d'accéder aux différents éléments :
  - **Service Cloud** : permet d'accéder à la liste des services Cloud du catalogue.
  - **Carte de capacités technologiques** : elle regroupe l'ensemble des capacités technologiques couvertes par les services Cloud du catalogue.  
  
☛ Pour plus de détails sur les fonctionnalités, voir [Décrire les fonctionnalités avec HOPEX IT Architecture](#).
  - **Editeur** : dans **HOPEX**, l'éditeur est représenté par un acteur.

Pour visualiser la couverture fonctionnelle couverte par les services Cloud d'une carte de capacités technologiques :

1. Sélectionnez la carte de fonctionnalités du catalogue qui vous intéresse.
2. Ouvrez la page de propriétés **Rapports > Rapport de décomposition des briques d'entreprise**.



### 3. Sélectionnez **Service Cloud** dans le champ **Afficher**.

Amazon WS Technical Functionality Map

Characteristics Structure Assignment **Reporting** Activity Feed

Building Block Breakdown Report ▾

Parameters

- Root Object
- Architecture Building Blocks
- Fulfilling Solution Building Blocks

Refresh the report

Levels All Show Cloud Service X ▾

Show legend

Amazon - App-Integration	Amazon - AR-VR	Amazon - AWS-Cost-Management	Amazon - Business-Application	Amazon - Customer-Enablement
Amazon-Managed-Workflows-for-Apache-Airflow	Amazon-Sumerian	AWS-Custom-Billing-Manager	Amazon-Connect	AWS-Professional-Services
Amazon-AppFlow		AWS-Cost-Explorer	Amazon-WorkDocs	AWS-Support
Amazon-Simple-Queue-Service		Savings-Plans	Amazon-Chime	AWS-Activate
AWS-Console-Mobile-Application		Reserved-Instance-Reporting	Alexa-For-Business	AWS-Managed-Services
Amazon-MQ		Amazon-Corretto	Amazon-WorkDocs-SDK	AWS-IQ
		AWS-Budgets	Amazon-Pinpoint	AWS-Training-Certification

## Les propriétés d'un service Cloud

La page de propriété **Caractéristiques** d'un service Cloud permet d'accéder à :

- son **Nom**,
- son **Type de Service**,
- un texte de **Description**,
- une section **Mises en œuvre** qui permet d'identifier les capacités technologiques couvertes par le service Cloud.
  - ☛ Pour plus de détails sur la mise en œuvre d'un composant, voir [Décrire la mise en œuvre d'une fonctionnalité](#).
- une section **Utiliser l'architecture de déploiement d'une application** présente sous forme d'arbre toutes les architectures de déploiement d'une application dont composant de type module de déploiement d'une application utilise le service Cloud.
  - ☛ Pour plus de détails sur l'utilisation d'un service Cloud par un module de déploiement d'une application, voir [Décrire un module de déploiement d'une application](#).



# ALIGNER L'INFORMATIQUE ET LES MÉTIERS



Cette étape, de niveau stratégique, a pour objectif de vérifier l'adéquation entre les *capacités métier* de l'entreprise et les éléments de l'architecture logique qui les réalisent.

Il s'agit de réaliser les tâches suivantes :

- ✓ Décrire l'architecture logicielle logique
- ✓ Décrire les capacités métier avec HOPEX IT Architecture,
- ✓ Utiliser les fonctionnalités avec HOPEX IT Architecture,
- ✓ Utiliser les mécanismes de mise en œuvre.

# DÉCRIRE L'ARCHITECTURE LOGICIELLE LOGIQUE

**HOPEX IT Architecture** offre des moyens de définir des architectures d'applications logiques qui représentent des architectures idéales. Ces représentations permettent de concevoir les structures logiques pour les architectures d'applications, de rationaliser les échanges entre ces structures et d'identifier les données utilisées. Les architectures d'applications logiques peuvent ensuite être comparées aux architectures mises en œuvre pour détecter les écarts entre le réel et l'idéal.

## Décrire un système applicatif logique avec HOPEX IT Architecture

Un projet de description de l'architecture logique d'un système d'information permet de recenser les *systèmes applicatifs logiques* existants et leurs échanges.



*Un système applicatif logique est un assemblage d'autres architectures, d'applications logiques et d'utilisateurs finaux interagissant avec les composants applicatifs, dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

Un *système applicatif logique* peut être décrit par deux types de diagramme :

- un diagramme de structure de système applicatif logique qui permet de représenter les différents composants du système et les échanges entre eux.



*Pour plus de détails l'utilisation des diagrammes de structure de système applicatif, voir [Décrire la structure d'un système applicatif logique](#).*

- un diagramme de scénario de flux de système applicatif logique qui permet de décrire les échanges réalisés à l'intérieur du système applicatif logique décrit dans un contexte particulier.



*Pour plus de détails l'utilisation des diagrammes de scénario de flux, voir [Utiliser un diagramme de scénario de séquence de flux](#).*

## Accéder à la liste des systèmes applicatifs logiques avec HOPEX IT Architecture

Pour accéder à la liste des systèmes applicatifs logiques à partir du menu de navigation **Inventaires** :

- 1. Sélectionnez **Logiciel > Architecture logicielle logique** dans le menu de navigation.  
L'arbre des systèmes applicatifs logiques apparaît.

## Créer un système applicatif logique

Pour créer un *système applicatif logique* :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Logiciel > Architecture logicielle logique**.



2. Cliquez sur le bouton **Nouveau**.  
La fenêtre **Création d'un Système applicatif logique** apparaît.
3. Saisissez le **Nom** de votre système applicatif logique ainsi que son **Détenteur** et cliquez sur **OK**.  
Le nouveau système applicatif logique apparaît dans la liste.

## Les propriétés d'un système applicatif logique

La page **Caractéristiques** des propriétés d'un système applicatif logique permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création du système applicatif logique, il s'agit de la bibliothèque courante.
- le texte de sa **Description**,

Avec **HOPEX IT Architecture** un système applicatif logique est décrit par les pages suivantes :

- la page **Propriétés** qui permet de spécifier les propriétés qui apparaissent dans les diagrammes, en bas du cadre de l'objet décrit.
- la page **Composants** permet d'accéder à la liste des composants du système applicatif logique et des communications définies entre eux dans les diagrammes du système applicatif logique.  
 Pour plus de détails les composants d'un système applicatif logique, voir [Décrire la structure d'un système applicatif logique](#).
- la page **Mise en œuvre** permet de spécifier les éléments logiques ou physiques qui réalisent le système applicatif logique décrit.  
 Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Décrire la structure d'un système applicatif logique

Avec **HOPEX IT Architecture**, les composants d'un système applicatif logique et leurs échanges sont décrits dans un **diagramme de structure système applicatif logique**.

Le diagramme de structure de système applicatif logique, chargé de gérer les "Demandes d'achats par Internet" présente différentes applications logiques, un accès à une

base de donnée logique ainsi que des points de service et de requête pour "Réserver" ou "Commander".

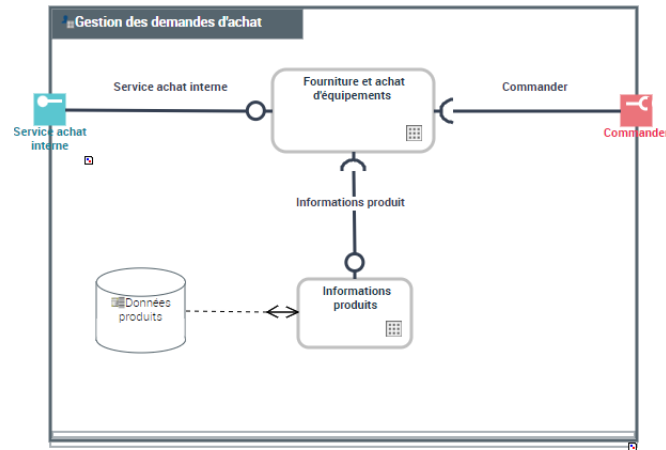


Diagramme de structure de système applicatif logique "Gestion des demandes d'achat"

Un diagramme de structure de système applicatif logique est composé des éléments suivant :

- d'**utilisateurs finaux**  
 L'utilisateur final représente une unité organisationnelle interagissant aux frontières d'un système applicatif ou d'un système applicatif logique.  
 Pour plus de détails sur l'ajout d'utilisateurs finaux, voir [Ajouter un utilisateur final au diagramme de structure de système applicatif logique](#).
- de **composants de système applicatif logique** et de **composants d'application logique**  
 Une application logique est un ensemble de fonctionnalités définies indépendamment d'une mise en œuvre particulière. Par exemple, la classification de toutes les applications de traitement des demandes d'achat implémentées dans une entreprise.  
 Pour plus de détails sur l'ajout d'applications, voir [Ajouter une application logique au diagramme de structure de système applicatif logique](#).
- d'**interactions de service** entre les composants représentant les demandes de services  
 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à

*l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*

☛ *Pour plus de détails sur les interactions de service entre les composants de système applicatif logique, voir [Gérer les interactions de service](#).*

- de **points de service**

📖 *Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.*

- de **points de requête**

📖 *Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.*

☛ *Pour plus de détails sur les points d'accès, voir [Décrire les points de service et de requête](#).*

## **Ajouter un utilisateur final au diagramme de structure de système applicatif logique**

Pour ajouter un **utilisateur final** :

1. Dans la barre d'objets du diagramme de structure de système applicatif logique, cliquez sur le bouton **Utilisateur final**.
2. Cliquez dans le cadre du système applicatif logique décrit.  
Une fenêtre d'ajout vous demande de choisir le **Type d'objet** que vous souhaitez utiliser.
3. Sélectionnez, par exemple, le type d'objet **Acteur**.

📖 *Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.*

4. Sélectionnez l'acteur qui vous intéresse et cliquez sur **OK**.  
L'acteur apparaît dans le diagramme.

## **Ajouter une application logique au diagramme de structure de système applicatif logique**

Pour décrire qu'un système applicatif logique utilise une application logique :

1. Dans la barre d'objets du diagramme de structure de système applicatif logique, cliquez sur le bouton **Composant d'application logique** et cliquez dans le cadre du système applicatif logique décrit.  
Une fenêtre d'ajout vous demande de choisir l'**Application logique** utilisée.
2. Sélectionnez une application logique existante.
3. Cliquez sur **OK**.  
L'application logique apparaît dans le diagramme.

## Décrire les applications logiques avec HOPEX IT Architecture



Une application logique est un ensemble de fonctionnalités définies indépendamment d'une mise en œuvre particulière. Par exemple, la classification de toutes les applications de traitement des demandes d'achat implémentées dans une entreprise.

### Accéder à la liste des applications logiques avec HOPEX IT Architecture

Pour accéder à la liste des applications logiques à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Logiciel > Architecture logicielle logique** et dépliez le dossier **Systèmes applicatifs logiques**.
2. Déplier l'arbre du système applicatif logique qui vous intéresse et dépliez le dossier **Composant application logique**.

### Créer une application logique

Pour créer une *application logique* à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Logiciel > Architecture logicielle logique**. La liste des systèmes applicatifs logiques apparaît dans la liste.
2. Ouvrez la page **Composants** du système applicatif logiques qui vous intéresse.
3. Dans la liste des types de composants, sélectionnez **Composant application logique** et cliquez sur **Nouveau**. La fenêtre **Création d'une application logique** apparaît.
4. Saisissez le **Nom** de votre application logique et cliquez sur **OK**. La nouvelle application logique apparaît dans la liste.

### Les propriétés d'une application logique

La page **Caractéristiques** des propriétés d'une application logique permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création d'une application logique, il s'agit de la bibliothèque courante.
- le texte de sa **Description**,

Avec **HOPEX IT Architecture** une application logique est décrite par les pages suivantes :

- la page **Propriétés** qui permet de spécifier les propriétés qui apparaissent dans les diagrammes, en bas du cadre de l'objet décrit.
- la page **Mise en œuvre** permet de spécifier les éléments logiques ou physiques qui mettent en œuvre l'application logique décrite.
- les pages **Composants**, **Processus exécutés** ainsi que **Rapports** sont également disponibles.

➡ Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).



## Décrire un environnement de système applicatif logique



Un environnement de système applicatif logique présente un contexte d'utilisation de systèmes applicatifs logiques. Il décrit les interactions de service, entre le système applicatif logique et ses partenaires externes, qui lui permettent de remplir sa mission et d'assurer les fonctionnalités attendues.

Un **environnement de système applicatif logique** peut être décrit par deux types de diagramme :

- un diagramme de scénario de flux d'environnement de système applicatif logique qui permet de décrire les échanges réalisés à l'intérieur de l'environnement de système applicatif logique décrit dans un contexte particulier.

☛ Pour plus de détails l'utilisation des diagrammes de scénario de flux, voir [Utiliser un diagramme de scénario de séquence de flux](#).

- un diagramme d'environnement de système applicatif logique qui permet de représenter les interactions de service entre les systèmes applicatifs logiques internes, ses utilisateurs et les systèmes logiques partenaires.

☛ Pour plus de détails l'utilisation des diagrammes d'environnement de système applicatif, voir [Utiliser le diagramme d'environnement de système applicatif logique](#).

### Exemple d'environnement de système applicatif logique

Un diagramme d'environnement de système applicatif logique décrit les interactions de service entre les principaux composants internes de l'environnement décrit et les composants externes.

Les demandes d'achat sont formulées par des clients dans des conditions définies par le service Ventes et le service Marketing.

Le système applicatif logique interne "Traitement des demandes d'achat" fait appel à un système applicatif

logique de "Livraison" qui est externe à l'environnement décrit.

s

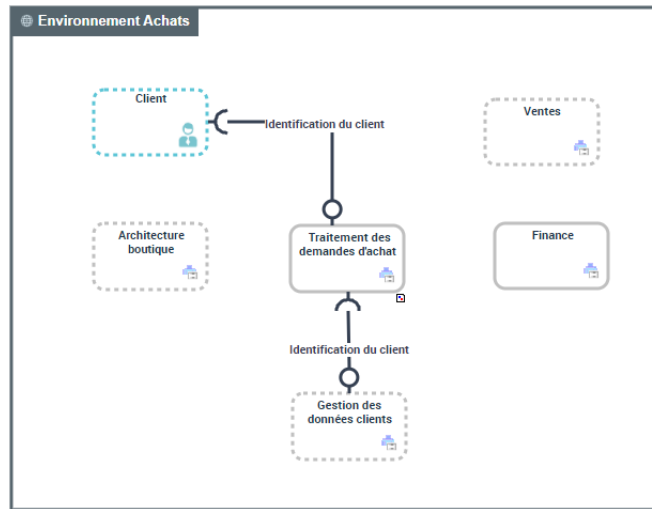


Diagramme d'environnement de système applicatif logique

## Accéder à la liste des environnements d'un système applicatif logique

Pour accéder à la liste des environnements de systèmes applicatifs logiques à partir du menu de navigation du menu de navigation **Inventaires** :

1. Sélectionnez **Logiciel > Architecture logicielle logique**.  
La liste des systèmes applicatifs logiques apparaît dans la liste.
2. Sélectionnez la page **Environnements** du système applicatif logique qui vous intéresse.  
La liste des environnements du système applicatif logique apparaît.

## Créer un environnement de système applicatif logique

Pour créer un *environnement de système applicatif logique* :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Logiciel > Architecture logicielle logique**.
2. Ouvrez la page **Environnement** du système applicatif logique qui vous intéresse et cliquez sur **Nouveau**.  
La fenêtre **Création d'un Environnement de système applicatif logique** apparaît.
3. Saisissez le **Nom** de votre environnement de système applicatif et cliquez sur **OK**.  
Le nouvel environnement de système applicatif logique apparaît dans la liste.

## Les propriétés d'un environnement de système applicatif logique

La page **Caractéristiques** des propriétés d'un environnement de système applicatif logique permet d'accéder à :

- son **Nom**,
- son **Détenteur**, par défaut lors de la création d'un environnement de système applicatif logique, il s'agit de la bibliothèque courante.
- le texte de sa **Description**,

☛ Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Utiliser le diagramme d'environnement de système applicatif logique

Un *environnement de système applicatif logique* permet de représenter les interactions de service entre les systèmes applicatifs logiques internes, ses utilisateurs et les systèmes logiques partenaires.

Le diagramme d'environnement de système applicatif logique est composé de :

- *de système applicatif logique* qui représentent les systèmes applicatifs logiques internes à l'environnement décrit.

Dans l'exemple, il s'agit du système applicatif logique "Traitement des demandes d'achat".

📖 Un système applicatif logique est un assemblage d'autres architectures, d'applications logiques et d'utilisateurs finaux interagissant avec les composants applicatifs, dans le but de mettre en œuvre une ou plusieurs fonctionnalités.

- *systèmes logiques partenaires* qui représentent les systèmes applicatifs logiques externes à l'environnement décrit.

Dans l'exemple, il s'agit du système applicatif logique de "Livraison".

📖 Un système logique partenaire est un système applicatif logique externe à l'environnement de service applicatif logique décrit. Le système logique partenaire peut être fournisseur ou consommateur de service vis à vis des composants du système applicatif logique.

- d'*acteurs* et de *positions types* qui représentent les catégories d'utilisateurs des services proposés par l'environnement.
- Des *interactions de service* entre les composants qui représentent les demandes de services.

📖 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

☛ Pour plus de détails, voir [Créer une interaction de service](#).

# DÉCRIRE LES CAPACITÉS MÉTIER AVEC HOPEX IT

## ARCHITECTURE

Cette étape, de niveau stratégique, permet de vérifier l'adéquation entre les capacités métier de l'entreprise, les fonctionnalités attendues et les applications qui les réalisent.

---

### Exemples de capacités métier avec HOPEX IT Architecture

Une *capacité métier* définit une aptitude escomptée.



*Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).*

Par exemple, pour répondre à l'objectif de satisfaction client, l'organisation doit être en mesure de livrer les services conformément aux engagements contractés.

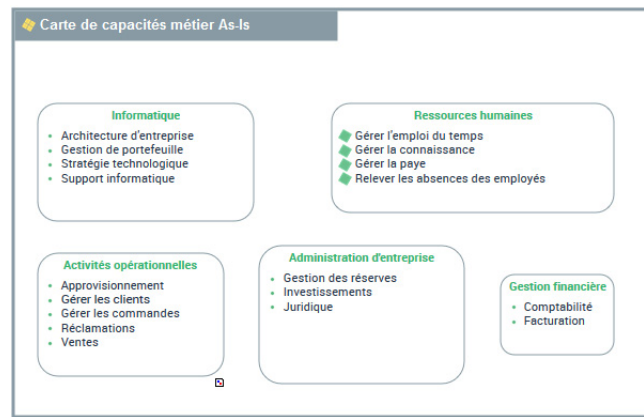
Une *carte de capacités métier* permet de décrire ce que l'entreprise est capable de produire pour ses besoins internes ou pour répondre aux attentes de ses clients. Elle s'appuie donc sur les capacités métier principales de son activité à un instant donné.



*Une carte de capacités métier est un assemblage de capacités métier avec leurs dépendances qui, conjointement, définissent un cadre pour une phase d'entreprise.*

Par exemple, la capacité standard à gérer les "Activités opérationnelles" s'appuie sur les capacités métier de traiter les "Approvisionnements", les "Ventes" et les

"Réclamations", de "Gérer les commandes" et de "Gérer les clients".



☛ Pour plus de détails sur la gestion d'une carte de capacités métier, voir le chapitre "Décrire une carte de capacités métier" du guide **HOPEX IT Business Management**.

La description des **capacités métier** et des **fonctionnalités** est particulièrement intéressante si les capacités métiers sont associées aux fonctionnalités qui permettent de les satisfaire.

Par ailleurs, si les **applications** sont reliées aux **fonctionnalités** qu'elles couvrent, elles sont indirectement reliées à des **capacités métier**. Un rapport **HOPEX IT Business Management** permet de vérifier la couverture fonctionnelle du parc applicatif.

☛ Pour plus de détails sur les rapports relatifs aux capacités métier, voir [Rapport de décomposition des briques d'entreprise](#).

## Utiliser les cartes de capacités métier avec HOPEX IT Architecture

📖 Une carte de capacités métier est un assemblage de capacités métier avec leurs dépendances qui, conjointement, définissent un cadre pour une phase d'entreprise.

### Accéder à la liste des cartes de capacité métier

Pour accéder à la liste des cartes de capacités à partir du menu de navigation **Capacités** :

- Sélectionnez **Capacités > Capacités métier**.  
La liste des cartes de capacités métier apparaît.

## Créer une carte de capacités métier

Pour créer une carte de capacités métier à partir du menu de navigation **Capacités** :


1. Sélectionnez **Capacités > Capacités métier**.  
La liste des cartes de capacités métier apparaît.
2. Cliquez sur le bouton **Nouveau**.  
La fenêtre **Création d'une carte de capacités métier** apparaît.
3. Saisissez le **Nom** de votre carte de capacité métier et cliquez sur **OK**.  
La nouvelle carte de capacité métier apparaît dans la liste.


## Les propriétés d'une carte de capacités métier


La page **Caractéristiques** des propriétés d'une carte de capacités permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de l'objet, il s'agit de l'entreprise courante,
- son **Nom**,
- le texte de sa **Description**.

Avec **HOPEX IT Architecture** une carte de capacités métier est décrite par les pages suivantes :

- la page **Structure** qui permet de spécifier la liste des composants de la carte de capacités détenus et les dépendances entre eux.  
 Pour plus de détails sur les composants d'une carte de capacité métier, voir [Créer une carte de capacités métier](#).
- la page **Mises en œuvre** qui permet d'identifier les environnements d'application, de systèmes applicatifs logiques qui mettent en œuvre la carte de capacités métier décrite.

 Pour plus de détails sur la mise en œuvre d'une carte de capacités métier, voir [Créer une mise en oeuvre de capacité métier](#).

 Pour plus de détails sur les autres pages de propriétés proposées par **HOPEX IT Architecture**, voir [Contenu d'une page de propriétés HOPEX IT Architecture](#).

## Créer un diagramme de carte de capacités métier

Une carte de capacités métier peut être décrite par deux types de diagramme :

- un arbre de décomposition de capacités métier qui permet une description arborescente d'une capacité métier. Focalisé sur une capacité métier particulière, ce type de diagramme permet une représentation synthétique de sa décomposition en sous-capacités.
- un diagramme de structure de capacités métier qui permet de représenter l'ensemble des capacités métiers de la carte décrite.

Pour créer un diagramme de carte de capacités métier :

1. Ouvrez la page **Diagrammes** de la carte de capacités métier qui vous intéresse.
2. Cliquez sur **Créer un diagramme**.

3. Sélectionnez **Structure de capacité**.

Le **diagramme de carte de capacités métier** s'ouvre dans la zone d'édition. Le cadre de la carte de capacités métier décrite apparaît dans le diagramme.

☛ Vous pouvez construire ce diagramme en mode saisie tabulaire.



☛ Pour plus de détails sur l'utilisation de la saisie tabulaire, voir le chapitre "Saisie de diagramme en mode tabulaire" du guide **HOPEX Common Features**.

---

## Utiliser les capacités métier avec HOPEX IT Architecture



Une capacité métier est un ensemble de fonctionnalités qui peuvent être rendues accessibles par un système (une entreprise ou un système automatisé).

### Accéder à la liste des capacités métier avec HOPEX IT Architecture

Les capacités métier sont accessibles à partir de la liste des cartes de capacités métier.

Pour accéder à la liste des cartes de capacités à partir du menu de navigation **Capacités** :

1. Sélectionnez **Capacités > Capacités métier**.  
La liste des cartes de capacités métier apparaît.
2. Ouvrez la page **Structure** de la carte de capacités métier qui vous intéresse.
3. Dépliez la section **Composant de capacité détenu**, sélectionnez **Composant application logique**.  
La liste des capacités métier apparaît dans la zone d'édition.

### Créer une capacités métier

Pour créer une capacité métier à partir du menu de navigation **Capacités** :

1. Sélectionnez **Capacités > Capacités métier**.  
La liste des cartes de capacités métier apparaît.
2. Ouvrez la page **Structure** de la carte de capacités métier qui vous intéresse.
3. Dans la section **Composant de capacité détenu** cliquez sur **Nouveau**.  
La fenêtre **Création d'une capacités métier** apparaît.
4. Saisissez le **Nom** de votre capacité métier et cliquez sur **OK**.  
La nouvelle capacité métier apparaît dans la liste.

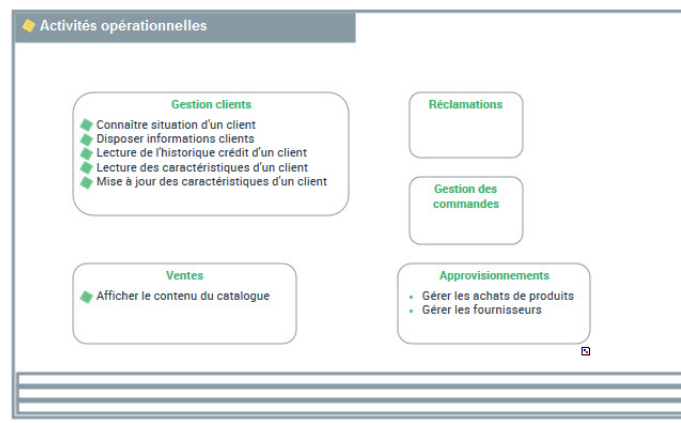
## Décrire une capacité métier

Une capacité métier est décrite plus précisément par les éléments suivants :

- la décomposition en capacité de granularité plus fine ;
- les effets attendus de la capacité ;
- les fonctionnalités requises, voir [Définir les fonctionnalités associées à une capacité métier](#) ;
- les dépendances entre capacités (l'effet attendu de l'une dépendant du résultat de l'autre).

✎ Pour plus de détails sur la gestion d'une capacité métier, voir le chapitre "Décrire une capacité métier" du guide **HOPEX IT Business Management**.

Par exemple, la capacité métier qui regroupe les activités opérationnelles se décompose en plusieurs capacités métier : "Gestion des clients", "Réclamations", "Ventes", "Approvisionnement" et "Gestion des commandes".



Une capacités métier peut être décrite par trois types de diagramme :

- un **arbre de décomposition des capacités métier** est un diagramme qui permet une description arborescente d'une capacité métier. Focalisé sur une capacité métier particulière, ce type de diagramme permet une représentation synthétique d'une décomposition en sous-capacités.
- un **arbre de capacités métier** est un diagramme montre la structure hiérarchique d'une capacité métier.
- un **diagramme de structure de capacités métier** qui permet de représenter l'ensemble des capacités métiers de la carte décrite.

## Définir les fonctionnalités associées à une capacité métier

📖 Une capacité technologique est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.



Chaque capacité métier est associée à des fonctionnalités qu'elle est en mesure de couvrir ainsi qu'aux compétences dont elle a besoin pour assurer ces fonctionnalités.

Par exemple, la capacité métier "Gérer les clients" requiert la fonctionnalité "Disposer informations clients".

☛ Pour plus de détails sur les fonctionnalités d'une entreprise, voir [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

Pour associer une **fonctionnalité** à une capacité métier :

1. Ouvrez la page de propriétés **Capacités attendues** de la capacité métier.
2. Dans la section **Fonctionnalité attendue**, cliquez sur **Nouveau**. Une fenêtre d'**Ajout de fonctionnalité** s'ouvre.
3. Vous pouvez relier une fonctionnalité existante ou en créer une nouvelle en spécifiant le nom de la nouvelle fonctionnalité.
4. Cliquez sur **OK**.  
La fonctionnalité attendue apparaît dans la liste des fonctionnalités associées à la capacité métier.

Les fonctionnalités et les effets attendus apparaissent, dans les diagrammes, en bas du cadre de la capacité décrite.

☛ Pour plus de détails sur les fonctionnalités d'une entreprise, voir [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

# UTILISER LES FONCTIONNALITÉS AVEC HOPEX IT

## ARCHITECTURE

Une **fonctionnalité** définit une aptitude attendue d'un équipement.

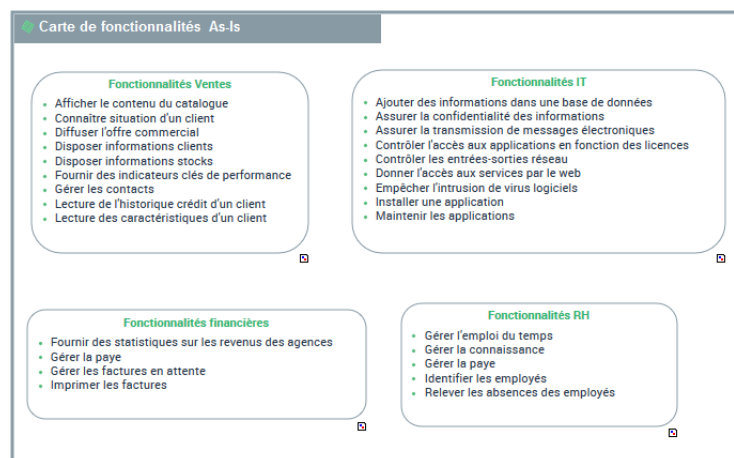


*Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.*

Une **carte de fonctionnalités** permet de décrire l'ensemble des fonctionnalités que l'entreprise est capable de couvrir pour ses besoins internes ou pour répondre aux attentes de ses clients.



*Une carte de fonctionnalités est un assemblage de fonctionnalités avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture matérielle ou logicielle.*



Exemple de carte de fonctionnalités



*Pour plus de détails sur la gestion d'une carte de fonctionnalités, voir le chapitre "Décrire la carte des fonctionnalités" du guide **HOPEX IT Business Management**.*

## Décrire une carte de fonctionnalités avec HOPEX IT Architecture



*Une carte de fonctionnalités est un assemblage de fonctionnalités avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture matérielle ou logicielle.*

## Accéder à la liste des cartes de fonctionnalités avec HOPEX IT Architecture

Pour accéder à la liste des capacités à partir du menu de navigation **Capacités** :

1. Sélectionnez **Capacités > Fonctionnalités**.  
La liste des cartes de fonctionnalités apparaît dans la zone d'édition.

## Créer une carte de fonctionnalités

Pour créer une carte des fonctionnalités à partir du menu de navigation **Capacités** :

1. Sélectionnez **Capacités > Fonctionnalités**.  
La liste des cartes de fonctionnalités apparaît dans la zone d'édition.
2. Cliquez sur **Nouveau**.
3. Saisissez le **Nom** de la carte de fonctionnalités et cliquez sur **OK**.
4. Sélectionnez **Carte de fonctionnalités**.  
La carte de fonctionnalités apparaît dans la liste.

## Créer un diagramme de carte de fonctionnalités

Une carte de fonctionnalités peut être décrite par deux types de diagramme :

- un arbre de décomposition de fonctionnalité qui permet une description arborescente d'une fonctionnalité (ou d'une carte de fonctionnalité). Focalisé sur une fonctionnalité particulière, ce type de diagramme permet une représentation synthétique de sa décomposition en sous-fonctionnalités.
- une carte de fonctionnalités qui permet de représenter l'ensemble des fonctionnalités de la carte décrite.

Pour créer un diagramme de carte de fonctionnalités :

1. Faites un clic droit sur la carte de fonctionnalités qui vous intéresse et cliquez sur **Créer Diagramme**.
2. Sélectionnez un type de diagramme.  
Le diagramme s'ouvre dans la zone d'édition. Le cadre de la carte de fonctionnalité décrite apparaît dans le diagramme.

Pour créer une fonctionnalité dans un diagramme de carte de fonctionnalités et définir les dépendances entre les fonctionnalités, voir le chapitre "Décrire la carte des fonctionnalités" du guide **HOPEX IT Business Management** .

## Les propriétés d'une carte de fonctionnalités

La page **Caractéristiques** des propriétés d'une carte de fonctionnalités permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de la carte de fonctionnalités, il s'agit de l'entreprise (ou de la bibliothèque) courante,
- son **Nom**,
- le texte de sa **Description**.

Avec **HOPEX IT Architecture** une carte de fonctionnalités est décrite par les pages suivantes :

- la page **Structure** qui permet de spécifier la liste des composants détenus et les dépendances entre eux.

☛ Pour plus de détails sur les composants d'une carte de fonctionnalités, voir [Créer un diagramme de fonctionnalités avec HOPEX IT Architecture](#).

- la page **Mises en œuvre** qui permet de spécifier les environnements qui permettent de réaliser la carte de fonctionnalités décrite.

☛ Pour plus de détails sur la mise en œuvre d'une fonctionnalité, voir [Créer une mise en œuvre de fonctionnalité](#).

---

## Décrire les fonctionnalités avec HOPEX IT Architecture



Une fonctionnalité est un service attendu pour effectuer un travail. Cette fonctionnalité peut être nécessaire généralement dans le cadre d'une activité ou plus précisément pour exécuter une opération particulière. Elle peut être fournie par une application s'il s'agit d'une fonctionnalité technique.

La page **Caractéristiques** des propriétés d'une fonctionnalité permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de la fonctionnalité, il s'agit de l'entreprise courante.
- son **Nom**,
- le texte de sa **Description**,
- ses **Effet de capacité attendu**.

☛ Pour plus de détails sur les effets attendus, voir [Créer un diagramme de fonctionnalités avec HOPEX IT Architecture](#).

## Créer un diagramme de fonctionnalités avec HOPEX IT Architecture

L'**arbre de décomposition d'une fonctionnalité** est un diagramme qui permet une description arborescente d'une fonctionnalité. Focalisé sur une fonctionnalité particulière, Ce type de diagramme permet une représentation synthétique de sa décomposition en sous- fonctionnalités.

Pour créer un diagramme de fonctionnalités :

1. Faites un clic droit sur la fonctionnalité qui vous intéresse et cliquez sur **Nouveau > Diagramme**.
2. Sélectionnez **Diagramme de fonctionnalités**.  
Le diagramme s'ouvre dans la fenêtre d'édition. Le cadre de la fonctionnalité décrite apparaît dans le diagramme.

Pour créer une fonctionnalité dans un diagramme de fonctionnalités, voir le chapitre "Créer une sous-fonctionnalité dans une carte de fonctionnalités" du guide **HOPEX IT Business Management** .

Pour définir les dépendances des sous-fonctionnalités, voir le chapitre "Définir les dépendances des sous-fonctionnalités" du guide **HOPEX IT Business Management** .

## Décrire une carte de capacités technologiques avec HOPEX IT Architecture


 Une carte de capacités technologiques est un assemblage de capacités technologiques avec leurs dépendances qui, conjointement, définissent le périmètre d'une architecture.

### Accéder à la liste des cartes de capacités technologiques avec HOPEX IT Architecture

Pour accéder à la liste des *cartes de capacités technologiques* à partir du menu de navigation **Inventaires** :

- 1 Sélectionnez **Déploiement > Capacités technologiques** dans le menu de navigation.  
La liste des cartes de capacités technologiques apparaît dans la zone d'édition.


### Décrire une capacité technologique

 Une capacité technologique est la capacité attendue d'un équipement (matériel ou logiciel) à assurer le fonctionnement d'un élément technique ou d'une application.

Avec **HOPEX IT Architecture**, l'utilisation des capacités technologiques et des cartes de capacités technologiques est identique à celle des fonctionnalités et des cartes de fonctionnalités.

 Pour plus de détails sur le fonctionnement des cartes de fonctionnalités et des fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#) et [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

### Décrire une capacité matérielle

 Une capacité matérielle est la capacité à fournir un résultat attendu par une ressource matérielle afin de réaliser un travail. Cette fonctionnalité matérielle est généralement nécessaire à un système informatique pour exécuter une opération spécifique.

Avec **HOPEX IT Architecture**, l'utilisation des fonctionnalités matérielles et des cartes de fonctionnalités matérielles est identique à celle des fonctionnalités et des cartes de fonctionnalités.

 Pour plus de détails sur le fonctionnement des cartes de fonctionnalités et des fonctionnalités, voir [Décrire une carte de fonctionnalités avec HOPEX IT Architecture](#) et [Décrire les fonctionnalités avec HOPEX IT Architecture](#).

Pour accéder à la liste des *cartes de fonctionnalités matérielles* à partir du menu de navigation **Inventaires** :

- 1 Sélectionnez **Matériel > Capacités matérielles**.  
La liste des cartes de capacités matérielles apparaît dans la zone d'édition.

# UTILISER LES MÉCANISMES DE MISE EN ŒUVRE

Le mécanisme de mise en œuvre permet de relier un élément qui correspond à ce que l'on sait faire ou que l'on cherche à faire, aux moyens de réalisation qui sont représentés par :

- des éléments concrets, tels que des *applications* ou des *systèmes applicatifs*.
- des éléments de niveau conceptuel, c'est-à-dire en amont des choix organisationnels et techniques.

---

## Décrire la mise en œuvre d'une capacités métier

Il s'agit ici de relier la *capacité métier*, qui correspond à ce que l'on sait faire ou que l'on cherche à faire, aux moyens de réalisation qui sont représentés par :

- des *applications logiques* ou des *Systèmes applicatifs logiques*, par exemple, à un niveau conceptuel, c'est-à-dire en amont des choix organisationnels et techniques.



*Un système applicatif logique est un assemblage d'autres architectures, d'applications logiques et d'utilisateurs finaux interagissant avec les composants applicatifs, dans le but de mettre en œuvre une ou plusieurs fonctionnalités.*

- des *applications* ou des *systèmes applicatifs*, par exemple, à un niveau technique.

Par exemple, la construction de la *carte de capacités de métier* d'une part et de l'*environnement de système applicatif logique* d'autre part permet de vérifier que les capacités métier sont bien mises en œuvre par des applications logiques.



*Les représentations conceptuelles se font en amont des choix organisationnels et techniques.*

## Créer une mise en oeuvre de capacité métier

Une capacité métier peut être mise en œuvre soit par une application ou un système applicatif, ou alors, à un niveau conceptuel, par une application logique ou un système applicatif logique.

Pour associer une application à une capacité, vous devez créer une mise en œuvre de capacité métier.



*Une mise en œuvre de capacité métier représente l'agent physique (Système applicatif) ou logique (Fonction métier) qui met en œuvre la capacité.*

Pour spécifier qu'une capacité métier est mise en œuvre par une application existante :

1. Ouvrez la page de propriétés **Mises en œuvre** de la capacité métier qui vous intéresse.
2. Cliquez sur **Nouveau**.  
La fenêtre de création d'une mise en œuvre de capacité métier s'ouvre.

3. Cochez **Ajouter un composant et le connecter à un type** et sélectionnez le type **Application**.
4. Dans la liste des applications qui apparaît, sélectionnez l'application qui vous intéresse et cliquez sur **OK**.  
La réalisation de capacité métier apparaît dans la liste avec le nom de l'application sélectionnée.

## Analyser la mise en œuvre des capacités de l'entreprise

**HOPEX IT Architecture** met à votre disposition des rapports qui vous présentent la couverture de réalisation des éléments des capacités métier par des éléments opérationnels tels que des applications, et ce selon différents axes : Organisationnel, Métier/Données, application logique/physique, etc.

☛ Pour plus de détails sur les rapports de mise en œuvre des capacités de l'entreprise, voir [Rapport de décomposition des briques d'entreprise](#).

---

## Décrire la mise en œuvre d'une fonctionnalité

Il s'agit ici de relier la **fonctionnalité**, qui contribue à l'objectif à atteindre, aux moyens de mise en œuvre qui sont représentés par des **applications** (ou des **applications logiques** à un niveau conceptuel) ou des **Systèmes applicatifs** (ou des **Systèmes applicatifs logiques**) ou des **services du Cloud**.

☛ Les représentations conceptuelles se font en amont des choix organisationnels et techniques.

## Créer une mise en œuvre de fonctionnalité

Une fonctionnalité peut être mise en œuvre soit par une application ou un système applicatif, ou alors, à un niveau conceptuel, par une application logique ou un système applicatif logique.

Pour associer une application à une fonctionnalité, vous devez créer une mise en œuvre de fonctionnalité.

📖 Une mise en œuvre décrit une relation entre une entité logique et l'entité physique qui la met en œuvre. L'entité physique donne la liste des entités logiques qu'elle implémente.

Pour spécifier qu'une fonctionnalité est mise en œuvre par une nouvelle application :

1. Ouvrez la page de propriétés **Mises en œuvre** de la fonctionnalité qui vous intéresse.
2. Cliquez sur **Nouveau**.  
La fenêtre de création d'une mise en œuvre de fonctionnalité s'ouvre.
3. Cochez **Ajouter un composant avec un nouveau type** et sélectionnez le type **Application**.
4. Cliquez sur **OK**.  
Une fenêtre de création d'application s'ouvre.

5. Spécifiez le **Nom** et le **Détenteur** de votre application et cliquez sur **OK**. La mise en œuvre de fonctionnalité apparaît dans la liste avec le nom de l'application sélectionnée.

☛ Les composants mis en œuvre par des capacités technologiques ou matérielles apparaissent dans les diagrammes qui présentent ces fonctionnalités.

## Identifier les applications associées aux fonctionnalités

Les *applications* couvrent des *fonctionnalités* associées à des *capacités métier*. Un rapport **HOPEX IT Architecture** permet de vérifier la couverture de réalisation des fonctionnalités par des éléments opérationnels tels que des composants applicatifs logiques ou physique.

☛ Pour plus de détails sur ce rapport de décomposition, voir [Rapport de décomposition des briques d'entreprise](#).

☛ Pour obtenir un exemple de mise en œuvre de capacités technologiques par des Services Cloud, voir [Accéder à la liste des services cloud](#).

---

## Accéder aux mises en œuvre à partir d'un point de service

Les services assurés par des briques logicielles (applications ou services applicatifs) sont accessibles par des *points de service*.

📖 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.

☛ Pour plus de détails sur les points de service, voir [Décrire les points de service et de requête](#).

☛ Les modalités de demande de service sont définies par une *interface de service* affecté au point de service. Pour plus de détails sur les interfaces de service, voir [Décrire une interface de service](#).

Les services assurés par une brique logicielle peuvent répondre à des mises en œuvre de fonctionnalités ou de capacité métier.

Ainsi un *point de service* peut également être relié à l'une des mises en œuvre de la brique logicielle qui le détient.

Pour spécifier les mises en œuvre de capacité couvertes par un point de service :

1. Ouvrez la page de propriétés **Mises en œuvre publiées** du point de service qui vous intéresse.
2. Sélectionnez l'onglet qui correspond au type de mise en œuvre qui vous intéresse.

☛ Seules les mises en œuvre de l'objet qui détient le point de service ne peuvent être reliées au point de service.



# MODÉLISER LES INFRASTRUCTURES INFORMATIQUES



Les fonctionnalités proposées par **HOPEX IT Architecture V2** pour la modélisation des infrastructures complexes permettent de représenter les ressources matérielles, informatiques et organisationnelles nécessaires au déploiement du système et à son fonctionnement : les interactions de service entre les composants, les moyens de communication qui supportent ces interactions de service ainsi que les services offerts et utilisés par l'architecture modélisée.

L'ensemble des éléments d'infrastructure modélisés sont accessibles à partir du menu de navigation **Infrastructure > Infrastructure**.

Les points traités ici sont les suivants :

- ✓ [Décrire les architectures de ressources](#) ;
- ✓ [Décrire les infrastructures informatiques](#) ;
- ✓ [Décrire les ordinateurs](#) ;
- ✓ [Décrire les communications dans une infrastructure informatique](#).

## DÉCRIRE LES ARCHITECTURES DE RESSOURCES

Une *architecture de ressources* est composée de l'ensemble des ressources matérielles, informatiques et organisationnelles nécessaires au fonctionnement d'une infrastructure complexe (système).

Les communications entre ces composants sont représentées par des interactions de service et les moyens matériels, supportant ces interactions de service, sont des canaux réseau.



*Une architecture de ressources est la combinaison d'équipements matériels et d'organisations configurés pour fournir une capacité.*

Les services offerts par le système à ses utilisateurs sont représentés par des points de service. Les points de service sont physiquement supportés par les ports de communication qui permettent l'accès aux moyens de communication du système.

---

### Décrire une architecture de ressources



*Une architecture de ressources est la combinaison d'équipements matériels et d'organisations configurés pour fournir une capacité.*

Pour créer une architecture de ressources à partir du menu de navigation **Infrastructure** :

1. Sélectionnez **Infrastructure > Architecture de ressources**.  
L'arbre des architectures de ressources apparaît.
2. Sélectionnez le dossier **Architecture de ressources** et cliquez sur **Nouveau > Architecture de ressources**.  
La fenêtre **Création d'une architecture de ressources** apparaît.
3. Saisissez le **Nom** de votre architecture ainsi que son **Détenteur** et cliquez sur **OK**.  
La nouvelle architecture de ressources apparaît dans l'arbre.

### Créer un diagramme d'assemblage de ressources

Pour créer un diagramme d'assemblage de ressources :

1. Sélectionnez **Infrastructure > Architecture de ressources**.  
L'arbre des architectures de ressources apparaît.
2. Sélectionnez l'architecture de ressources qui vous intéresse et cliquez sur **Créer un diagramme**.
3. Sélectionnez **Diagramme structuré**.  
Le diagramme d'assemblage de ressources s'ouvre dans la zone d'édition.  
Le cadre de la carte de l'architecture de ressources décrite apparaît dans le diagramme.

## Utiliser un diagramme d'assemblage de ressources

### Ajouter une architecture de ressources

Pour décrire qu'une architecture de ressources, telle qu'un « centre d'appels », met en œuvre une autre architecture de ressources, telle qu'un « service de gestion des clients » par exemple, vous allez ajouter l'architecture de ressources utilisée dans le diagramme de l'architecture de ressources utilisatrice.

Pour ajouter une architecture de ressources dans un diagramme d'assemblage de ressources :

1. Dans la barre d'objets du diagramme d'assemblage de ressources, cliquez sur **Architecture de ressources**.
2. Cliquez dans le cadre du diagramme d'assemblage de ressources. Une fenêtre d'ajout vous demande de choisir l'architecture de ressources.
3. Sélectionner une architecture de ressources existante et cliquez sur **OK**.

☛ Vous pouvez aussi créer une nouvelle architecture de ressources, il vous suffit de saisir son nom et de cliquer sur le bouton **Créer**.

### Ajouter une infrastructure informatique ou une configuration de ressources

📖 Une infrastructure informatique est composée de différents matériels informatiques tels que : des terminaux informatiques techniques, des serveurs ou des réseaux informatiques.

☛ Pour plus de détails sur les infrastructures informatiques, voir [Décrire les infrastructures informatiques](#).

📖 Une configuration de ressources est un ensemble de ressources physiques et humaines configurées pour fournir une capacité métier.

☛ Pour plus de détails sur les configurations de ressources, voir [Décrire une configuration de ressource](#).

Pour décrire qu'une architecture de ressources s'appuie sur des ressources informatiques telles que un réseau de communication, des postes de travail hébergeant des applications, vous allez ajouter au diagramme d'assemblage de ressources des composants de type **Infrastructure informatique**.

📖 Une infrastructure informatique est composée de différents matériels informatiques tels que : des terminaux informatiques techniques, des serveurs ou des réseaux informatiques.

☛ Pour plus de détails sur les infrastructures informatiques, voir [Décrire les infrastructures informatiques](#).

Pour créer une **Infrastructure informatique** :

1. Dans la barre d'objets du diagramme, cliquez sur **Infrastructure informatique**.
2. Cliquez dans le cadre du diagramme. Une fenêtre d'ajout vous demande de choisir l'infrastructure informatique à déployer.

3. Sélectionnez l'infrastructure informatique qui vous intéresse et cliquez sur **OK**.

L'infrastructure informatique apparaît dans le diagramme.

☛ Pour créer une infrastructure informatique, il vous suffit de saisir son nom et de cliquer sur le bouton **Créer**.

☛ De la même manière, vous pouvez ajouter une **configuration de ressources** dans le diagramme d'assemblage de ressources. Pour plus de détails sur les configurations de ressources, voir [Décrire une configuration de ressource](#).

### Ajouter un acteur ou un type de position

📖 Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.

📖 Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.

Pour décrire que l'architecture de ressources du centre d'appels, s'appuie sur des opérateurs pour prendre les appels et traiter les demandes, vous allez créer un **type de position**.

Pour ajouter un type de position dans un diagramme d'assemblage de ressources :

1. Dans la barre d'objets du diagramme d'assemblage de ressources, cliquez sur **Type de position**.
2. Cliquez dans le cadre de l'architecture de ressources décrite. Une fenêtre d'ajout vous demande de choisir le **Type de position** que vous souhaitez utiliser.
3. Sélectionnez le type de position qui vous intéresse et cliquez sur **OK**. Le type de position apparaît dans le diagramme.

☛ Pour créer un type de position, il vous suffit de saisir son nom et de cliquer sur le bouton **Créer**.

☛ De la même manière, vous pouvez ajouter un **acteur** dans le diagramme d'assemblage de ressources.

### Décrire les services dans un diagramme d'assemblage de ressources

Une architecture de ressources est créée pour assurer un ou plusieurs services.

Les services attendus et réalisés sont représentés par :

- des **points de service**

📖 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.


- des **points de requête**

📖 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.


☛ Pour plus de détails, voir [Les points de service](#) et [Les points de requête](#).

## Décrire les interactions de service dans un diagramme d'assemblage de ressources

Dans un diagramme d'assemblage de ressources, les **interactions de service** permettent de représenter les échanges prévus entre les entités organisationnelles en vue de réaliser ces services.

 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

Les modalités d'échange sont définies par un **interface de service** qui est affectée à l'interaction de service.

 Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

Vous pouvez définir des interactions de service entre :

- deux composants de type **Architecture des ressources** pour représenter les échanges prévus entre ces entités,
- un composant de type **Architecture des ressources** et une **infrastructure informatique** pour représenter les modalités d'utilisation de la ressource matérielle par la ressource organisationnelle. Par exemple, vous pouvez représenter que l'utilisation du matériel opérateur est utilisé sur réservation.
- deux composants de type **infrastructure informatique** pour représenter les modalités d'utilisation d'une ressource informatique par une autre dans le contexte de l'architecture de ressources modélisée.
- un **point de service** et un ou plusieurs composants de type **Architecture de ressources** pour représenter la mise en œuvre du service au sein de l'architecture de ressources,
- un composant de type **Architecture de ressources** et un **point de requête** pour représenter que l'entité fait appel à une ressource organisationnelle extérieure.

➡ Pour plus de détails, voir [Les interactions de service](#).

## Les canaux et les ports de communication


Dans une architecture de ressources, les canaux réseau supportent le transfert des informations d'un équipement matériel à un autre.

➡ Pour plus de détails sur la création de ces canaux et des protocoles de communication associés, voir [Les canaux réseau](#).

Les Ports de communication permettent de connecter les équipements de l'architecture de ressources avec des éléments matériels extérieurs.

---

## Décrire un environnement d'architecture de ressources

 Un environnement d'architecture métier représente les relations d'un domaine fonctionnel métier avec ses partenaires.

## Créer un environnement d'architecture de ressources

Pour créer un environnement d'architecture de ressources à partir du menu de navigation **Infrastructure** :

1. Sélectionnez **Infrastructure > Architecture de ressources**.  
L'arbre des architectures de ressources apparaît.
2. Ouvrez les propriétés de l'architecture de ressources qui vous intéresse.
3. Cliquez sur l'onglet **Environnements**.  
La liste des environnements de l'architecture de ressources apparaît.
4. Cliquez sur **Nouveau**.  
Le nouvel environnement d'architecture de ressources apparaît dans la liste.
5. Ouvrez les propriétés de l'environnement d'architecture de ressources pour modifier son **Nom** ainsi que son **Détenteur**.

## Les propriétés d'un environnement d'architecture de ressources

La page **Caractéristiques** des propriétés d'un environnement d'architecture de ressources permet d'accéder à :

- son **Détenteur**, par défaut lors de la création de l'objet, il s'agit de l'entreprise courante,
- son **Nom**,
- le texte de sa **Description**.

Avec **HOPEX IT Architecture V2** un environnement d'architecture de ressources est décrit par les pages suivantes :

- la page **Composants** qui permet de spécifier la liste des composants de l'environnement d'architecture de ressources internes et partenaires.
- la page **Diagrammes** qui permet de créer un diagramme de l'objet.
- la page **Rapports** qui permet d'accéder à la liste des rapports disponibles sur l'objet.

## Créer un diagramme d'environnement d'architecture de ressources

Pour créer un diagramme d'environnement d'architecture de ressources :

1. Ouvrez la page de propriétés **Environnements** de l'architecture de ressources qui vous intéresse.
2. Sélectionnez l'environnement d'architecture de ressources et cliquez sur **Créer diagramme**.
3. Sélectionnez **Diagramme Structuré**.  
Le diagramme d'environnement d'architecture de ressources s'ouvre dans la zone d'édition.

## Décrire un diagramme d'environnement d'architecture de ressources

### **Ajouter une architecture de ressources**



*Une architecture de ressources est la combinaison d'équipements matériels et d'organisations configurés pour fournir une capacité.*

Pour ajouter une architecture de ressources dans un diagramme d'environnement d'architecture de ressources :

1. Dans la barre d'objets du diagramme d'environnement d'architecture de ressources, cliquez sur **Architecture de ressources**.
2. Cliquez dans le cadre de l'environnement d'architecture de ressources décrit.  
Une fenêtre d'ajout vous demande de choisir l'architecture de ressources mise en œuvre. Vous pouvez sélectionner une architecture de ressources existante ou en créer une nouvelle.

☛ Dans le cas où l'architecture de ressources que vous souhaitez utiliser n'existe pas encore dans le référentiel, il vous suffit de saisir son nom.

3. Cliquez sur **OK**.

### Créer une architecture de ressources partenaire

Pour décrire qu'une architecture de ressources externe est mise en œuvre dans l'environnement décrit, vous allez ajouter un composant de type **architecture de ressources partenaire** dans le diagramme d'environnement .

📖 Une architecture de ressources partenaire est la mise en place d'une architecture de ressources externe dans une autre architecture de ressources ou dans un environnement d'architecture de ressources.

Pour créer une **Architecture de ressources partenaire** :

1. Dans la barre d'objets du diagramme d'environnement d'architecture de ressources, cliquez sur le bouton **Ressource partenaire** et sélectionnez **Architecture de ressources**.
2. Cliquez dans le cadre de l'environnement d'architecture de ressources décrit.  
Une fenêtre d'ajout vous demande de choisir l'architecture de ressources à ajouter. Vous pouvez saisir le nom d'une nouvelle architecture de ressources.
3. Cliquez sur **OK**.

### Créer une ressource humaine

Pour décrire que les services de l'environnement d'architecture de ressources sont utilisés par des clients, par exemple, vous allez créer un **type de position** ou un **acteur**.


📖 Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.

📖 Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.


☛ Pour plus de détails sur la création d'une ressource humaine, voir [Ajouter un acteur ou un type de position](#).

## Décrire les communications

Les éléments suivants permettent de décrire que les communications techniques et organisationnelles :


- des ports et des canaux réseau,  
 Pour plus de détails, voir [Décrire les communications techniques](#).
- des interactions de service, des points de service et des points de requête,

 Pour plus de détails, voir [Décrire les communications liées aux services](#).

 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

---

## Décrire une configuration de ressource

 Une configuration de ressources est un ensemble de ressources physiques et humaines configurées pour fournir une capacité métier.

### Créer une configuration de ressource

Pour créer une configuration de ressource à partir du menu de navigation **Infrastructure** :

1. Sélectionnez **Infrastructure > Configuration de ressource**.  
La liste des configuration de ressource apparaît.
2. Cliquez sur **Nouveau**.  
La fenêtre **Création d'une Configuration de ressource** apparaît.
3. Saisissez le **Nom** de votre configuration de ressource ainsi que son **Détenteur** et cliquez sur **OK**.  
La nouvelle configuration de ressource apparaît dans la liste.

### Créer un diagramme de configuration de ressource








Pour créer un diagramme de configuration de ressource :

1. Sélectionnez la configuration de ressource et cliquez sur **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré**.  
Le diagramme de configuration de ressource s'ouvre dans la zone d'édition.



## Utiliser un diagramme de configuration de ressource

Vous pouvez insérer dans un diagramme de configuration de ressource :

- des **infrastructures informatiques**, voir [Décrire une infrastructure informatique](#),  
 Une infrastructure informatique est composée de différents matériels informatiques tels que : des terminaux informatiques techniques, des serveurs ou des réseaux informatiques.
- des **objets connectés**,  
 Un objet connecté est à la fois un équipement matériel et un ordinateur, qui fournit des services matériels et informatiques combinés à ses utilisateurs. En tant qu'équipement matériel, il embarque des capteurs (accéléromètre, par exemple) qui fournissent des données aux ordinateurs embarqués. En tant qu'ordinateur, il peut héberger des bases de données ou exécuter des applications. Exemples d'objets connectés : montre connectée avec traqueur GPS, caméra de vidéo surveillance en ligne avec flux vidéo IP, pèse personne connectés avec gestion de l'historique du poids.
- des **réseaux informatiques**, voir [Décrire un réseau informatique](#),  
 Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur informatique). Un réseau informatique peut se décomposer en sous-réseaux.
- des éléments **matériel**, voir [Décrire un matériel](#),  
 Un matériel non-informatique peut embarquer des ordinateurs. De façon conjointe avec ses ordinateurs, il fournit les services d'information et les données. Exemples : camion connecté avec une application de calendrier de livraison, drone connecté avec application de paiement en ligne. Un matériel fournit des services matériels. Par exemple : un réfrigérateur connecté fournit des fonctionnalités de commandes et bien entendu une fonctionnalité matérielle de refroidissement, les drones connectés volent et fournissent une fonctionnalité de paiement en ligne.
- des **types de position** ou des **acteurs**.  
 Un type de position représente un statut affecté à un individu ou à un groupe d'individus, afin de définir un poste dans une organisation ou une hiérarchie.  
 Un acteur représente une personne ou un groupe de personnes qui interviennent dans les processus ou dans le système d'information de l'entreprise. Un acteur peut être interne ou externe à l'entreprise. Un acteur interne représente un élément de l'organisation d'une entreprise tel qu'une direction, un service ou un poste de travail. Il est défini à un niveau plus ou moins fin en fonction de la précision à fournir sur l'organisation (cf type d'acteur). Ex : la direction financière, la direction commerciale, le service marketing, l'agent commercial. Un acteur externe représente un organisme qui échange des flux avec l'entreprise. Ex : Client, Fournisseur, Administration.  
 Pour plus de détails sur la création d'une ressource humaine, voir [Ajouter un acteur ou un type de position](#).

- des ports et des canaux réseau, voir [Décrire les communications techniques](#).
- des points de service et des points de requête, voir [Décrire les communications liées aux services](#).
- des interactions de service, voir [Décrire les communications techniques](#).



*Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*

---

## Décrire un matériel



*Un matériel non-informatique peut embarquer des ordinateurs. De façon conjointe avec ses ordinateurs, il fournit les services d'information et les données. Exemples : camion connecté avec une application de calendrier de livraison, drone connecté avec application de paiement en ligne. Un matériel fournit des services matériels. Par exemple : un réfrigérateur connecté fournit des fonctionnalités de commandes et bien entendu une fonctionnalité matérielle de refroidissement, les drones connectés volent et fournissent une fonctionnalité de paiement en ligne.*

### Créer un matériel

Pour créer une pièce matérielle à partir du menu de navigation **Inventaires** :

1. Sélectionnez **Matériel > Matériels**.  
La liste des éléments matériels apparaît.
2. Cliquez sur **Nouveau**.  
Le matériel crée apparaît dans la liste.

### Créer un diagramme de structure d'éléments matériels

Pour créer un diagramme de structure d'éléments matériels :

1. Sélectionnez le matériel qui vous intéresse et cliquez sur **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré**.  
Le diagramme de structure d'éléments matériels s'ouvre dans la zone d'édition.

### Utiliser un diagramme de structure d'élément matériel

Vous pouvez insérer dans un diagramme de structure d'élément matériel :

- des *serveurs informatiques* et des *terminaux informatiques*, voir [Décrire un ordinateur](#),



*Un serveur informatique est un matériel informatique fournissant un service à des utilisateurs connectés via un réseau informatique. Ce*

matériel informatique peut héberger des bases de données ou exécuter des applications.



Un terminal informatique est un ordinateur qui fournit un service directement à l'utilisateur final. Cet ordinateur peut héberger des bases de données et exécuter des applications. Il s'agit par exemple d'un poste de travail, d'un ordinateur portable ou d'un téléphone intelligent.

- des **objets connectés**, voir [Décrire un ordinateur](#),



Un objet connecté est à la fois un équipement matériel et un ordinateur, qui fournit des services matériels et informatiques combinés à ses utilisateurs. En tant qu'équipement matériel, il embarque des capteurs (accéléromètre, par exemple) qui fournissent des données aux ordinateurs embarqués. En tant qu'ordinateur, il peut héberger des bases de données ou exécuter des applications. Exemples d'objets connectés : montre connectée avec traqueur GPS, caméra de vidéo surveillance en ligne avec flux vidéo IP, pèse personne connectés avec gestion de l'historique du poids.

- des **Composants matériels**, voir [Décrire un matériel](#),
- des ports de communication et des canaux réseau, voir [Décrire les communications techniques](#).
- des points de service et des points de requête, voir [Décrire les communications liées aux services](#).
- des interactions de service.




Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.



Pour plus de détails sur les interactions de service, voir [Décrire les communications liées aux services](#).

# DÉCRIRE LES INFRASTRUCTURES INFORMATIQUES

## Décrire une infrastructure informatique

 Une infrastructure informatique est composée de différents matériels informatiques tels que : des terminaux informatiques techniques, des serveurs ou des réseaux informatiques.

Vous pouvez décrire les composants d'une *infrastructure informatique* dans un diagramme de structure d'assemblage d'infrastructure.

### Créer une infrastructure informatique

Pour créer une infrastructure informatique à partir du menu de navigation **Infrastructure** :

1. Sélectionnez **Infrastructure > Infrastructures informatiques**.  
L'arbre des infrastructures informatiques apparaît.
2. Sélectionnez le dossier **Infrastructures informatiques** et cliquez sur **Nouveau > Infrastructures informatiques**.  
La fenêtre **Création d'une Infrastructure informatique** apparaît.
3. Saisissez le **Nom** de votre infrastructure ainsi que son **Détenteur** et cliquez sur **OK**.  
La nouvelle infrastructure informatique apparaît dans l'arbre.

### Créer un diagramme de structure d'assemblage d'infrastructure

Pour créer un diagramme de structure d'assemblage d'infrastructure :


1. Sélectionnez l'infrastructure informatique et cliquez **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré**.  
Le diagramme de structure d'assemblage d'infrastructures s'ouvre dans la zone d'édition.


### Utiliser un diagramme de structure d'assemblage d'infrastructure

Vous pouvez insérer dans un diagramme de structure d'assemblage d'infrastructure :

Vous pouvez insérer dans ce diagramme :


- des *serveurs informatiques* et des *terminaux informatiques*, voir [Décrire un ordinateur](#),

 Un serveur informatique est un matériel informatique fournissant un service à des utilisateurs connectés via un réseau informatique. Ce matériel informatique peut héberger des bases de données ou exécuter des applications.


 Un terminal informatique est un ordinateur qui fournit un service directement à l'utilisateur final. Cet ordinateur peut héberger des bases

de données et exécuter des applications. Il s'agit par exemple d'un poste de travail, d'un ordinateur portable ou d'un téléphone intelligent.


- des **objets connectés**, voir [Décrire un ordinateur](#),

 Un objet connecté est à la fois un équipement matériel et un ordinateur, qui fournit des services matériels et informatiques combinés à ses utilisateurs. En tant qu'équipement matériel, il embarque des capteurs (accéléromètre, par exemple) qui fournissent des données aux ordinateurs embarqués. En tant qu'ordinateur, il peut héberger des bases de données ou exécuter des applications. Exemples d'objets connectés : montre connectée avec traqueur GPS, caméra de vidéo surveillance en ligne avec flux vidéo IP, pèse personne connectés avec gestion de l'historique du poids.


- des **périphériques réseau**, voir [Décrire un périphérique réseau](#),


 Un périphérique informatique peut héberger et exécuter un logiciel. De façon conjointe avec ses logiciels hébergés, il fournit des services technologiques. Il s'agit, par exemple d'une borne wifi, d'un parefeu, d'une imprimante, d'un disque dur.

- des **Composants de réseau informatique**, voir [Décrire un réseau informatique](#),

 Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur informatique). Un réseau informatique peut se décomposer en sous-réseaux.

- des ports et des canaux réseau, voir [Décrire les communications techniques](#).
- des points de service et des points de requête, voir [Décrire les communications liées aux services](#)
- des interactions de service.

 Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.


 Pour plus de détails sur les interactions de service, voir [Décrire les communications liées aux services](#).

Le rapport **Description d'infrastructure** permet d'analyser les composants des infrastructures et les relations prévues entre eux afin de vérifier que chaque élément décrit dans le diagramme héberge un composant logiciel ou matériel.

 Pour plus de détails, voir [Rapport de description d'infrastructure](#).

---

## Décrire un réseau informatique

 Un réseau informatique est un assemblage de terminaux informatiques (ex: routeurs, commutateurs, pare-feux) qui permet la communication distante entre des ordinateurs (ex: serveur informatique). Un réseau informatique peut se décomposer en sous-réseaux.

## Créer un réseau informatique

Pour créer un réseau informatique :


1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructures > Réseaux informatiques**.  
La liste des réseaux informatiques apparaît.
2. Cliquez sur **Nouveau**.  
La fenêtre **Création d'un réseau informatique** apparaît.
3. Saisissez le **Nom** de votre réseau ainsi que son **Détenteur** et cliquez sur **OK**.

## Créer un diagramme de réseau informatique

Un réseau informatique est décrit par un diagramme de structure d'assemblage d'infrastructure.

Pour créer un diagramme de structure d'assemblage d'infrastructures à partir d'un réseau informatique :

1. Faites un clic droit sur le réseau informatique et sélectionnez **Créer un diagramme**.
2. Sélectionnez **Diagramme structuré**.  
Le diagramme de structure d'assemblage d'infrastructure s'ouvre dans la zone d'édition.

 Pour plus de détails sur ce type de diagramme, voir [Utiliser un diagramme de structure d'assemblage d'infrastructure](#).

---

## Décrire une installation



Une installation est un modèle de site d'intérêt pour l'entreprise.  
Par exemple : un centre de données, une usine ou une agence.

## Créer une installation

Pour créer une installation :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructures > Installations**.  
La liste des installations apparaît.
2. Cliquez sur **Nouveau**.  
La fenêtre **Création d'une installation** apparaît.
3. Saisissez le **Nom** de votre installation ainsi que son **Détenteur** et cliquez sur **OK**.

## Créer un diagramme de configuration de ressources pour une installation

Une installation est décrite par un diagramme de configuration de ressources.

Pour créer un diagramme de configuration de ressources à partir d'une installation :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructures > Installations**.

2. Faites un clic droit sur une installation et sélectionnez **Créer un diagramme**.
3. Sélectionnez **Diagramme structuré**.  
Le diagramme de configuration de ressources s'ouvre dans la zone d'édition.

☞ Pour plus de détails sur ce type de diagramme, voir [Utiliser un diagramme de configuration de ressource](#).

# DÉCRIRE LES ORDINATEURS

## Décrire un ordinateur

### Accéder à la liste des ordinateurs

Pour accéder à l'ensemble des différents types d'ordinateurs :

- 1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructure > Ordinateurs**.

La liste de tous les différents types d'ordinateur apparaît :

- les **objets connectés**,



Un objet connecté est à la fois un équipement matériel et un ordinateur, qui fournit des services matériels et informatiques combinés à ses utilisateurs. En tant qu'équipement matériel, il embarque des capteurs (accéléromètre, par exemple) qui fournissent des données aux ordinateurs embarqués. En tant qu'ordinateur, il peut héberger des bases de données ou exécuter des applications. Exemples d'objets connectés : montre connectée avec traqueur GPS, caméra de vidéo surveillance en ligne avec flux vidéo IP, pèse personne connectés avec gestion de l'historique du poids.

- les **serveurs informatiques**,



Un serveur informatique est un matériel informatique fournissant un service à des utilisateurs connectés via un réseau informatique. Ce matériel informatique peut héberger des bases de données ou exécuter des applications.

- les **services cloud**,



Le service Cloud (considéré comme un objet connecté) peut être utilisé dans des architectures de déploiement. L'image du service cloud apparaît alors dans la forme du module de déploiement.



Pour plus de détails sur les services Cloud, voir [Utiliser les services cloud](#).

- les **terminaux informatiques**.



Un terminal informatique est un ordinateur qui fournit un service directement à l'utilisateur final. Cet ordinateur peut héberger des bases de données et exécuter des applications. Il s'agit par exemple d'un poste de travail, d'un ordinateur portable ou d'un téléphone intelligent.

### Créer un terminal informatique

Pour créer un terminal informatique de type **ordinateur** :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructure > Ordinateurs**  
Les différents types d'ordinateurs sont classés dans des dossiers.
2. Sélectionnez le dossier qui correspond au type d'ordinateur que vous souhaitez créer et cliquez sur **Nouveau**.
3. Saisissez le **Nom** de votre ordinateur ainsi que son **Détenteur** et cliquez sur **OK**.  
Le nouvel ordinateur apparaît dans la liste.




## Créer un diagramme d'assemblage d'ordinateurs

Pour créer un diagramme d'assemblage d'ordinateurs :


1. Faites un clic droit sur l'ordinateur et sélectionnez **Créer diagramme**.
  2. Sélectionnez **Diagramme structuré**.
- Le diagramme d'assemblage d'ordinateurs s'ouvre dans la zone d'édition.

Vous pouvez insérer dans un diagramme d'assemblage d'ordinateurs :


- des *hébergements de module de déploiement*, voir [Ajouter un module de déploiement à un diagramme d'architecture de déploiement d'une application](#),

 *Un module de déploiement d'une application représente un élément de découpage du code d'une application en fonction de critères techniques ou d'objectifs d'hébergement. Par exemple, ce peut être le découpage en N tiers, la séparation Front End/Back End/... ou IHM/Logique métier/Base de données etc... Chaque module de déploiement d'une application est associé à des technologies requises (pour son exécution) et peut contenir le code de plusieurs services applicatifs. Un architecte peut préconiser un type d'hébergement du module (IaaS/PaaS cloud service ou un modèle de serveur).*


- des *hébergements de technologie logicielle*,

 *Une technologie logicielle est un composant de base nécessaire au fonctionnement des applications métiers. Parmi les technologies logicielles figurent tous les logiciels de base tels que : serveur d'application, serveur de messagerie, composants logiciels pour la présentation, la saisie, le stockage, le partage de l'information métier, les systèmes d'exploitation, les middlewares, les navigateurs.*


- des *hébergements de microservices*,

 *Un microservice est un composant logiciel qui peut se déployer de manière autonome, mais qui ne fournit pas directement un service à l'utilisateur final. Il peut interagir avec d'autres services applicatifs, applications ou systèmes applicatifs. C'est un composant logiciel déployable qui utilise des technologies logicielles. Par exemple : service d'authentification, service d'impression de fichiers PDF.*

- des *hébergements de dépôt de données*,

 *Un dépôt de données fournit un mécanisme pour mettre à jour ou consulter des données qui vont persister au-delà de la portée du processus en cours. Il permet de stocker les données des flux en entrée et de les retransmettre par l'intermédiaire d'un ou plusieurs flux en sortie.*

- des ports et des canaux réseau,
- des points de service et des points de requête,
- des interactions de service.

 *Pour plus de détails sur la création des interactions de service, points de service et de requête, des canaux et des protocoles de communication associés, voir [Décrire les communications dans une infrastructure informatique](#).*

---

## Décrire un périphérique réseau

### Accéder à la liste des périphériques réseau

Les différents types de périphériques réseau proposés en standard par **HOPEX IT Architecture V2** est la suivante :

- Hub,
- Imprimante,
- Modem,
- Pare-feu,
- Point d'accès wifi,
- Pont,
- Périphérique réseau,
- Routeur,
- Satellite,
- Switch.

Pour accéder à l'ensemble des périphériques réseau d'un type donné :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructure >Périphériques réseau**.
2. Dépliez le dossier qui correspond au type de périphérique qui vous intéresse.

La liste de tous les périphériques réseau du type choisi apparaît.

### Créer un périphérique réseau

Pour créer un périphérique réseau de type **Imprimante**, par exemple :

1. A partir du menu de navigation **Inventaires**, sélectionnez **Infrastructure >Périphériques réseau**.
2. Sélectionnez le dossier qui correspond au type de périphérique réseau qui vous intéresse.
3. Cliquez sur **Nouveau**.
4. Saisissez le **Nom** de votre périphérique ainsi que son **Détenteur** et cliquez sur **OK**.

Le nouveau périphérique apparaît dans la liste.

## DÉCRIRE LES COMMUNICATIONS DANS UNE INFRASTRUCTURE INFORMATIQUE

Dans une infrastructure informatique, les communications s'appuient sur :


- des points de service, des points de requête et des interactions de service, pour les communications liées aux services,
- des ports et des canaux réseau, pour les communications techniques.

---


### Décrire les communications liées aux services

#### Les interactions de service

Les *interactions de service* permettent de représenter les échanges de services prévus entre les entités organisationnelles.

 Une *interaction de service* représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

Les modalités d'échange sont définies par une *interface de service* qui est affectée à l'interaction de service.

 Une *interface de service* est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).


Vous pouvez définir des interactions de service entre :

- deux composants de type Architecture des ressources pour représenter les échanges prévus entre ces entités,
- un composant de type Architecture des ressources et une infrastructure informatique pour représenter les modalités d'utilisation de la ressource matérielle par la ressource organisationnelle. Par exemple, vous pouvez représenter que l'utilisation du matériel opérateur est utilisé sur réservation.
- deux composants de type infrastructure informatique pour représenter les modalités d'utilisation d'une ressource informatique par une autre dans le contexte de l'architecture de ressources modélisée.
- un point de service et un ou plusieurs composants de type Architecture de ressources pour représenter la mise en œuvre du service au sein de l'architecture de ressources,
- un composants de type Architecture de ressources et un point de requête pour représenter que l'entité fait appel à une ressource organisationnelle extérieure.


Pour plus de détails sur les modalités de gestion des interactions de service, voir [Gérer les interactions de service](#).

## Les points de service

Les services assurés par les éléments d'une infrastructure sont représentés par des *points de service*.

 *Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.*

Le service est demandé selon des modalités précises définies par une *interface de service* affectée au point de service.


 *Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).*

Les ressources activées pour assurer un service sont liées au point de service par des interactions de service. Si l'activation de plusieurs ressources est nécessaire, alors plusieurs interactions de service doivent être créées entre le point de service et les ressources de l'architecture.


Pour créer un point de service, voir [Décrire les points de service et de requête](#).

## Les points de requête

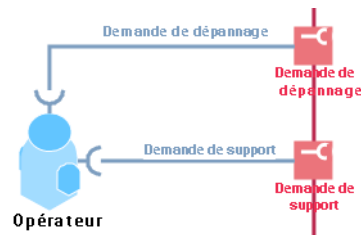
Un *point de requête* permet de représenter l'utilisation d'un service externe.

 *Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.*

Le service est demandé selon des modalités précises définies par une *interface de service* affectée au point de requête.

 *Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).*

Les ressources qui émettent une requête sont liées au point de requête par une interaction de service.




Dans l'exemple, les points de requête représentent les demandes de service effectuées entre les opérateurs du centre d'appels et les autres organisations.

La procédure de création d'un point de requête est identique à celle d'un point de service. Pour plus de détails, voir [Décrire les points de service et de requête](#).

## Décrire les communications techniques

### Les ports de communication

Les **ports de communication** sont des points physiques de communication qui peuvent être définis dans des infrastructures techniques et dans les architectures de ressources.


 Un port de communication est un point physique de communication avec une ressource. Il se conforme à un protocole de communication particulier. Un port de communication implémente des points de service et de requête.

Les **ports de communication** assurent le transfert physique des informations échangées à partir des points de service et les points de requête.

Les ports de communication se conforment à des « Protocoles de communication » particuliers. Voir [Les protocoles de communication réseau](#).


### Les canaux réseau

Les **canaux réseau** permettent de connecter les ressources matérielles entre elles, ou à des ressources organisationnelles, ou encore à des ports de communication.

 Un canal réseau permet d'établir une connexion physique entre deux ressources matérielles. Il supporte les interactions de service qui définissent les protocoles de communication entre ces ressources. Les canaux réseau connectent les ressources avec l'extérieur par le biais des ports de communication.

#### Créer un canal réseau

Pour créer un canal réseau :


1. Dans la barre d'objets du diagramme d'assemblage de ressources, cliquez sur **Canal réseau** .
2. Tirez un lien entre les deux entités en communication.  
Le canal apparaît directement dans le diagramme.

Pour définir le protocole de communication associé au canal :

1. Ouvrez la page de propriétés **Protocoles supportés** et cliquez sur **Relier**.
2. Dans la fenêtre de recherche qui apparaît, sélectionnez le protocole de communication qui vous intéresse et cliquez sur **Relier**.  
Le nom du protocole apparaît à côté du canal.

### Les protocoles de communication réseau

Un **protocole de communication** est supporté par un canal réseau.

 Un protocole de communication est un ensemble de règles normalisées pour la transmission d'informations (voix, données, images) sur un canal de communication. Les différentes couches de protocoles peuvent prendre en charge la détection et le traitement des erreurs, l'authentification des correspondants, la gestion du routage.

Par exemple, le protocole "HTTPS" est basé sur le protocole "HTTP" pour le transport, ces protocoles s'appuient sur TCP qui s'appuie sur Ethernet.

Un utilisateur peut être amené à construire sa propre couche de protocoles de communication et à les assigner aux ports et canaux réseau.

☛ *Les protocoles de communication supportés par un port de communication doivent être compatibles avec les ports de communication auxquels il est connecté.*

---

## Relier une interaction de service à un canal réseau

Pour indiquer qu'une interaction de service est supportée physiquement par un canal réseau :

1. Dans la barre d'objets du diagramme d'assemblage de ressources, cliquez sur la bobine.
2. Tirez un lien entre l'interaction de service et le canal réseau qui la supporte.  
Une ligne en pointillé apparaît dans le diagramme.

Pour accéder à la liste des interactions de service supportée par un canal réseau :

- 1. Ouvrez la page de propriétés **Interactions de service gérées** du canal réseau qui vous intéresse.  
Le nom de l'interaction de service apparaît dans la liste.

☛ *Vous pouvez utiliser le bouton **Relier** pour connecter d'autres interactions de service au canal réseau.*

# ACCÉDER À LA CONCEPTION DE LOGICIELLE



**HOPEX IT Architecture** offre des facilités pour assister les architectes dans la spécification des évolutions de leur système d'information.

☛ *Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.*

Les points spécifiques traités ici sont les suivants :

- ✓ [Modélisation UML des données](#) ;
- ✓ [Décrire les traitements par lots](#) ;
- ✓ [Décrire les interfaces utilisateur](#).

# MODÉLISATION UML DES DONNÉES

**HOPEX IT Architecture** fournit les outils nécessaires à la modélisation des données logiques via les diagrammes de classes et les modèles de données.

☛ *Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.*

A travers les concepts de *Domaine de données* et de *Vue de données*, vous pouvez détailler une structure de données logique dans un contexte d'utilisation particulier.



*Une vue de données permet de représenter le périmètre couvert par un élément de modèle de données ou de domaine de données. Une vue de données est construite à partir d'une sélection de plusieurs classes reliées dans le contexte spécifique de la vue.*

☛ *Pour plus de détails sur la création et la mise à jour d'un modèle de données, voir le guide **HOPEX Information Architecture**.*

Les différents concepts de la vue logique sont décrits dans ce paragraphe.

---

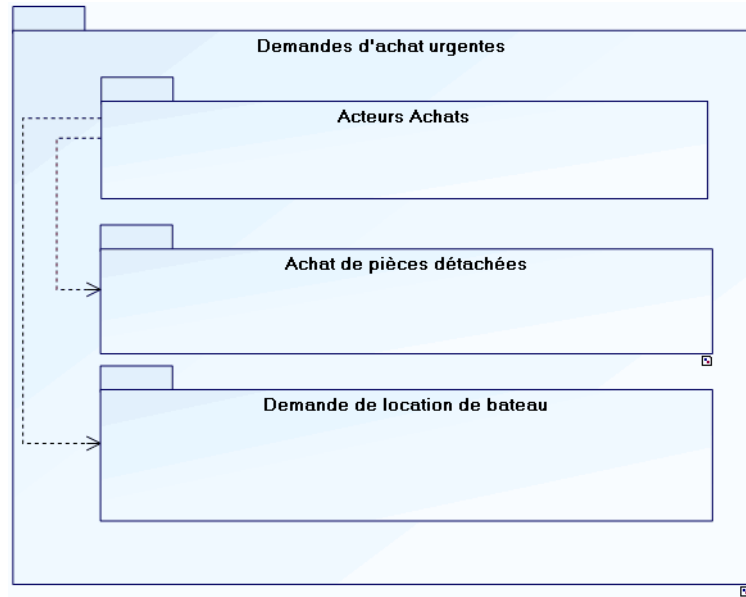
## Paquetage (UML)

Un paquetage permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

Le paquetage est un élément détenteur, il fournit un espace de nommage pour les éléments qu'il regroupe.



Le paquetage vous permet de classer les éléments référencés dans un projet. Vous pouvez créer des sous-paquetages dans un paquetage afin de classer plus finement les objets, par exemple les acteurs d'un projet.



Les demandes d'achat urgentes sont prévues pour traiter les achats de pièces détachées et les demandes de location de bateau. Dans ces deux cas, les utilisateurs sont des acteurs du domaine des achats.

La représentation graphique des éléments d'un paquetage se fait dans un diagramme de classes.

➡ Plus plus de détails sur la construction d'un diagramme de classes, voir [Le diagramme de classes](#).

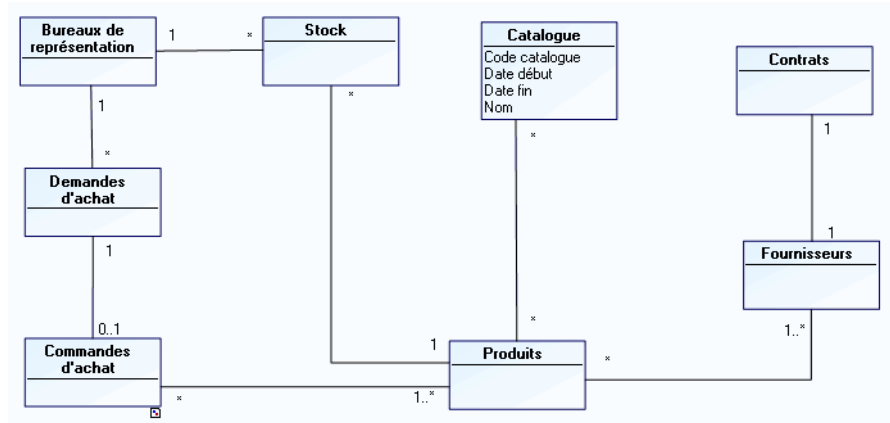
## Modèles de données

Tout comme le paquetage, le modèle de données permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et les relations qui existent entre eux.

La représentation graphique des éléments d'un modèle de données se fait dans un diagramme de données.

## Exemple

Le modèle de données du projet "Automatisation des demandes d'achat" est présenté ci-dessous.



L'application gère les demandes d'achat, les commandes et les stocks de produits dans chacun des bureaux de représentation.

Un catalogue centralisé des produits et des fournisseurs est mis en place.

Les contrats avec les fournisseurs référencés sont également accessibles depuis l'application.

---

## Les domaines de données

Un domaine de données représente une structure de données restreinte, dédiée à la description d'un stockage de données logicielles (voir [Gérer les données](#) ). Il est constitué de classes et/ou de vue de classes et peut être décrit par un diagramme de domaine de données.

Un domaine de données logique permet de définir une structure de données logiques constituée de classes et de vue de classes.

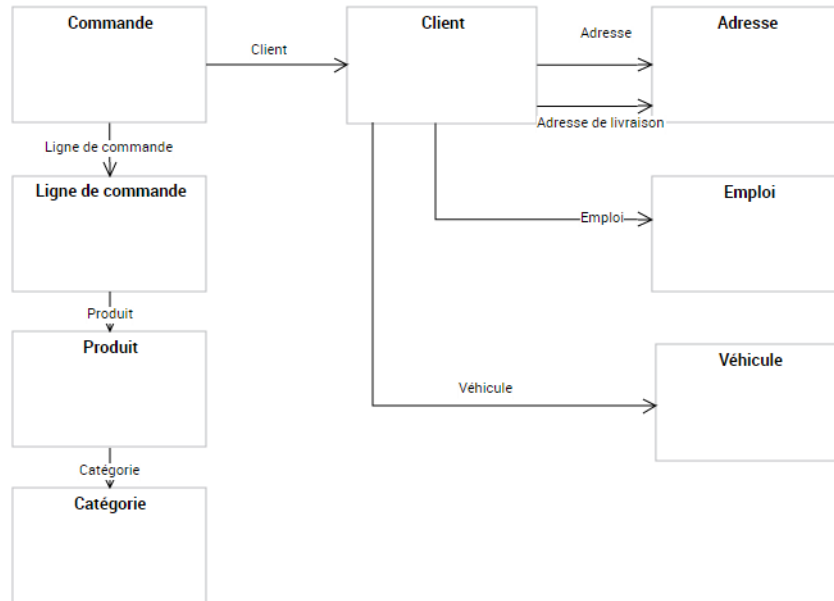
Un domaine de données logique est détenu par un paquetage, il peut référencer des objets détenus dans d'autres paquetages.

Il est possible de définir le mode d'accès (création, suppression, etc.) aux objets référencés par un domaine de données en les intégrant comme composants du domaine de données.


Lors d'une intégration avec HOPEX Database Builder, une structure physique correspondante peut être définie via un domaine de données physique. Celui-ci est constitué de tables et de vues de tables.

## Exemple

Le diagramme de domaine de données suivant représente une structure de données relative aux Commandes ; il décrit sous forme de tout/partie les classes et leurs relations.



Pour répondre à des cas d'utilisation précis, vous pouvez créer des **Vues de données** dans lesquelles vous pouvez visualiser et modifier le périmètre couvert par des classes.

 Une vue de données permet de représenter le périmètre couvert par un élément de modèle de données ou de domaine de données. Une vue de données est construite à partir d'une sélection de plusieurs classes reliées dans le contexte spécifique de la vue.

# DÉCRIRE LES TRAITEMENTS PAR LOTS

**HOPEX IT Architecture** vous permet de décrire l'enchaînement des traitements automatisés dans des **diagrammes de structure de chaîne de traitement par lots**.

☛ Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.

☛ Pour visualiser les **traitements par lot**, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Fonctionnalité de modélisation d'IHM et de traitements par lots (ADES)** est activée.

Ce type de diagramme permet de représenter le planning d'exécution des batch, les programmes de batch et leur organisation.

---

## Définition d'un traitement par lots

Un traitement par lots est un ensemble de traitements informatiques exécutés par un ordinateur sans intervention humaine, généralement la nuit ou le week-end.

Un **Traitement par lots** est décrit par un **Planning d'exploitation** ou par un **Programme**.

📖 Un **planning d'exploitation** définit l'ensemble des traitements informatiques à effectuer sur une ou plusieurs machines pendant une période de temps donnée.

📖 Un **programme** est une étape élémentaire d'exécution d'un traitement par lots qui consiste à lancer l'exécution d'un programme avec les paramètres appropriés.

Un **Planning d'exploitation** est un ensemble de **Traitements de batch**. Chacun d'eux est associé soit à un **Programme** soit à un autre **Planning d'exploitation**. Un **Planning d'exploitation** est décrit par un **diagramme de structure de chaîne de traitement de lot**.

☛ Pour plus de détails voir [Construire un diagramme de structure de chaîne de traitement de lot](#).

Un **Programme** est un ensemble de **Traitements de batch**. Chacun d'eux ne peut être associé qu'à un **Programme**. Un **Programme** est un décrit par un **diagramme de structure de programme batch**.

☛ Pour plus de détails voir [Créer un diagramme de structure de programme batch](#).

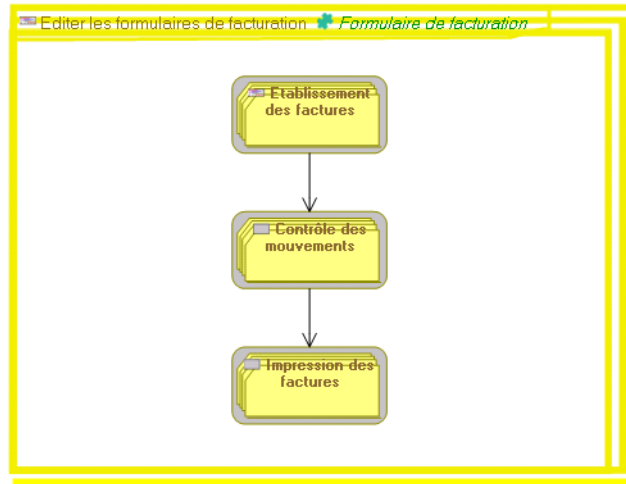
---

## Construire un diagramme de structure de chaîne de traitement de lot

📖 Un **planning d'exploitation** définit l'ensemble des traitements informatiques à effectuer sur une ou plusieurs machines pendant une période de temps donnée.

## Créer un diagramme de structure de chaîne de traitement de lot

L'enchaînement de traitements automatisés peut être décrit dans un **diagramme de structure de chaîne de traitement de lot**.



Pour créer un diagramme de structure de chaîne de traitement de lot :

1. Cliquez sur le menu de navigation **Conception (UML)**.
2. Sélectionnez **Mise en œuvre de traitements par lots et de programmes > Batch et Programme**.  
La liste des plannings d'exploitation apparaît.
3. Ouvrez la page de propriétés **Diagrammes** du planning d'exploitation qui vous intéresse et sélectionnez **Créer Diagramme**.
4. Dans la fenêtre de choix, sélectionnez **Diagramme de structure de chaîne de traitement de lot**.  
Le diagramme s'ouvre.

## Ajouter un appel de traitement de batch dans le diagramme

Les composants d'un **Planning d'exploitation** sont définis à partir des **Appels de traitement de batch** qui sont positionnés dans le diagramme. Il peut s'agir de planning d'exploitation ou de programmes.

Pour ajouter un composant de type Planning d'exploitation dans le diagramme de structure de chaîne de traitement de lot :

1. Sélectionnez le bouton **Appel de traitement de batch** et cliquez dans le diagramme.  
La fenêtre **d'ajout d'un appel de traitement de batch** s'ouvre.
2. Cliquez sur la flèche à droite du champ **Type d'objet** et sélectionnez **Planning d'exploitation** dans la liste déroulante.
3. Cliquez sur la flèche à droite du champ **Nom court** et sélectionnez le planning d'exploitation qui vous intéresse dans la liste déroulante.
4. Cliquez sur **OK**.  
L'appel de traitement batch apparaît dans le diagramme avec l'icône d'un planning d'exploitation.


## Définir les séquences des batches

Pour préciser l'ordre d'exécution des traitements :

1. Cliquez sur le bouton **Séquence batch**.
2. Cliquez sur l'appel de traitement batch initial et, en maintenant enfoncé le bouton gauche de la souris, tirez un lien vers l'appel de traitement batch suivant.
3. Lâchez le bouton de la souris.  
Le lien représentant l'ordonnancement des traitements apparaît dans le diagramme.

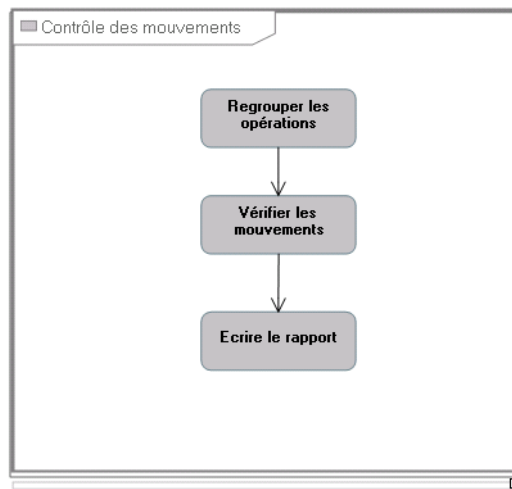
---

## Créer un diagramme de structure de programme batch

 *Un programme est une étape élémentaire d'exécution d'un traitement par lots qui consiste à lancer l'exécution d'un programme avec les paramètres appropriés.*

### Créer un diagramme de structure de programme batch

L'enchaînement de traitements d'un programme peut être décrit dans un **diagramme de structure de programme batch**.



Pour créer le diagramme de structure de programme batch :

1. Ouvrez la page de propriétés **Diagrammes** du programme qui vous intéresse et cliquez sur **Créer un diagramme**.
2. Dans la fenêtre de choix, sélectionnez **Diagramme de structure de programme batch**.  
Le diagramme s'ouvre.

### Ajouter un appel de programme dans le diagramme

Les composants d'un **Programme** sont définis à partir des **Appels de programme** qui sont positionnés dans le diagramme.

Pour ajouter un composant dans le diagramme :

1. Sélectionnez le bouton **Appel de programme** et cliquez dans le diagramme.  
La fenêtre **d'ajout d'un appel de programme** s'ouvre.
2. Cliquez sur la flèche à droite du champ **Nom** et sélectionnez le **Programme** qui vous intéresse.
3. Cliquez sur **OK**.  
L'appel de programme apparaît dans le diagramme.

L'ordre d'exécution des programmes est défini par des **séquences de batch**, voir [Définir les séquences des batchs](#).

---

## Utiliser les réalisations batch de processus applicatif

Un mécanisme de réalisation permet de spécifier qu'un processus applicatif décrit le traitement mise en œuvre par un **Planning d'exploitation** ou un **Programme**.

Pour décrire qu'un planning d'exploitation est associé à un processus applicatif :

1. Ouvrez la page de propriétés **Caractéristiques > Réalisation** du planning d'exploitation qui vous intéresse.
2. Cliquez avec le bouton **Nouveau**.  
La fenêtre de création d'une réalisation apparaît.
3. Dans le champs **Type d'objet**, sélectionnez **Réalisation Batch de processus applicatif** et cliquez sur **Suivant**.
4. Sélectionnez le processus applicatif qui vous intéresse et cliquez sur **Nouveau**.  
La réalisation batch de processus système apparaît dans la page de propriétés du planning d'exploitation.

# DÉCRIRE LES INTERFACES UTILISATEUR

Il est possible de décrire les interfaces reliant les services ou les opérations avec l'extérieur. Cette description s'effectue dans un diagramme d'IHM.

☛ Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.

☛ Pour visualiser les **interfaces utilisateur**, ouvrez la fenêtre des **Options** et vérifiez que **IT Architecture > Fonctionnalité de modélisation d'IHM et de traitements par lots (ADES)** est activée.

---

## Créer une interface homme-machine

Pour créer une interface utilisateur à partir du menu de navigation **Conception (UML)** :

1. Sélectionnez **Interfaces utilisateur**.  
La liste des interfaces utilisateur apparaît.
2. Cliquez sur **Nouveau**.
3. Saisissez le nom de l'interface.
4. Cliquez sur **OK**.

---

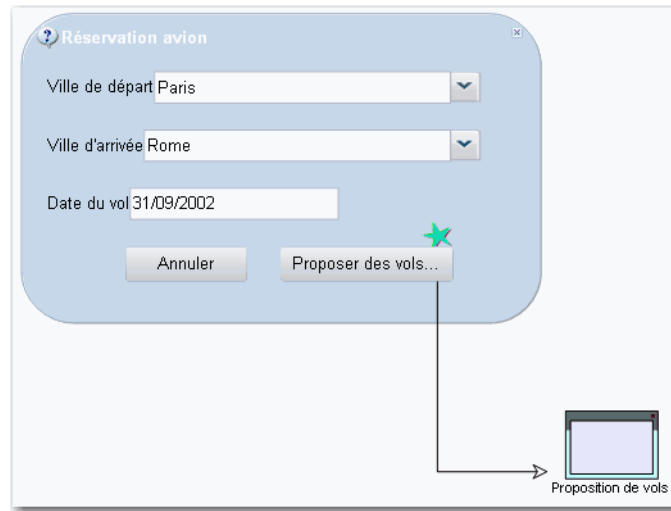
## Construire un diagramme d'Interface Homme Machine

Pour créer un diagramme d'IHM :

1. Sélectionnez l'interface utilisateur qui vous intéresse et cliquez sur le bouton **Créer un diagramme**.
2. Dans la fenêtre de choix, sélectionnez **Diagramme d'IHM**.  
Le diagramme d'IHM s'ouvre dans la fenêtre d'édition.



Prenons l'exemple du diagramme d'IHM du service "Réservation Avion".



L'interface se présente sous forme d'une boîte de dialogue, dans laquelle doivent être renseignés différents champs :

- La ville de départ
- La ville d'arrivée
- La date du vol

Un bouton annule la demande, un autre ouvre une seconde interface.

---

## Dessiner l'IHM

Le diagramme d'IHM vous permet de dessiner l'interface de l'opération ou du service.

### Élément D'IHM

Un certain nombre de boutons vous permettent de définir l'apparence de l'interface :

- Champ de texte
- Liste
- Bouton radio
- Case à cocher
- Etc.

Pour créer un élément :

1. Dans la barre d'objets du diagramme, sélectionnez le bouton correspondant à l'élément voulu puis cliquez dans le diagramme.
2. Dans la fenêtre qui apparaît, tapez le nom de l'élément.
3. Cliquez sur **OK**.

Vous pouvez également cliquer sur le bouton **Élément d'IHM** et indiquer le type de l'élément dans sa fenêtre de propriétés.

Caractéristiques

Nom Local  
Élément d'IHM

Défendeur  
▼

#Tags  
▼

Type d'élément d'IHM  
▼

- ☐ Fenêtre
- ☐ Formulaire
- ☐ Champ éditable
- ☐ Dialogue
- ☐ Champ de texte
- ☐ Champ non éditable

## Événement d'IHM

Vous pouvez relier un *événement* à un élément d'IHM. Dans notre exemple, le bouton « Proposer des vols » est relié à un événement, qui, lorsqu'il est actionné, ouvre une autre interface.

Pour créer un événement :

1. Cliquez sur le bouton **Événement d'IHM** puis cliquez dans le diagramme.
2. Saisissez le nom de l'événement et cliquez sur **OK**.

### Type d'événement

Il existe différents types d'événement. Il peut s'agir :

- D'un clic sur un bouton
- D'une entrée dans un champ
- Etc.

Pour spécifier le type d'un événement :

1. Ouvrez la page de propriétés **Caractéristiques** de l'événement.

2. Dans le champ **Événement type d'IHM**, cliquez sur la flèche puis sur **Rechercher Événement type d'IHM**.  
La fenêtre de sélection apparaît.
3. Cliquez sur **Chercher**.  
La liste des types d'événement apparaît.
4. Sélectionnez le type voulu et cliquez sur **OK**.

### ***Relier l'événement à un élément***

Pour relier l'événement d'IHM à un élément d'IHM, deux possibilités s'offrent à vous :

- Sélectionnez l'événement dans le diagramme et faites-le glisser sur l'élément.
- Ou ouvrez la page de propriétés **Caractéristiques** de l'événement et renseignez le champ **Élément d'IHM**.



# DÉCRIRE LES ÉCHANGES D'INFORMATIONS




Ce chapitre présente comment décrire les interfaces de service entre les composants d'une architecture métier ou informatique.

Pour simplifier la création d'interfaces de service, des modèles d'opération de service et des modèles d'interface de service peuvent être utilisés.


- ✓ Gérer les interactions de service ;
- ✓ Décrire une interface de service ;
- ✓ Décrire les opérations de service ;
- ✓ Utiliser un modèle d'interface de service.


## GÉRER LES INTERACTIONS DE SERVICE

Une *interaction de service* représente l'échange d'informations entre des composants d'architecture.

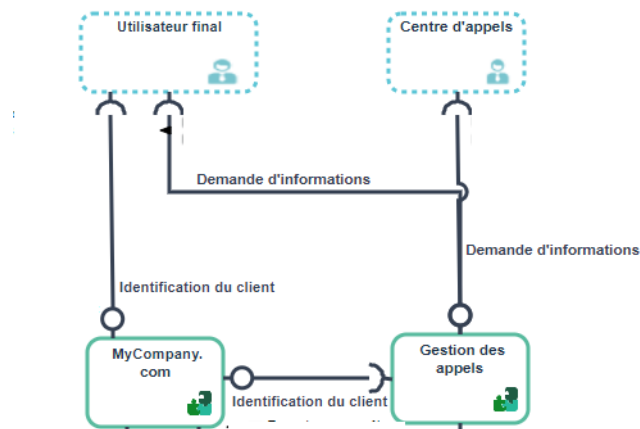
 Une *interaction de service* représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.

Le contenu d'une interaction de service est décrit par une *interface de service*.

 Une *interface de service* est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

 Pour plus de détails sur les notions d'interface de service, voir [Décrire une interface de service](#).

Dans un diagramme de structure de système applicatif de "Traitement des demandes d'achat", deux interfaces de service sont utilisées par les différentes interactions de service.





*Interactions de service dans le diagramme de structure de système applicatif*

Les clients doivent être identifiés avant de passer une commande. Ils peuvent passer des commandes directement à partir d'une application "MyCompagny.com" ou en passant par un Centre d'appels. Le centre d'appels passe par l'application de "Gestion des appels" qui utilise le service d'identification du client proposé par l'application "MyCompagny.com".

---

## Créer une interaction de service


Pour créer une interaction de service :


1. Dans la barre d'objets d'un diagramme, cliquez sur le bouton  
**Interaction de service** 
2. Cliquez sur l'entité qui demande le service et tirez un lien l'entité qui fourni le service.
3. Dans la fenêtre d'ajout d'interaction de service, sélectionnez l'interface de service que vous souhaitez utiliser.  
 Vous pouvez également créer une nouvelle interface de service. Pour plus de détails, voir [Créer une interface de service](#).
4. Cliquez sur **Ajouter**.

---

## Décrire les points de service et de requête


Dans une architecture orientée service, les communications s'appuient sur des points d'accès : *points de service* et des *points de requête*.

 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.


 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.


### Les points de service

Un système applicatif, par exemple, est créé pour assurer un ou plusieurs services. Ces services sont représentés par des *points de service*.

 Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.

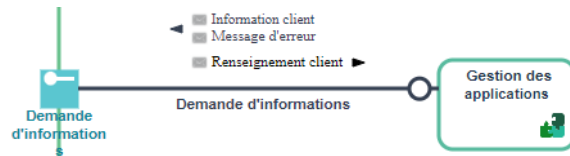
Le service est demandé selon des modalités précises définies par une *interface de service* affectée au point de service.

 Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

 Pour plus de détails sur les interfaces de service, voir [Décrire une interface de service](#).

Les composants activés pour assurer un service sont liés au point de service par des interactions de service. S'il est nécessaire d'activer plusieurs composants, vous

devez créer plusieurs interactions de service entre le point de service et les composants du système.




Dans l'exemple présenté ici, le service applicatif de "Gestion des clients" est activé par à l'interaction de service "Demande d'information".

✎ Pour créer un point de service, voir [Créer un point de service ou un point de requête](#).

✎ La page de propriétés **Mises en œuvre publiées** du point de service permet d'accéder aux capacités mises en œuvre par le point de service. qui vous intéresse. Pour créer un point de service, voir [Accéder aux mises en œuvre à partir d'un point de service](#).

## Les points de requête

Un **point de requête**  permet de représenter l'utilisation d'un service externe à l'entité décrite.

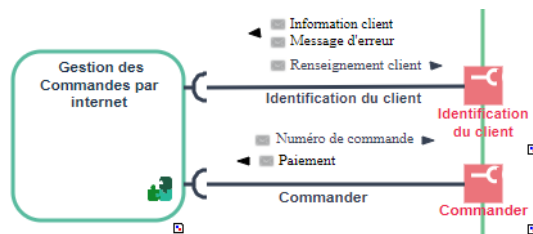
📖 Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.

Le service est demandé selon des modalités précises définies par une **Interface de service** affectée au point de requête.

📖 Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

✎ Pour plus de détails sur les interfaces de service, voir [Décrire une interface de service](#).

Les composants qui émettent une requête sont liés au point de requête par une interaction de service.




Dans l'exemple, les points de requête représentent les demandes de service effectuées par le service applicatif "Gestion des commandes par internet" pour identifier le client et passer une commande.


✎ Pour créer un point de requête, voir [Créer un point de service ou un point de requête](#).





## Créer un point de service ou un point de requête

Le processus de création d'un *point de service* ou d'un *point de requête* est identique.

 *Un point de requête est un point d'échange par lequel un agent requiert un service à des fournisseurs potentiels.*

 *Un point de service est un point d'échange par lequel un agent offre un service à des clients potentiels.*

Pour créer un point de service :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Point de service** .
2. Positionnez l'objet sur le bord du cadre de l'objet décrit.  
Une fenêtre de création s'ouvre.
3. Cliquez sur la flèche à droite du champ **Interface de service** pour définir l'interface de service qui permet l'activation de ce point de service et sélectionnez, par exemple, **Relier Interface de service**.  
La fenêtre de recherche s'ouvre.
4. Sélectionnez l'interface de service qui est associée à ce point de service et cliquez sur **Relier**.
5. Cliquez sur **Suivant**.  
Une nouvelle fenêtre s'ouvre pour vous proposer la liste des rôles de l'interface de service pouvant être associés au point de service.  
 *Cette fenêtre n'est pas proposée s'il n'y a qu'un seul rôle candidat à être associé au point de service.*
6. Sélectionnez le rôle qui vous intéresse et cliquez sur **OK**.  
Le point de service apparaît dans le diagramme.

## DÉCRIRE UNE INTERFACE DE SERVICE

Une *interaction de service* représente l'échange d'informations entre les composants de l'architecture.



*Une interaction de service représente un contrat conclu dans un contexte précis entre des entités autonomes à l'intérieur ou à l'extérieur d'une entreprise. Ces entités peuvent être des acteurs, des applications, des activités, des processus de l'entreprise, ou des acteurs externes à l'entreprise. Le contenu de ce contrat est décrit par une interface de service.*

Le contenu d'une interaction de service est décrit par une *interface de service*.



*Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).*

Une interface de service est décrite comme un enchaînement d'opérations qui sont représentées :

- soit par des *interfaces de service utilisées*



*Une interface de service utilisée est associée à une interface de service. Elle permet de représenter des échanges complexes.*

- soit par des *opérations de service utilisées*



*Une opération de service utilisée représente l'utilisation d'une opération de service dans une interface de service.*

➡ Pour plus de détails sur les opérations de service, voir [Décrire les opérations de service](#).

---

## Exemples de diagrammes d'interface de service (BPMN)

Une interface de service est décrite par un enchaînement d'étapes qui sont représentées soit :

- par des *opérations de service utilisées*
- par des *interfaces de service utilisées*



*Une interface de service utilisée est associée à une interface de service. Elle permet de représenter des échanges complexes.*

Les rôles de l'interface de service, présentés à la frontière du cadre, représentent les intervenants :

- client/fournisseur, ou
- émetteur/récepteur

Il est possible de décrire une interface de service faisant intervenir plus de deux intervenants. Dans ce cas, un rôle est initiateur de l'interface de service et les autres sont des contributeurs.

## Exemple de diagramme d'interface de service (BPMN)

Le diagramme d'interface de service associé à l'interface de service "Identification du client" décrit, dans un formalisme BPMN, les opérations réalisées.

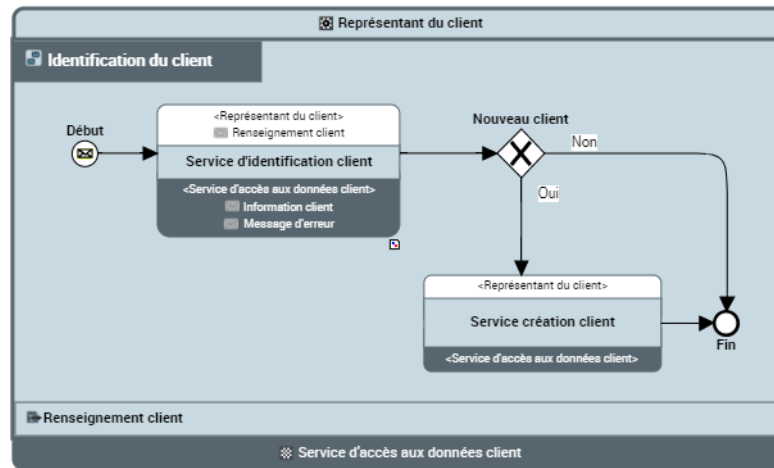


Diagramme d'interface de service (BPMN) "Identification du client"

Le déroulement de l'identification d'un client commence par une étape de recherche du client. Si le client est trouvé, l'interface de service retourne les renseignements client, sinon une interface de service de "Création de client" est activée.

## Exemple d'interface de service de communication avancé

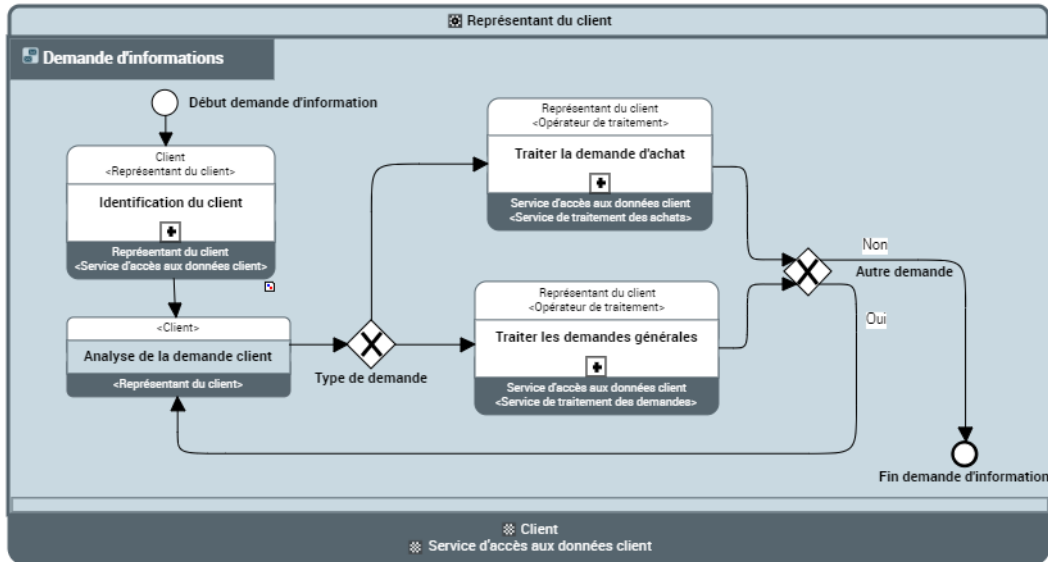


Diagramme d'interface de service (BPMN) "Demande d'informations"

L'interface de service de "Demande d'information" est utilisée par le centre d'appels pour prendre en compte la demande d'un client en ligne. Il existe donc trois intervenants dans cette interface de service : le client, les applications IT et le représentant du client qui est le demandeur effectif du service (il s'agit ici du centre d'appel).

Cette interface de service consiste à identifier le client, puis à analyser sa demande. La demande est ensuite traitée comme demande d'achat ou comme une autre demande s'il s'agit d'une demande de renseignements, par exemple.

☛ La page de propriétés **Rôles** donne la liste des rôles contributeur d'une interface de service ainsi que le rôle initiateur.

## Accéder à la liste des interfaces de service

Pour accéder à la liste des interfaces de service d'une bibliothèque :

1. A partir du menu de navigation **Environnement**, ouvrez la zone d'exploration **Conteneurs > Bibliothèques**.
2. Dépliez la bibliothèque qui vous intéresse, puis le dossier **Interface de service**.

La liste des interfaces de service de la bibliothèque apparaît.

---

## Créer une interface de service

Vous pouvez créer une interface de service :


- à partir d'une bibliothèque,
- à partir d'un diagramme en utilisant une interaction de service, par exemple.

Quelque soit le point d'origine, vous pouvez créer une interface de service en mode standard ou en utilisant un *modèle d'interface de service*.

➡ Pour plus de détails sur les modèles d'interface de service, voir [Utiliser un modèle d'interface de service](#).

### Créer une interface de service en mode standard à partir d'un diagramme

Pour créer une interface de service en mode standard, dans un diagramme, à partir d'une interaction de service :

1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Interaction de service** .
2. Tirez un lien entre les deux entités en communication.
3. Dans la fenêtre d'ajout d'interaction de service, cliquez sur la flèche à droite du champ **Interface de service** et sélectionnez **Créer une interface de service**.  
La fenêtre **Création d'une interface de service** s'ouvre.
4. Sélectionnez le **Mode de création** : **Création standard**.

➡ Pour plus de détails sur l'utilisation d'un modèle d'interface de service, voir [Créer une interface de service à partir d'un modèle d'interface de service](#).

5. Saisissez le nom de l'interface de service dans le champ **Nom**.
6. Cliquez sur le bouton **OK**.
7. Dans la fenêtre de création de l'interaction de service, spécifiez le nom de l'interaction de service à partir du nom de l'interface de service et cliquez sur **Ajouter**.  
L'interaction de service et l'interface de service sont créés.

---

## Construire un diagramme d'interface de service (BPMN)

### Créer un diagramme d'interface de service (BPMN)

Une interface de service est représentée par un **Diagramme d'interface de service (BPMN)**.

Pour créer un diagramme d'interface de service (BPMN) à partir d'une interaction de service :

1. A partir du menu de navigation **Environnement**, ouvrez la zone d'exploration **Conteneurs > Bibliothèques**.
2. Dépliez la bibliothèque qui vous intéresse, puis le dossier **Interface de service**.

La liste des interfaces de service de la bibliothèque apparaît.

3. Sélectionnez l'interface de service associé et, dans son menu contextuel, cliquez sur **Créer Diagramme**.
4. Dans la fenêtre de choix, sélectionnez **Diagramme d'interface de service (BPMN)**.

Le diagramme s'ouvre avec le cadre de l'interface de service et les deux **rôles** qui représentent le consommateur et le fournisseur.



*Un rôle est un participant à une interaction de service, un workflow ou un processus. Il peut en être l'initiateur, c'est-à-dire le demandeur du service, ou représenter un sous-traitant réalisant un traitement extérieur au service. Un rôle fait partie intégrante de l'objet qu'il décrit et n'est pas réutilisable. Il peut par la suite être affecté à un acteur interne ou externe à l'organisation ou à un composant informatique. Exemples : Demandeur, voyageur.*

Les **événements**, les **branchements** et les **enchaînements** de votre diagramme respectent les standards de la norme BPMN.



*Pour plus de détails sur les événements, branchements et enchaînements, voir [Gérer les événements, les branchements et les enchaînements](#).*

## Définir un opération de service ou une interface de service utilisée

Dans un diagramme d'interface de service (BPMN), les opérations sont décrites par :

- des **interfaces de service utilisées**
- des **opérations de service utilisées**



*Une interface de service utilisée est associée à une interface de service. Elle permet de représenter des échanges complexes.*



*Une opération de service utilisée représente l'utilisation d'une opération de service dans une interface de service.*

Pour créer une **interface de service utilisée** :

1. Sélectionnez le bouton **Interface de service utilisée**  et cliquez dans le diagramme à l'intérieur de la forme de l'interface de service. L'interface de service apparaît dans le diagramme.
2. Ouvrez la page de propriétés **Caractéristiques** de la nouvelle interface de service.
3. Cliquez sur la flèche à droite du champ **Spécification de l'interface de service utilisée**.
4. Sélectionnez **Relier Interface de service** dans la liste déroulante et choisissez l'interface de service que vous souhaitez utiliser.



*La page de propriétés **Opérations de service** donne la liste des composants d'une interface de service.*

## DÉCRIRE LES OPÉRATIONS DE SERVICE

Le contenu d'une interaction de service est décrit par une *interface de service*.

Une interface de service est un modèle de contrat entre des entités (organisationnelles, IT ...). Ce contrat est décrit par les opérations déclenchables au travers de messages échangés entre deux rôles (par exemple acheteur et vendeur).

Pour plus de détails sur les interfaces de service, voir [Décrire une interface de service](#).

Une interface de service peut être décrite par un enchaînement d'opérations de services ou d'interfaces de service.

Une opération de service décrit les échanges de flux entre les participants.

Un diagramme d'opération de service décrit l'enchaînement des flux d'une *opération de service*.

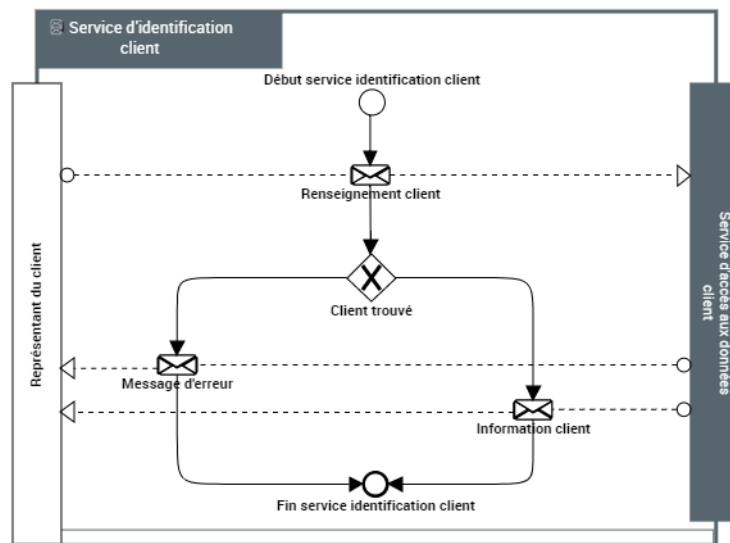


Diagramme d'opération de service de "Service d'identification client"

Le déroulement du service d'identification de client commence par l'envoi d'information permettant d'identifier le client. Un message d'erreur est émis si le client n'a pas été trouvé, sinon les renseignements client sont envoyés (identification client, état des commandes, etc.).

---

## Accéder à la liste des opérations de service

Pour accéder à la liste des opérations de service d'une bibliothèque :


1. A partir du menu de navigation **Environnement**, ouvrez la zone d'exploration **Conteneurs > Bibliothèques**.
2. Dépliez la bibliothèque qui vous intéresse, puis le dossier **Opération de service**.

La liste des opérations de service de la bibliothèque apparaît.


---

## Créer une opération de service

Vous pouvez créer une *opération de service* à partir d'un diagramme d'interface de service (BPMN).

 Pour plus de détails sur les modèles d'opération de service, voir [Utiliser un modèle d'interface de service](#).

Pour créer une *opération de service* à partir d'un diagramme d'interface de service (BPMN) :

1. Sélectionnez le bouton **Opération de service utilisée**  et cliquez dans le diagramme à l'intérieur de la forme de l'interface de service décrite.



Une opération de service utilisée représente l'utilisation d'une opération de service dans une interface de service.

L'opération de service apparaît dans le diagramme.

2. Ouvrez la page de propriétés **Caractéristiques** de la nouvelle opération de service.
3. Cliquez sur la flèche à droite du champ **Spécification de l'opération de service** et sélectionnez **Créer une Opération de service** dans la liste déroulante.

La fenêtre de création d'une opération de service s'ouvre.

4. Saisissez le **Nom** de votre opération de service et cliquez sur **OK**. L'opération de service est automatiquement créée.

---

## Décrire une opération de service

### Créer un diagramme d'opération de service (BPMN)

Une *opération de service* est décrit par un diagramme d'opération de service qui présente l'enchaînement des messages échangés.

Pour créer un diagramme d'opération de service :


1. A partir du menu de navigation **Environnement**, ouvrez la zone d'exploration **Conteneurs > Bibliothèques**.
2. Dépliez la bibliothèque qui vous intéresse, puis le dossier **Opération de service**.




3. Sélectionnez l'opération de service qui vous intéresse et cliquez sur **Créer Diagramme**.
4. Dans la fenêtre de choix, sélectionnez **Diagramme d'opération de service (BPMN)**.  
Le diagramme s'ouvre. Le cadre de l'opération de service est positionné et les deux rôles (Consommateur et Fournisseur) sont créés.

## Créer un flux avec contenu

Vous devez préciser les *flux* et leur *contenu* qui sont échangés entre les deux rôles de l'opération de service.

 Un flux représente la circulation d'information à l'intérieur d'une interface de service. Un flux transporte son contenu.


 Le contenu désigne le contenu d'un message ou d'un événement indépendamment de sa structure. Cette dernière peut être représentée par un schéma XML relié au contenu. Un contenu peut être utilisé par plusieurs messages puisqu'il n'est pas associé à un émetteur et à un destinataire. Il ne peut y avoir qu'un contenu par message ou par événement, mais un même contenu peut être utilisé par plusieurs messages ou événements.

Pour créer le flux et son contenu :


1. Dans le diagramme d'opération de service, cliquez sur le bouton **Flux avec contenu**.
2. Cliquez sur le rôle qui représente l'émetteur du flux et tirez un trait jusqu'au rôle qui représente le récepteur du flux en maintenant le bouton de la souris enfoncé.  
La fenêtre de **Création de Flux** s'ouvre.
3. Sélectionnez le contenu que vous souhaitez associer au flux à partir de la liste déroulante du champ **Contenu**.  
Le flux s'affiche avec son contenu dans le diagramme.

## Gérer les événements, les branchements et les enchaînements


Les *événements* "Début" et "Fin" sont nécessaires à la description du service assuré par l'interface de service.


 Un événement représente un fait ou une action se produisant dans le système, par exemple - modification de l'adresse client. Il est géré par un broker. Pour signaler qu'elle peut produire un événement, une application déclare qu'elle le publie. Si elle est intéressée par un événement, une application déclare qu'elle y souscrit.

Conformément à la norme BPMN, dans la barre d'objets, plusieurs types de *branchement* vous sont proposés.

 Un branchement est un élément de modélisation utilisé pour contrôler des enchaînements qui convergent ou qui divergent à l'intérieur d'un processus.

Un *enchaînement* est un lien orienté qui représente l'organisation chronologique des différentes étapes de traitement.

 Un enchaînement est utilisé pour montrer l'ordre d'exécution des étapes d'une interface de service. Un enchaînement n'a qu'une source et une cible.

 Pour plus de détails sur les événements, branchements et enchaînements, voir [Gérer les événements, les branchements et les enchaînements](#).

# UTILISER UN MODÈLE D'INTERFACE DE SERVICE

Les *modèles d'interface de service*, les *modèles d'opération de service* et les *modèles de contenu* facilitent la création de interfaces de service en dupliquant les composants du modèle utilisé.

La nouvelle interface de service peut ensuite être mise à jour et complétée si nécessaire.

## Présentation des modèles d'interface de service livrés en standard

Des modèles d'interface de service sont livrés en standard pour simplifier la création de vos interfaces de service. Ces modèles d'interface de service s'appuient sur des modèles d'opération de service.

Des exemples de *modèles d'interface de service* sont livrés avec la solution.

### Le modèle d'interface de service "Communication simple"

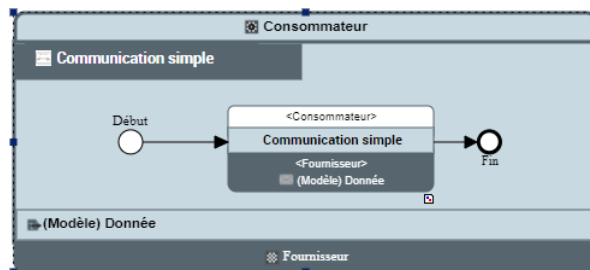


Diagramme du modèle d'interface de service (BPMN) "Communication simple"

Cette *interface de service* s'appuie sur une *opération de service utilisée* notée "Communication simple" entre le Consommateur et le Fournisseur.

 Une opération de service utilisée représente l'utilisation d'une opération de service dans une interface de service.

L'*opération de service utilisée* représente le contenu "(Modèle) Donnée" échangé entre le consommateur et le Fournisseur.

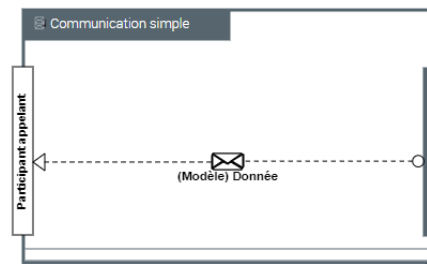


Diagramme du modèle d'opération de service (BPMN) "Communication simple"

## Le modèle d'interface de service "Question - Réponse"

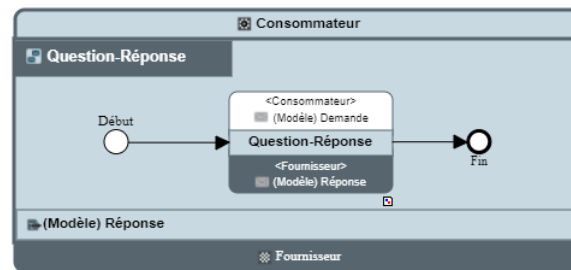



Diagramme du modèle d'interface de service (BPMN) "Question - Réponse"

Cette *interface de service* s'appuie sur une *opération de service utilisée* notée "Question-Réponse" entre le consommateur et le fournisseur.

 Une opération de service utilisée représente l'utilisation d'une opération de service dans une interface de service.

L'*opération de service utilisée* représente les opérations de services des contenus "(Modèle) Demande" et "(Modèle) Réponse" échangés entre le consommateur et le fournisseur.

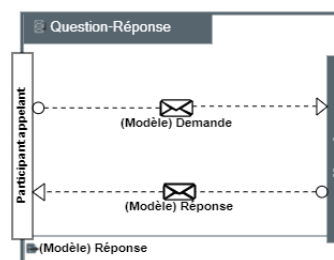


Diagramme du modèle d'opération de service (BPMN) "Question-Réponse"

Cette opération de service représente un envoi de contenu en demande et l'envoi du contenu de réponse.

## Le modèle d'interface de service "Souscription - Publication"

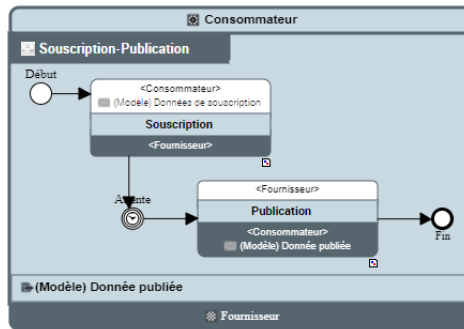


Diagramme d'interface de service (BPMN) "Souscription - Publication"

Cette *interface de service* s'appuie sur deux *opérations de service utilisées* notées "Souscription" et "Publier" entre le consommateur et le fournisseur. La demande de souscription est envoyée. Un événement représente le délai d'attente avant l'acceptation pour publication.

---

## Accéder à la liste des modèles d'interface de service

Pour accéder à la liste des *modèles d'interface de service* de votre référentiel :

- 1. A partir du menu de navigation **Administration**, sélectionnez **Modèles > Interfaces de service**.

La liste des modèles d'interface de service apparaît.

De même pour accéder à la liste des *modèles d'opération de service* :

- 1. A partir du menu de navigation **Administration**, sélectionnez **Modèles > Opérations de service**.

La liste des modèles d'opération de service apparaît.

Enfin pour accéder à la liste des *modèles de contenu* :

- 1. A partir du menu de navigation **Administration**, sélectionnez **Modèles > Contenus**.

La liste des modèles de contenu apparaît.

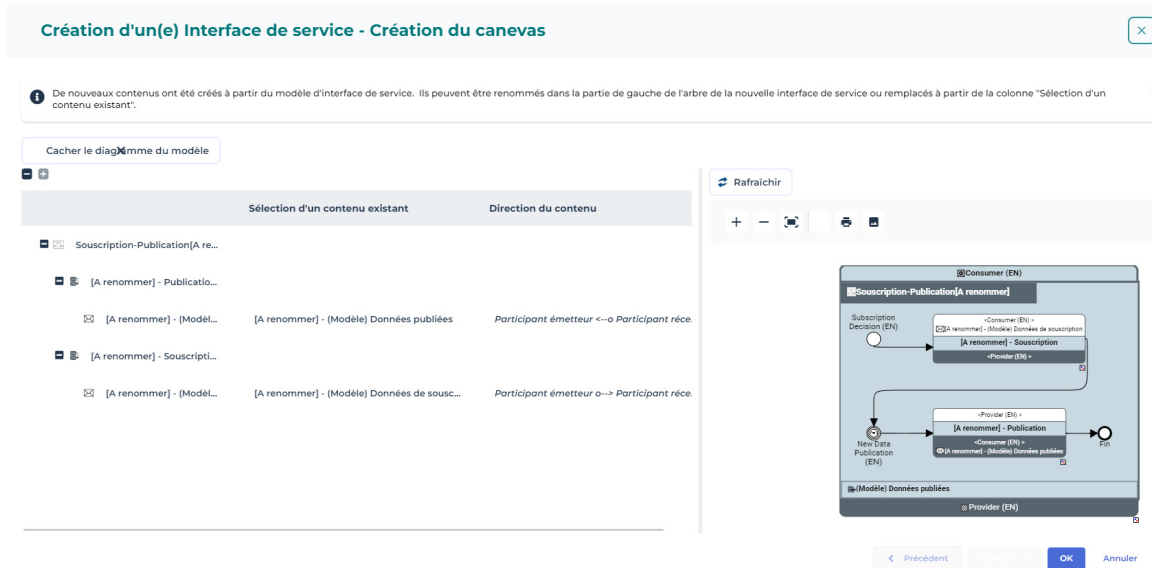
---

## Créer une interface de service à partir d'un modèle d'interface de service

Pour créer une interface de service à partir d'une liste en utilisant un modèle d'interface de service :

1. A partir du menu de navigation **Environnement**, ouvrez la zone d'exploration **Conteneurs > Bibliothèques**.

2. A partir de la bibliothèque qui vous intéresse créez une nouvelle interface de service.
3. Dans la fenêtre de création d'une interface de service, sélectionnez le **Mode de création : Création à partir du modèle**.
4. Sélectionnez le modèle qui vous intéresse et cliquez sur **Suivant**. Une fenêtre vous présente la liste des composants de la nouvelle interface de service.



Le nom des composants dupliqués est précédé de la chaîne "[A renommer]". Si des modèles de contenus sont utilisés, ils sont dupliqués.

5. Double-cliquez sur le nom des éléments que vous souhaitez renommer.
6. (Optionnel) Dans la colonne **Sélection de contenu existant**, sélectionnez le contenu que vous souhaitez réutiliser. Le contenu "[A renommer]" créé à partir du modèle est alors détruit.
7. Cliquez sur le bouton **OK**. L'interface de service est créé.

*Vous pouvez ensuite modifier les éléments de la nouvelle interface de service, par exemple à partir de son diagramme, voir [Construire un diagramme d'interface de service \(BPMN\)](#).*

## Créer un modèle d'interface de service

Vous pouvez utiliser une interface de service existante pour créer un nouveau modèle d'interface de service.

Pour définir qu'une interface de service est un modèle :

1. Sélectionnez l'interface de service qui vous intéresse.
2. Ouvrez la page de propriétés **Caractéristiques**.

3. Cochez la case **Modèle de comportement d'interaction**.  
L'interface de service est ajoutée à la liste des modèles d'interface de service.

Seuls les composants du modèle d'interface de service déclarés comme des modèles sont dupliqués lors de l'utilisation du modèle.

Pour accéder à la liste des composants du modèle d'interface de service déclarés comme des modèles :

1. A partir du menu de navigation **Administration**, sélectionnez **Modèles > Interfaces de service**.
2. Ouvrez la page de propriétés **Définition du modèle** de l'interface de service qui vous intéresse.
3. Cochez la case **Modèle** des composants que vous souhaitez dupliquer.

---

## Créer un modèle d'opération de service

Pour définir qu'une opération de service est un modèle d'opération de service :

1. Sélectionnez l'opération de service qui vous intéresse.
2. Ouvrez la page de propriétés **Caractéristiques**.
3. Cochez la case **Modèle de comportement d'interaction de service**.  
L'opération de service est ajoutée à la liste des modèles d'opération de service.

☛ Pour accéder à la liste des modèles d'opération de service : à partir du menu de navigation **Administration**, sélectionnez **Modèles > Interfaces de service**.

# LES RAPPORTS HOPEX IT ARCHITECTURE



**HOPEX IT Architecture** offre des facilités d'analyse et de suivi de la mise en œuvre des évolutions de l'architecture de votre infrastructure informatique. Dans la Suite **HOPEX** les rapports permettent de regrouper des ensembles d'objets du référentiel et d'étudier leurs interactions.

☛ *Pour plus de détails sur le fonctionnement des rapports, voir le chapitre "Générer des rapports" dans le guide **HOPEX Common Features**.*

Les types de rapports proposés en standard par **HOPEX IT Architecture** offrent différentes possibilités de présentation des analyses. Certains de ces rapports sont communs avec d'autres solutions, **HOPEX IT Business Management** par exemple.

Enfin les **Rapports de diagrammes éclatés** sont disponibles pour plusieurs types d'objets. Ce type de rapport élabore une vue récapitulative d'une architecture complexe dans un diagramme unique ; elle consiste à ouvrir le diagramme de chaque complexe pour l'inclure dans le diagramme décrivant l'objet racine. Les diagrammes qui décrivent les objets du diagramme source sont affichés de manière récursive, de sorte que plusieurs niveaux de diagramme peuvent apparaître dans une image.

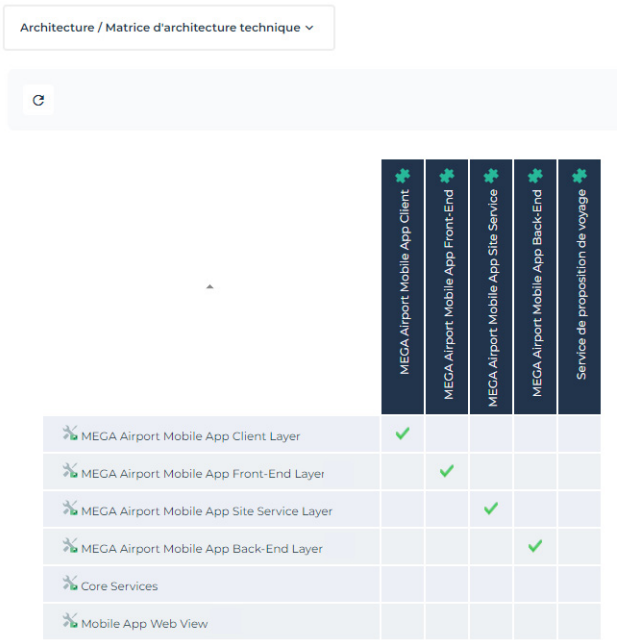
☛ *Pour plus de détails, voir "Lancer un rapport de diagrammes éclatés" dans le chapitre concernant les diagrammes du guide **HOPEX Common Features**.*

# RAPPORTS SUR LES ARCHITECTURES APPLICATIVES

## Matrice d'architecture technique

Ce rapport affiche la répartition des services informatiques d'une application sur ses domaines techniques (Application ou donnée).

### Exemple de rapport



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Application.	Un objet obligatoire.



## Densité des échanges entre applications

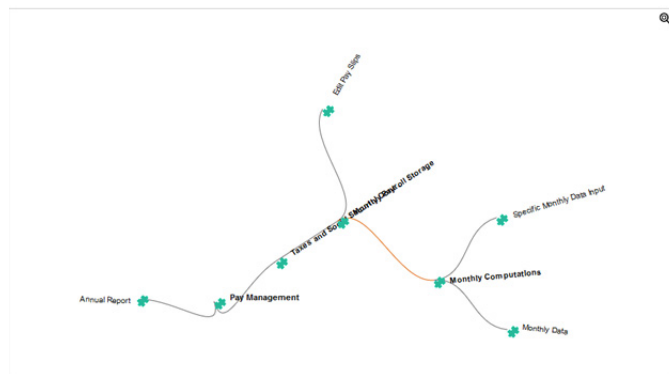
Ce rapport affiche la densité des échanges autour de l'objet racine dans le but de mieux définir les systèmes applicatifs.

Les lignes permettent d'indiquer qu'il existe au moins un contenu échangé entre deux applications.

La couleur de la ligne indique le nombre de contenus échangés.

- Gris : 1 ou 2 contenus,
- Vert : entre 3 et 5 contenus,
- Orange : entre 6 et 10 contenus,
- Brun : entre 11 et 15 contenus,
- Rouge : plus de 15 contenus.

### Exemple de rapport



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application ou service applicatif.	Un objet obligatoire.
Profondeur	Entier	

## Cohérence des échanges (Structure / Scénarios)

Ce rapport permet la vérification de la cohérence des échanges entre les descriptions de scénario et de structures via les flux d'application ou les interactions de service.

Il analyse tous les contenus émis/reçus par un agent via les flux d'application et vérifie leur présence dans les interactions de service associées.

### Exemple de rapport de Cohérence des échanges (Structure / Scénario)

Payroll - Exchange Consistency Structure Scenario									
External	Content	Interaction	External Target	Context	Flow	External Target	Scenario	Source (Context, Agent)	
+	Aggregated Monthly Taxes	Aggregated Monthly Taxes	Healthcare (External)	Payroll Management System	Aggregated Monthly Taxes	Healthcare (External)	Payroll Management System	Payroll Management System	Payroll Management System
+	Aggregated Monthly Taxes	Aggregated Monthly Taxes	Healthcare (External)	Payroll Management System	Aggregated Monthly Taxes	Healthcare (External)	Payroll Management System	Payroll Management System	Payroll Management System
+	Monthly Time Counters Update	Vacation Rights	Time Management	Payroll Management System	Monthly Time Counters Update	Time Management	Payroll Management System	Payroll Management System	Payroll Management System
+	Vacation Rights Monthly Update	Vacation Rights	Time Management	Payroll Management System	Vacation Rights Monthly Update	Time Management	Payroll Management System	Payroll Management System	Payroll Management System
+	Monthly Timesheet Data Request	Timesheets	Time Management	Payroll Management System	Monthly Timesheet Data Request	Time Management	Payroll Management System	Payroll Management System	Payroll Management System
+	Monthly Pay Data Update	Pay Data	Accounting	Payroll Management System	Monthly Pay Data Update	Accounting	Payroll Management System	Payroll Management System	Payroll Management System
+	Additional Hours	Timesheets	Time Management	Payroll Management System	Additional Hours	Time Management	Payroll Management System	Payroll Management System	Payroll Management System

### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application ou service applicatif.	Un objet obligatoire.

## Cohérence du contenu (Structure)

Ce rapport permet de vérifier la cohérence des échanges entre les "vues internes" et les "vues externes" dans le contexte de description de structures (interactions de service).

Une interface de service établie entre l'objet analysé et un autre (via une interaction de service) dans une "vue externe" doit aussi être établi avec un composant de l'objet analysé (via une autre interaction) dans une "vue interne" et vice versa.

☛ Toutes les interactions de service/interface de service avec d'autres agents doivent être gérés par un composant interne et toutes les

interactions de service/interface de service gérés par un composant interne doivent être utilisées par un autre agent.

### Exemple de rapport de Cohérence du contenu(Structure)

Payroll - IT Architecture - Structured Diagrams Exchange Consistency

External / Boundary Exchange Contract Consistency

External Object	Exchange Contract	Internal Object	Boundary Exchange Contract
Payroll			
Time Management	Timesheets	Pay Management	Timesheets
Accounting	Pay Data	Pay Management	Pay Data
Time Management	Vacation Rights	Pay Management	Vacation Rights
Time Management	Vacation Rights	Pay Management	Vacation Rights
Healthcare (External)	Aggregated Monthly Taxes	Taxes and Social Security Contributions	Aggregated Monthly Taxes
Healthcare (External)	Aggregated Monthly Taxes	Taxes and Social Security Contributions	Aggregated Monthly Taxes
Manager	Monthly Input	Pay Management	Monthly Input
Time Management	Vacation Rights	Pay Management	Vacation Rights
Time Management	Vacation Rights	Pay Management	Vacation Rights
Time Management	Timesheets	Pay Management	Timesheets
Accounting	Pay Data	Pay Management	Pay Data
Manager	Monthly Input	Pay Management	Monthly Input

### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, appli-cation ou service applica-tif.	Un objet obligatoire.

## Cohérence du contenu (Scenario)

Ce rapport permet de vérifier la cohérence des échanges entre les "vues internes" et les "vues externes" dans le contexte de description de scénario (flux d'application).

Un contenu envoyé/reçu dans une "vue externe" avec un autre agent doit aussi apparaître dans la "vue interne" d'un composant et vice versa.

☛ Tous les flux échangés avec d'autres agents doivent être gérés par un composant interne et tous les flux gérés par un composant interne doivent être utilisés par un autre agent.

### Exemple de rapport de Cohérence du contenu(Scenario)

Pay Management - IT Architecture - Content Consistency (Scenario)		
Pay Management		
Sent Content Internal/External Balance		
Content	External Target (Scenario)	Internal Source (Scenario)
Aggregated Monthly Taxes	Taxes and Social Data Security Contributions (Payroll)	Monthly Payroll Storage (Pay Management)
Monthly Time Counters Update	Monthly Time Counters Update (Payroll)	Monthly Computations (Pay Management)
Aggregated Annual Figures	Annual Report (Payroll)	Monthly Payroll Storage (Pay Management)
Vacation Rights Monthly Update	Vacation Rights Monthly Update (Payroll)	Monthly Computations (Pay Management)
Monthly Pay Data Update	Monthly Pay Data Update (Payroll)	Monthly Payroll Storage (Pay Management)
Internal Content has no External Target		
No Internal Source		
Received Content Internal/External Balance		
Content	External Source (Scenario)	Internal Target (Scenario)
Additional Hours	Additional Hours (Payroll)	Monthly Payroll Storage (Pay Management)
Vacation Dates	Vacation Dates (Payroll)	Monthly Payroll Storage (Pay Management)
Monthly Accounting Data	Monthly Accounting Data (Payroll)	Monthly Payroll Storage (Pay Management)
Monthly Bonus Update	Monthly Bonus Update (Payroll)	Specific Monthly Data Input (Pay Management)
Business Absences	Business Absences (Payroll)	Monthly Payroll Storage (Pay Management)
Internal Content has no External Source		
No Internal Target		

### Paramètres du rapport

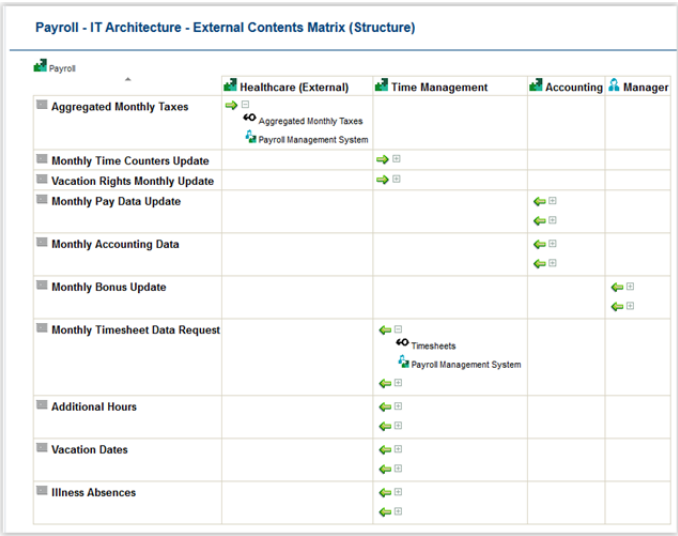
Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application ou service applicatif.	Un objet obligatoire.

## Matrice des contenus externes (Structure)

Ce rapport présente la matrice des contenus émis ou reçus par l'agent analysé dans le contexte de la description d'une structure (interactions de service).

Exemple de matrice des contenus externes (Structure)



Paramètres du rapport

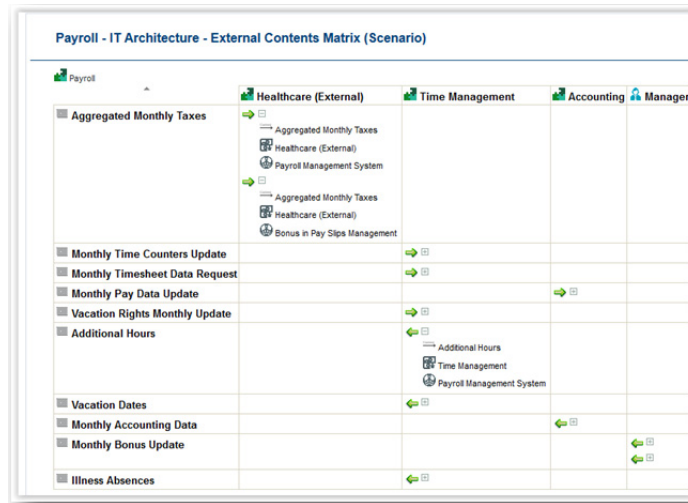
Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, appli- cation ou service applica- tif.	Un objet obligatoire.

Matrice des contenus externes (Scénario)

Ce rapport affiche la matrice des contenus émis ou reçus par l’agent analysé avec les autres agents dans le périmètre de la description d’un scénario (flux applicatifs).

### Exemple de matrice des contenus externes (Scénario)



### Paramètres du rapport

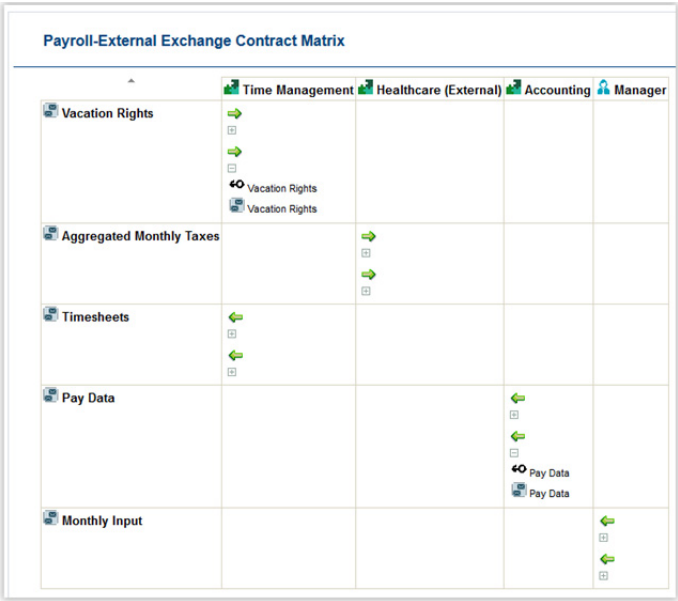
Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application ou service applicatif.	Un objet obligatoire.

## Matrice des interfaces de service externes

Ce rapport affiche la matrice des interfaces de service entre l'agent analysé et les autres agents.

Exemple de matrice des interfaces de service externes



Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, appli- cation ou service applica- tif.	Un objet obligatoire.

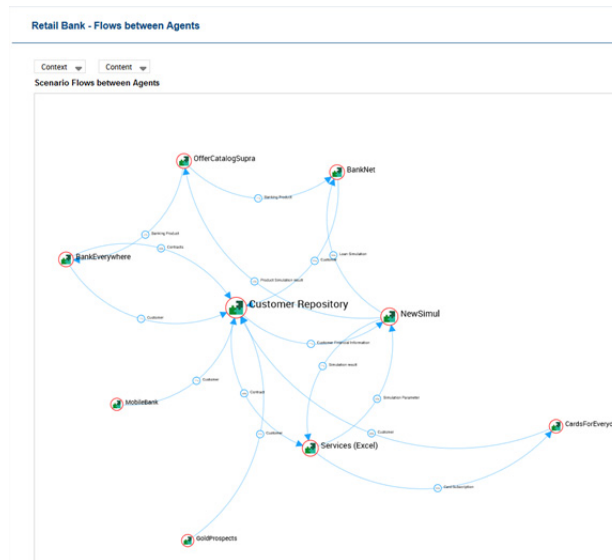
Graphe des flux entre des agents

Ce rapport de type «graphe» présente une synthèse des flux applicatifs échangés entre des agents (Système Applicatif, Application, Service Applicatif, Microservice, etc.) sélectionnés par l'utilisateur.

Il est possible d'appliquer des filtres sur le résultat en sélectionnant des interfaces de service et/ou des contextes d'interactions de service.

Ce rapport peut être visualisé en 2D ou en 3D.

## Exemple de graphe de flux entre des agents



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application, service applicatif ou microservice.	Un objet obligatoire.

## Graphe des flux d'un agent

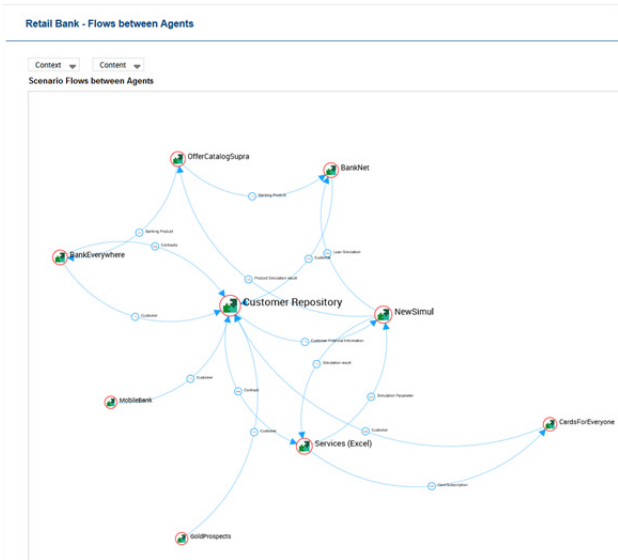
Ce rapport de type « graphe » présente une synthèse des flux applicatifs échangés par un agent (Système Applicatif, Application, Service Applicatif, Microservice, etc.).

Il est possible d'appliquer des filtres sur le résultat en sélectionnant des contenus échangés et/ou des contextes des flux.

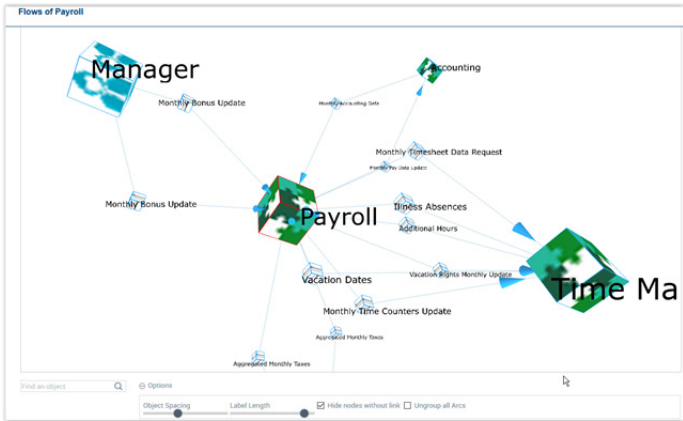
Ce rapport peut être visualisé en 2D ou en 3D.



### Exemple de rapport de flux d'un agent en 2D



### Exemple de rapport de flux d'un agent en 3D



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, appli- cation, service applicatif ou microservice.	Un objet obligatoire.

---

## Rationalisation des flux traités

Ce graphique en bâton présente la répartition des contenus multi-chaînes de communication applicative. Il permet d'identifier rapidement les contenus ayant le plus de chaînes de communication associées et donc potentiellement les moins urbanisées en terme de flux.

### ***Paramètres du rapport***

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système de communication	Un objet obligatoire.

---

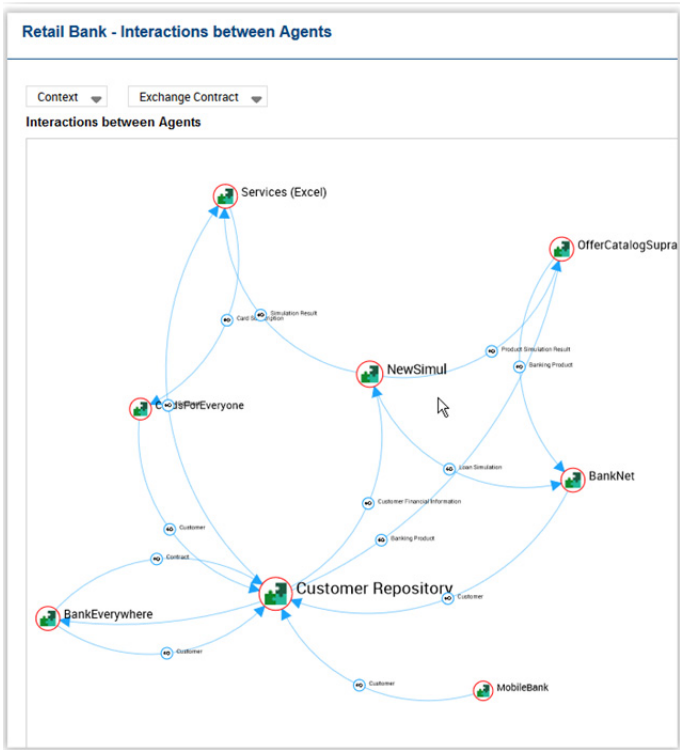
## Graphe des interactions de service entre des agents

Ce rapport de type « graphe » présente une synthèse des interactions de service échangées entre des agents (Système Applicatif, Application, Service Applicatif, Microservice, etc.) sélectionnés par l'utilisateur.

Il est possible d'appliquer des filtres sur le résultat en sélectionnant des interactions de service échangées et/ou des contextes d'interactions de service.

Ce rapport peut être visualisé en 2D ou en 3D.

Exemple de graphe d'interactions de service entre des agents



Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, appli- cation, service applicatif ou microservice.	Un objet obligatoire.

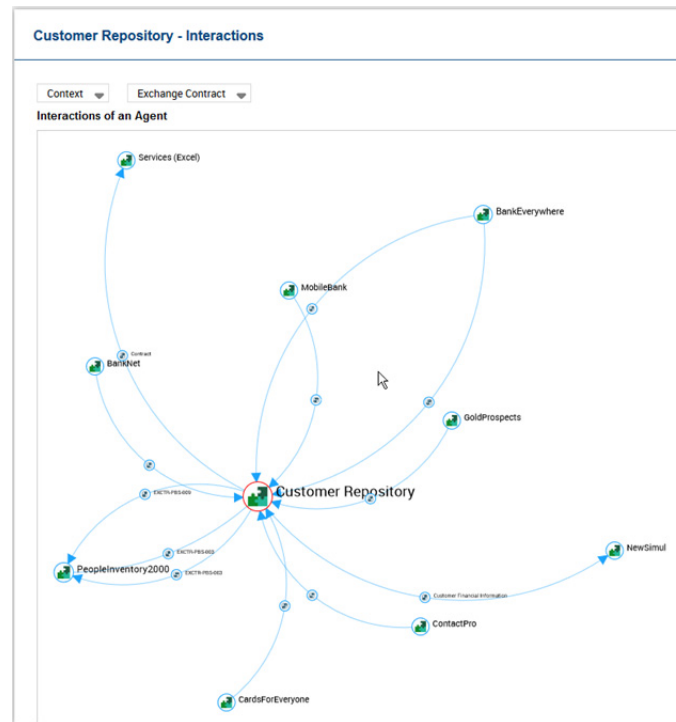
Graphe des interactions de service d'un agent

Ce rapport de type «graphe» présente une synthèse des interactions de service auxquelles participe un agent (Système Applicatif, Application, Service Applicatif, Microservice, etc.) sélectionnés par l'utilisateur.

Il est possible d'appliquer des filtres sur le résultat en sélectionnant des interfaces de service et/ou des contextes d'interactions de service.

Ce rapport peut être visualisé en 2D ou en 3D.

### Exemple de graphe d'interaction d'un agent



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application, service applicatif ou microservice.	Un objet obligatoire.

# RAPPORTS SUR LA COUVERTURE FONCTIONNELLE DE L'ARCHITECTURE

---

## Rapport de décomposition des briques d'entreprise

L'objectif de ce rapport est de détailler Classificateur composite / Objets qui composent ses composants de classificateur en mettant l'accent sur la réalisation par des équipements EA de chacun des composants.

- La profondeur de l'analyse peut être définie,
- Les types des composants analysés peuvent être affichés ou filtrés (Dimension EA),
- Les types des éléments réalisateurs peuvent être affichés ou filtrés (Couche EA),
- L'apparence peut être améliorée (palette de couleur, nombre de colonnes affichées).

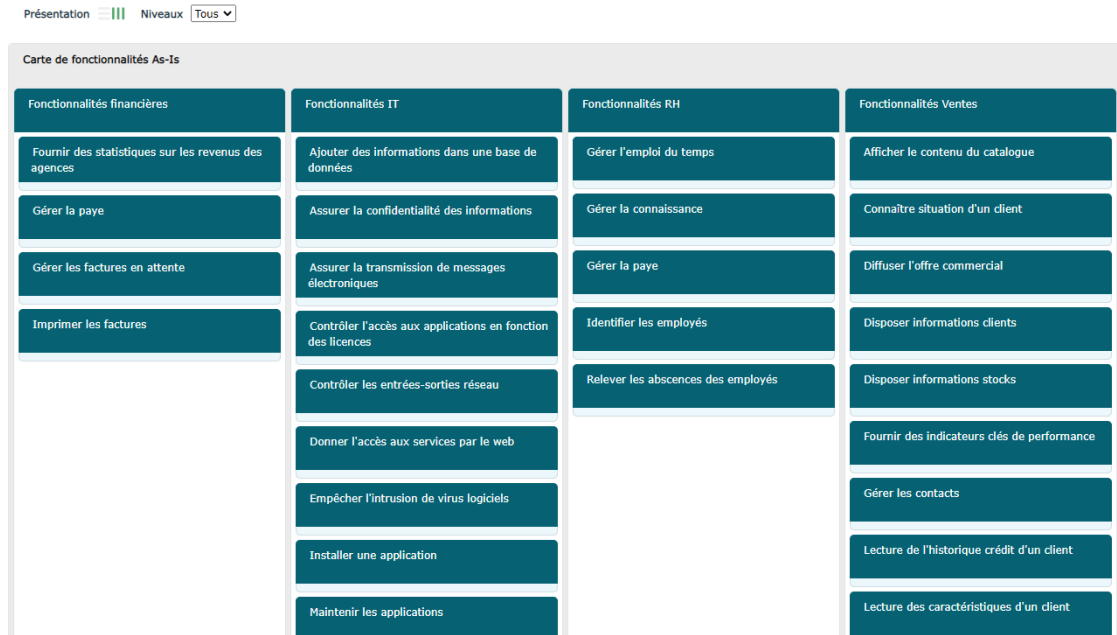


*Pour plus de détails sur l'utilisation d'un rapport de décomposition, voir le chapitre "Manipuler un rapport de décomposition" du guide **HOPEX Common Features**.*

## Exemples de rapports

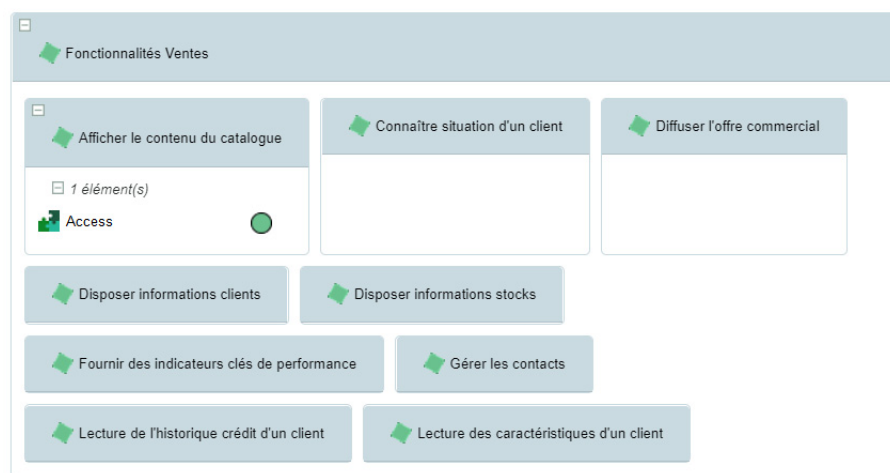
L'exemple ci-dessous permet de visualiser la décomposition des fonctionnalités de la carte de fonctionnalités passée en paramètre.

### Rapport de décomposition des briques d'entreprise-1



#### Exemple de carte de décomposition des fonctionnalités.

Dans l'exemple ci-dessous, les applications qui mettent en œuvre les fonctionnalités sont présentées.

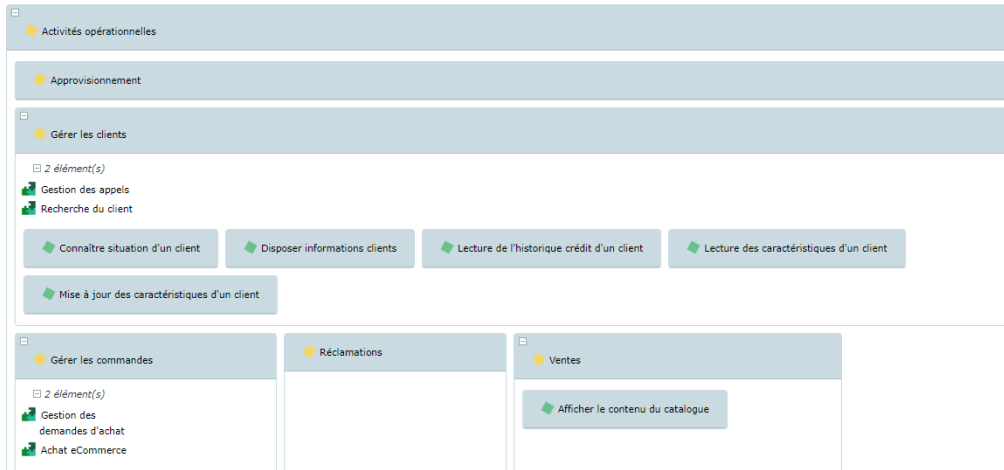


Une autre présentation permet de visualiser directement la mise en œuvre des capacités métier par des applications.

#### Rapport de décomposition des briques d'entreprise ①



Présentation Niveaux Tous ☒ Déplier tous les éléments



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Brique d'entreprise,	Un objet obligatoire.
Niveau de profondeur	Entier	Définit le niveau de décomposition de la carte de capacité métier ou de la capacité passée en paramètre
Nombre de colonnes	Entier	Définit le nombre de colonnes affichées par niveau de décomposition (par ex. 2 ou 3)
Palette de couleur	Palette HOPEX	Obligatoire. La palette livrée par défaut est « BoxInBox Report Monochrome Grey »
Niveau EA	Choix multiple : - niveau métier, - niveau organisationnel, - niveau application, - niveau technique.	Définit quels objets de quel type de niveau d'architecture sont affichés pour les réalisations des capacités ;  <i>Par exemple l'activation du « niveau application » va faire apparaître les réalisations des capacités métier sur les Environnements de Systèmes applicatifs, les Systèmes Applicatifs ou les Applications</i>
Dimension EA	Choix multiple : - modèles de capacité, - modèles d'agent, - modèle de processus, - modèles d'information, - modèles de performances, - modèles de résultats	Définit quels types d'objets sont examinés dans le cadre de l'analyse de la décomposition  <i>Par exemple l'activation des « modèles de capacités » vont afficher les compétences métier ou les fonctionnalités requises par les capacités métier décomposées</i>

➡ Pour plus de détails sur la manière d'associer une capacité métier à une fonctionnalité, voir [Décrire la mise en œuvre d'une capacités métier](#).

## Chevauchement d'applications

Ce rapport présente un tableau des systèmes applicatifs, applications, services applicatifs et microservices qui ont le même périmètre fonctionnel que l'élément décrit.

➡ Pour plus de détails sur la manière d'associer un élément concret à une fonctionnalité, voir [Décrire la mise en œuvre d'une fonctionnalité](#).



## Exemple de rapport chevauchement d'applications

### Overlapping Applications

Overlapping Applications	Business Capabilities	Fonctionnalités
 BankNet	 Electronic Banking	
 CardsForEveryone	 Electronic Banking	 Bank
 BankEverywhere	   	  

### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Système applicatif, application, service applicatif ou microservice.	Un objet obligatoire.

## Rapport de décomposition de capacité métier

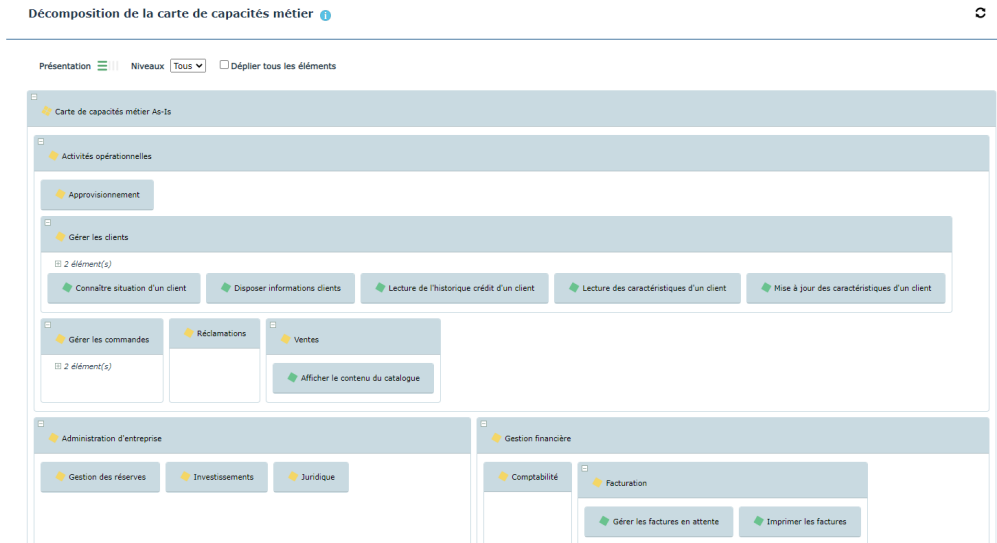
Vous pouvez utiliser ce rapport pour visualiser la couverture de réalisation des éléments des capacités métier par des éléments opérationnels tels que des applications logiques et physiques, des systèmes applicatifs, etc.

☛ Pour plus de détails sur la manière d'associer une capacité métier à une application, voir [Créer une mise en oeuvre de capacité métier](#).

☛ Pour plus de détails sur l'utilisation d'un rapport de décomposition, voir [Rapport de décomposition des briques d'entreprise](#).

## Exemples de rapport

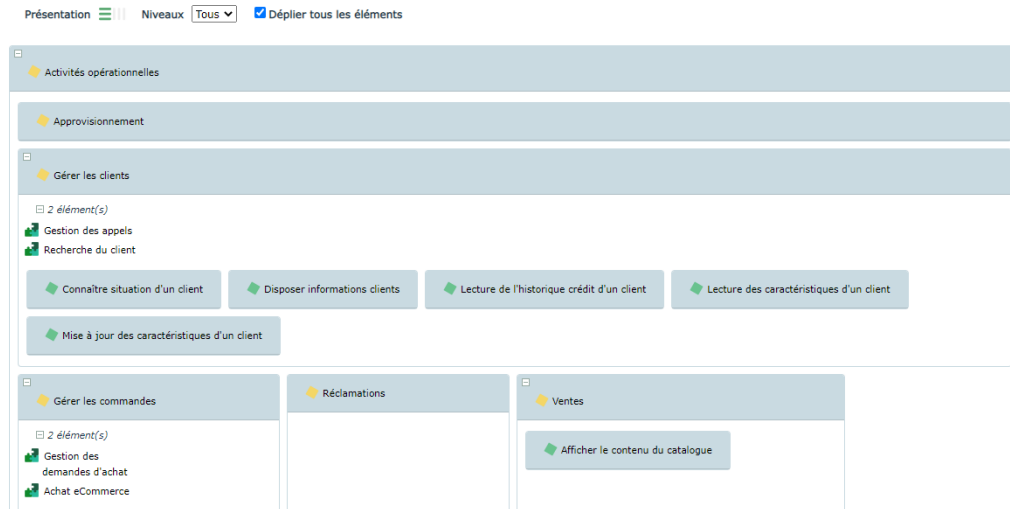
L'exemple ci-dessous permet de visualiser le taux de couverture fonctionnelle de la carte de capacité passée en paramètre.



L'exemple ci-dessous permet de visualiser comment les fonctionnalités associées aux capacités sont mises en œuvre par des composants applicatifs.

✎ Pour plus de détails sur la manière d'associer une capacité métier à une fonctionnalité, voir [Décrire la mise en œuvre d'une capacité métier](#).

## Rapport de décomposition des briques d'entreprise ⓘ



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Capacité Métier et Carte de capacités métier	Un objet obligatoire.

🔍 Pour plus de détails sur la manière d'associer une capacité métier à une fonctionnalité, voir [Décrire la mise en œuvre d'une capacités métier](#).

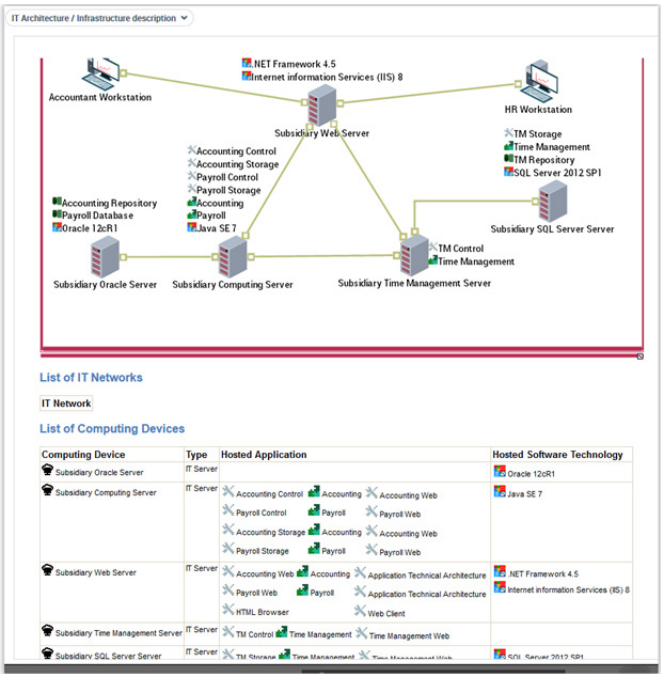
# RAPPORTS SUR LES INFRASTRUCTURES

## Rapport de description d'infrastructure

Ce rapport affiche une description de l'infrastructure et dresse la liste des canaux réseau définis entre les composants.

✎ Pour plus de détails, voir [Modéliser les infrastructures informatiques](#).

### Exemple de rapport



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Infrastructure informa- tique ou réseau informa- tique	Un objet obligatoire.

## Matrice des exigences technologiques applicatives x technologies de l'infrastructure informatique

Ce rapport présente la matrice des exigences technologiques applicatives requises par les applications x Technologies fournies par l'infrastructure informatique.

- La couleur verte indique que le support est conforme aux exigences attendues.
- La couleur rouge indique que le support ne couvre pas l'exigence.
- La couleur orange indique que la capacité du support n'est pas requise.

### Exemple de rapport

IT Infrastructure x Application Technical Area Required Software Technologies		
		Payroll
		Application Technical Architecture
Subsidiary IT Infrastructure	Subsidiary Oracle Server	Web Client Internet Explorer 11 Chrome Oracle 12cR1
	Subsidiary Computing Server	Web Client Internet Explorer 11 Chrome Java SE 7
	Subsidiary Web Server	Web Client Internet Explorer 11 Chrome .NET Framework 4.5 Internet Information Services (IIS) 8
	Subsidiary Time Management Server	Web Client Internet Explorer 11 Chrome
	Subsidiary SQL Server Server	Web Client Internet Explorer 11 Chrome SQL Server 2012 SP1
	HR Workstation	Web Client Internet Explorer 11 Chrome
	Accountant Workstation	Web Client Internet Explorer 11 Chrome

### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Application	Un objet obligatoire.

## Canaux réseau x Interactions de service

Ce rapport affiche une matrice des canaux réseau d'une infrastructure par rapport aux interactions applicatives.

### Exemple de rapport

	Time sheets	Vacation Rights	Expense Sheet
Subsidiary Computing Server – Subsidiary Web Server			
Subsidiary Oracle Server – Subsidiary Computing Server			
Subsidiary Web Server – Subsidiary Time Management Server			
Subsidiary Time Management Server – Subsidiary SQL Server Server			
Subsidiary Web Server – Accountant Workstation			
Subsidiary Web Server – HR Workstation			
Subsidiary Computing Server – Subsidiary Time Management Server	✓	✓	✓

### Paramètres du rapport

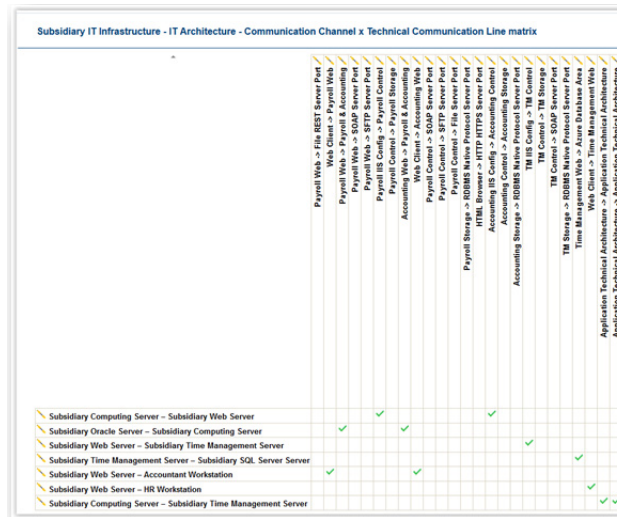
Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Infrastructure informatique ou réseau informatique	Un objet obligatoire.

## Canaux réseau x connexions de module

Ce rapport affiche une matrice des canaux réseau de l'infrastructure x les connexions de module des architectures techniques.

### Exemple de rapport



### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Infrastructure informa- tique ou réseau informa- tique	Un objet obligatoire.

## RAPPORTS SUR LES ARCHITECTURES DE DÉPLOIEMENT

---

### Rapport d'architecture de déploiement

Ce rapport présente sous forme de tableaux, les caractéristiques des éléments suivants :

- Les modules de déploiement de l'architecture décrite,
- Les connexions de module,
- Les supports d'hébergement recommandés.

#### ***Paramètres du rapport***

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Architecture de déploiement d'une application Architecture de déploiement d'un système applicatif	Un objet obligatoire.

---

### Matrice d'architecture de déploiement

Ce rapport présente la matrice des services applicatifs d'une application sur les domaines techniques (Application et données).



Exemple de rapport

Payroll-Technical Architecture Matrix									
	Monthly Input GUI	Monthly Data	Pay Management	Taxes and Social Security Contributions	Annual Report	Specific Monthly Data Input	Monthly Payroll Storage	Monthly Computations	Edit Pay Slips
Web Client									
Payroll GUI									
Payroll Web									
Payroll IIS Config									
Payroll Control									
Payroll Storage									
HTML Browser									

Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Application	Un objet obligatoire.

Matrice des connexions de module x Interactions de service

Ce rapport permet l'analyse des interactions de service supportées par les connexions de module.

Une case cochée indique qu'une connexion de module supporte l'interaction de service correspondante dans une architecture de déploiement.

### Exemple de rapport

Payroll - IT Architecture - Technical Communication Line x Interaction matrix												
	Timesheets	Pay Data	Vacation Rights	Vacation Rights	Aggregated Monthly Taxes	Aggregated Annual Figures	Aggregated Monthly Taxes	Timesheets	Pay Data	Vacation Rights	Monthly Input	
Web Client -> Payroll Web												
Payroll Web -> Payroll & Accounting												
Payroll Web -> SOAP Server Port												
Payroll Web -> SFTP Server Port												
Payroll Web -> File REST Server Port												

### Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

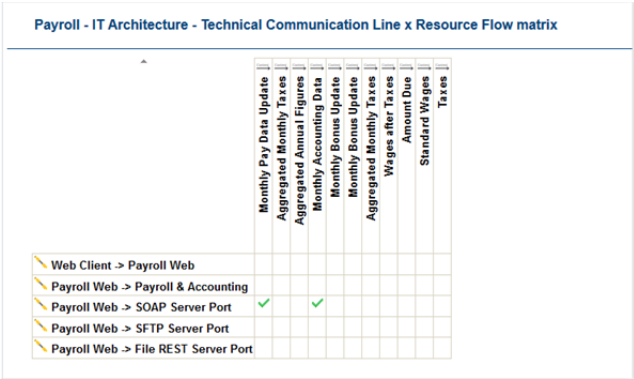
Paramètres	Type du paramètre	Contraintes
Objet racine	Architecture de déploiement d'une application Architecture de déploiement d'un système applicatif	Un objet obligatoire.

## Matrice des connexions de module x Flux applicatifs

Ce rapport permet l'analyse des flux applicatifs supportés par les connexions de module.

Une case cochée indique qu'une connexion de module supporte le flux applicatif correspondant dans une architecture de déploiement.

Exemple de rapport

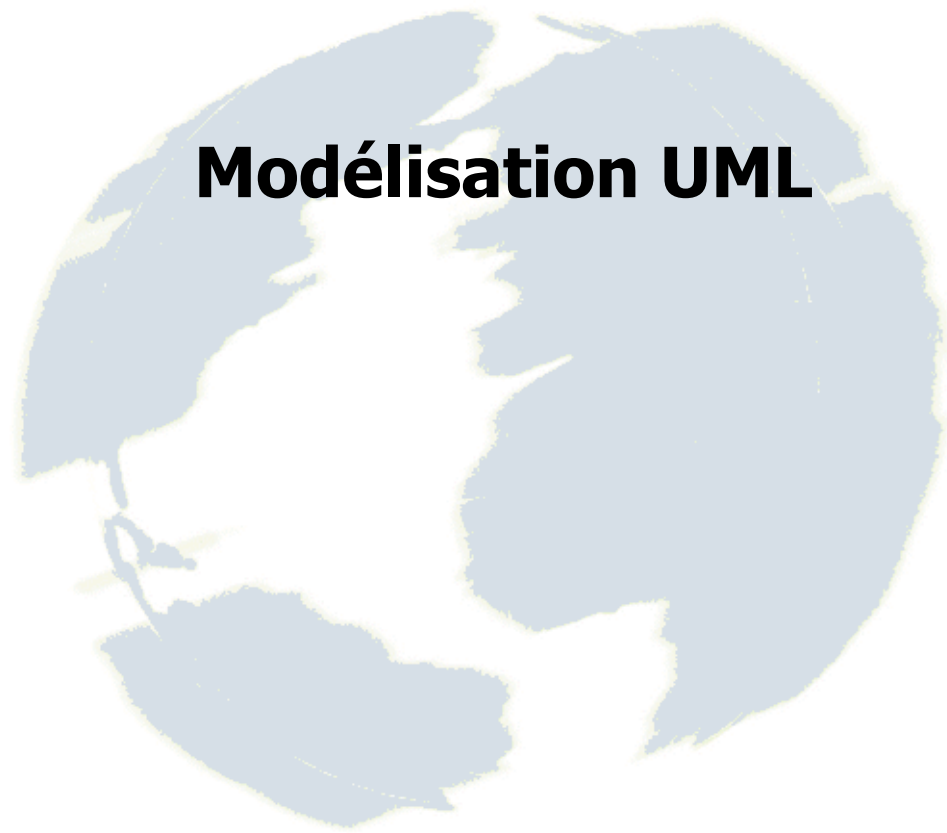


Paramètres du rapport

Il s'agit ici de définir les données en entrée du rapport.

Paramètres	Type du paramètre	Contraintes
Objet racine	Architecture de déploie- ment d'une application Architecture de déploie- ment d'un système appli- catif	Un objet obligatoire.







# A PROPOS DE L'IMPLÉMENTATION UML



Le langage de modélisation UML (Unified Modeling Language) s'est érigé comme standard pour la modélisation graphique des systèmes d'information. **HOPEX IT Architecture V2** offre un ensemble d'outils vous qui vous permettent de modéliser votre SI conformément à la version 2.3 de ce standard.

✎ Pour accéder à cette fonctionnalité UML, vous devez être connecté avec le profil **Architecte Solution** ou avec le profil **Administrateur fonctionnel d'architecture de solution**.

Cette partie a pour objectif de vous faire découvrir les principales fonctionnalités d'**HOPEX IT Architecture V2** spécifiques à UML.

- ✓ ["Présentation", page 266](#)
- ✓ ["Organisation des diagrammes d'UML", page 268](#)

# PRÉSENTATION

Les facilités offertes par **HOPEX IT Architecture V2** pour la modélisation UML sont introduites ci-dessous.

## Analyser les cas d'utilisation

Une réflexion sur les fonctionnalités attendues du futur système est nécessaire avant sa conception. Les composants de ce système vont en effet être utilisés par les acteurs de l'organisation pour effectuer leur mission. Les divers "cas d'utilisation" de ce système vont être présentés dans les **diagrammes de cas d'utilisation**.

Ils serviront de point de départ pour la découverte des objets.

Ils permettront ensuite de valider l'utilisation de ces objets à l'aide des diagrammes d'interaction.

Ils donneront enfin des critères de regroupement en "paquetage" pour les objets découverts.

Voir ["Le diagramme de cas d'utilisation"](#), page 271.

## Identifier les objets

Les objets ayant une structure semblable, le même comportement et les mêmes types de relations avec d'autres objets sont réunis dans une classe.

Le **diagramme de classes** permet d'identifier les objets mis en jeu à l'intérieur du système et de définir leur structure en termes d'attributs et d'opérations ainsi que les relations entre eux. Le **diagramme d'objets** montre les instances compatibles avec un diagramme de classes particulier qu'on peut ainsi valider avec un exemple.

Voir ["Le diagramme de classes"](#), page 285.

## Décrire les comportements

Le **diagramme de machine à états** permet de définir le comportement d'un objet vis-à-vis des sollicitations internes ou externes auxquelles il peut être soumis. Chacun des états dans lequel peut se trouver l'objet est indiqué ainsi que les réactions de l'objet à un événement donné en fonction de l'état où il se trouve.

Le **diagrammes d'activité** décrit également un comportement, mais en termes d'actions.

Voir :

- ["Le diagramme de machine à états"](#), page 351
- ["Le diagramme d'activités"](#), page 361

## Représenter les interactions entre les objets

Le dialogue qui s'instaure entre les différents objets concernés par l'événement pour y répondre peut être représenté dans des **diagrammes d'interaction**.

Les diagrammes d'interaction insistent sur les échanges qui ont lieu entre les objets.



Le **diagramme de séquence** présente ces mêmes échanges en mettant en évidence leur chronologie.

Le **diagramme de communication** met l'accent sur l'organisation structurelle des objets qui envoient et reçoivent des messages.

Le **diagramme de vue générale d'interaction** donne une vue d'ensemble du flot de contrôle.

Voir ["Les diagrammes d'interaction", page 369](#).

## Répartir les classes entre les paquetages

Une fois les objets identifiés, il est aisé de répartir les classes qui les implémentent entre les différents paquetages. Les regroupements de ces classes effectués dans le **diagramme de paquetages** sont faits de manière à minimiser les échanges entre les différents paquetages. Ils obéissent à deux critères : l'un, technique, lié à leur environnement d'exécution et l'autre, organisationnel, lié à l'emploi qui en sera fait par les différents utilisateurs à équiper pour chaque cas d'utilisation.

Voir ["Le diagramme de paquetages", page 338](#).

## Définir les interfaces

Pour respecter le principe d'encapsulation, la répartition des éléments entre les composants est stricte. Il est donc nécessaire de prévoir les interfaces entre les éléments ayant des relations entre eux et qui appartiennent cependant à des composants différents.

Le **diagramme de composants** et le **diagramme de structure composite** présentent l'interdépendance des composants ou éléments d'un composant.

La définition des interfaces des objets ainsi que l'adhésion à un protocole d'échange normalisé (CORBA2, DCOM/OLE) sont des facteurs clés de l'interopérabilité, c'est-à-dire la capacité à faire coopérer les objets développés et exploités dans des environnements hétérogènes.

Voir :

- ["Le diagramme de composants", page 341](#)
- ["Le diagramme de structure composite", page 346](#)

## Spécifier la mise en oeuvre

La mise en oeuvre des objets dans un environnement de travail concret peut être spécifiée les **diagrammes de déploiement**.

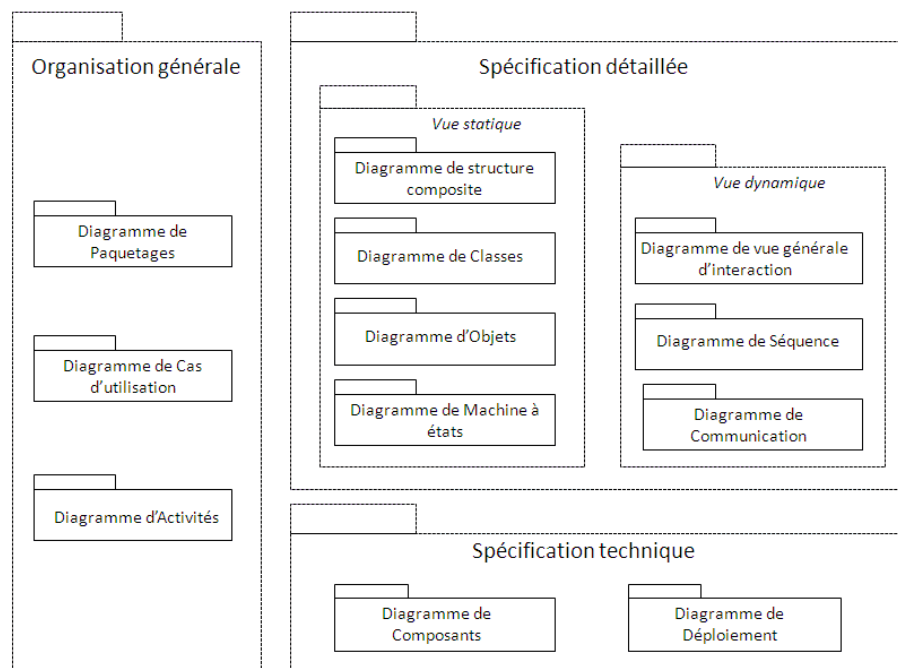
Voir ["Le diagramme de déploiement", page 387](#).

# ORGANISATION DES DIAGRAMMES D'UML

## Organisation générale

Les **diagrammes de cas d'utilisation** vous permettent de montrer les principales interactions entre le système étudié et son environnement, et de mettre en évidence ses principaux sous-systèmes.

Les **diagrammes de paquetages** constituent un découpage du système. Le découpage du système en paquetages est relativement structurant dans la mesure où un objet ne peut être que dans un seul paquetage. Vous pouvez commencer à dessiner des diagrammes de paquetages dès que vous avez identifié les principaux composants de votre système (Commercial, Production, Facturation, ...).



## Spécification détaillée

Le diagramme principal est le **diagramme de classes**. Il décrit l'essentiel de la sémantique des objets composant le système. C'est le diagramme sur lequel les concepteurs passeront généralement le plus de temps. La découverte des classes se fait généralement par des aller-retour entre les diagrammes de classes et les diagrammes de séquence.

Le **diagramme de machine à états** permet de décrire l'aspect statique d'un objet, c'est à dire les différents états dans lesquels il peut se trouver et les transitions possibles entre ces états. Il permet ainsi de compléter la description d'une classe.

Les **diagrammes d'interaction** permettent de spécifier l'aspect dynamique du système en montrant l'interaction des objets entre eux. Ils permettent en particulier de décrire de façon détaillée les différents scénarios de fonctionnement d'un cas d'utilisation. Le diagramme de séquence explicite plutôt le déroulement d'un scénario dans le temps, tandis que le diagramme de communication insiste plutôt sur l'interaction entre les objets.

## Spécification technique et mise en oeuvre

Le **diagramme de composants** décrit les différents composants techniques d'une application et leurs interactions.

Le **diagramme de structure composite** précise les collaborations entre les composants ou éléments d'un composant dans l'exécution d'une tâche commune.

Le **diagramme de déploiement** permet de préciser l'architecture technique du système en indiquant sur quels postes de travail ou sur quels nœuds du système informatique seront installés les différents composants de l'application.

## Points d'entrée des diagrammes d'UML

Diagramme	Points d'entrée
Diagramme de classes	Paquetage, Classe, Cas d'utilisation
Diagramme d'objets	Classe, Composant, Paquetage, Cas d'utilisation
Diagramme de composants	Composant, Paquetage
Diagramme de structure composite	Composant, Classe, Collaboration
Diagramme de déploiement	Paquetage
Diagramme de paquetages	Paquetage, Bibliothèque
Diagramme de cas d'utilisation	Paquetage, Cas d'utilisation, Environnement d'application (ADES)
Diagramme de séquence (UML2)	Interaction
Diagramme de communication	Interaction
Diagramme de vue générale d'interaction	Interaction
Diagramme d'activités (UML2)	Activité
Diagramme de machine à états	Machine à état, Machine à état de protocole

Dans **HOPEX IT Architecture V2**, les différents points d'entrée ci-dessus sont accessibles dans le menu de navigation **Conception (UML)**.



# LE DIAGRAMME DE CAS D'UTILISATION



Le diagramme de cas d'utilisation constitue une première étape dans la description d'un système d'information. Il permet d'identifier les fonctionnalités que doit fournir le système afin de répondre aux besoins des acteurs de l'organisation ; il décrit en ce sens les interactions entre le système et les acteurs.

Les points suivants sont abordés ici :

- ✓ [Créer un diagramme de cas d'utilisation](#)
- ✓ [Les éléments du diagramme de cas d'utilisation](#)

# CRÉER UN DIAGRAMME DE CAS D'UTILISATION

Un diagramme de cas d'utilisation permet de décrire les interactions entre les acteurs de l'organisation et le système dans chacun des *cas d'utilisation* envisagés.



*Un cas d'utilisation est une suite d'actions qui amène un résultat observable pour un acteur particulier. Des scénarios illustrent les cas d'utilisation par l'exemple.*

Ces cas d'utilisation sont regroupés dans des *paquetages* représentant les frontières du système.



*Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.*

Vous pouvez créer un diagramme de cas d'utilisation à partir d'un paquetage. Cependant, dans le cadre de systèmes complexes, vous pouvez créer ce type de diagramme depuis un cas d'utilisation afin de détailler ce dernier.

Dans **HOPEX IT Architecture** vous pouvez également créer un diagramme de cas d'utilisation sur l'environnement d'application d'un projet. Voir [Créer un diagramme de cas d'utilisation d'une application](#).

---

## Créer un paquetage

Pour créer un paquetage dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Cliquez sur le sous-menu **Paquetages**.
2. Dans la zone d'édition, cliquez sur **Nouveau**.  
La fenêtre de création d'un paquetage s'ouvre.
3. Saisissez son **Nom**.
4. Indiquez éventuellement la bibliothèque ou le paquetage détenteur.



*La bibliothèque par défaut est utilisée pour ranger un objet s'il n'y a pas de bibliothèque courante au moment de sa création.*

5. Cliquez sur **OK**.

Le paquetage est créé et ajouté dans la liste des paquetages.

---

## Créer le diagramme de cas d'utilisation du paquetage

Pour créer un diagramme de cas d'utilisation :

1. Dans la ligne du paquetage qui vous intéresse, cliquez sur le bouton **Nouveau Diagramme.**
2. Dans la fenêtre de choix, sélectionnez **Diagramme de cas d'utilisation.**


Le diagramme créé s'ouvre dans la fenêtre d'édition. Le cadre du paquetage est positionné à l'intérieur du dessin.

# LES ÉLÉMENTS DU DIAGRAMME DE CAS D'UTILISATION

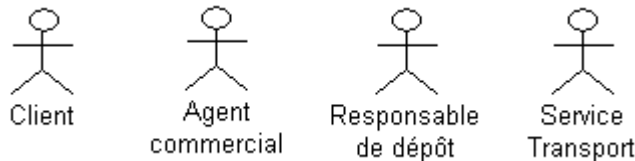
- ✓ Les acteurs
- ✓ Les cas d'utilisation
- ✓ Les paquetages
- ✓ Les participations
- ✓ Les associations entre cas d'utilisation : extension et inclusion
- ✓ Les généralisations
- ✓ Les interfaces

---


## Les acteurs

 Un acteur représente le rôle joué par quelque chose ou quelqu'un se trouvant dans l'environnement de l'entreprise ou du système étudié. Il est en relation avec le métier de l'entreprise et interagit avec le système dans différents cas d'utilisation. Ce peut être un élément de la structure de l'entreprise tel qu'une direction, un service ou un poste de travail.

Exemples d'acteurs :



Pour créer un **acteur** dans un diagramme de cas d'utilisation :

1. Dans la barre d'objets, cliquez sur le bouton **Acteur** .
2. Cliquez sur le plan de travail du diagramme.  
La fenêtre **Ajout d'un acteur** s'ouvre.
3. Saisissez le nom de l'acteur, "Standardiste", par exemple.
4. Cliquez sur le bouton **Ajouter**.  
L'acteur apparaît alors dans le diagramme.

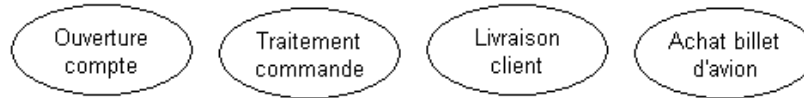
😊 Vous pouvez créer plusieurs éléments à la suite sans revenir à la barre d'outils en effectuant un double-clic sur le bouton **Acteur**

😊 Pour revenir ensuite au mode normal, utiliser la touche <Echap>, ou cliquez sur un autre bouton de la barre d'outils,




## Les cas d'utilisation

Exemples de **cas d'utilisation** : traitement d'une commande, livraison d'un client, ouverture d'un compte, envoi d'une facture, établissement d'un crédit, achat d'un billet d'avion, etc.



Pour créer un cas d'utilisation dans un diagramme :

1. Dans la barre d'objets du diagramme de cas d'utilisation cliquez sur le bouton **Cas d'utilisation** .  
La fenêtre **Ajout d'un cas d'utilisation** s'ouvre.
2. Saisissez le **Nom** du cas d'utilisation et cliquez sur le bouton **Ajouter**.

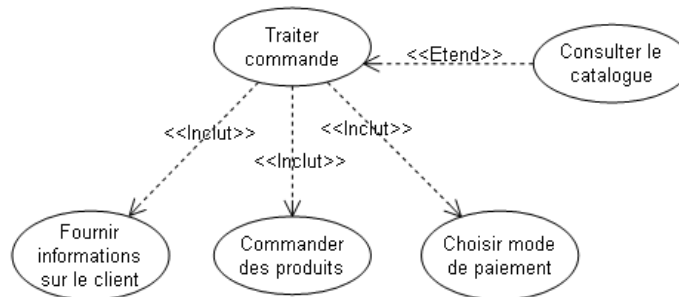
Le cas d'utilisation apparaît sur le diagramme.

### Faire un zoom sur un cas d'utilisation

Pour ouvrir directement, depuis le diagramme du packaging, le diagramme qui décrit un cas d'utilisation :


1. Faites un clic droit sur le cas d'utilisation.
2. Sélectionnez **Diagramme de cas d'utilisation**.

Le diagramme de cas d'utilisation s'ouvre.



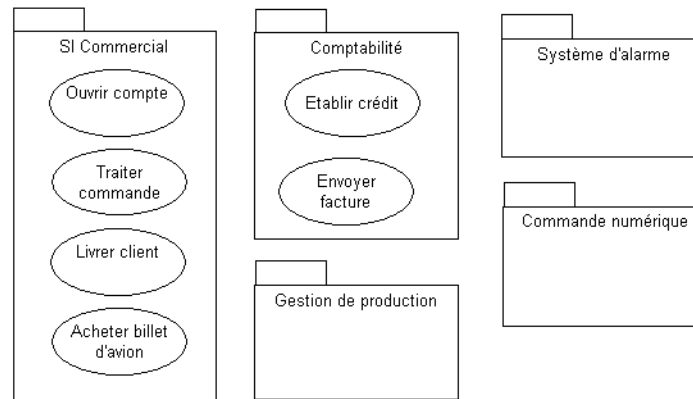
☺ Ce zoom sur la description d'un objet est disponible sur tous les éléments décrits par un diagramme.


## Les paquetages

 Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas

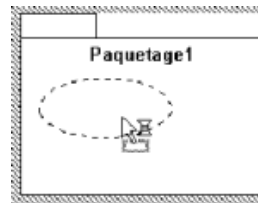
*d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.*

Exemples de **paquetage** : Le système d'information commercial, la comptabilité, la gestion de production, la commande numérique d'une machine, un système d'alarme, etc.



Vous pouvez créer un paquetage à l'aide du bouton **Paquetage**  de la barre d'outils. Vous pouvez l'agrandir pour pouvoir poser des cas d'utilisation dessus.

Vous pouvez relier un cas d'utilisation à un paquetage simplement en le posant dessus. Lorsque vous déplacez l'objet sur le paquetage, la forme du paquetage est mise en relief pour indiquer que l'objet va bien lui être relié.



☛ Si les objets reliés disparaissent sous le paquetage, cliquez sur le paquetage puis cliquez sur le bouton **Arrière-plan** de la barre de dessin.

Lorsque vous déplacez un cas d'utilisation d'un paquetage à l'autre avec la souris, il est délié du premier et relié au deuxième. En revanche, si vous le déplacez avec les flèches du clavier, les liens ne sont pas modifiés.

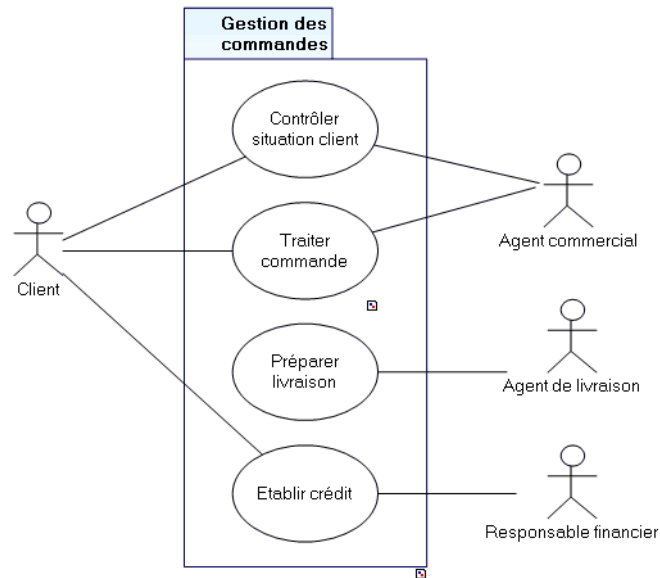
## Les participations

Vous pouvez indiquer quel acteur participe à chacun des différents cas d'utilisation.



Une participation indique qu'un acteur joue un rôle dans un cas d'utilisation.


### Exemple de participation




- L'agent commercial intervient dans le traitement d'une commande et dans le contrôle de la situation du client ;
- L'agent de livraison intervient dans la livraison ;
- Le responsable financier intervient dans l'établissement des crédits, etc.

### Créer une participation


Pour créer une **participation** dans le diagramme de cas d'utilisation :



1. Cliquez sur le bouton **Participation**  de la barre d'objets.
2. Cliquez sur l'acteur concerné, et faites glisser la souris jusqu'au cas d'utilisation, avant de relâcher votre pression. Une fenêtre apparaît.
3. Indiquez le nom de la participation et indiquez si l'acteur en est l'initiateur.


 Il est possible de préciser le début du cas d'utilisation en activant la case à cocher **Initiateur** dans la fenêtre de propriétés de la participation correspondante. Une flèche apparaît sur le trait représentant la participation.

4. Cliquez sur **OK**.

Le lien représentant la participation apparaît dans le diagramme.

 **Une participation est représentée par un lien mais il s'agit d'un objet, avec des propriétés qui lui sont propres.**

 La bobine  n'est pas utilisée pour créer des participations. Elle sert à créer certains types de liens comme ceux entre Paquetage et les autres objets.

 En cas d'erreur, vous pouvez supprimer un objet en cliquant avec le bouton droit sur cet objet, et en sélectionnant la commande **Supprimer** dans le menu contextuel. Vous pouvez de même supprimer un lien en cliquant avec le bouton droit sur le lien, et en sélectionnant la commande **Supprimer** ou **Délier** dans le menu contextuel du lien.

## Multiplicités d'une participation

Sur une participation, il est possible de préciser la multiplicité :

- De l'acteur, afin d'indiquer que plusieurs instance de l'acteur sont impliquées dans une même instance du cas d'utilisation (exemple : les participants à une réunion).
- Du cas d'utilisation, afin d'indiquer qu'une même instance d'acteur intervient dans plusieurs instances du cas d'utilisation (exemple : un agent commercial qui traite plusieurs commandes d'un même client).

Pour définir les multiplicités sur une participation :

1. Sélectionnez la participation et affichez ses **Propriétés**.
2. Dans la page de propriétés, cliquez sur la liste déroulante puis sur **Caractéristiques**.

Une première section vous permet de définir la multiplicité de l'acteur, une seconde section permet de définir celle du cas d'utilisation.

Une fois définie, la multiplicité apparaît dans le diagramme.

---

## Les associations entre cas d'utilisation : extension et inclusion

Lorsque la taille du système à décrire est importante, il est utile d'avoir des mécanismes de représentation adaptables au niveau de détail désiré. C'est ce que permettent les associations entre cas d'*utilisation*.

Lorsqu'un cas d'utilisation comprend trop de possibilités et d'exceptions, ces dernières sont représentées à part dans des extensions du cas d'utilisation standard.

### Relation d'inclusion

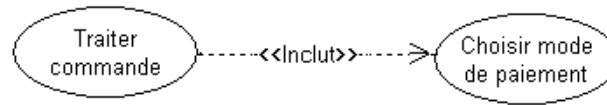
Un cas d'utilisation peut être sollicité automatiquement à la suite d'un autre, par exemple, la validation d'une commande inclut obligatoirement la sélection d'un moyen de paiement.

Pour indiquer qu'un cas d'utilisation en inclut un autre :

1. Dans le diagramme de cas d'utilisation, cliquez sur le bouton **Lien** 

2. Cliquez sur le cas utilisateur, par exemple "Traiter commande" et faites glisser la souris jusqu'au cas utilisé, par exemple "Choisir mode de paiement", avant de relâcher votre pression.
3. Sélectionnez le type de lien "Cas inclus" et cliquez sur **OK**.

Le lien apparaît dans le dessin accompagné du mot "Inclut".



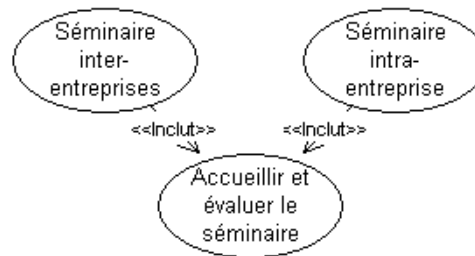
### Exemples d'inclusion

Dans un organisme de formation, les cas d'utilisation :

- Séminaire inter-entreprises (dont les participants viennent de plusieurs entreprises différentes)
- Séminaire intra-entreprise (dont les participants viennent tous de la même entreprise)

peuvent avoir en commun le cas d'utilisation :

- Accueillir et évaluer le séminaire



Dans une entreprise commerciale, le cas d'utilisation :

- Passer une commande

peut réutiliser les cas d'utilisation suivants :

- Fournir des informations sur le client
- Passer un ordre de production
- Proposer un mode de paiement

### Relation d'extension

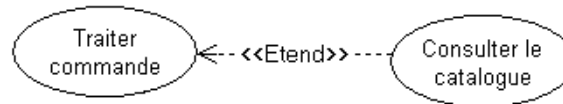
Un cas d'utilisation peut entraîner l'exécution d'un autre. Contrairement à l'inclusion qui est automatique, l'extension est optionnelle.

Pour indiquer qu'un cas d'utilisation est l'extension d'un autre :

1. Dans le diagramme de cas d'utilisation, cliquez sur la flèche associée au bouton **Lien** et cliquez sur **Extension**.

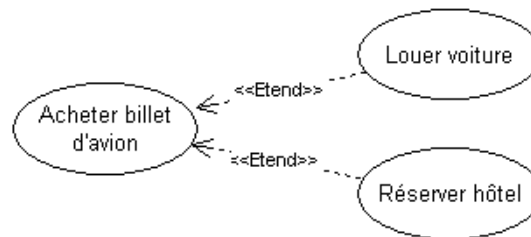
2. Cliquez dans un cas d'utilisation, par exemple "Consulter le catalogue" et faites glisser la souris jusqu'au cas étendu, par exemple "Traiter commande" avant de relâcher votre pression.  
La fenêtre de création d'une extension apparaît. Vous pouvez définir une contrainte ou un point d'extension.
3. Cliquez sur **OK**.

Le lien apparaît dans le diagramme accompagné du mot "Etend".



### **Exemple d'extension**

L'achat d'un billet d'avion peut être complété par la réservation d'un hôtel ou la location d'une voiture.



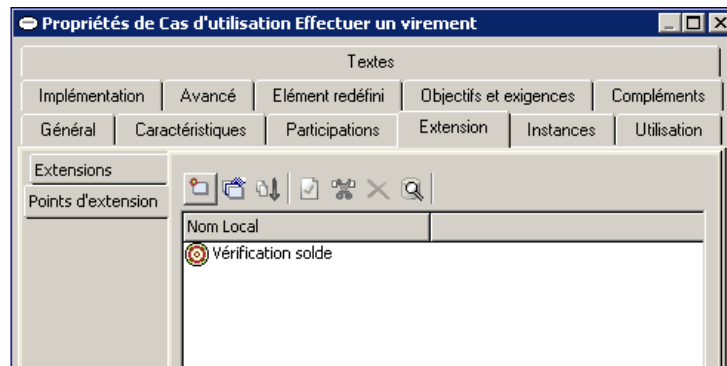
### **Point d'extension**

L'extension peut intervenir à un point précis du cas étendu. Ce point est appelé point d'extension.


Pour créer un point d'extension sur le cas étendu :

1. Affichez la fenêtre de propriétés de l'extension.
2. Sélectionnez la page **Caractéristiques**.
3. Sous la section **Point Extension**, cliquez sur **Ajouter**.  
La fenêtre de recherche apparaît.

4. Sélectionnez le point d'extension voulu et cliquez sur **Relier**.  
Le point d'extension apparaît dans la fenêtre de propriétés de l'extension.

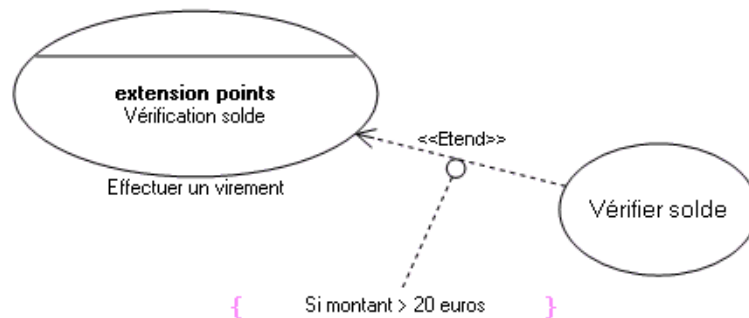


Un point d'extension peut être associé à une **contrainte** qui indique le moment où l'extension intervient. Vous pouvez ajouter une contrainte lors de la création de l'extension, ou ultérieurement, dans la fenêtre de propriétés du lien d'extension.


 Une contrainte est une déclaration qui établit un contrôle ou une règle de gestion impliquant généralement plusieurs classes.

### Exemple de point d'extension

L'exemple suivant présente le cas d'utilisation d'un virement bancaire ; au delà de la somme de 20 euros, la vérification de solvabilité du client est déclenchée.

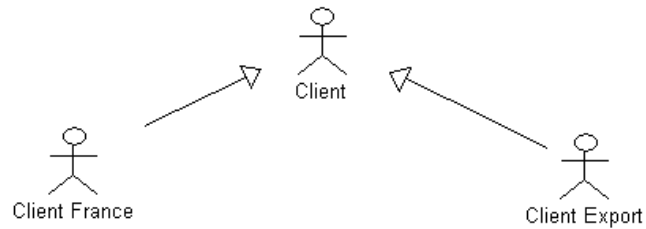


## Les généralisations

 Une généralisation représente une relation d'héritage entre une classe générale et une classe plus spécifique. La classe spécifique est cohérente avec la classe plus générale et en hérite ses caractéristiques et son comportement. Elle comporte cependant des informations supplémentaires. Toute instance de la classe spécifique est aussi une instance de la classe générale.


La notion de *généralisation*, initialement utilisée pour les classes, a été étendue à d'autres concepts d'UML comme acteur et cas d'utilisation.

Exemples de généralisation entre acteurs :

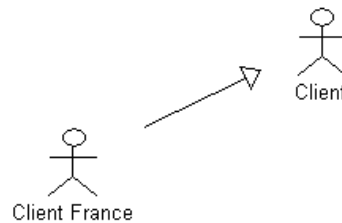


L'acteur "Client" peut être spécialisé en France et à l'Export.

Pour créer une généralisation entre acteurs dans un diagramme de cas d'utilisation :

- 1 Cliquez sur le bouton  et tirez le lien en partant de l'acteur particulier (Ex : Client France) vers l'acteur général (Ex : Client).

La généralisation apparaît dans le dessin.



☛ Vous pouvez créer de la même manière une généralisation entre deux cas d'utilisation.

Lors de la création d'une deuxième généralisation, une fenêtre vous propose de réutiliser la généralisation existante si elle porte sur le même sujet.

---

## Les interfaces

Il est possible de compléter la description d'un cas d'utilisation ou d'un acteur par la description des *interfaces* par lesquelles il communique avec son environnement.



## Créer une interface

Pour créer une interface dans un diagramme de cas d'utilisation :

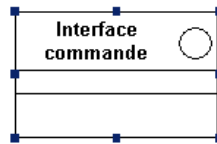
1. Dans la barre d'objets du diagramme, cliquez sur le bouton **Interface**



☛ Si le bouton **Interface** n'apparaît pas dans la barre d'objets, cliquez sur **Affichage > Vues et détails** et dans la fenêtre qui s'affiche, cochez la case **Classes**.

2. Cliquez sur le plan de travail du diagramme.
3. Dans la fenêtre qui s'ouvre, saisissez le nom de l'interface et cliquez sur le bouton **Ajouter**.


L'interface apparaît sur le diagramme.



## Relier une interface à un cas d'utilisation

Lorsque vous reliez une interface à un cas d'utilisation, vous devez préciser s'il s'agit d'une interface supportée ou d'une interface requise par le cas d'utilisation.

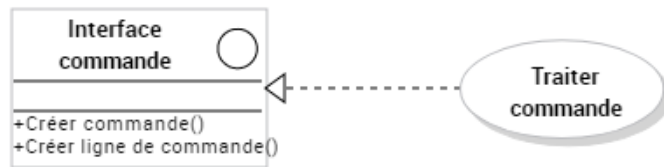
Pour préciser le type de lien entre une interface et un cas d'utilisation :

1. Cliquez sur le bouton **Lien**  et tirez le lien en partant du cas d'utilisation (Ex : Traiter Commande) vers l'interface (Ex : Interface Commande). Une fenêtre apparaît.
2. Sélectionnez le type du lien à créer :
  - Interface requise
  - Interface supportée
3. Cliquez sur **OK**.

Le lien apparaît dans le dessin.



Vous pouvez préciser quelles sont les opérations que ce cas d'utilisation peut réaliser par l'intermédiaire de cette interface.



- ✓ Pour plus de détails sur les interfaces requises et supportées, voir aussi [Relier les interfaces aux autres objets](#).

# LE DIAGRAMME DE CLASSES



Un diagramme de classes permet de représenter la structure statique d'un système, en particulier les types d'objets manipulés dans le système, leur structure interne et leurs relations. Un diagramme d'objets montre des exemples illustrant ce diagramme de classes.

La spécification des diagrammes de classes est souvent considérée comme la partie la plus importante dans la modélisation d'un système d'information. Les points abordés dans ce chapitre sont :

- ✓ [Présentation du diagramme de classes](#)
- ✓ [Créer un diagramme de classes](#)
- ✓ [Les classes](#)
- ✓ [Les attributs](#)
- ✓ [Les opérations](#)
- ✓ [Les signaux](#)
- ✓ [Les associations](#)
- ✓ [Les généralisations](#)
- ✓ [Spécifier les interfaces](#)
- ✓ [Spécifier les dépendances](#)
- ✓ [Spécifier des classes paramétrées](#)
- ✓ [Les contraintes](#)
- ✓ [Le diagramme d'objets](#)

# PRÉSENTATION DU DIAGRAMME DE CLASSES

Un diagramme de classes permet de représenter la structure statique d'un système, en particulier les types d'*objets* manipulés dans le système, leur structure interne et leurs relations.



*Un objet est une entité avec une identité et des frontières clairement définies dont l'état et le comportement sont encapsulés. Son état est défini par les valeurs de ses attributs et de ses liens avec d'autres objets. Son comportement est représenté par ses opérations et ses méthodes. Un objet est une instance de classe.*

Exemples d'objets :

- Objets de gestion :
  - Jacques Dupond, Pierre Durand, Paul Smith sont des instances de la classe "personne".
  - Les commandes no 10533 et 7322 sont des instances de la classe "commande".
  - Ecran SPD-1730 est une instance de la classe "article".
- Objets techniques utilisés pour la programmation :
  - Dlg\_Order\_Create, Dlg\_Customer\_Query sont des instances de la classe "fenêtre".
  - Str\_Customer\_Name, Str\_Product\_Comment sont des instances de la classe "chaîne".

Modéliser les données consiste à identifier les classes considérées d'intérêt pour représenter l'activité de l'entreprise, et à définir les associations qui existent entre elles.

Il faut que les classes et les associations qui constituent le diagramme de classes associé à un secteur de l'entreprise suffisent à le décrire complètement sur le plan sémantique.

En d'autres termes, on doit pouvoir décrire l'activité de l'entreprise en utilisant seulement ces classes et associations.

Ceci n'implique pas que, pour chaque mot ou verbe utilisé pour cette explication, il y ait un correspondant direct dans le diagramme de classes. Il s'agit de pouvoir traduire ce que l'on veut exprimer, au travers des classes et des associations.

La spécification des diagrammes de classes est souvent considérée comme la partie la plus importante dans la modélisation d'un système d'information.

Un diagramme d'objets montre des exemples illustrant ce diagramme de classes.

En particulier, il est possible d'illustrer un diagramme de classes par le diagramme d'objets correspondant dans un même dessin.

## Le diagramme de classes : synthèse

Un diagramme de classes inclut :

- Des classes, qui représentent les concepts de base (client, compte, produit, etc.).
- Des associations, qui définissent les relations entre les différentes classes.
- Des attributs, qui décrivent les caractéristiques des classes et, dans certains cas, des associations.
- Des opérations, qui peuvent être effectuées sur les objets de la classe.

☛ *Les opérations ne sont pas prises en compte par les outils de **HOPEX Information Architecture** (synchronisation, génération etc.).*

Le diagramme de classes est complété avec la définition des multiplicités.

A la fin de ce guide, un glossaire rappelle la définition de chacun des concepts utilisés.

---

## Créer un diagramme de classes

Un diagramme de classe se crée à partir d'un paquetage.

Pour créer un diagramme de classes :

- 】 Cliquez sur l'icône du paquetage concerné et sélectionnez **Nouveau > Diagramme de classes.**

Le diagramme créé s'ouvre dans la fenêtre d'édition.

Un diagramme de classes peut décrire un paquetage, un cas d'utilisation, une classe ou une instance.

# LES CLASSES

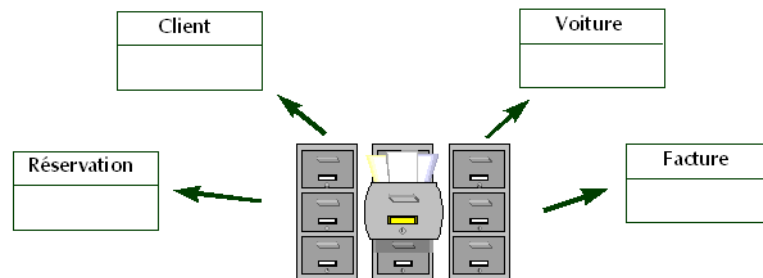
- ✓ Définition d'une classe
- ✓ Créer une classe
- ✓ Propriétés d'une classe
- ✓ Stéréotype d'une classe

## Définition d'une classe

Une *classe* est décrite par une liste d'attributs et d'opérations.

Une classe est reliée à d'autres classes via des *associations*. L'ensemble des classes et des associations constitue le noyau d'un diagramme de classes.

Nous pouvons illustrer la notion de classe en comparant les classes à des fiches dans des tiroirs.



Les classes peuvent être des objets techniques utilisés pour la programmation.

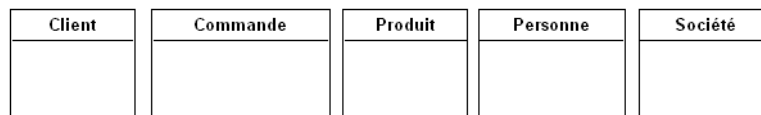
Exemples : fenêtre, rectangle, chaîne, tableau, etc.

Les classes peuvent représenter des objets techniques utilisés dans l'industrie.

Exemples : Alarme, Capteur, Zone

Les classes peuvent également représenter des objets de gestion.

Exemples : client, commande, produit, personne, société, etc.




Une classe peut aussi être la traduction d'un processus comme "Valider demande client" ou l'implémentation d'une règle de gestion fonctionnelle comme "Cohérence comptes analytiques".

☛ A la fin de ce guide, un glossaire rappelle la définition de chacun des concepts utilisés.

---


## Créer une classe

Pour créer une classe :

1. Dans le diagramme de classes, cliquez sur le bouton **Classe**  de la barre d'insertion d'objets.
2. Cliquez sur le plan de travail du diagramme.  
La fenêtre **Ajout d'une classe** s'ouvre.
3. Saisissez le **Nom** de la classe et cliquez sur le bouton **Ajouter**.  
La classe est alors posée dans le diagramme.

☛ Dans les exemples présentés dans ce guide, les noms des objets incluent, entre autres, des blancs, des majuscules et des caractères accentués. Il est important de noter que, lorsqu'il y a, pour un outil de génération qui utilise les spécifications effectuées avec **HOPEX UML**, des restrictions sur les caractères autorisés et sur la longueur des noms, il est préférable de se conformer à ces règles dès la spécification avec **HOPEX UML**.

Vous pouvez créer plusieurs classes à la suite sans revenir à la barre d'outils en double-cliquant sur le bouton **Classe**.

Pour revenir ensuite au mode normal, cliquez sur la flèche .

Vous pouvez utiliser partout le nom complet d'une classe en indiquant le paquetage auquel elle appartient séparé par deux fois deux points.

Exemple :

```
Entreprise::Gestion Commerciale::Client
```

Si l'un des paquetages cités n'existe pas, il est automatiquement créé et relié à la classe.

## Retrouver une classe existante

Pour retrouver une classe existante:

1. Dans la fenêtre **Ajout d'une classe**, sélectionnez la commande **Lister** à l'aide de la flèche.  
La liste des classes apparaît.
2. Sélectionnez la classe qui vous intéresse et cliquez sur **OK**.  
Le nom de la classe sélectionnée apparaît dans la fenêtre d'ajout d'une classe.
3. Cliquez sur **Ajouter**.  
La classe apparaît dans le diagramme.

---

## Propriétés d'une classe

Les propriétés affichées varient en fonction du stéréotype de la classe.

Pour ouvrir la fenêtre de propriétés d'une classe :

- 1 Sélectionnez la classe et cliquez sur le bouton **Propriétés** associé à la fenêtre d'édition.  
Plusieurs pages permettent de définir les propriétés d'une classe.

## Page Caractéristiques

La page **Caractéristiques** permet de saisir différentes caractéristiques de la classe :

- Son **Nom**, que vous pouvez modifier.

☺ Vous pouvez également modifier le nom d'une classe en cliquant directement sur son nom dans le dessin.

- Le détenteur de la classe (par exemple le paquetage).
- La **Visibilité** de la classe (par rapport au paquetage) :
  - "Publique" : la classe est visible par tout élément situé à l'extérieur du paquetage. C'est la visibilité sélectionnée par défaut.
  - "Protégée" : la classe est visible par les éléments héritiers ou importateurs et amis.
  - "Privée" : la classe est seulement visible par les éléments importateurs et amis.
  - "Non spécifiée".

📖 Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Son **Stéréotype** : voir [Stéréotype d'une classe](#).
- Un **Commentaire** : les commentaires permettent de compléter le diagramme lorsque des détails utiles n'apparaissent pas dans le dessin. Ces commentaires sont repris dans le document qui décrit le diagramme de classes.



Les autres caractéristiques que vous pouvez préciser sont l'abstraction, la persistance et l'activité :

- Une classe **Abstraite** n'a pas d'instance. Elle n'est utilisée que pour mettre en commun des opérations ou des attributs communs à ses sous-classes.
- La **Persistance** précise si cette classe doit être conservée dans le temps ou si elle ne vit que le temps du traitement en mémoire dans l'ordinateur.
- Les instances d'une classe **Active** sont capables de déclencher des flux de contrôle sans qu'il y ait d'intervention de l'utilisateur.

Ex : Une instance de la classe imprimante peut envoyer un message "Plus de papier disponible" sur l'écran de l'administrateur du réseau.

- Une classe **Racine** est une classe ne possédant pas de sur-classe dans l'arborescence des généralisations de classe.
- Une classe **Feuille** est une classe ne possédant pas de sous-classe dans l'arborescence des généralisations de classe.

Vous pouvez également préciser les Paramètres d'une classe paramétrée (pour C++).

☛ Voir [Pour spécifier une classe paramétrée](#) : pour plus de détails.

## Autres pages de propriétés

D'autres pages vous permettent de définir ou de visualiser :


- Les attributs d'une classe (voir [Les attributs](#))
- Les opérations d'une classe (voir [Les opérations](#))
- Les classes associées (voir [Les associations](#))
- Les instances d'une classe (voir [Le diagramme d'objets](#))
- Les éléments redéfinis

---

## Stéréotype d'une classe

Un stéréotype est un type d'élément de modélisation qui permet d'étendre la sémantique du métamodèle. Les stéréotypes doivent être basés sur des types ou des classes existantes dont ils reprennent la structure. D'autres stéréotypes peuvent être créés par l'utilisateur.

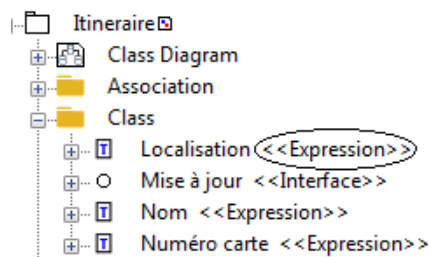
Les stéréotypes disponibles pour une classe sont :

- **Acteur** : représente le rôle joué par quelque chose ou quelqu'un se trouvant dans l'environnement du système étudié.
- **Auxiliaire** : classe qui supporte une autre classe centrale ou fondamentale, généralement en implémentant un flux logique ou de contrôle secondaire.
- **Classe d'implémentation** : permet de caractériser les classes qui sont nécessaires à l'implémentation physique du système.
- **Classe Meta** : classe dont les instances sont des classes. En règle générale, les méta-classes sont utilisées pour construire des méta-modèles.
- **Contrôle** : permet de caractériser les classes effectuant des traitements internes au système. Ceux-ci nécessitent généralement le concours de plusieurs classes.
- **Entité** : permet de caractériser des classes passives qui ne génèrent aucune interaction par elles-mêmes. Elles peuvent participer à plusieurs cas d'utilisation et survivent généralement à une interaction unique. Elles représentent des objets partagés entre les différents acteurs qui les manipulent.
- **Énumération** : type de données contenant une liste de valeurs tabulées.
- **Expression** : expressions de types de données complexes basés sur des types.
- **Focus** : classe qui définit le flux logique ou de contrôle principal pour la ou les classes auxiliaires qui la supportent.
- **Frontière** : permet de caractériser les classes qui sont en prise directe avec l'environnement du système. Les interfaces hommes-machines en font partie.
- **Interface** : une interface est constituée d'un ensemble d'opérations qui décrivent le comportement d'un élément. En particulier, une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.  
 *Ce sont des interfaces entre les différents composants du système informatique. Ce ne sont pas des interfaces avec les utilisateurs du système qui sont du stéréotype frontière. Voir [Spécifier les interfaces](#) pour plus de détails.*
- **Opérateur** : représente un acteur humain qui interagit avec le système. Un opérateur interagit avec d'autres opérateurs et manipule des entités lorsqu'il participe à la réalisation d'un cas d'utilisation.
- **Opérateur externe** : un opérateur externe interagit directement avec des acteurs extérieurs au système.
- **Opérateur interne** : un opérateur interne interagit avec d'autres opérateurs et entités à l'intérieur du système.
- **PowerType** : méta-type dont les instances sont des sous-types d'un autre type.
- **Structure** : classe servant à décrire une structure utilisée dans les programmes.
- **Thread** : stéréotype utilisé dans l'implémentation d'un objet actif en tant que processus léger.
- **Type élémentaire** : permet de caractériser les types de données.

- **Utilitaire** : une classe de ce stéréotype regroupe des variables globales et des procédures utiles à la programmation décrites sous forme des attributs et opérations de cette classe.
- **Schema group** : classe décrivant un type d'élément XML, dont les sous-éléments forment un groupe.
- **XML Document Definition Root** : classe servant à décrire la structure d'un message échangé entre deux systèmes en utilisant la syntaxe du langage XML.

## Option d'affichage des stéréotypes

Une option vous permet d'afficher les stéréotypes dans la fenêtre de navigation des objets.



Pour activer cette option :

1. Ouvrez la fenêtre des **Options**.
2. Dans la partie gauche de la fenêtre des options, sélectionnez le dossier **Espace de travail**.
3. Dans la partie droite, cochez l'option **Afficher le stéréotype des objets UML dans le navigateur**.
4. Cliquez sur **OK**.

# LES ATTRIBUTS

- ✓ Définition d'un attribut
- ✓ Spécifier les attributs d'une classe
- ✓ Propriétés des attributs

## Définition d'un attribut

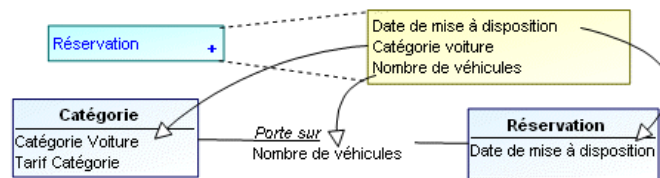
Un attribut est une propriété nommée d'une classe. C'est la donnée élémentaire mémorisée dans le système d'information de l'entreprise.

Exemples :

"Nom du client" (Attribut de la classe client).

"No client" (Identifiant de la classe client).

"Solde du compte" (Attribut de la classe compte).

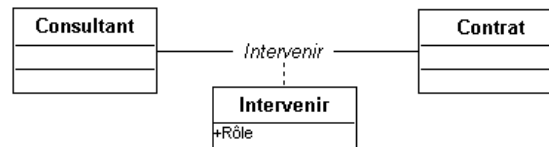


Les classes et les classes d'associations peuvent être caractérisées par des attributs.

Ces attributs ont pu, par exemple, être révélés par l'étude du contenu des messages qui circulent à l'intérieur de l'entreprise.

Un attribut est porté par une association quand sa valeur dépend de l'ensemble des classes participant à cette association.

Dans le diagramme présenté ci-après, le "rôle" qu'un "Consultant" a joué sur un "Contrat" dépend du consultant et du contrat, donc de l'association "Intervenir".



## Spécifier les attributs d'une classe

### Créer un attribut standard


Pour définir un attribut sur une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre des propriétés, cliquez sur la liste déroulante puis sur **Composants**.
3. Sous la section **Attributs**, cliquez sur **Ajouter un attribut**.  
Le nouvel attribut apparaît.
4. Cliquez sur le nom pour le modifier.

Pour chaque attribut, il est possible de préciser :


- Son **Type**, qu'il est possible d'exprimer sous forme d'une expression.

Ex : Integer.

 L'expression doit être conforme à la syntaxe UML. Voir [Signature d'une opération ou d'un signal](#) pour plus de détails.

 Voir aussi : [Type d'un attribut](#).

- Sa **Visibilité** :
  - "Public" : c'est la visibilité par défaut. L'attribut est visible par tous.
  - "Protégé" : l'attribut est visible par les héritiers de son paquetage ou ses amis.
  - "Privé" : l'attribut est visible par sa classe ou ses *amis*.

 Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Sa **Multiplicité**, c'est à dire le nombre de répétitions de cet attribut dans la classe.

### Créer un attribut calculé

Un attribut calculé est relié à une règle de calcul.

La règle de calcul définit les objets en entrée et en sortie ainsi que l'expression de la règle.

Les objets en entrée de la règle peuvent être des classes, des types ou vues de données. Les objets en sortie sont uniquement des classes.

Pour définir un attribut calculé sur une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre des propriétés, cliquez sur la liste déroulante puis sur **Composants**.
3. Sous la section **Attributs**, cliquez sur **Ajouter un attribut calculé**.  
Le nouvel attribut apparaît.
4. Ouvrez les propriétés de l'attribut pour spécifier :
  - les paramètres d'entrée et de sortie
  - la description de la règle

## Attributs hérités

Lorsqu'une généralisation existe entre une classe générale et une classe particulière, la classe particulière hérite des attributs de la classe générale.


- 1 Cliquez sur le bouton **Attributs hérités** pour visualiser les attributs hérités d'autres classes.

---

## Propriétés des attributs

Pour ouvrir la fenêtre de propriétés d'un attribut :

1. Dans la page de propriétés **Composants** de la classe détentrice, sous la section **Attributs** sélectionnez l'attribut en question.
2. Cliquez sur **Propriétés**.

 Le bouton  permet d'afficher les commandes cachées.

Dans la page **Caractéristiques**, vous pouvez préciser :

- Le **Type** de l'attribut sous la forme d'une **expression** (voir [Type d'un attribut](#)).
- S'il s'agit d'un attribut **statique** : indique si l'attribut peut prendre des valeurs spécifiques pour chacune des instances de la classe ou bien avoir une valeur qui caractérise l'ensemble de la classe.
  - "Oui" : l'attribut a une valeur qui caractérise l'ensemble de la classe. Par exemple, l'attribut "Longueur des numéros de téléphone" de la classe "Client France" est de 10 chiffres.
  - "Non" : l'attribut peut prendre une valeur différente pour chacune des instances de la classe. Par exemple, l'attribut "Numéro de téléphone" prend une valeur différente pour chaque instance de la classe "Client".
- S'il s'agit d'un attribut **Persistant**, c'est-à-dire si la valeur de cet attribut doit être conservée dans le temps ou si elle ne vit que le temps du traitement en mémoire dans l'ordinateur.
- Sa **Multiplicité**, c'est-à-dire le nombre de répétitions de cet attribut dans la classe.
- S'il est en **Lecture seule**, c'est-à-dire si sa valeur peut être modifiée après avoir été renseignée une première fois.
- S'il s'agit d'un **Attribut Calculé**, c'est-à-dire que sa valeur est déduite de la valeur d'un ou plusieurs autres attributs.
- La **Valeur initiale** de l'attribut, qu'il prendra lors de la création d'une instance de la classe.

## Type d'un attribut

Un type de données définit la nature des valeurs que peut prendre une donnée. Il peut être simple (entier, caractère, text, booléen, date...) ou plus élaboré et composé.

Un type est implémenté sous forme de classe.

Toute classe peut être utilisée pour typer un attribut ou un paramètre.

Exemple : Client, Commande, Fenêtre, Tableau

Les classes de stéréotype "Type élémentaire" sont créées uniquement afin de typer des attributs ou des paramètres. Elles sont invariables.

Exemples de types élémentaires :

Chaîne.

Entier.

Adresse Export.

Montant en devises.

Vous pouvez lister les types existants ou en créer de nouveaux.

☛ Les types proposés sont les classes détenues ou utilisées par le paquetage courant.

☛ Pour renseigner la structure d'un type, posez la classe correspondante dans le même diagramme ou dans un autre diagramme et activez la commande Propriétés de son menu contextuel.

# LES OPÉRATIONS

- ✓ Définition d'une opération
- ✓ Spécifier les opérations d'une classe
- ✓ Propriétés d'une opération
- ✓ Signature d'une opération ou d'un signal
- ✓ Paramètres d'une opération
- ✓ Méthodes d'une opération (comportement opaque)
- ✓ Le diagramme d'objets
- ✓ Exceptions d'une opération
- ✓ Afficher les attributs et les opérations d'une classe

---

## Définition d'une opération

Une opération est un service qui peut être demandé à un objet pour mettre en oeuvre un comportement défini. Une opération possède une signature qui permet de préciser les paramètres qui lui sont nécessaires.

Exemples :

"Calcul Age" (Opération de la classe client).

"Imprimer" (Opération de la classe dessin).

"Calculer échéancier" (Opération de la classe prêt).

☛ *Les opérations ne sont pas prises en compte par les outils de  
**HOPEX Information Architecture** (synchronisation, génération etc.).*

---

## Spécifier les opérations d'une classe

Pour spécifier les opérations d'une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la page de propriétés **Composants**, sous la section **Opérations**, cliquez sur **Nouveau** pour créer une opération ou sur **Relier** pour relier une opération existante.

L'opération apparaît dans les propriétés de la classe.

Vous pouvez préciser sa signature.

## Opérations héritées

Lorsqu'une généralisation existe entre une classe générale et une classe particulière, la classe particulière hérite des opérations de la classe générale.



1. Cliquez sur le bouton **Opérations héritées** pour visualiser les opérations héritées d'autres classes.



## Propriétés d'une opération

Pour ouvrir la fenêtre de propriétés d'une opération :

1. Dans la page de propriétés **Composants** de la classe détentrice, sous la section **Opérations**, sélectionnez l'opération en question.
2. Cliquez sur **Propriétés**.


 Le bouton  permet d'afficher les commandes cachées.

Vous pouvez indiquer pour chaque opération :

- Son **Stéréotype** afin de préciser son utilisation :
  - **Constructeur** : créer une instance de la classe.
  - **Destructeur** : détruire une instance de la classe.
  - **Itérateur** : parcourir les instances de la classe.
  - **Sélecteur** : effectuer une sélection parmi les instances de la classe.
- S'il s'agit d'une opération **Statique** : si l'opération qui peut prendre des valeurs spécifiques pour chacune des Instances de la classe ou bien avoir une valeur qui caractérise l'ensemble de la Classe.
- La **Concurrence** permet de préciser le comportement de l'opération lorsqu'elle est appelée plusieurs fois simultanément.
  - **Concurrente** : l'opération répond simultanément aux différents appels.
  - **Protégée** : l'opération répond au premier appel et rejette les suivants.
  - **Séquentielle** : l'opération répond successivement à chacun des appels.
- Si c'est une opération en **Consultation**, c'est-à-dire qui ne modifie pas l'état de l'objet.
- Si cette opération est **Polymorphe**, c'est-à-dire que des méthodes peuvent être redéfinies pour cette opération dans des sous-classes.

Les indications suivantes sont utilisées pour compléter la signature de l'opération.

- Le **Type expression** de l'opération.

 Le type expression d'une opération précise le type de la variable retournée par l'opération à la fin de son exécution.

- Sa **Signature**.

## Signature d'une opération ou d'un signal

La signature d'une opération ou d'un signal est constituée du nom de l'opération (du signal), de son type de retour, et de ses paramètres avec leurs types. La syntaxe UML standard est utilisée pour cette signature, elle est du type : Ope0 (Param0 : M-Bool) : M-Bool.

Il est possible de définir la signature :

- Soit dans la fenêtre de propriétés de l'opération ou du signal.
- Soit dans la fenêtre de propriétés de la classe à laquelle appartient l'opération, sous la section des **Opérations**.

Operations		
<div>Ajouter Réordonner Enlever InheritedOp</div>		
Nom Local ↑	Signature	Visibilité (UML)
EmpruntPar	EmpruntPar (UnAdherent : Inte	Public
MiseEnRayon	MiseEnRayon ()	Public

La signature qui est conservée inclut une référence au type : si le type est renommé, les signatures qui l'utilisent reflètent cette évolution.

## Syntaxe des signatures

La syntaxe standard des signatures est :

```
nomopération (paramètre1:expressiondutype1,paramètre2:expres  
siondutype2,...):expressiondutyperetour
```

Les noms comportant des blancs ou des caractères spéciaux doivent être placés entre apostrophes ('Nom du client'). Lorsqu'un nom contient une apostrophe, il faut dupliquer l'apostrophe : 'Nom de l' 'acheteur'

Exemples de signatures :

```
Déstocker (Produit0 : Entier(3), Quantité0 : Entier) :  
Booléen  
'Création de commande' ('Nom du client' : Client) : Byref  
Variable
```

Dans la spécification d'une signature, il est possible de préciser le paquetage auquel appartient une classe, séparé par deux fois deux points.

Exemple : `Entreprise::'Gestion Commerciale'::Client.`

La classe citée est reliée au paramètre ou au type de retour. Si elle n'existe pas, elle est créée. De même, les paquetages cités dans le parcours qui n'existent pas sont créés et reliés à la classe.

Si le paquetage n'est pas précisé, une fenêtre vous propose de choisir entre les éventuelles classes homonymes.

---

## Paramètres d'une opération

Un paramètre est la spécification d'une variable qui peut être modifiée, transmise ou retournée. Un paramètre peut préciser un nom, un type et une direction. Les paramètres sont utilisés pour les opérations, les messages et les événements.

Un argument est une valeur spécifique correspondant à un paramètre.

Dans la fenêtre de propriétés d'une opération, la section **Paramètres** vous permet de préciser :

- Son **Type** sous forme d'une expression Ex : Integer(5).
- Sa **Valeur par défaut** Ex : 0.
- Sa **Direction** : en entrée et/ou en sortie de l'opération.

Pour créer un *paramètre* sur une opération :

1. Affichez les propriétés de l'opération.
2. Sélectionnez la page **Caractéristiques**.
3. Sous la section **Paramètres**, cliquez sur **Nouveau**.  
La fenêtre de création d'un paramètre apparaît.
4. Saisissez le nom du paramètre et cliquez sur **OK**.

---

## Méthodes d'une opération (comportement opaque)

Une méthode - ou comportement opaque - est une représentation textuelle de l'implémentation d'une opération, d'une classe ou d'un composant. Elle spécifie l'algorithme ou la procédure qui produit les résultats d'une opération ou le comportement d'un élément.

Pour définir la méthode qui implémente une opération :

1. Ouvrez la page **Caractéristiques** de l'opération.
2. Sous la section **Méthode**, cliquez sur **Ajouter**.  
La fenêtre d'ajout d'une méthode apparaît.
3. Saisissez le nom de la méthode à créer ou recherchez une méthode existante.
4. Cliquez sur **OK**.

Pour saisir le corps du texte de la méthode qui implémente l'opération :

1. Ouvrez la page **Caractéristiques** de la méthode.
2. Définissez la méthode dans le cadre **Corps**.

Quand une classe possède plusieurs sous-classes, chaque sous-classe est susceptible d'implémenter l'opération sous forme d'une méthode différente.

La section **Méthode** présente la méthode relative à la classe sélectionnée.

---

## Conditions d'une opération

Vous pouvez définir les conditions d'une opération sous la forme de contraintes.

Les types de condition sont :



- La **Pré-Condition**, qui doit être vérifiée avant que l'opération ne s'exécute.
- La condition sur le **Corps**, qui doit être vérifiée lors de l'exécution de l'opération.
- La **Post-Condition** qui doit être vérifiée après l'exécution de l'opération.

Pour saisir une condition sur une opération :

1. Ouvrez la page **Conditions** de l'opération.
2. Sous la section **Conditions**, sélectionnez le type de condition :
  - pré-condition
  - condition sur le corps
  - post-condition
3. Cliquez sur **Nouveau**.  
La fenêtre d'ajout d'une contrainte apparaît.
4. Saisissez le nom de la contrainte à créer ou recherchez une contrainte existante.
5. Cliquez sur **OK**.

Pour saisir le corps de la condition :

1. Dans la fenêtre de propriétés de l'opération détentrice, sélectionnez la condition.
2. Cliquez sur le bouton **Propriétés**.

 Le bouton  permet d'afficher les commandes cachées.

3. Cliquez sur la page **Caractéristiques**.
4. Sous la section **Corps expression**, saisissez l'expression de la condition.

## Exceptions d'une opération

Si une condition n'est pas respectée, une exception est générée.

La section **Exceptions** vous permet de définir les messages d'erreur envoyés par l'opération lorsqu'une exception se produit et de préciser leur signature.

---

## Afficher les attributs et les opérations d'une classe

Pour modifier l'affichage des attributs et des opérations de la classe :

1. Faites un clic droit sur la ou les classes dont vous voulez afficher des attributs.
2. Sélectionnez **Formes et détails**.  
La fenêtre de sélection des éléments à afficher apparaît.
3. Dans l'arbre de gauche, cliquez sur **Attribut**.
4. Sélectionnez les attributs que vous voulez afficher.

Vous pouvez les afficher **Tous**, en afficher **Certains** que vous allez sélectionner dans la liste, ou n'en afficher **Aucun**.

Vous pouvez demander l'affichage de la **Visibilité**, du *type*, ... de chacun des attributs.



*Un type de données permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types de données sont implémentés sous forme de classe.*



*Vous pouvez cacher ou rendre visible le compartiment présentant les attributs dans le dessin de la classe, en activant ou désactivant la case à cocher devant "Affichage de".*

L'affichage des opérations est indiqué de la même manière, en choisissant **Opération** dans l'arbre qui vous est présenté.

# LES SIGNAUX

---

## Définition d'un signal

Un signal est un événement qui peut être invoqué explicitement. Un signal peut posséder des paramètres. Un signal peut être envoyé à un objet ou à un ensemble d'objets. Il peut être invoqué dans le cadre de la participation d'un acteur à un cas d'utilisation.

Un signal peut être émis ou reçu par une classe. Il peut être également émis par une opération à la suite d'une exception.

---

## Spécifier les signaux d'une classe

### Créer un signal émis ou reçu


Pour préciser quels sont les signaux qui peuvent être émis ou reçus par une classe :

1. Sélectionnez la classe et affichez ses propriétés.
2. Dans la fenêtre de propriétés, cliquez sur la liste déroulante puis sur **Compléments**.
3. Dans l'arbre qui vous est présenté, sélectionnez la branche **Signal émis** ou **Signal reçu**, puis cliquez sur le bouton **Ajouter**.
4. Indiquez le nom du signal et cliquez sur **OK**.

### Propriétés d'un signal

Pour ouvrir la fenêtre de propriétés d'un signal :


1. Dans la fenêtre de propriétés d'une classe, dans la page **Compléments**, sélectionnez le signal et cliquez sur **Propriétés**.

 Le bouton permet d'afficher les commandes cachées.


La fenêtre de propriétés du signal apparaît.

Vous pouvez indiquer pour un signal :

- Son **Stéréotype** afin de préciser son utilisation :
  - **Exception** : signal d'erreur généré lorsqu'une exception se produit pendant l'exécution d'une opération.
- Sa **Visibilité** par rapport au paquetage :
  - **Publique** : c'est la visibilité par défaut. Le signal est visible par tout élément situé à l'extérieur du paquetage.
  - **Protégée** : le signal est visible par les éléments héritiers ou amis.
  - **Privée** : le signal est visible par sa classe ou ses amis.

 Les amis d'une classe sont des classes autorisées à connaître les internes de cette classe. Il est possible de préciser les amis d'une classe dans l'onglet Compléments de la fenêtre de propriétés de la classe.

- Le **Type expression** du signal (voir *type expression*).

 Le type expression d'un signal précise le type de la variable retournée par le signal lors de sa réception par son destinataire.


Un signal peut être une demande de **Vote** envoyé à chacun des objets actifs pour leur demander s'il est possible d'effectuer une action, comme par exemple fermer une session Windows.


Un signal peut donner lieu à une **Diffusion** générale à tous les objets actifs.

## Paramètres d'un signal

Les *Paramètres* du signal sont renseignés sous l'onglet **Paramètres** de sa fenêtre de propriétés. Vous pouvez préciser :

- Son **Type** sous forme d'une expression. Ex : Integer(5).
- Sa **Valeur** par défaut. Ex : 0.
- Sa **Direction** : en entrée et/ou en sortie de l'opération.

 Un paramètre est la spécification d'une variable qui peut être modifiée, transmise ou retournée. Un paramètre peut préciser un nom, un type et une direction. Les paramètres sont utilisés pour les opérations, les messages et les événements.

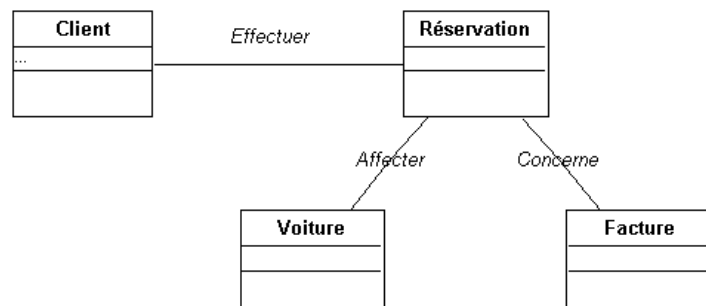
 Un argument est une valeur spécifique correspondant à un paramètre.

# LES ASSOCIATIONS

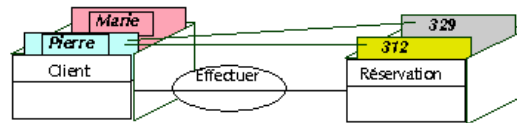
Une association est une relation existant entre deux classes.

Une association est binaire quand elle relie deux classes, ternaire quand elle en relie trois, etc.

Les associations peuvent être comparées à des liens entre des fiches.



Le dessin suivant permet de visualiser "en trois dimensions" les situations qu'un diagramme de classes permet de mémoriser.



Pierre et Marie sont des clients. Pierre a effectué les réservations numéros 312 et 329.

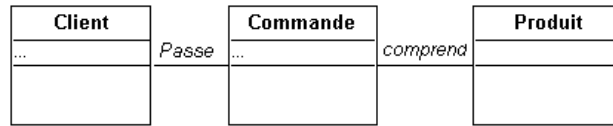
Un diagramme de classes doit permettre de mémoriser toutes les situations du contexte de l'entreprise.

☛ *Le diagramme ne doit pas permettre de représenter des situations irréalistes ou aberrantes.*



#### Exemples d'association

- Un client *pass*e une commande.
- Une commande *comprend* plusieurs produits.



- Une personne *travaille* pour une société.






- Une alarme *est déclenchée* par un capteur.

Un capteur *couvre* une zone.

Une fenêtre *affiche* une chaîne de caractères.

## Créer une association

Pour créer une *association*:

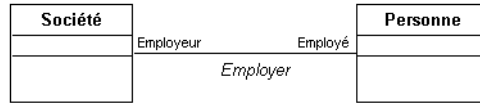
1. Dans le diagramme de classes, cliquez sur le bouton **Association**  de la barre d'objets.
2. Cliquez sur une des classes concernées, et faites glisser la souris jusqu'à la deuxième classe avant de relâcher votre pression.  
La fenêtre de création d'une association apparaît.
3. Saisissez le nom de l'association à créer.  
 Vous pouvez également sélectionner une association existante.
4. Cliquez sur **Ajouter**.  
L'association apparaît dans le diagramme sous forme d'un trait.  
 En cas d'erreur, vous pouvez supprimer un élément ou un lien en cliquant avec le bouton droit sur cet élément, et en sélectionnant la commande Supprimer dans son menu contextuel.

## Les rôles des associations

Il est possible de décrire les différents *rôles* joués par les classes dans les associations, et de préciser leur multiplicité et leur navigabilité.

Chaque extrémité d'une association permet de préciser le rôle joué par chaque classe dans l'association.

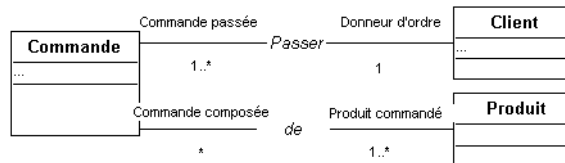
Visuellement, le nom du rôle se distingue du nom d'une association, car il est placé près de son extrémité. De plus, il apparaît en caractères droits, alors que le nom de l'association est en italique.



Lorsque deux classes sont reliées par une seule association, le nom des classes suffit souvent à caractériser le rôle ; nommer les rôles prend tout son intérêt lorsque plusieurs associations relient deux classes.

Exemples de rôles :

- Un client est le *donneur d'ordre* d'une commande.
- Une commande est *passée* par un client.
- Une commande est *composée* de produits.
- Un produit est *commandé*.



Une personne est un *employé* dans une société.

Une société est *l'employeur* de ces personnes.

Une alarme est *déclenchée* par un ou plusieurs capteurs.

Une zone est *couverte* par un capteur.

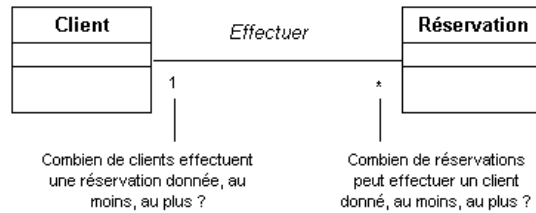
Une ou plusieurs chaînes de caractères sont *affichées* dans une fenêtre.

## Multiplicité d'un rôle

La multiplicité précise l'intervalle entre les valeurs minimum et maximum des cardinalités possibles pour un ensemble. On l'indique en particulier pour chacun des rôles que jouent les classes dans une association. Elle peut prendre les valeurs \*, 0..1, 1, 1..\*, 2..\*, 4..10, etc. La valeur proposée par défaut est \*.

La cardinalité est le nombre d'éléments contenus dans un ensemble.

La **multiplicité** exprime le nombre minimum et maximum d'instances d'une classe pouvant être reliées par l'association à chaque instance de l'autre classe.

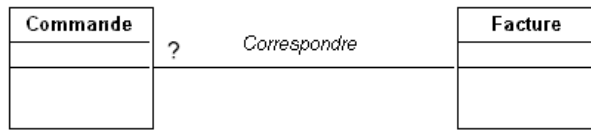


Les multiplicités usuelles sont "1", "0..1", "\*" ou "0..\*", "1..\*", et "M..N" où "M" et "N" sont des entiers :

- La multiplicité "1" indique qu'une et une seule instance de la classe est reliée par cette association à chaque instance de l'autre classe.
- La multiplicité "0..1" indique qu'au plus une instance de la classe peut être reliée par cette association à chaque instance de l'autre classe.
- La multiplicité "\*" ou "0..\*" indique qu'un nombre quelconque d'instances de la classe peuvent être reliées par l'association à chaque instance de l'autre classe.
- La multiplicité "1..\*" indique qu'au moins une instance de la classe est reliée par l'association à chaque instance de l'autre classe.
- La multiplicité "M..N" indique qu'au moins M instances de la classe et au plus N sont reliées par l'association à chaque instance de l'autre classe.

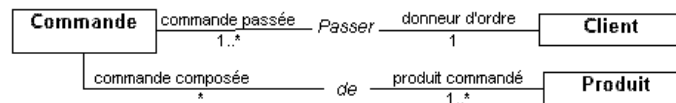
1	Un et un seul
0..1	Zéro ou un
M..N	De M à N (entiers naturels)
*	De zéro à plusieurs
0..*	De zéro à plusieurs
1..*	De un à plusieurs

L'exemple suivant va nous permettre d'illustrer la signification de chacune des multiplicités.



- 0..1 : A une commande correspond une facture au maximum ou aucune.
- \* : Aucune restriction n'est imposée sur le nombre de factures correspondant à une commande.
- 1 : A chaque commande correspond une facture et une seule.
- 1..\* : A chaque commande correspond une ou plusieurs factures.

Autres exemples de multiplicité :



- 1..\* : Un client peut passer une ou plusieurs commandes.
- 1 : Une commande est passée par un et un seul client.
- 1..\* : Une commande comprend un ou plusieurs produits.
- \* : Un produit peut faire partie de plusieurs commandes, ou d'aucune.
- 0..1 : Une personne travaille pour une société.
- 1..\* : Une alarme est déclenchée par un ou plusieurs capteurs.
- 1 : Un capteur couvre une et une seule zone.
- 1..\* : Une fenêtre affiche une ou plusieurs chaînes de caractères.

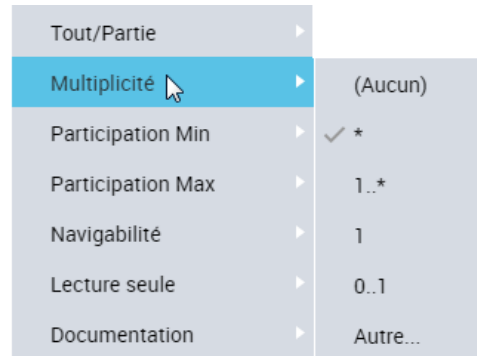
## Préciser la multiplicité d'un rôle

Pour préciser la multiplicité d'un rôle :

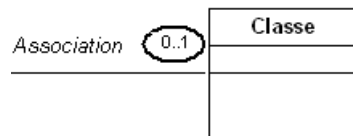
1. Faites un clic droit sur la partie du trait de l'association qui se trouve au plus près de la classe.

- Sélectionnez **Multiplicité** puis la valeur voulue.

☛ Si le menu affiché ne propose pas les multiplicités, vérifiez que vous avez bien cliqué sur le trait qui matérialise le rôle, et non sur l'association.



La multiplicité apparaît alors sur le rôle.



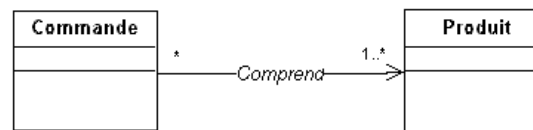
☺ Toutes les informations précisées avec le menu contextuel peuvent être consultées et modifiées dans la fenêtre de propriétés du rôle.

## Navigabilité d'un rôle

La navigabilité précise le sens dans lequel l'association entre deux classes peut être parcourue. Pour ne pas encombrer le dessin, on n'indiquera la navigabilité que lorsqu'elle n'a lieu que dans un seul sens.

Exemple de navigabilité :

- Il est nécessaire de connaître tous les produits contenus dans une commande.
- Par contre, il est rarement utile de connaître toutes les commandes qui portent sur un produit.



## Préciser la navigabilité d'un rôle

Pour indiquer qu'une association n'est navigable que dans un sens :

1. Faites un clic droit sur le rôle non navigable.
2. Sélectionnez **Navigabilité > Non**.

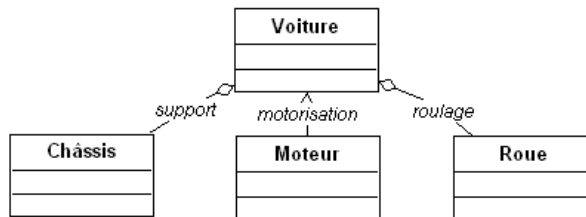
Une flèche représentant la navigabilité apparaît alors sur le rôle opposé.

---

## Agrégation d'un rôle

L'agrégation est une forme particulière d'association qui indique que l'une des classes contient l'autre.

Exemple : Une voiture comprend un châssis, un moteur et des roues.



## Préciser l'agrégation d'un rôle

Pour préciser l'agrégation d'un rôle :

1. Faites un clic droit sur le rôle.
2. Sélectionnez **Tout/Partie > Agrégat**.

☛ Si le menu affiché ne propose pas l'agrégation, vérifiez que vous avez bien cliqué sur le trait qui matérialise le rôle, et non sur l'association.

Un losange représentant l'agrégation apparaît alors sur le rôle.

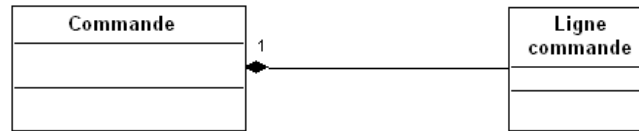
---

## Composition d'un rôle

La composition est une agrégation forte pour laquelle la durée de vie des composants coïncide avec celle du composé. Une composition est une agrégation immuable avec une multiplicité 1.

Exemple : Une commande est composée de plusieurs lignes de commande qui n'existent plus si la commande est supprimée.

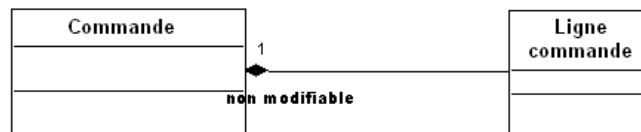
La composition est matérialisée par un losange noir.



## Rôle modifiable

La caractéristique **Lecture seule** permet de préciser si le rôle joué par une classe dans une association est modifiable après qu'il a été créé ou non. Par défaut, le rôle d'une classe dans une association est considéré comme modifiable.

Exemple : Une commande comprend une ligne de commande pour chacun des produits commandés. On ne peut plus changer ces différentes lignes de commande après l'enregistrement de la commande.



Il est possible d'indiquer si un rôle est modifiable à l'aide du menu contextuel du rôle ou dans la fenêtre de propriétés du rôle.

La caractéristique **Lecture seule** du rôle peut avoir les valeurs suivantes :

- **Ajout seul** : il est toujours possible de relier de nouveaux objets par cette association, mais il n'est pas possible de délier les objets déjà reliés.
- **Lecture seule** : les instances reliées ne peuvent plus être déliées. Il n'est pas possible non plus d'ajouter un nouveau lien.
- **Pas de restriction** : de nouvelles instances peuvent être reliées ou déliées à tout moment sans aucune contrainte.

## Ordre d'un rôle

Il est possible de préciser si un rôle est ordonné ou non. Par exemple, pour une commande, il peut être intéressant de mémoriser l'ordre de ses lignes de commande.

Pour spécifier qu'un rôle est ordonné :

1. Ouvrez la fenêtre de **Propriétés** du rôle.
2. Dans la page **Caractéristiques**, cochez la case **Ordonnée**.

---

## Propriété statique d'un rôle

Comme pour un attribut, il est possible de préciser si un rôle peut prendre des valeurs spécifiques pour chacune des instances de la classe ou bien s'il a une valeur qui caractérise l'ensemble de la classe :

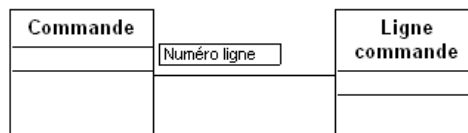
1. Ouvrez la fenêtre de propriétés du rôle.
2. Cliquez sur la page **Caractéristiques**.
3. Dans le champ **Statique**, sélectionnez :
  - "Oui" : pour que le rôle ait une valeur qui caractérise l'ensemble de la classe.
  - "Non" : pour que le rôle puisse prendre une valeur différente pour chacune des instances de la classe.

---

## Qualificatif d'un rôle

Un qualificatif est un attribut dont les valeurs partitionnent l'ensemble des objets reliés à un objet à travers une association.

Exemple : Une commande comprend plusieurs lignes de commande. Le numéro de ligne de commande peut servir de qualificatif pour identifier chacune des lignes.

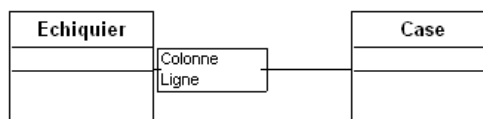


Pour définir un qualificatif :

1. Faites un clic droit sur le rôle et sélectionnez **Propriétés**. La fenêtre de propriétés du rôle s'ouvre.
2. Sélectionnez la page **Qualificatifs**.
3. Pour ajouter un nouveau qualificatif au rôle, cliquez sur **Ajouter**.
4. Saisissez le nom du qualificatif.
5. Cliquez sur **Ajouter**.

Plusieurs qualificatifs peuvent être nécessaires pour identifier de manière unique chacun des objets de la classe.

Par exemple, chaque case d'un échiquier est identifiée par son numéro de ligne et son numéro de colonne dans l'échiquier.





## Surcharger un rôle

Un rôle peut hériter d'un rôle défini au niveau supérieur. La surcharge permet de définir des propriétés supplémentaires sur un rôle hérité.

Pour surcharger un rôle :

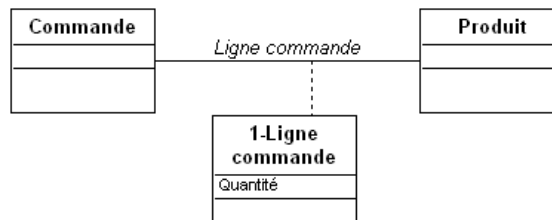
1. Ouvrez la fenêtre de propriétés du rôle.
2. Dans la fenêtre de propriétés, cliquez sur la liste déroulante puis sur **Caractéristiques**.
3. Sous la section **Rôles**, sélectionnez **Rôle surchargé**.
4. Cliquez sur **Ajouter**.  
La fenêtre de sélection apparaît.
5. Recherchez et sélectionnez le rôle en question.
6. Cliquez sur **OK**.

## Les classes d'association

Une classe d'association est une association qui possède aussi les propriétés d'une classe comme des attributs.

Il est utile de créer une classe d'association pour préciser des caractéristiques de l'association.

Par exemple, il est nécessaire de préciser la quantité de produit demandée pour chacune des lignes d'une commande.



Pour créer une classe d'association :

1. Créez une nouvelle classe.
2. A l'aide du bouton **Lien**, créez un lien entre la classe et l'association.  
La classe d'association est reliée à l'association par un trait pointillé.

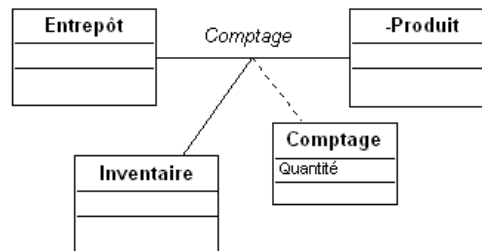
☛ Comme pour les classes standard, il est possible de cacher les compartiments et de retailler la classe d'association à l'aide de la commande **Affichage de son menu contextuel**.

---

## Définir une association "plus que binaire"

Certaines associations associent non pas deux, mais davantage de classes. Ces associations sont, en principe, rares.

Exemple : Lors d'un inventaire, une certaine quantité de produit a été comptée dans chaque entrepôt.



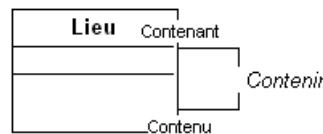
Pour créer une association ternaire :

1. Créez tout d'abord l'association entre deux des classes.
2. Cliquez sur le bouton **Rôle de l'association** .
3. Tirez un lien entre l'association et la troisième classe.

Vous pouvez ensuite créer la classe d'association éventuelle comme précédemment.

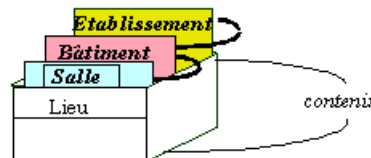
---

## Les associations réflexives



Certaines associations mettent en jeu plusieurs fois la même classe.

Une salle de classe, un bâtiment, un établissement scolaire sont tous des lieux.




Une salle de classe est contenue dans un bâtiment, lui-même contenu dans un établissement scolaire.

Une association réflexive porte deux fois sur la même classe.

## Créer une association réflexive

Pour créer une association réflexive :

1. Cliquez sur le bouton **Association**  de la barre d'outils.
2. Cliquez dans la classe concernée et faites glisser la souris en dehors de cette classe, puis revenez-y ; relâchez enfin votre pression.  
L'association réflexive apparaît sous forme d'un crochet.

☛ Dans le cas d'une association entre une classe et elle-même, il est indispensable de préciser les rôles afin de distinguer les liens correspondants dans le dessin.

# LES PARTIES

Dans un diagramme de classes, une partie représente un rôle joué par une instance d'une classe ou d'un composant lors de l'exécution d'une tâche.

Une partie appartient à une classe. L'appartenance est indiquée sur le lien de la partie.

Par exemple, ci-dessous, La Classe "Commande" est composée de la classe "Personne".

La partie est détenue par la classe "Commande" et référence la classe "Personne".



---

## Créer une partie entre deux classes

Une partie associe deux classes uniquement et elle est orientée.

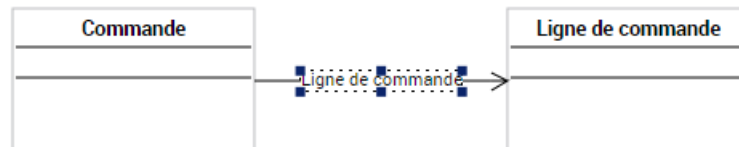
Pour définir une partie entre deux classes :

1. Dans la barre d'objets du diagramme de classes, cliquez sur **Partie**.
2. Tirez un lien de la classe détentrice vers la classe référencée.  
Le nom de la partie est défini automatiquement.

---

## Définir l'identifiant d'une classe à travers une partie

Dans l'exemple ci-dessous, l'identifiant de la classe "Ligne de commande" peut être défini à partir de la classe "Commande" à travers la partie "Ligne de commande".



Pour définir l'identifiant de la classe "Ligne de commande" :

1. Affichez les propriétés de la classe "Ligne de commande".

2. Sélectionnez la page **Identifiant**.
3. Faites un clic droit sur le dossier **Membres** et sélectionnez **Relier** > **Partie**.



4. Sélectionnez la partie proposée.
5. Cliquez sur **OK**.

## Multiplicités des classes associées

A travers les multiplicités vous pouvez indiquer le nombre minimum et maximum d'instances des classes reliées par la partie.

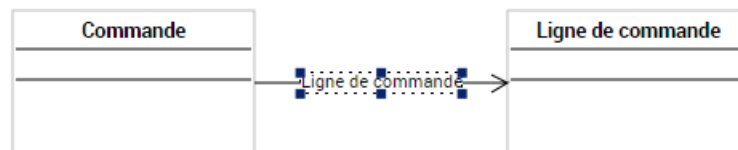
Exemple : 1 commande contient 1 ou plusieurs ligne(s) de commande.

### Multiplicité de la classe référencée par la partie

La multiplicité de la classe référencée est à indiquer sur le lien de la partie.

Pour définir la multiplicité de la classe référencée :

1. Faites un clic droit sur le lien de la partie.

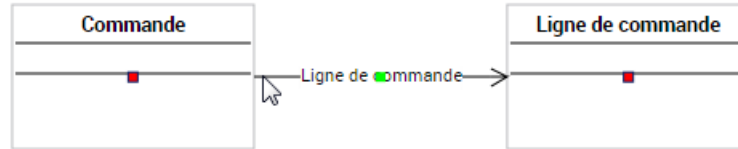


2. Sélectionnez **Multiplicité** puis la valeur souhaitée.

## Multiplicité de la classe détentrice de la partie

Pour définir la multiplicité sur la classe détentrice de la partie :

1. Faites un clic droit sur le rôle de la partie associé à la classe détentrice.





2. Dans le menu contextuel qui apparaît, sélectionnez **Multiplicité** puis la valeur souhaitée.

---

## Relations d'agrégation et de composition

Sur la partie qui relie deux classes vous pouvez définir une relation d'agrégation ou de composition.

 *L'agrégation est une forme particulière d'association qui indique que l'une des entités contient l'autre.*


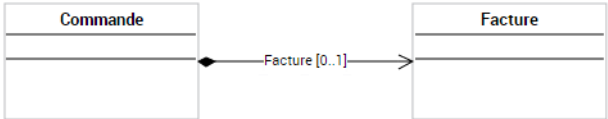

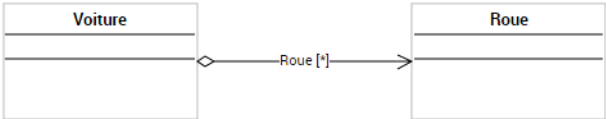

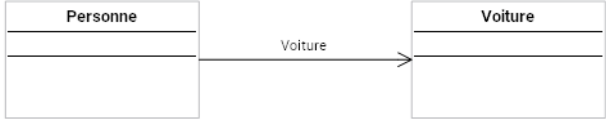
 *La composition est une agrégation forte pour laquelle la durée de vie des composants coïncide avec celle du composé. Une composition est une agrégation immuable avec une multiplicité 1.*

Pour définir un lien de composition ou d'agrégation entre les classes :

1. Faites un clic droit sur la partie.
2. Sélectionnez **Tout/Partie** puis la valeur souhaitée :
  - **Agrégat**
  - **Composé**

## Multiplicités associées

Le tableau ci-dessous présente les multiplicités associées automatiquement aux agrégations et compositions.

		Multiplicité correspondante	Exemple
Composition		1	 1, 1..*
Agrégation		0..1	 0..1, 1..*
Non renseigné		*	 *, *

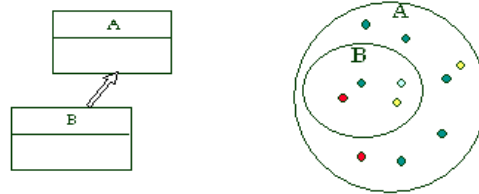
# LES GÉNÉRALISATIONS

Une généralisation représente une relation d'héritage entre une classe générale et une classe plus spécifique. La classe spécifique est cohérente avec la classe plus générale et en hérite ses caractéristiques et son comportement. Elle comporte cependant des informations supplémentaires. Toute instance de la classe spécifique est aussi une instance de la classe générale.

- ✓ Qu'est-ce qu'une généralisation
- ✓ Cas de plusieurs sous-classes
- ✓ Intérêt des sous-classes
- ✓ Héritage Multiple
- ✓ Créer une généralisation
- ✓ Discriminant

---

## Qu'est-ce qu'une généralisation



La classe A est une généralisation de la classe B. Cela suppose que tous les objets de la classe B sont aussi des objets de la classe A. Autrement dit, B est un sous-ensemble de A.

B est alors la sous-classe, A la super-classe.

Exemple A : Personne, B : Parisien.

B étant un sous-ensemble de A, les objets de la classe B "héritent" des caractéristiques de ceux de la classe A.

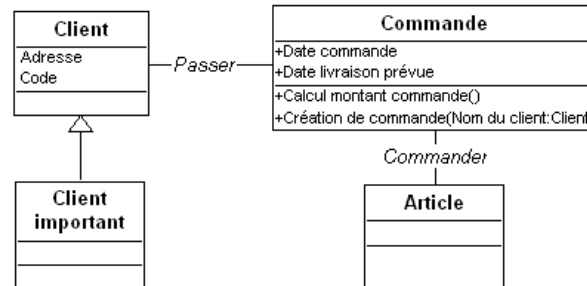
Il n'est donc pas nécessaire de décrire de nouveau pour la classe B :

- Ses attributs
- Ses opérations
- Ses associations



## Exemple

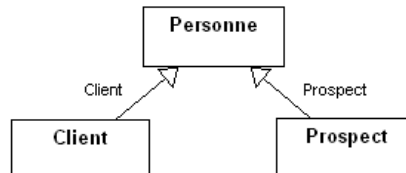
La classe "Client important" qui représente les clients dont le "C.A. sur les 12 derniers mois" dépasse 1 MF, peut être une spécialisation de la classe client (origine).



Dans l'exemple qui précède, les associations et les attributs spécifiés pour "Client" sont aussi valables pour "Client important".

Autres exemples de généralisations :

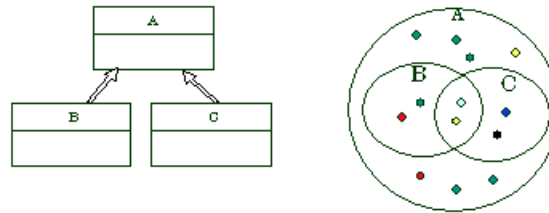
- Prospect et client sont deux sous-classes de "personne".



- Commande export est une sous-classe de la classe "commande".
- Personne physique et personne morale sont deux sous-classes de la classe "personne".
- Polygone, ellipse et cercle sont des sous-classes de la classe "forme".
- Chêne, orme, et bouleau sont des sous-classes de la classe "arbre".
- Véhicule à moteur, véhicule tout-terrain et véhicule amphibie sont des sous-classes de la classe "véhicule".
- Camion est une sous-classe de la classe "véhicule à moteur".

---

## Cas de plusieurs sous-classes

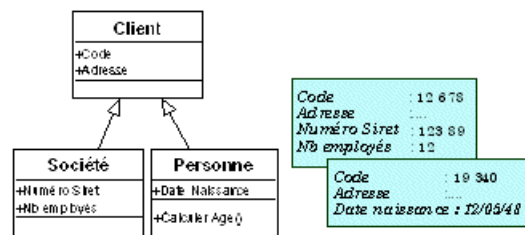


Plusieurs sous-classes d'une même classe :

- Ne sont pas forcément exclusives.
- Ne forment pas nécessairement une partition.

---

## Intérêt des sous-classes

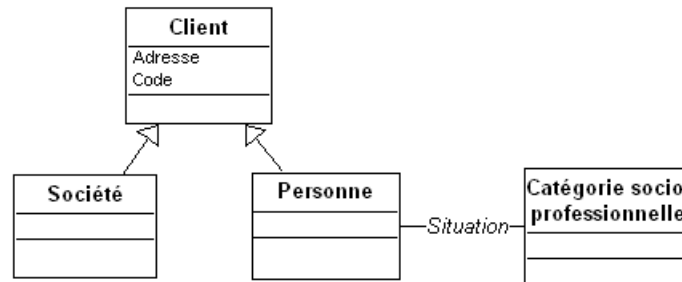


Une sous-classe hérite de tous les attributs, opérations et associations de sa super-classe, mais elle peut avoir des attributs ou des associations que ne possède pas sa super-classe.

Une sous-classe peut ainsi avoir des attributs spécifiques. Ceux-ci n'ont de sens que pour une sous-classe particulière. Dans l'exemple ci-dessus :

- Le "numéro de Siret" et le "nombre d'employés" n'ont de sens que pour une "société".
- La "date de naissance" est caractéristique d'une "personne", pas d'une "société".
- De même, il est utile de calculer l'"âge" d'une "personne". Cet attribut et cette opération n'ont généralement pas d'intérêt pour une "société".

Une sous-classe peut également avoir des *associations* spécifiques.



- Une "personne" entre dans une "catégorie socio-professionnelle" : "cadre", "employé", "commerçant", "agriculteur", etc. Cette classification n'a pas de sens pour une "société". (Il existe également une classification pour les entreprises, mais ce n'est pas la même que pour les personnes.)

## Héritage Multiple

Il est parfois utile de spécifier une classe ayant plusieurs super-classes. La sous-classe hérite alors de toutes les caractéristiques des deux super-classes. Cette possibilité doit être utilisée avec précaution.

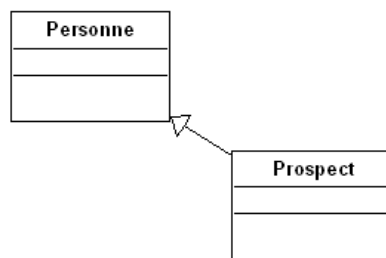
☛ L'héritage multiple n'est pas pris en compte pour la génération des tables.

## Créer une généralisation

Pour créer une généralisation :

1. Cliquez sur le bouton **Généralisation**  de la barre d'outils
2. Cliquez dans la sous-classe et faites glisser la souris jusqu'à la super-classe, avant de relâcher votre pression.

La généralisation apparaît dans le diagramme, matérialisée par une flèche.

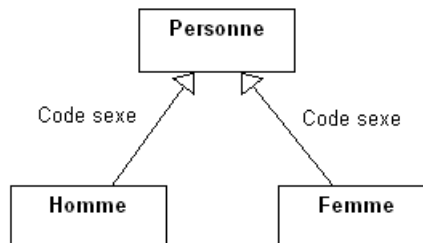


---

## Discriminant

Le discriminant est l'attribut d'une généralisation dont la valeur permet de répartir les objets entre les sous-classes associées à la généralisation.

Par exemple, l'attribut code-sexe permet de répartir les objets de la classe personne entre la sous-classe homme ou femme.



Vous pouvez définir le ou les discriminants dans la fenêtre de propriétés de la généralisation, sous la page **Discriminants**.

☛ Il est également possible de préciser si une généralisation est :

- *Disjointe* : Une instance ne peut pas appartenir simultanément à deux sous-classes de cette généralisation.
- *Complète* : Toutes les instances de la super-classe appartiennent à au moins une des sous-classes de cette généralisation.

## SPÉCIFIER LES INTERFACES

Une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur. L'interface est un stéréotype de classe.


Une interface est constituée d'un ensemble d'opérations qui décrivent le comportement d'un élément. En particulier, une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.

Ce sont des interfaces entre les différents composants du système informatique.

---

### Créer une interface

Pour créer une classe interface dans un diagramme de classes :

1. Dans la barre d'outils, sélectionnez le bouton **Interface** .
2. Cliquez sur le plan de travail du diagramme.
3. Dans la fenêtre qui apparaît, saisissez le nom de l'interface et cliquez sur le bouton **Ajouter**.  
La classe interface apparaît dans le diagramme.

Vous pouvez spécifier les opérations de l'interface comme pour n'importe quelle classe.


### Relier une interface à une classe


Lorsque vous reliez une classe à une interface, vous devez préciser s'il s'agit d'une interface requise ou d'une interface fournie par la classe.

Une interface requise est une interface nécessaire au fonctionnement de l'objet.

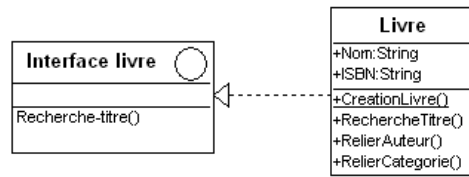
Une interface fournie est une interface mise à disposition par un objet à destination d'autres objets.

Pour relier une interface à une classe :

1. Cliquez sur le bouton **Lien** .
2. Effectuez le lien en partant de la classe pour aller vers l'interface.  
Une fenêtre apparaît.
3. Sélectionnez le type de lien à créer : interface fournie ou interface requise.

 D'autres types de liens, spécifiques aux classes, peuvent apparaître.



4. Cliquez sur **OK**.



## SPÉCIFIER LES DÉPENDANCES

Dans le diagramme de classes, pour indiquer qu'un paquetage référence une classe ou un autre paquetage :

1. Cliquez sur le bouton **Lien** 
2. Effectuez le lien en partant d'un paquetage vers le paquetage ou la classe qu'il référence.

 Le bouton **Vues**  vous permet de préciser les boutons que vous voulez voir apparaître dans la barre d'objets.

## SPÉCIFIER DES CLASSES PARAMÉTRÉES

Une classe paramétrée permet de définir des caractéristiques et un comportement modulable en fonction de la valeur de certains paramètres. Par exemple, une classe paramétrée peut être utilisée pour gérer des listes d'objets. Le paramètre sera alors le type d'objet que l'on veut gérer sous forme de liste. Ce type de classe est implémenté en particulier dans le langage C++.

Pour spécifier une classe paramétrée :

1. Ouvrez la fenêtre des **Propriétés** de la classe et sélectionnez la page **Caractéristiques**.
2. Vous pouvez saisir les paramètres et préciser éventuellement leur type.


Paramètres:

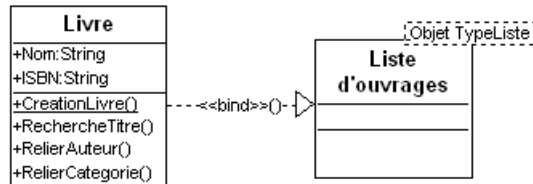


Objet TypeListe : int

Les paramètres de la classe s'affichent en haut à droite.

Pour relier une classe à une classe paramétrée :

1. Cliquez sur le bouton **Lien** .
2. Créez le lien en partant de la classe pour aller vers la classe paramétrée.





## LES CONTRAINTES

Une contrainte est une déclaration qui établit un contrôle ou une règle de gestion impliquant généralement plusieurs classes.

La plupart des contraintes impliquent les associations entre les classes.


Exemples de contraintes :

- La personne responsable d'un service doit appartenir à ce service.
- Toute commande facturée doit avoir été livrée auparavant.
- La date de livraison doit être postérieure à la date de commande.

Un capteur couvrant une zone ne peut déclencher qu'une alarme protégeant cette même zone.

Pour créer une contrainte dans le diagramme de classes :

1. Cliquez sur le bouton **Contrainte**  de la barre d'objets.



 *S'il n'est pas affiché, cliquez sur le menu **Affichage** > **Vues et détails** et cochez la case "Contraintes".*

2. Cliquez dans une des associations concernées par la contrainte, et faites glisser la souris jusqu'à la deuxième association, avant de relâcher votre pression.

La fenêtre d'ajout d'une contrainte s'ouvre.

3. Saisissez le nom de la contrainte puis cliquez sur **Ajouter**.

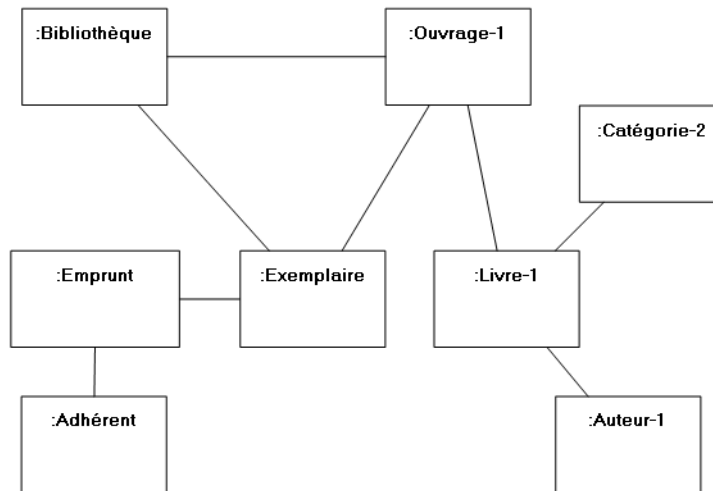
La contrainte apparaît dans le diagramme.

 *Vous pouvez relier une contrainte à d'autres classes ou associations à l'aide du bouton **Lien** .*

# LE DIAGRAMME D'OBJETS

Un diagramme d'objets, ou diagramme d'instances, contient des objets avec des valeurs exemples pour leurs attributs, et des liens. Il montre en détail l'état du système à un instant précis.

Vous pouvez créer le diagramme d'objets d'une classe, d'un composant, d'un paquetage ou d'un cas d'utilisation.



---

## Les objets

Un objet est une entité avec une identité et des frontières clairement définies dont l'état et le comportement sont encapsulés. Son état est défini par les valeurs de ses attributs et de ses liens avec d'autres objets. Son comportement est représenté par ses opérations et ses méthodes. Un Objet est une instance de Classe.


Exemples d'objets :

- Objets de gestion :
  - Jacques Dupond, Pierre Durand, Paul Smith sont des instances de la classe personne.
  - Les commandes no 10533 et 7322 sont des instances de la classe commande.
  - Ecran Sony SPD-1730, Compaq Deskpro 200 sont des instances de la classe article.
  - Dupond de Nemours, Burger King sont des instances de la classe société.
- Objets techniques utilisés pour la programmation :
  - Dlg\_Order\_Create, Dlg\_Customer\_Query sont des instances de la classe fenêtre.
  - Str\_Customer\_Name, Str\_Product\_Comment sont des instances de la classe chaîne.

☛ Les objets représentés dans un diagramme d'objets peuvent être des instances de classe, de paquetage, de cas d'utilisation, de composant ou de nœud, ce qui permet de définir des diagrammes de séquence au niveau de détail souhaité.

## Créer un objet (une instance)

Pour créer un objet :

1. Cliquez sur le bouton **Instance** .  
Vous pouvez créer des objets de différents types. La flèche située à droite du bouton offre un raccourci vers les types d'objets Classe ou Composant, plus fréquemment utilisés.
2. Puis cliquez sur le plan de travail du diagramme.  
La fenêtre d'ajout d'une instance s'ouvre.
3. Saisissez le **Nom** de l'instance.
4. Précisez si nécessaire le **Type d'instance**.
5. Cliquez sur **Ajouter**.

L'instance est posée dans le diagramme.

## Propriétés d'une instance

Pour ouvrir la fenêtre de propriétés d'une instance :

1. Sélectionnez l'instance en question et cliquez sur le bouton **Propriétés** de la fenêtre d'édition s'il n'est pas activé.

Plusieurs pages permettent de définir les propriétés d'une instance.

**Propriétés de instance1**

Caractéristiques

Nom Local: instance1

Défendeur: Package-1 (EN)

Visibilité (UML): Public

Stéréotype: Entité

Type instance: Instance de classe

Élément instancié: Exemplaire

Mot-clé:

Multiplicité:

Sous la page **Caractéristiques**, vous pouvez :

- Sélectionner le **Type** de l'instance (Acteur, Classe, etc.).
- Préciser de quelle **Classe**, quel **Acteur**, etc. cet objet est une instance.
- Indiquer un nom pour cette Instance.
- Préciser son **Stéréotype**.

## Valeur d'un attribut

Pour renseigner la valeur d'un attribut :

1. Affichez les propriétés de l'instance de la classe qui contient l'attribut.
2. Sélectionnez la page **Attributs**.
3. Dans la colonne correspondante, indiquez la valeur de l'attribut. Vous pouvez renseigner une valeur instanciée ou une valeur constante.
  - **Valeur instanciée** : cliquez dans cette colonne pour afficher la liste des instances possibles pour l'attribut sélectionné. Il s'agit de valeurs variables.
  - **Valeur** : cliquez dans la colonne et entrez la valeur de l'attribut.

---

## Les liens


Un lien entre objets représente une instance d'association entre deux objets.

Exemples de liens entre objets :

- La commande n° 10733 a été passée par Jacques Dupond.
- La commande 10733 comprend les produits Ecran Sony SPD-1730 et Compaq Deskpro 200.
- Mr Jacques Dupond travaille pour la société Dupond de Nemours.
- La fenêtreDlg\_Customer\_Query affiche la chaîne de caractères Str\_Customer\_Name.

## Créer un lien

Pour créer un lien :

1. Cliquez sur le bouton **Lien**  de la barre d'outils du diagramme.
2. Cliquez sur l'un des objets concernés, et faites glisser la souris jusqu'au deuxième objet, avant de relâcher votre pression.


Le lien apparaît dans le dessin.

Si un lien entre les deux objets existe déjà, une fenêtre s'ouvre pour vous permettre de choisir parmi les liens existants ou en créer un nouveau.

## Propriétés d'un lien

Pour ouvrir la fenêtre de propriétés d'un lien :

- Sélectionnez le centre du lien pour afficher ses **Propriétés**.


 Si vous ne cliquez pas au centre du lien, c'est la fenêtre Propriétés d'un des rôles qui va s'ouvrir.

Sous la page **Caractéristiques**, vous pouvez préciser :

- Le **Nom** du lien.
- Le **Stéréotype** du lien.
- L'**Association** correspondant au lien.
- La **Visibilité** du lien.
- Le **Paquetage** détenteur du lien.

Et sous la page **Rôle du lien** :



- Pour chaque **Instance** reliée par ce lien, le nom du **Rôle** et la **Multiplicité** de ce rôle.

 Parmi les associations proposées ne figurent que celles qui figurent entre les classes des deux objets.

## Propriétés d'un rôle

Pour ouvrir la fenêtre de propriétés d'un rôle :

1. Dans la fenêtre de propriétés d'un lien, sélectionnez la page **Rôle du lien**.
2. sélectionnez le rôle en question et cliquez sur **Propriétés**.

 Le bouton  permet d'afficher les commandes cachées.

La fenêtre de propriétés du rôle s'affiche.

Dans cette fenêtre vous pouvez préciser :

- Un **Nom** pour l'instance de rôle.
- Le **Rôle** de cette instance.
- La **Multiplicité** de l'instance de rôle.
- Pour cette instance de rôle, les valeurs des *qualificatifs* définis au niveau de la classe.

# LES DIAGRAMMES DE STRUCTURE ET DE DÉPLOIEMENT



Outre les diagrammes de classes et d'objets, les diagrammes structurels comprennent :

- ✓ Le diagramme de paquetages, qui permet d'organiser les éléments du modèle.
- ✓ Le diagramme de composants, qui met en évidence les relations de dépendance entre composants.
- ✓ Le diagramme de structure composite, qui décrit les interactions entre les composants et leurs parties.

# LE DIAGRAMME DE PAQUETAGES

Un diagramme de paquetages permet d'organiser les éléments de modélisation, de manière à assurer une partition des travaux de spécifications et de développement.

Un élément ne doit apparaître que dans un seul paquetage.

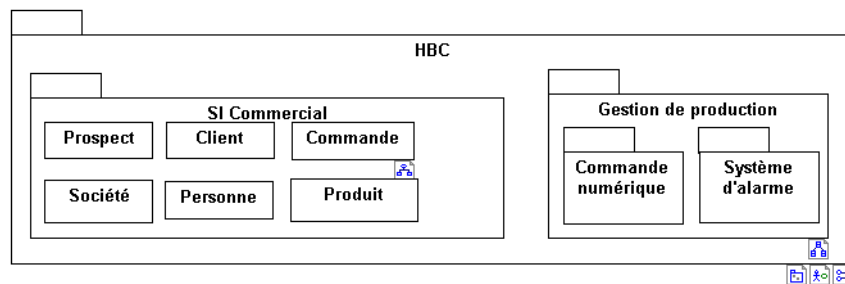
Le découpage en paquetage est généralement fait de manière à minimiser les interactions entre les différents paquetages.

## Exemple de diagramme de paquetages

Le paquetage "HBC" contient les paquetages "SI Commercial" et "Gestion de production".

Le paquetage "Gestion de production" se décompose en deux paquetages "Commande numérique" et "Système d'alarme".

Le paquetage "SI Commercial" contient les classes "Prospect", "Client", "Société", "Personne", "Commande" et "Produit".



## Créer un diagramme de paquetages

Un diagramme de paquetage se crée depuis un paquetage.

Pour créer un diagramme de paquetages dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Cliquez sur le sous-menu **Paquetages**.
2. Sélectionnez la ligne du paquetage concerné et cliquez sur le bouton **Nouveau Diagramme**.
3. Sélectionnez **Diagramme de paquetages**.  
Le diagramme créé s'ouvre dans la fenêtre d'édition.



---


## Définir les paquetages

Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.

Exemples de *paquetages* :

- Le système d'information commercial.
- La comptabilité.
- La gestion de production.
- La commande numérique d'une machine.
- La gestion des stocks.
- La gestion du système d'alarme et du téléphone.

Pour ajouter dans le diagramme un paquetage existant :

1. Dans le diagramme de paquetages, cliquez sur le bouton **Paquetage** de la barre d'objets puis cliquez sur le plan de travail.
2. Dans la fenêtre **Ajout d'un paquetage**, sélectionnez la commande **Lister** à l'aide de la flèche   
La liste des paquetages apparaît.
3. Sélectionnez le paquetage qui vous intéresse et cliquez sur **OK**.  
Le nom du diagramme apparaît dans la fenêtre **Ajout d'un paquetage**.
4. Cliquez sur **Ajouter**.


Le paquetage apparaît dans le diagramme.

---

## Définir les classes

Le diagramme de paquetages permet de répartir les classes entre les paquetages.

Pour ajouter rapidement un ensemble de classes dans le diagramme de paquetages :


1. Cliquez sur **Menu principal** > **Recherche avancée** pour ouvrir l'assistant de recherche.
2. Dans l'assistant, sélectionnez la métaclasse "Classe" et cliquez sur **Chercher**   
La liste des classes du référentiel apparaît.
3. Sélectionnez les classes qui vous intéressent et glissez-les dans le diagramme.

---

## Spécifier les dépendances dans un diagramme de paquetage


Des liens vous permettent d'indiquer si un paquetage détient ou référence une classe ou un autre paquetage.


Pour indiquer qu'un paquetage référence une classe ou un autre paquetage :

1. Cliquez sur le bouton .
2. Effectuez le lien en partant d'un paquetage vers le paquetage ou la classe qu'il référence.  
Une fenêtre vous demande le type de lien à créer.
3. Sélectionnez "Paquetage référencé" ou "Classe référencée" selon qu'il s'agisse d'un paquetage ou d'une classe.

## LE DIAGRAMME DE COMPOSANTS

Un diagramme de **composants** présente l'interdépendance des composants logiciels et des **interfaces** (il définit qui utilise quoi).

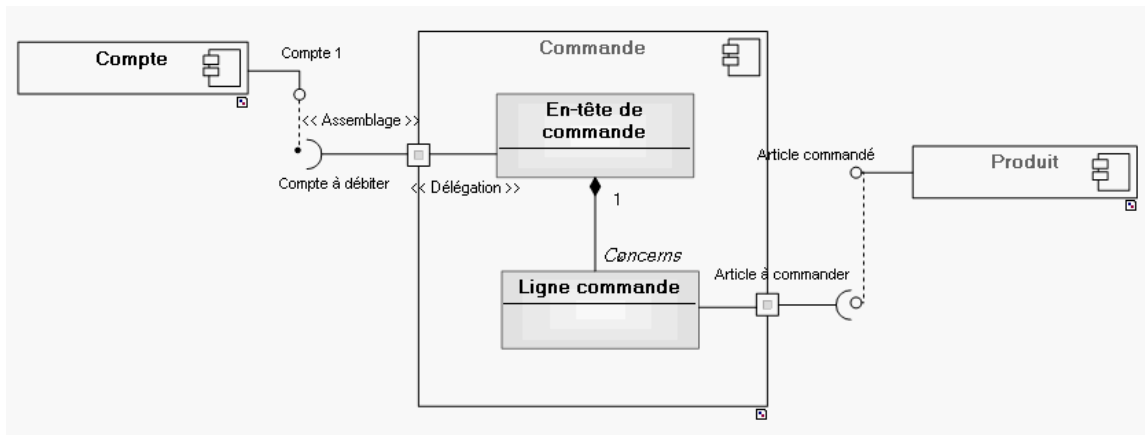
 Une **composant** est un élément de la mise en oeuvre du système : ce peut être un logiciel, un programme, un élément de code, mais aussi un élément physique tel qu'un document de travail.

 Une **interface** représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur. L'interface est un stéréotype de classe.

Un diagramme de composants contient des composants et des classes de stéréotype interface. Il est également possible d'y préciser les paquetages implémentés par les composants.

### Exemple de diagramme de composants

Ce diagramme décrit les éléments détenus par le composant "Commande" et les interactions de ces éléments avec des composants externes.



## Créer un diagramme de composants

Dans **HOPEX IT Architecture** vous pouvez créer un diagramme de composants à partir d'un composant ou d'un paquetage.

Pour créer un diagramme de composants à partir d'un paquetage dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Dans le sous-menu de navigation, cliquez sur **Paquetages**.
2. Sélectionnez la ligne du paquetage concerné et cliquez sur le bouton **Nouveau Diagramme**.
3. Sélectionnez **Diagramme de composants**.  
Le diagramme apparaît dans la fenêtre d'édition.


---

## Les composants

Un composant représente une partie modulaire d'un système qui encapsule son contenu et qui est remplaçable dans son environnement. Un composant définit son comportement par les interfaces qu'il fournit et celles qu'il requiert.

Un composant peut être remplacé par un autre si leurs interfaces sont conformes.

Un composant peut être un logiciel, un programme, un élément de code, etc.

Il est représenté par l'icône suivante : 

---


## Les interfaces


### Créer les interfaces des composants

Une interface représente la partie visible d'une classe ou d'un paquetage dans une relation contractuelle de type client - fournisseur.

L'interface est un type particulier de classe.

Pour créer une classe de stéréotype "Interface" dans le diagramme de structure composite :

1. Cliquez sur le bouton **Interface**  puis cliquez dans le diagramme.
2. Dans la fenêtre qui apparaît, saisissez le nom de la classe.
3. Cliquez sur **Ajouter**.

 Vous pouvez spécifier le détail de l'interface en termes d'attributs et d'opérations dans le diagramme de classes de la même manière que pour une classe.

### Relier les interfaces aux autres objets

Deux types de lien permettent de différencier les interfaces requises des interfaces fournies.

Une interface requise est une interface nécessaire au fonctionnement de l'objet.


Exemple : le composant « Gestion des achats » a besoin pour son fonctionnement de l'interface « Produit » pour pouvoir associer une commande d'achat aux produits commandés.

Une interface fournie est une interface mise à disposition par un objet à destination d'autres objets.

Exemple : le composant « Gestion des produits » met à disposition l'interface « Produit ».

Vous pouvez définir les interfaces requises et les interfaces fournies par un objet indépendamment des autres objets.

Pour préciser qu'une interface est supportée ou requise par un objet :

1. Cliquez sur le bouton **Lien**  et tirez le lien en partant de l'objet client vers l'interface.  
Une fenêtre apparaît.
2. Sélectionnez le type du lien à créer :
  - Interface requise
  - Interface supportée
3. Cliquez sur **OK**.

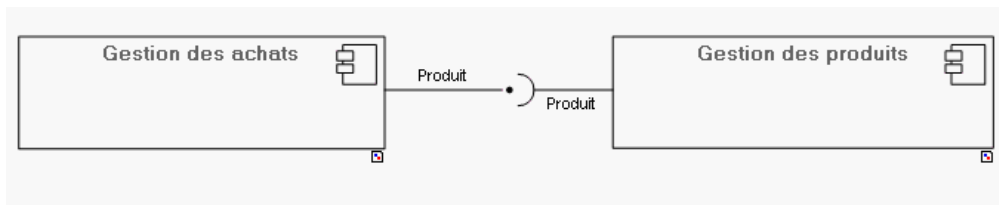
Le lien apparaît dans le dessin.

Selon le type de lien, la forme de l'interface change.

## Relier des interfaces

Deux interfaces peuvent être reliées l'une à l'autre. Cette connexion est modélisée par un connecteur.

Vous pouvez également indiquer qu'une interface fournie par un objet est requise par un autre. Il s'agit ici d'une seule et même interface.



## Les ports

Les ports permettent de connecter un composant à ses parties ou à son environnement.

Les ports sont symbolisés par un carré dans le diagramme, et posés en bordure du composant décrit lorsqu'ils assurent la connexion avec l'extérieur.

Ils sont reliés aux composants par des connecteurs.

Les ports peuvent spécifier les requêtes envoyées et les services fournis par le composant ainsi que les requêtes et services qu'ils peuvent requérir d'autres parties du système. Ces requêtes et services sont représentés par des classes de type Interface.

Vous pouvez visualiser les interfaces associées à un port dans la fenêtre de propriétés d'un port, sous l'onglet **Interfaces fournies et requises**.

## Les connecteurs

Les connecteurs permettent de relier les objets du diagramme.

Les connecteurs de type simple ne spécifient aucun type de connexion particulier, ils sont utilisés notamment pour relier les instances d'objets décrits dans des collaborations.

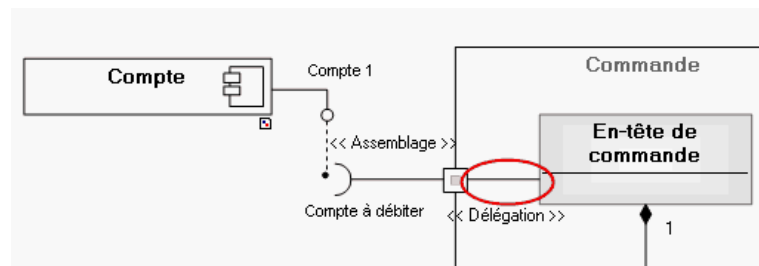
Dans le diagramme de structure composite, il est possible de spécifier le type de connecteur qui relie deux composants : Assemblage ou Délégation.

### Connecteur de délégation

Un connecteur de type "Délégation" montre le réacheminement de requêtes vers un élément du composant chargé de les réaliser.

Le lien de délégation peut se faire directement entre le port du composant et l'élément du composant ou entre le port du composant et le port de l'élément.

Ci-dessous, le composant "Commande" délègue la gestion des comptes à débiter à la classe "En-tête de commande".



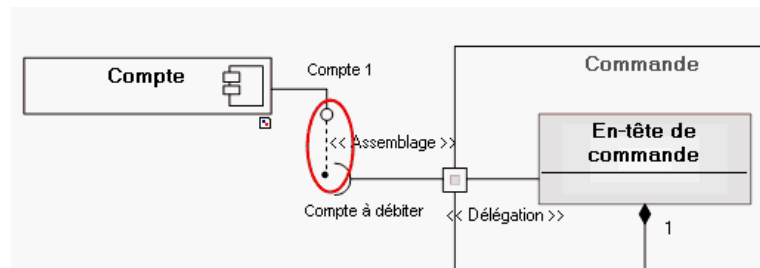
### Connecteur d'assemblage

Un connecteur de type "Assemblage" est un connecteur entre deux ou plusieurs composants ou ports qui indique qu'un ou que plusieurs composants fournissent les services que d'autres utilisent.

☛ *Il peut s'agir d'autres objets que de composants.*

Pour relier des ports ou des composants qui partagent une interface, vous pouvez également utiliser les liens "Interface fournie" et "Interface requise".

Un connecteur de type "Assemblage" relie l'interface fournie par le composant "Compte" à l'interface requise par la classe "En-tête Commande".



## LE DIAGRAMME DE STRUCTURE COMPOSITE

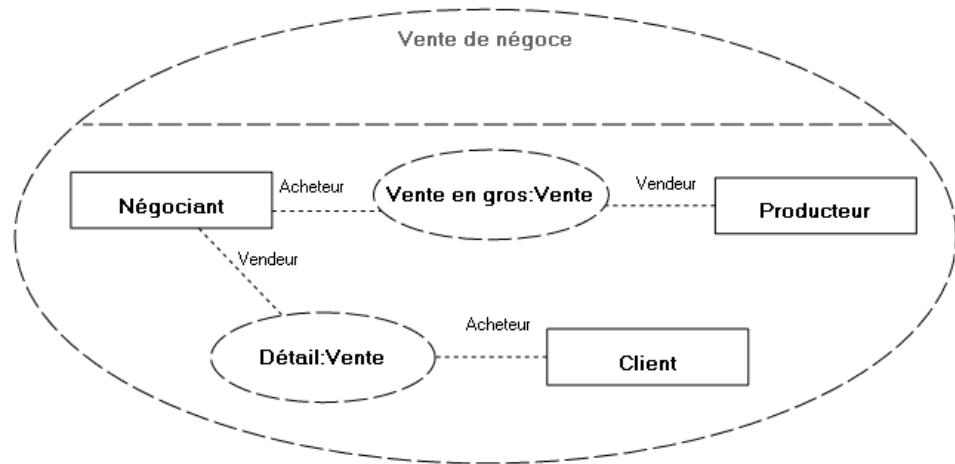
Le diagramme de structure composite permet de décrire la structure interne d'un composant, d'un paquetage ou d'une classe structurée.

Il permet également de préciser les collaborations qui interviennent entre les éléments de la structure dans l'exécution d'une tâche, en mettant en évidence le rôle joué par chaque élément dans la collaboration.

Les éléments de ce diagramme sont les parties (les *parts* en anglais), les ports par les biais desquels les parties interagissent avec l'extérieur, et les connecteurs reliant les parties entre elles ou avec les ports.

### Exemple de diagramme de structure composite

Ce diagramme décrit le rôle joué par les parties dans la collaboration "Vente de négoce".



## Créer un diagramme de structure composite

Pour créer un diagramme de composants à partir d'un composant dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Dans le sous-menu de navigation, cliquez sur **Composants**.
2. Sélectionnez la ligne du composant concerné et cliquez sur le bouton **Nouveau Diagramme**.
3. Sélectionnez **Diagramme de structure composite**.  
Le diagramme créé s'ouvre dans la fenêtre d'édition.



---

## Les parties

Une partie représente un rôle joué par une instance d'une classe ou d'un composant lors de l'exécution d'une tâche.

Les parties sont reliées entre elles par des connecteurs ou des dépendances.

Une partie peut également être reliée - via un connecteur- à un port qui assure l'interface entre le composant décrit et l'extérieur.

Pour plus de détails sur ces éléments, voir :

- ✓ [Les connecteurs](#)
- ✓ [Les liens de dépendance](#)
- ✓ [Les ports.](#)

Les multiplicités définies sur les parties indiquent le nombre d'instances qui sont créées. Les multiplicités sur les rôles de connecteur indiquent le nombre de liens qui peuvent être créés pour chacune de ces instances.


Pour définir la multiplicité d'une partie :

1. Ouvrez la fenêtre de propriétés de la partie.
2. Sélectionnez la page **Caractéristiques**.
3. Cliquez sur la flèche située à l'extrémité du champ **Multiplicité** et sélectionnez la multiplicité voulue.
4. Cliquez sur **OK**.

---


## Les collaborations

Dans le diagramme de structure composite, une *collaboration* décrit les rôles joués par chaque partie (instance) dans la réalisation d'une tâche.

 Une collaboration (UML) décrit une structure collaborative entre plusieurs éléments (rôles), qui accomplissent chacun une fonction spécialisée et qui réalisent collectivement une fonctionnalité attendue du système. Son objectif est de montrer comment un système fonctionne indépendamment d'une utilisation spécifique. On en retirera donc généralement l'identité précise des classes ou des instances qui y participent.

Elle est représentée par un ovale en pointillé contenant les instances de la collaboration.

Ces instances sont reliées entre elles par des *connecteurs*. A chaque extrémité du connecteur s'affiche le rôle qui correspond au nom de l'instance.

 Un connecteur est un lien qui permet d'établir une communication entre plusieurs objets. Un connecteur de délégation relie le contrat externe de l'objet (tel qu'il est spécifié par ses ports et/ou ses interfaces) aux objets internes qui vont le réaliser. Un connecteur d'assemblage entre plusieurs objets (ou leurs ports) permet de spécifier comment un des objets fournit l'interface requise par un autre.

Il est possible d'appliquer le modèle d'une collaboration à différentes instances.

## Utilisation de collaboration

Une utilisation de collaboration représente l'application de la structure décrite par une collaboration à une situation particulière mettant en oeuvre des classes ou des instances spécifiques. Ces classes ou ces instances jouent alors les rôles définis dans la collaboration.

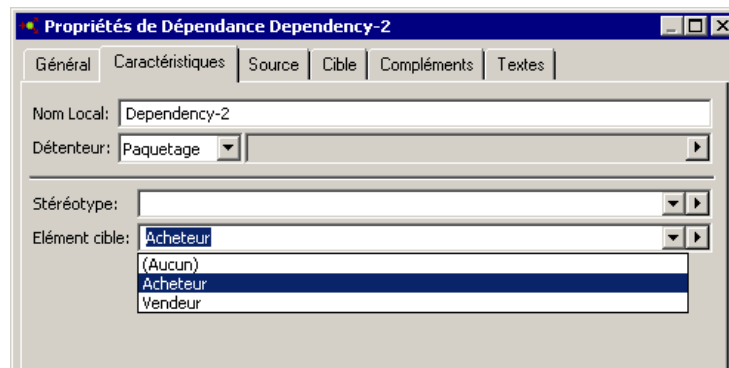
Les instances sont reliées à l'utilisation d'une collaboration par un lien de *dépendance* sur lequel doit être précisé le rôle joué par l'instance.



*Une dépendance précise que l'implémentation ou le fonctionnement d'un ou de plusieurs éléments nécessite la présence d'un ou de plusieurs autres éléments. Il existe plusieurs stéréotypes de dépendance.*

## Exemple d'utilisation de collaboration

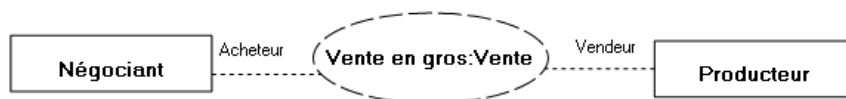
Dans le cas d'une demande d'achat entre deux instances d'acteur, une collaboration est utilisée. Cette collaboration relie deux rôles : le rôle d'acheteur et le rôle de vendeur. Sur la dépendance qui relie chaque instance à la collaboration, vous pouvez indiquer le rôle que joue l'instance.



## Les liens de dépendance

Une dépendance précise que l'implémentation ou le fonctionnement d'un ou de plusieurs éléments nécessite la présence d'un ou de plusieurs autres éléments.

Une dépendance est également une relation de type fournisseur / client qui indique quel est l'élément source et l'élément cible dans la collaboration.



Un stéréotype sur la dépendance permet de spécifier la nature de la dépendance :

- Binding : le binding est une relation entre un template et un élément de modélisation généré à partir du template. Il inclut une liste d'arguments en correspondance avec les paramètres du template.
- Derive : indique une relation de dérivation entre des éléments de modélisation qui sont généralement, mais pas nécessairement, de même type. Une telle relation de dépendance implique que l'un des éléments peut être calculé à partir de l'autre.
- Mapping UML/XML : une expression de mapping qui définit la relation entre les éléments (classes, attributs, ...) d'un schéma ou d'un diagramme de classes et ceux d'un autre schéma ou diagramme de classes.
- Refine : spécifie une relation de dépendance entre des éléments de modélisation à différents niveaux sémantiques, tels que l'analyse et la conception.
- Trace : spécifie une relation de traçabilité entre des éléments de modélisation ou des ensembles d'éléments de modélisation qui représentent le même concept dans différents modèles.

Pour préciser la nature d'une dépendance :

1. Ouvrez la fenêtre de propriétés de la dépendance.
2. Sélectionnez la page **Caractéristiques**.
3. Dans le champ **Stéréotype**, déroulez la liste et sélectionnez l'un des stéréotypes proposés.

La flèche  vous permet également de créer de nouveaux stéréotypes.



# LE DIAGRAMME DE MACHINE À ÉTATS



Un diagramme de machine à état permet de décrire les comportements possibles d'un objet, suivant les événements auxquels il est soumis au cours de son cycle de vie.

Les points suivants sont abordés ici :

- ✓ [Présentation du diagramme de machine à états](#)
- ✓ [Créer un diagramme de machine à états](#)
- ✓ [Les états](#)
- ✓ [Les transitions entre états](#)

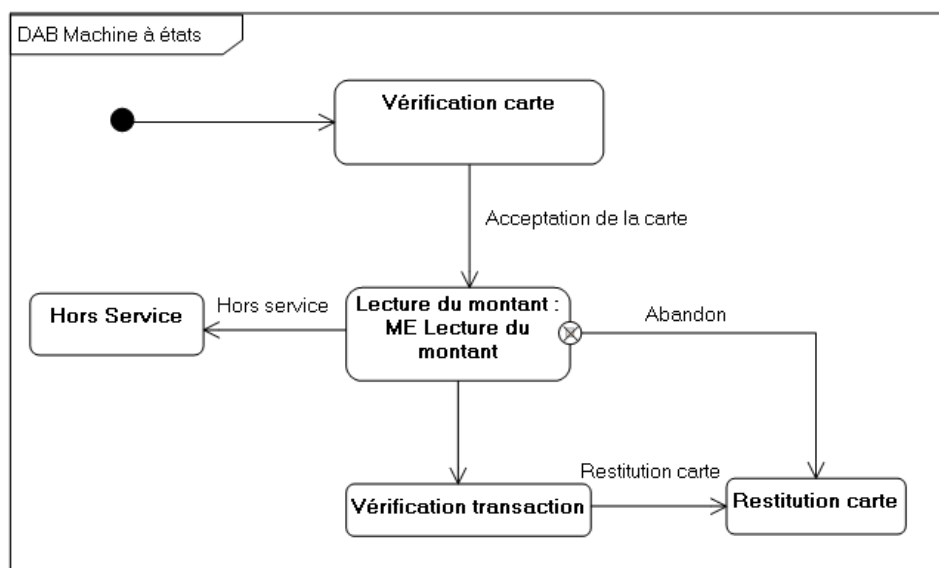
## PRÉSENTATION DU DIAGRAMME DE MACHINE À ÉTATS

Une machine à état est l'ensemble des états et des transitions entre états qui définissent le cycle de vie d'un objet variable dans le temps.

Le diagramme de machine à état permet de représenter cet enchaînement d'états que peut prendre un objet en réponse aux interactions avec les objets (internes ou externes au système étudié) qui peuplent son environnement.

### Exemple de diagramme de machine à d'états

Le diagramme suivant décrit les comportements possibles d'un distributeur automatique.



## Créer un diagramme de machine à états

Un diagramme de machine à états se crée à partir d'une machine à état.

Vous pouvez créer une machine à état à partir d'un paquetage, d'une classe ou d'un composant.

Pour créer un diagramme de machine à états dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Dans le sous-menu de navigation, cliquez sur **Machines à états**.
2. Sélectionnez la ligne de la machine à états concernée et cliquez sur le bouton **Nouveau Diagramme**.

3. Sélectionnez **Diagramme de machine à états**.

Le diagramme créé s'ouvre dans la fenêtre d'édition.

Le diagramme est initialisé par la création d'une région. Une région est une partie d'un état composite ou d'une machine à états qui contient des états et des transitions et dont l'exécution est autonome.

# LES ÉTATS

Un état d'objet est une condition ou une situation au cours de la vie d'un objet durant laquelle il satisfait à certaines conditions, exerce une certaine activité ou attend un événement. Un état d'objet représente un intervalle de temps dont les bornes sont deux événements. Un état d'objet est une phase par laquelle passe l'objet au cours de son cycle de vie.


## Exemples d'état d'objet

- Une personne peut être :
  - Célibataire
  - Mariée
  - Divorcée
- Un article peut être :
  - Disponible
  - En stock
  - En alerte
  - En rupture de stock.
  - Etc.

---





## Créer un état

Pour créer un état dans le diagramme de machine à états :

1. Cliquez sur la flèche noire associée au bouton **Etat**  de la barre d'insertion du diagramme.
2. Sélectionnez un type d'état.
3. Cliquez sur le plan de travail.  
La fenêtre **Ajout d'un état** s'ouvre.
4. Indiquez le **Nom** de l'état et cliquez sur **Créer**.  
L'état apparaît dans le diagramme.

## Les types d'état

Il est nécessaire de préciser le type de l'état lors de sa création. Ce peut être :

-  Un état normal : ne possède pas de sous-structure.
-  Un état composite : se compose de plusieurs états, décrits dans le diagramme.
-  Une sous-machine à état : appelle la description d'une machine à état décrite par ailleurs. Voir [Précision comportementale d'un état](#).
-  Un état final



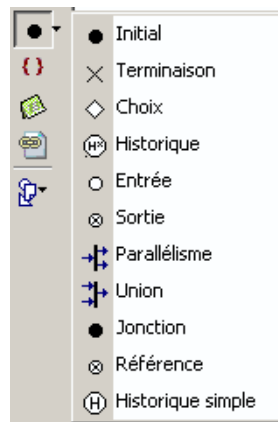
Lorsque vous posez un état dans un autre, il est automatiquement relié comme composant de cet état.

## Pseudo-états

Les pseudo-états sont utilisés pour spécifier des chemins complexes en combinant plusieurs transitions entre états.

Ils peuvent être de différents types : initial, terminaison, choix, historique (deep history), historique simple (shallow history), point d'entrée, point de sortie, parallélisme (fork), union (join), jonction ou référence.

### ***Initial***



Le pseudo-état initial a une seule transition en sortie vers l'état Initial de l'objet lors de sa création.

### ***Historique***

Un pseudo-état Historique représente la dernière configuration active de l'état composite qui le contient ; c'est-à-dire, la configuration active quand l'état composite a été quitté pour la dernière fois.

### ***Historique simple***

Un pseudo-état historique simple représente le plus récent sous-état actif d'un état composite (sans les sous-états de ce sous-état).

### ***Parallélisme***

Un parallélisme (fork) sépare une transition en plusieurs transitions concurrentes.

### ***Union***

Une union (join) est le regroupement de plusieurs transitions en une seule.

### **Choix**

Représente le choix d'une transition entre plusieurs transitions possibles.

### **Jonction**

Une jonction est utilisée pour définir des chemins de transition complexes entre plusieurs états.

### **Entrée**

C'est un point d'entrée d'une machine à état ou d'un état composite.

### **Sortie**

C'est un point de sortie d'une machine à état ou d'un état composite.

### **Référence**

C'est une référence à une entrée ou à une sortie d'une machine à état ou d'un état composite

### **Terminaison**

L'entrée dans ce pseudo-état implique une terminaison complète de la machine à état.

### **Historique**

Un état **Historique** représente la dernière configuration active d'un état composite ; c'est-à-dire la configuration active quand l'état composite a été quitté pour la dernière fois.

Un état **Historique simple** représente le plus récent sous-état actif de l'état composite.

Exemple :

Prenons l'état "Marié" comme dernière configuration active. Cet état a pour sous-états "Avec enfants" et "Sans enfants". Dans le cas d'un historique, le sous-état "Avec enfants" ou "Sans enfant" est précisé. Dans le cas d'un historique simple, seul l'état "Marié" est pris en compte.

---

## **Précision comportementale d'un état**

Un état peut être composé de sous-états.

Pour décrire la composition d'un état dans un diagramme :

1. Ouvrez le menu contextuel d'un état et cliquez sur la commande **Nouveau > Précision comportementale**.  
La fenêtre de création d'un diagramme de machine à états apparaît.
2. Cliquez sur **Créer**.  
Le diagramme correspondant s'ouvre.

Vous pouvez également définir la composition d'un état en lui associant une machine à état, nouvelle ou existante :

1. Ouvrez la page de propriétés **Caractéristiques** de l'état décrit.
2. Dans le champ **Précision comportementale**, créez une machine à état ou recherchez une machine à état existante.

---


## Propriétés d'un état

Pour accéder aux propriétés d'un état :

1. Cliquez avec le bouton droit sur l'état.
2. Sélectionnez **Propriétés**.  
La fenêtre de propriétés de l'état s'ouvre.

Elle vous permet :

- De modifier le **Nom** de l'état.
- D'indiquer si ses sous-états sont **Concurrents**, c'est-à-dire s'ils peuvent être exécutés simultanément ou non.
- D'indiquer la **Précision comportementale** (dans le cas d'un état complexe). Voir [Précision comportementale d'un état](#).
- De préciser les **Activités** qui peuvent être effectuées en entrée, en sortie ou pendant que l'objet est dans cet état.

 Le contenu de la fenêtre de propriétés d'un état varie en fonction du type de l'état.

# LES TRANSITIONS ENTRE ÉTATS

Le passage d'un état à un autre est matérialisé par une **transition**.



*Une transition est le passage d'un objet d'un état dans un autre. Une transition est une réponse d'un objet à un événement qu'il reçoit. Quand un événement se produit et que certaines conditions sont satisfaites, l'objet va effectuer certaines actions tandis qu'il est encore dans le premier état puis passer au deuxième état.*

Vous devez définir toutes les transitions qui sont autorisées. Celles qui ne sont pas définies sont interdites.

Exemples de transitions :

En ce qui concerne l'état civil d'une personne, certaines transitions sont possibles :

- Elle peut passer de l'état "célibataire" à l'état "marié".
- Elle peut passer de l'état "marié" à l'état "divorcé".

D'autres ne sont pas possibles :

- Elle ne peut pas passer de l'état "célibataire" à l'état "divorcé".

---

## Créer une transition

Pour créer une transition entre deux états :

1. Dans le diagramme de machine à états, cliquez sur le bouton **Transition**

**(UML)**



de la barre d'insertion.

2. Cliquez sur l'état de départ et déplacez la souris jusqu'à l'état d'arrivée.
3. Relâchez le bouton : la transition est créée.

---

## Les types de transition

Une transition peut être de type externe, interne ou locale.

Vous pouvez préciser le type de la transition dans la page de propriétés **Caractéristiques** de la transition.

### Transition externe

Une transition externe est une transition qui modifie l'état actif.

### Transition interne

Une transition interne à un objet permet de prendre en compte l'arrivée d'un événement qui ne provoque pas de changement d'état de l'objet, mais une action comme l'appel d'une opération ou l'émission d'un message. Par exemple, lors d'un

mouvement de stock, un article peut ne pas changer d'état si la quantité restant disponible en stock est suffisante et ne passe pas le seuil d'alerte ou de rupture.

## Transition locale

Une transition locale s'applique aux sous-états d'un état composite. Elle peut provoquer un changement d'état uniquement à l'intérieur de l'état composite.

---

## Effet d'une transition

Le déclenchement d'une transition peut être accompagné d'un effet. L'effet peut être représenté par :

- Une activité
- Une collaboration
- Une interaction
- Une machine à état

Pour définir l'effet d'une transition :

1. Ouvrez la page de propriétés **Caractéristiques** de la transition.
2. Cliquez sur la flèche située à l'extrémité du champ **Effet (Comportement)** et créez ou reliez l'objet qui définit l'effet.

## Affichage des effets d'une transition

Pour modifier l'affichage des effets de la transition.

1. Dans le diagramme de machine à état, cliquez avec le bouton droit sur la transition puis cliquez sur **Formes et détails**.
2. Sélectionnez "Effet" dans l'arbre qui s'affiche.

Vous pouvez choisir d'afficher tout ou partie des effets de la transition, avec leurs caractéristiques.

---

## Événement déclencheur d'une transition

Dans la fenêtre de propriétés d'une transition, sous l'onglet **Événement**, vous pouvez indiquer le **Type d'Événement** qui déclenche une transition.

Ce peut être :

- Un événement quelconque
- L'appel d'une opération
- Un changement de l'objet concerné par la transition
- La création d'un objet
- La destruction d'un objet
- L'envoi d'un signal
- L'envoi d'une opération
- L'émission d'un signal par l'objet
- La réception d'un signal
- La réception d'une opération
- Un *temporisateur*



*Un temporisateur est un événement déterminé uniquement par le temps qui s'écoule. Ex : Le lundi, à quatre heures, etc.*

Les champs affichés sous le champ **Type d'événement** varient selon le type d'événement sélectionné.

Vous pouvez sélectionner l'objet concerné par l'effet.

Dans le cas d'une opération ou d'un signal, il est possible de préciser les valeurs des paramètres transmis.

# LE DIAGRAMME D'ACTIVITÉS



Un diagramme d'activités est proche du diagramme de machine à états. A la différence du diagramme de machine à états qui décrit le comportement d'un objet via l'enchaînement d'états, le diagramme d'activités décrit le comportement d'un élément en termes d'actions.

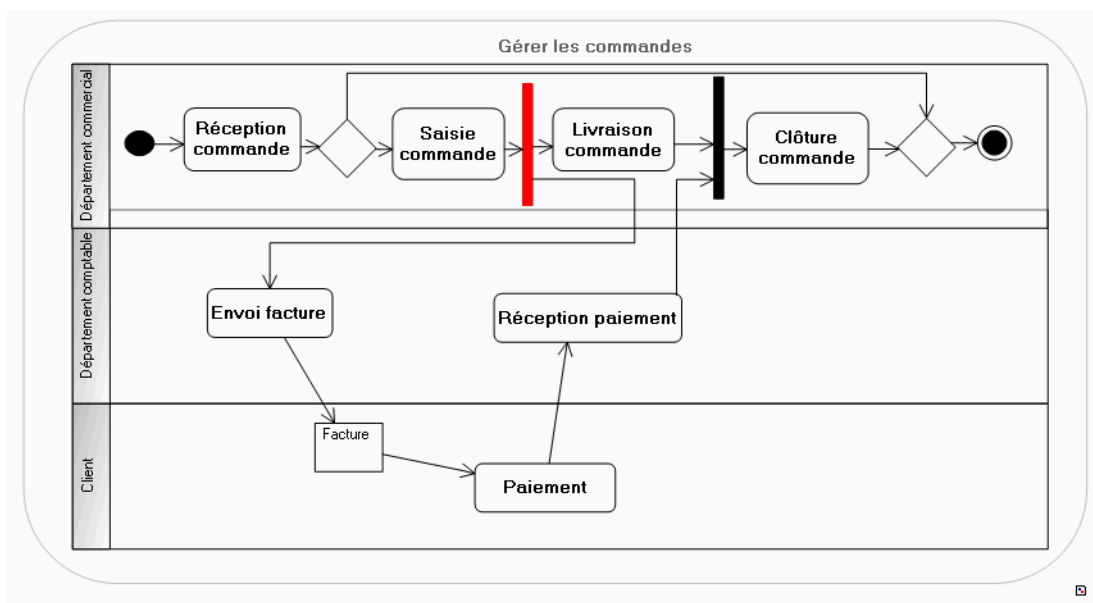
- ✓ [Présentation du diagramme d'activités](#)
- ✓ [Les partitions](#)
- ✓ [Les noeuds](#)
- ✓ [Les Flux](#)

## PRÉSENTATION DU DIAGRAMME D'ACTIVITÉS

Un diagramme d'activités représente un séquençement d'étapes décrivant le comportement d'un élément du système.

Les étapes sont modélisées par des noeuds - noeuds d'action, de paramétrage ou de contrôle - coordonnés par des flux de données ou de contrôle.

### Exemple de diagramme d'activités



## Créer un diagramme d'activités

Dans **HOPEX IT Architecture**, un diagramme d'activités se crée à partir d'un paquetage ou d'une activité.

Vous pouvez créer une activité sur un paquetage, un composant ou une classe.

Pour créer un diagramme d'activités :

1. Faites un clic droit sur le paquetage ou l'activité concerné(e).
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'activités**.

Le nouveau diagramme d'activités s'ouvre.

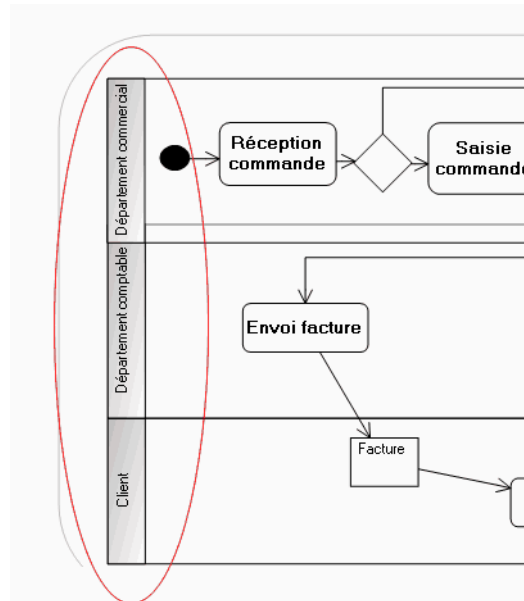


## LES PARTITIONS

Un diagramme d'activités peut être découpé en partitions. Chaque partition contient des nœuds ou des actions ainsi que les flux entre ces éléments.

Vous pouvez utiliser des partitions pour organiser les tâches ou pour spécifier l'élément responsable de la mise en œuvre d'un ensemble de tâches.

Pour plus de détails sur les couloirs, voir le guide **HOPEX Common Features**, chapitre "Manipuler les diagrammes", section "Utiliser les couloirs".



### Créer une partition

Pour créer une partition dans le diagramme d'activités :

1. Cliquez sur le bouton **Partition**  de la barre d'insertion d'objets.
2. Indiquez son nom.
3. Cliquez sur **Ajouter**.

### Propriétés d'une partition

La page **Etats** présente les états contenus dans la partition.

La page **Compléments** permet d'indiquer l'élément représenté par la partition. Il s'agit de l'élément qui met en œuvre les éléments de la partition. Ce peut être un acteur, une classe ou un composant.

## LES NOEUDS

Les nœuds permettent de modéliser les étapes de l'activité. Il existe différents types de nœuds dans **HOPEX**.

- Les nœuds d'actions
- Les nœuds de paramétrage
- Les nœuds de contrôle
- Les nœuds d'objets : pins d'entrée, de sortie et d'échange

---


### Les nœuds d'actions

Les actions sont les étapes élémentaires du comportement représenté par l'activité.

La coordination des actions est réalisée à l'aide de flux de contrôle et de flux de données.

#### Créer une action

Pour créer une action dans un diagramme d'activités :

1. Dans la barre d'insertion d'objets du diagramme, sélectionnez le bouton correspondant au type d'action, puis cliquez sur le plan de travail.  
La fenêtre d'ajout d'une action du type choisi s'ouvre.  
 La barre d'insertion propose les trois types d'action principaux.
2. Indiquez son nom et cliquez sur **Ajouter**.

#### Modifier le type d'action

Dans la page de propriétés **Caractéristiques** de l'action, vous pouvez modifier le type d'action. Ce peut être :

- L'appel d'une opération d'un autre objet
- La création d'un objet
- La destruction d'un objet
- L'exécution d'une opération locale à l'objet
- L'émission d'un signal par l'objet
- La destruction finale de l'objet
- Etc.

---

### Les nœuds de paramétrage

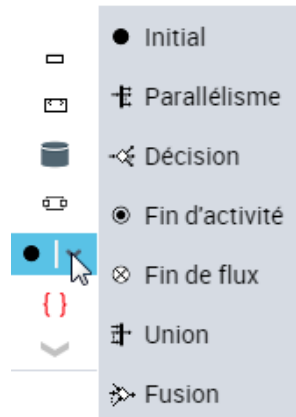
Les nœuds de paramétrage d'une activité décrivent les entrées ou les sorties de cette activité.

Ils transmettent les paramètres à l'activité par l'intermédiaire des flux qu'ils émettent ou qu'ils reçoivent.

## Les nœuds de contrôle

Un nœud de contrôle coordonne les flux entre les noeuds d'une activité.

Un nœud de contrôle peut être de type initial, final, décision, fusion (merge), parallélisme (fork) ou union (join).



### Types de noeud de contrôle

#### ***Initial***

Un noeud initial indique où débute le flux de contrôle lorsque l'activité est invoquée. Une activité peut avoir plusieurs nœuds initiaux.

#### ***Final***

Lorsqu'un jeton atteint un nœud final d'activité, tous les flux de l'activité sont stoppés. Au contraire, un nœud final de flux détruit les jetons qui lui arrivent mais n'a aucun effet sur les autres jetons de l'activité.

#### ***Décision***

Une décision fait le choix d'un seul flux entre plusieurs flux sortants possibles. Les flux sortants sont sélectionnés en fonction de leur condition de garde.

#### ***Fusion***

Une fusion (merge) rassemble plusieurs flux alternatifs entrants en un seul flux sortant. Elle n'est pas utilisée pour synchroniser des flux concurrents mais pour accepter un seul flux parmi plusieurs.

#### ***Parallélisme***

Un parallélisme (fork) sépare un flux en plusieurs flux concurrents. Les jetons arrivant à un parallélisme sont dupliqués à travers les flux sortants.

### **Union**

Une union (join) synchronise des flux multiples. Quand tous les flux en entrée sont disponibles, le flux en sortie est déclenché.

---

## **Les nœuds d'objets : pins d'entrée, de sortie et d'échange**

Pour spécifier les valeurs en entrée d'une action et les valeurs de retour, on utilise des nœuds d'objets appelés pins (pin en anglais) d'entrée ou de sortie. L'action ne peut débuter que si une valeur est affectée au pin d'entrée. De même, quand l'action se termine, une valeur doit être affectée au pin de sortie.

### **Pin d'entrée**

Un pin d'entrée supporte les valeurs d'entrée qui doivent être consommées par une action et qu'il reçoit de la part d'autres actions.

### **Pin de sortie**

Un pin de sortie supporte les valeurs de sortie qui sont produites par une action et fournit ces valeurs à d'autres actions à travers des flux.

### **Pin d'échange**

Un pin d'échange est utilisé pour représenter les données échangées entre deux actions.

---

## **Les Flux**

Le passage d'un nœud à un autre est matérialisé par un flux.

### **Flux de contrôle**

Un flux de contrôle démarre un nœud d'action lorsque le précédent est terminé. Les objets et les données ne peuvent pas être transmis par un flux de contrôle.

### **Flux d'objets**

Un flux d'objets permet de transmettre des données ou objets d'un nœud à un autre à l'intérieur d'une activité.



# LES DIAGRAMMES D'INTERACTION



Les diagrammes d'interaction, c'est-à-dire le diagramme de séquence, le diagramme de communication et le diagramme de vue générale d'interaction, représentent une série d'interactions entre objets, ordonnée dans le temps. Ils montrent une ou plusieurs histoires possibles du système.

Les points suivants sont abordés ici :

- ✓ [Les interactions](#)
- ✓ [Le diagramme de séquence](#)
- ✓ [Le diagramme de communication](#)
- ✓ [Le diagramme de vue générale d'interaction](#)

# LES INTERACTIONS

Une interaction décrit le comportement d'un système dans un contexte particulier par les échanges de messages entre les éléments de ce système.

Quand les diagrammes de machines à état ou d'activités étudient des comportements individuels, les diagrammes d'interaction se concentrent sur la coopération d'un ensemble d'objets.

---

## Créer une interaction

Vous pouvez créer une interaction à partir d'un composant, d'un paquetage ou d'une classe.

Pour créer une interaction dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Cliquez sur le sous-menu **Interactions**.
2. Cliquez sur le bouton **Nouveau**.
3. Indiquez le nom de l'interaction et éventuellement le détenteur.
4. Cliquez sur **OK**.

---

## Créer un diagramme d'interaction

Le diagramme de séquence, le diagramme de communication et le diagramme de vue générale d'interaction se crée à partir d'une interaction.

Pour créer un diagramme d'interaction :

1. Faites un clic droit sur une interaction.
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'interaction**.



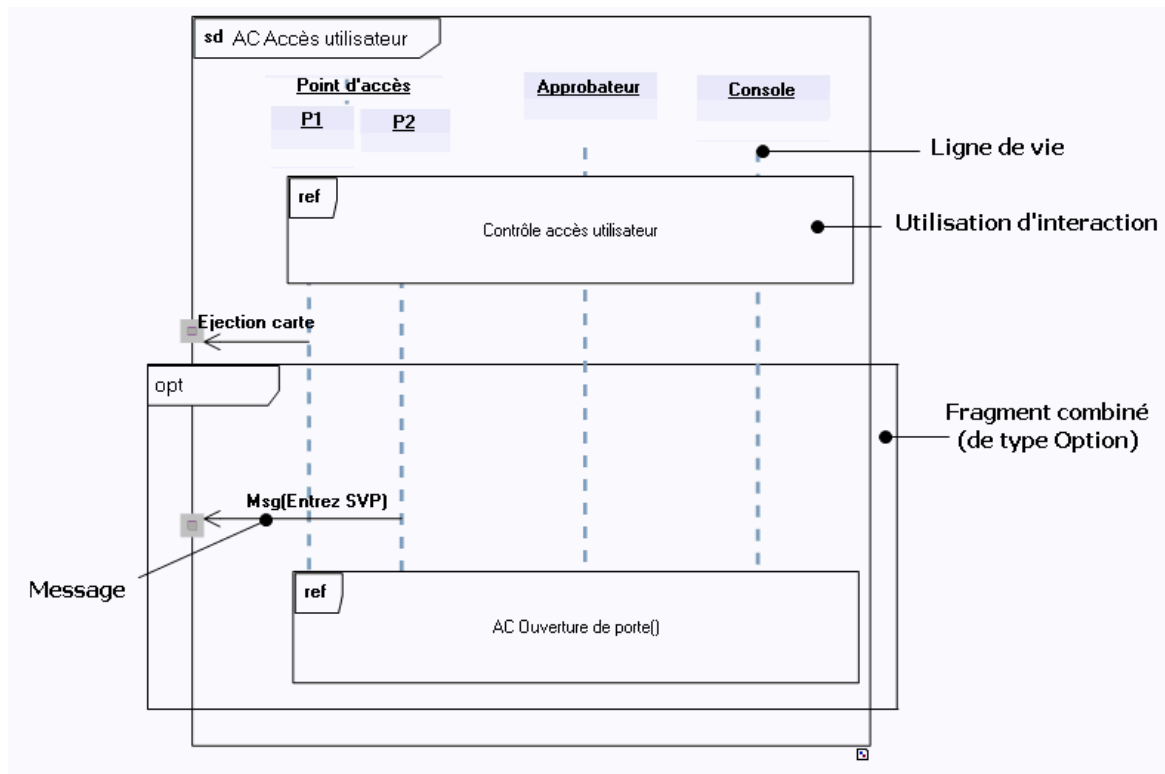
## LE DIAGRAMME DE SÉQUENCE

Le diagramme de séquence met en évidence la chronologie des messages échangés entre les objets participant à une interaction. Ces objets sont représentés dans le diagramme par leurs lignes de vie.

### Exemple de diagramme de séquence

Le diagramme ci-dessous décrit le comportement d'un distributeur automatique :

- Deux points d'entrée (représentés par des lignes de vie) donnent lieu à un contrôle d'accès de l'utilisateur. Ce contrôle est décrit dans une interaction.
- Selon le résultat du contrôle, l'accès est refusé et la carte de l'utilisateur éjectée ou l'ouverture de la porte est actionnée.
- Un comportement optionnel (représenté par un fragment combiné) peut influencer l'ouverture de la porte.



---

## Créer un diagramme de séquence

Pour créer un diagramme de séquence dans **HOPEX IT Architecture** :

1. Faites un clic droit sur une interaction.
2. Dans le menu contextuel qui s'affiche, cliquez sur **Nouveau > Diagramme d'interaction**.

Voir aussi [Créer une interaction](#).

---

## Les lignes de vie

Une ligne de vie représente un participant dans une interaction.


Les lignes de vie sont des instances de différents types (de classe, d'acteurs, etc.).

Dans un diagramme de séquence, le temps est représenté comme s'écoulant du haut vers le bas le long des lignes de vie de ces objets. Entre ces objets transitent des instances de messages.

☛ *Les instances représentées dans un diagramme de séquence peuvent être des instances de classe, d'acteur, de paquetage, de cas d'utilisation, de composant ou de nœud, ce qui permet de définir des diagrammes de séquence au niveau de détail souhaité.*

### Créer une ligne de vie

Pour créer une ligne de vie :

1. Cliquez sur le bouton **Ligne de vie** .
2. Cliquez dans le diagramme.  
Une fenêtre s'ouvre.
3. Saisissez le nom de la ligne de vie.
4. Cliquez sur **Ajouter**.  
La ligne de vie apparaît dans le diagramme.

### Propriétés d'une ligne de vie

Pour accéder aux propriétés d'une ligne de vie :

1. Sélectionnez l'instance et cliquez sur le bouton **Propriétés** de la fenêtre d'édition s'il n'est pas activé.

Vous pouvez sélectionner le **Type** de l'objet (Acteur, Classe, etc.), préciser de quelle **Classe**, **Acteur**, etc. cet objet est une instance et indiquer son *stéréotype*.

---

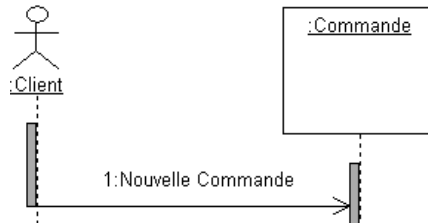
## Les messages

Un message définit une communication particulière entre les lignes de vie d'une interaction. Il spécifie l'émetteur et le récepteur par l'intermédiaire de spécifications d'occurrence, ainsi que le type de communication. Cette communication peut être,

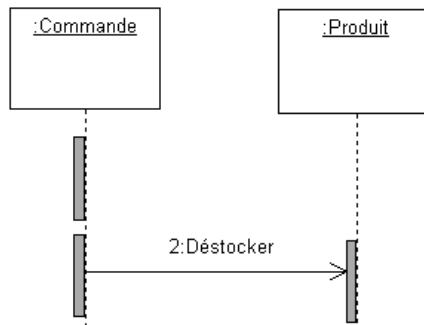
par exemple, l'émission d'un signal, l'invocation d'une opération, la création ou la destruction d'une instance.

## Exemples de messages échangés

1) Le message envoyé par l'acteur "Client" à la classe "Commande" transporte le signal "Nouvelle commande".



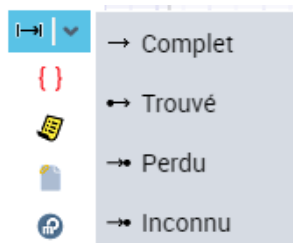
2) Le message envoyé par la classe "Commande" à la classe "Produit" appelle l'opération "Déstocker".



## Créer un message

Pour créer un message dans le diagramme de séquence :

1. Cliquez sur le bouton **Message** de la barre d'insertion d'objets en sélectionnant le type de message voulu.



2. Allez de la ligne pointillée sous le premier objet à celle qui est sous le deuxième objet en maintenant le bouton gauche de la souris enfoncé. Le message échangé entre les deux objets se dessine.

## Types de messages

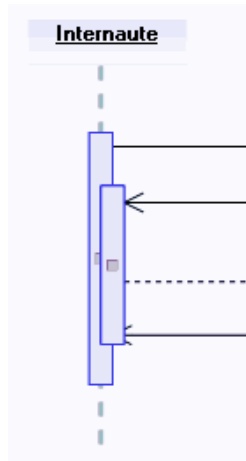
Vous pouvez créer quatre types de messages :

- Dans un message de type "Complet", l'émetteur et le destinataire sont tous les deux définis.
- Dans un message "Perdu", seul l'émetteur est connu. On considère ici que le message n'atteint jamais sa destination.
- Dans un message "Trouvé", seul le destinataire est connu. C'est le cas lorsque l'origine du message se situe en dehors du contexte de description.
- Dans un message de type "Inconnu", ni l'émetteur ni le destinataire ne sont définis.

---


## Occurrence d'exécution

Une occurrence d'exécution d'une ligne de vie (execution specification) représente une unité d'action ou de comportement qui se déroule à partir d'une occurrence d'événement de début jusqu'à une occurrence d'événement de fin.



## Créer une occurrence d'exécution

Pour créer une occurrence d'exécution :

1. Dans le diagramme de séquence, cliquez sur le bouton **Occurrence d'exécution**  de la barre d'insertion d'objets.
2. Positionnez-la sur la ligne de vie concernée. L'occurrence apparaît dans le diagramme.

## Occurrence d'événement

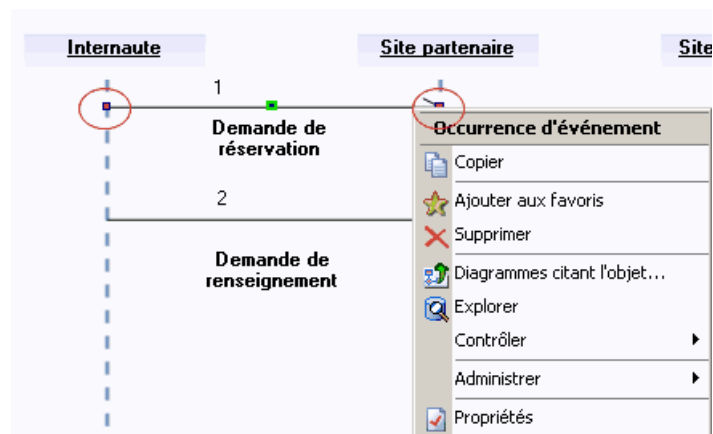
La création d'un message ou d'une occurrence d'exécution entraîne automatiquement la création d'occurrences d'événements.

Une occurrence d'événement (Occurrence Specification) est un point syntaxique à l'extrémité d'un message ou au début ou à la fin d'une occurrence d'exécution.

Les occurrences d'événement sont ordonnées le long d'une ligne de vie.

Ce sont les unités sémantiques de base d'une interaction.

Vous pouvez accéder au menu contextuel d'une occurrence d'événement en cliquant avec le bouton droit sur l'une des extrémités d'un message.



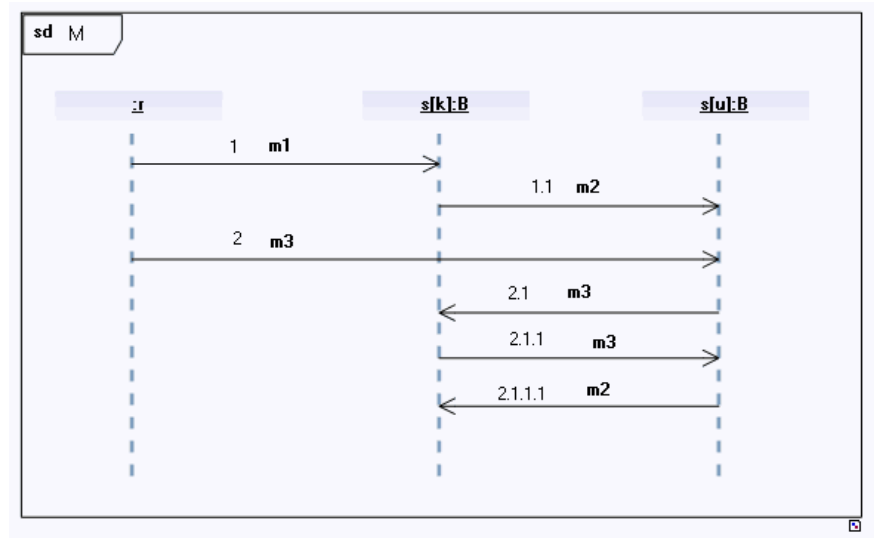
## Calcul des numéros de séquence

A partir du positionnement des occurrences d'événement, un outil de calcul permet d'ordonner les messages et les occurrences d'exécution.

Pour ordonner les messages circulant entre des lignes de vie d'une interaction :

1. Ouvrez le menu contextuel de l'interaction décrite.
2. Cliquez sur **Calculer les numéros de séquence**.  
L'outil applique automatiquement des numéros aux messages.

## Exemple



Vous pouvez modifier manuellement le numéro de séquence d'un message dans la fenêtre de propriétés du message :

- 1 Cliquez sur l'onglet **Caractéristiques** et modifiez la valeur du champ **Sequence Expression**.

Lorsque vous relancez le calcul des numéros de séquence, ce dernier met à jour le séquençement en fonction des modifications apportées.

---

## Fragment combiné

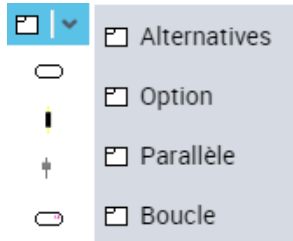
Un fragment combiné permet de décrire de manière concise plusieurs séquences d'exécution.

Un fragment combiné est défini par un opérateur d'interaction et les opérandes d'interactions correspondants.

## Créer un fragment combiné

Pour créer un fragment combiné :

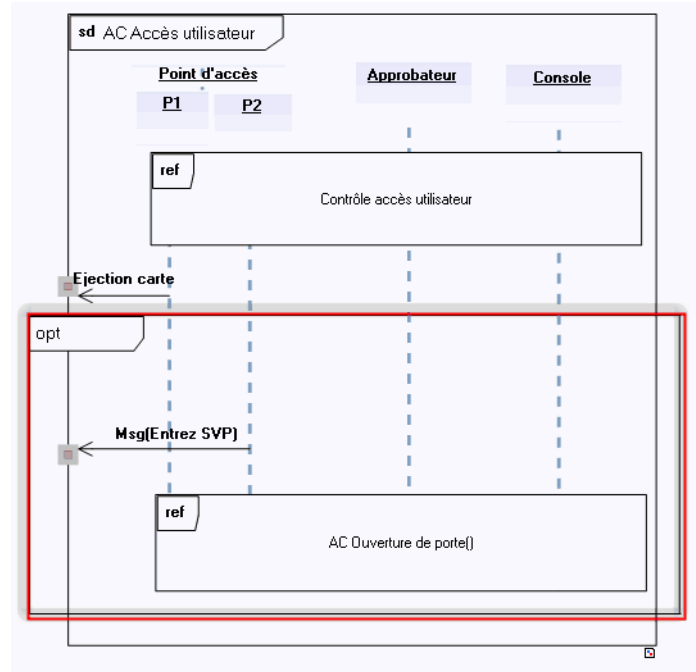
1. Dans la barre d'insertion d'objets du diagramme de séquence, cliquez sur le bouton **Fragment combiné**.  
Vous pouvez associer au fragment combiné différents types d'opérateur d'interaction. La flèche située à droite du bouton offre un raccourci vers quatre d'entre eux. Voir [Type d'opérateur d'interaction](#).



2. Cliquez dans le diagramme.  
La fenêtre de création du fragment combiné apparaît.
3. Indiquez son **Nom** et le **Type d'opérateur d'interaction** si ce n'est pas déjà fait.
4. Cliquez sur **Terminer**.

Un fragment combiné est représenté par un rectangle dont l'angle supérieur gauche affiche le type d'opérateur d'interaction.

Dans l'exemple ci-dessous, un fragment combiné de type optionnel traduit un comportement susceptible de contrarier le séquençement normal (l'ouverture de la porte).



## Type d'opérateur d'interaction

Le type d'opérateur d'interaction conditionne la signification du fragment combiné. Il existe différents types d'opérateurs : seq, alt, opt, break, par, strict, loop, region, neg, assert, ignore et consider.

### **Alternatives**

Alt exprime la possibilité de choisir entre différents comportements possibles en évaluant les conditions de garde associées à chacun des opérandes. Au plus un des opérandes pourra être exécuté.

L'opérande gardé par Else est choisi lorsqu'aucune des autres conditions n'est réalisée.

### **Option**

Opt représente un choix entre l'unique opérande proposé ou aucun.

### **Arrêt (Break)**

Break représente un scénario d'arrêt qui est exécuté à la place du reste du fragment d'interaction englobant.



### **Parallèle**

Par implique que les différents opérandes peuvent être exécutés en parallèle. Les occurrences des événements des divers opérandes d'interaction peuvent être entrelacées de toutes les façons tant que l'ordre imposé par chaque opérande est préservé.

### **Séquence faible (Weak Sequencing)**

Seq désigne un entrelacement faible entre les comportements des opérandes défini par trois propriétés :

- L'ordre des occurrences d'événements à l'intérieur de chacun des opérandes est maintenu dans le résultat.
- Les occurrences d'événements de différentes lignes de vie venant de différents opérandes peuvent apparaître dans n'importe quel ordre.
- Les occurrences d'événements d'une même ligne de vie venant de différents opérandes sont ordonnés de telle sorte que l'occurrence d'événement du premier opérande apparaît avant celle du deuxième.

### **Séquence stricte (Strict Sequencing)**

Strict définit un séquençement strict entre les comportements des opérandes.

### **Négation (Negative)**

Neg représente un opérande invalide.

### **Région critique**

Critical représente une région qui doit être traitée de manière atomique, ce qui signifie que des occurrences d'événement ne peuvent pas être entrelacées avec celles de la région critique.

### **Ignorer / Considérer**

Consider et Ignore nécessitent qu'une liste de messages pertinents soient spécifiés.

Ignorer indique que les types de certains messages sont ignorés dans le fragment combiné.

Consider signifie que seuls certains messages vont être considérés à l'intérieur du fragment combiné. C'est équivalent à définir tous les autres messages comme 'ignorés'.

### **Assertion**

Assert représente une séquence qui est la seule valide pour un message donné.

Ainsi, toute séquence définie par un fragment d'interaction qui commence par les messages qui aboutissent à la séquence définie par le bloc assert et qui continue par un échange de messages ne respectant pas le bloc assert doit être définie comme invalide.

Les assertions sont fréquemment utilisées en combinaison avec les types Ignore et Consider.

### ***Boucle***

Loop permet d'indiquer que l'opérande d'interaction sera répété un certain nombre de fois. Il est possible de spécifier un nombre minimum et un nombre maximum de boucles, ainsi qu'une expression de continuation de la boucle.

## **Opérande d'interaction**

Un opérande d'interaction est contenu dans un fragment combiné et représente un opérande de l'expression donnée par le fragment combiné englobant. Il peut être conditionné par une contrainte d'interaction qui sert de condition de garde.

### ***Créer un opérande d'interaction***

Pour créer un opérande d'interaction :

1. Faites un clic droit sur le fragment combiné qui contient l'opérande d'interaction.
2. Sélectionnez **Nouveau > Opérande d'interaction**.
3. Nommez l'opérande et cliquez sur **OK**.

### ***Créer une contrainte d'interaction***

Pour créer la contrainte d'interaction qui va conditionner l'opérande :

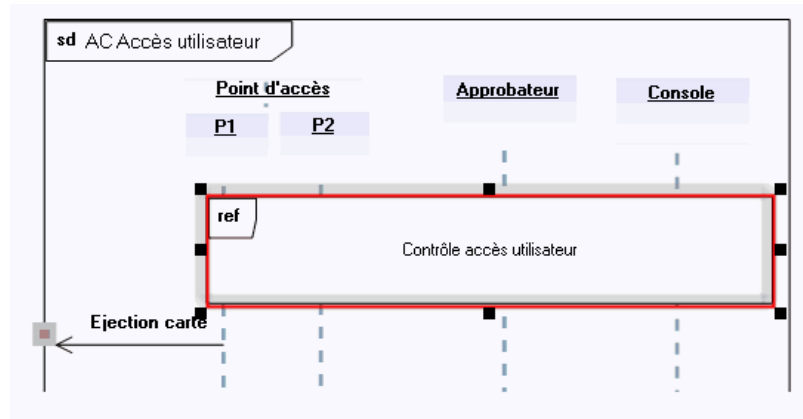
1. Ouvrez la fenêtre de propriétés de l'opérande d'interaction.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le cadre **Condition**, cliquez sur **Nouveau**.
4. La condition est représentée par une contrainte. Définissez la contrainte et cliquez sur **OK**.

---


## **Utilisation d'interaction**

Une utilisation d'interaction se réfère à une interaction. C'est un moyen de copier le contenu de l'interaction référencée à l'endroit de l'occurrence d'interaction.

## Exemple



Pour créer une utilisation d'interaction :

1. Cliquez sur le bouton **Utilisation d'interaction** .
2. Cliquez dans le diagramme.
3. Dans la fenêtre qui apparaît, indiquez son nom et l'interaction appelée.
4. Cliquez sur **Terminer**.

Vous pouvez préciser les arguments d'une utilisation d'interaction. Un argument est une valeur spécifique correspondant à un paramètre de l'interaction appelée. Aussi, une fois l'argument créé sur l'utilisation d'interaction, vous devez le mettre en correspondance avec le paramètre de l'interaction appelée.

Pour créer un argument :

1. Ouvrez la page de propriétés **Caractéristiques** de l'utilisation d'interaction.
2. Dans le cadre **Arguments**, cliquez sur le bouton **Nouveau**.  
Une spécification de valeur est créée.  
Vous pouvez la renommer et préciser ses caractéristiques en ouvrant sa fenêtre de propriétés.

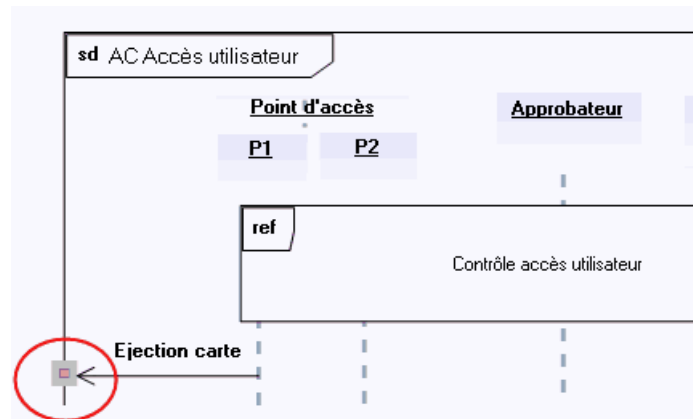
Pour mettre l'argument en correspondance avec le paramètre de l'interaction appelée :

1. Dans la page de propriétés **Caractéristiques** de l'utilisation d'interaction.
2. Cliquez sur la flèche située à l'extrémité du champ **Interaction called** et sélectionnez **Modifier**.  
Une fenêtre affiche les paramètres de l'interaction appelée.
3. Pour chaque paramètre, cliquez dans la colonne valeur et sélectionnez la spécification de valeur qui lui correspond.


## Porte

Une porte (gate) est un point de connexion entre un message extérieur à un fragment d'interaction et un message appartenant à ce fragment d'interaction.

## Exemple



Pour créer une porte dans le diagramme de séquence :

1. Cliquez sur le bouton **Porte**  de la barre d'insertion d'objets.
2. Cliquez sur le cadre délimitant l'interaction, là où vous souhaitez positionner la porte.  
La porte apparaît dans le diagramme.

---

## Continuation

Une continuation est un moyen syntaxique de définir le prolongement des séquences de différentes branches de fragment combiné alternatif. Les continuations sont similaires à des labels représentant des points intermédiaires dans un flot de contrôle.

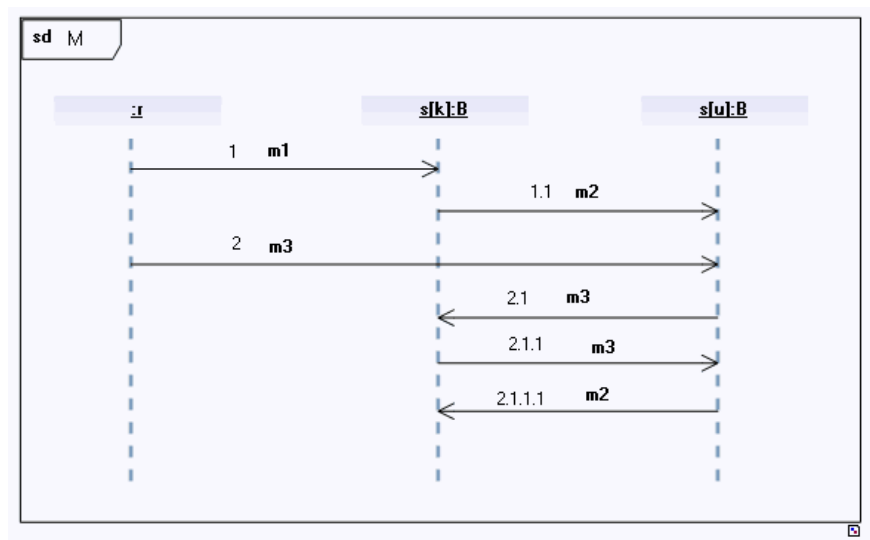
## LE DIAGRAMME DE COMMUNICATION

Le diagramme de communication est une représentation simplifiée d'un diagramme de séquence ; il se concentre sur les échanges de messages entre les objets au sein d'une interaction.

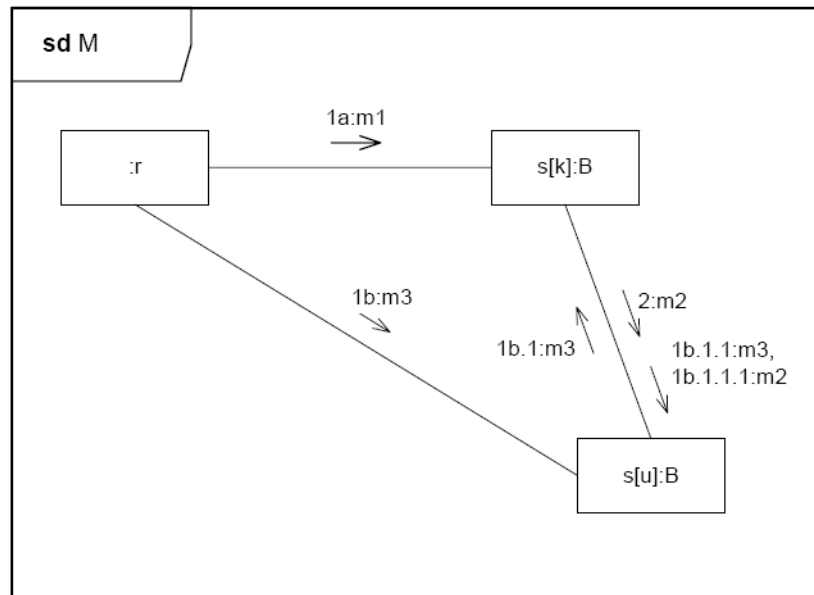
Le diagramme de séquence et le diagramme de communication sont isomorphes. Lorsqu'un diagramme de communication porte sur une interaction déjà décrite dans un diagramme de séquence, il est automatiquement initialisé à partir des informations contenues dans le diagramme de séquence.

### Exemple

#### *Diagramme de séquence*




## Diagramme de communication



### Objets du diagramme

Les objets du diagramme de communication sont les lignes de vie et les messages transmis par des connecteurs.

Lorsque vous reliez deux lignes de vie par un connecteur , la boîte de création du connecteur propose les messages susceptibles d'être transmis.

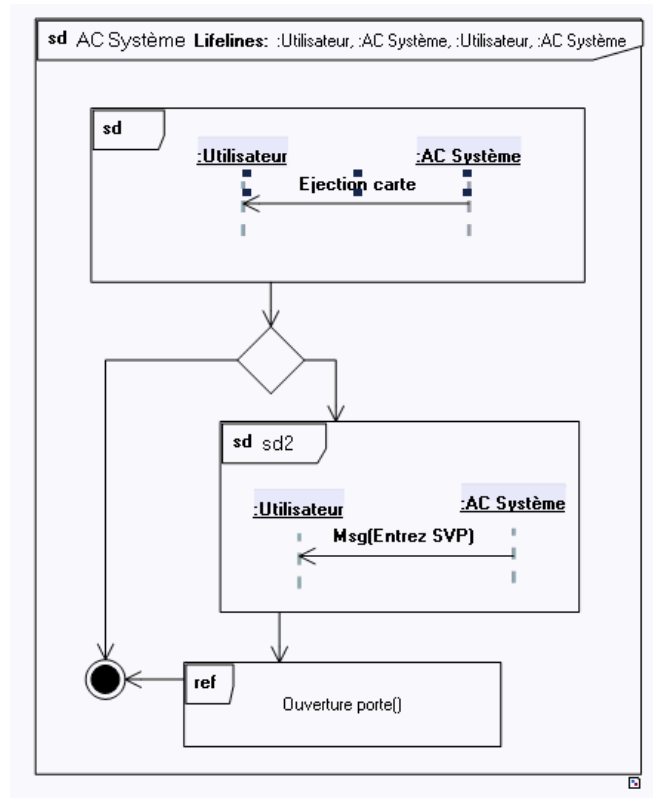
Une fois le connecteur créé, vous pouvez également lui associer de nouveaux messages via sa fenêtre de propriétés, sous l'onglet **Message**.

La séquence des messages est donnée par un numéro de séquence associé aux messages. Voir [Calcul des numéros de séquence](#).

Pour plus de détails sur les connecteurs, voir [Les connecteurs](#).

## LE DIAGRAMME DE VUE GÉNÉRALE D'INTERACTION

Le diagramme de vue générale d'interaction permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences. Il vise à fournir une vue d'ensemble du flux de contrôle.



Les objets représentés dans le diagramme de vue générale d'interaction sont les interactions et utilisations d'interaction, les lignes de vie, les messages, les noeuds de contrôle et les flux de contrôle.





# LE DIAGRAMME DE DÉPLOIEMENT



Le diagramme de déploiement complète le diagramme de composants en mettant en avant les ressources matérielles sur lesquelles s'exécutent les composants.

- ✓ [Présentation du diagramme de déploiement.](#)

# PRÉSENTATION DU DIAGRAMME DE DÉPLOIEMENT

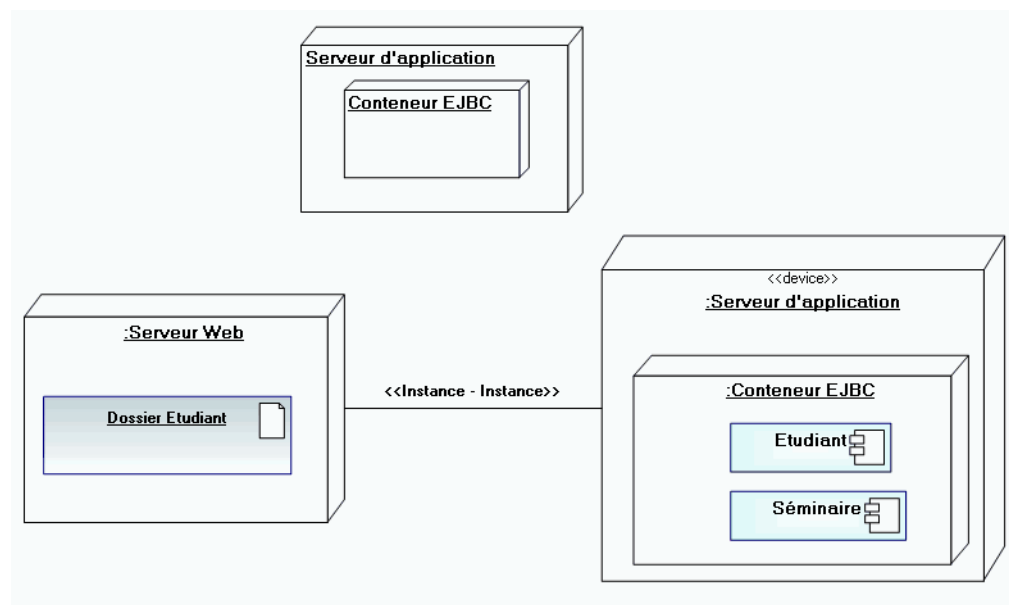
Le diagramme de déploiement complète le diagramme de composants. Il décrit les ressources matérielles (ordinateur, routeur etc.) qui composent le système et montre la répartition des composants sur ces matériels.

Il décrit également les connexions entre les composants ou les nœuds.

Ce diagramme permet également de préciser les interfaces requises et implémentées pour l'enchaînement des composants.

Il peut être illustré et complété par l'ajout d'instances de noeud, de composant ou de classe.

## Exemple de diagramme de déploiement



## Créer un diagramme de déploiement

Dans **HOPEX IT Architecture**, un diagramme de déploiement se crée à partir d'un paquetage.

Pour créer un diagramme de déploiement dans **HOPEX IT Architecture** à partir du menu de navigation **Conception (UML)** :

1. Cliquez sur le sous-menu **Paquetages**.
2. Sélectionnez la ligne du paquetage concerné et cliquez sur le bouton **Nouveau Diagramme**.

3. Sélectionnez **Diagramme de déploiement**.  
Le nouveau diagramme de déploiement s'ouvre dans la fenêtre d'édition.

---

## Objets du diagramme de déploiement


### Noeud

Un nœud est un objet physique représentant une ressource informatique disposant généralement d'une mémoire et souvent de capacités de calcul et sur lesquels des composants peuvent être déployés

Les nœuds peuvent se composer d'autre nœuds ou d'artefacts. Pour montrer qu'un composant est affecté à un nœud, il faut soit placer le composant dans le nœud, soit relier le composant au nœud par une relation de dépendance.

Voir [Les liens de dépendance](#).

Vous pouvez créer un nœud dans le diagramme de déploiement à l'aide du bouton

**Nœud (UML)**  de la barre d'insertion d'objets.

### Chemin de communication

Les connexions entre nœuds sont représentées par des chemins de communication par lesquels sont échangés des signaux et des messages.


### Composant

Un composant représente une partie modulaire d'un système qui encapsule son contenu et qui est remplaçable dans son environnement. Un composant définit son comportement par les interfaces qu'il fournit et celles qu'il requiert.


Un composant peut être remplacé par un autre si leurs interfaces sont conformes.

Un composant peut être un logiciel, un programme, un élément de code, etc.

### Artefact

Un artefact  représente un élément d'information physique qui est utilisé ou produit par le processus de développement d'un logiciel, ou par le déploiement ou la mise en œuvre d'un système. Ex: fichiers sources, scripts, fichiers binaires exécutables, les livrables issus d'un développement, un document produit par un traitement de texte, un message électronique, etc.

### Manifestation


Une manifestation  est la restitution physique concrète dans un artefact d'un ou de plusieurs éléments de modélisation tels que des composants ou des classes.

Une dépendance de manifestation a pour source un artefact et pour cible un composant ou une classe.

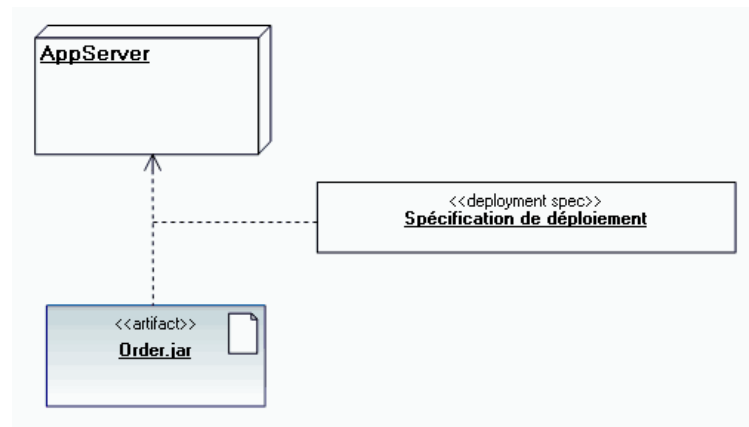
## Spécification de déploiement

La spécification d'un déploiement permet d'indiquer l'ensemble des caractéristiques qui déterminent les paramètres d'exécution d'un artefact ou d'un composant déployé sur un nœud.

## Configuration

Le bouton Configuration  permet de créer le lien entre une spécification de déploiement et un déploiement.

### Exemple



# ANNEXE : TYPE DES ATTRIBUTS



Les points suivants sont abordés ici :

- ✓ Types élémentaires
- ✓ Paquetages et types élémentaires
- ✓ Définir de nouveaux types élémentaires

## TYPES ÉLÉMENTAIRES

Un type élémentaire permet de mettre en commun des caractéristiques communes à plusieurs attributs. Les types élémentaires sont implémentés sous forme de classe.

---

### Conditions préalables : importer les types élémentaires

Pour disposer des types élémentaires dans **HOPEX IT Architecture**, l'administrateur doit importer le module "ISQL ANSI" dans votre environnement. Pour importer un module dans **HOPEX**, voir [Importer un module dans HOPEX](#).

---

### Définir un type élémentaire

Les types élémentaires sont définis dans un diagramme de classes.

Ce sont des classes pour lesquelles on précise les points suivants :

- Ce sont des classes de stéréotype "Type élémentaire".
- Ce sont des classes "Abstraites" car elles ne sont pas destinées à être instanciées.
- Ce sont des classes "Non persistantes". Elles ne doivent en effet pas donner lieu à une table dans la base de données.

Pour indiquer le type des attributs des classes :

1. Dans la fenêtre de propriétés de la classe, cliquez sur la page **Caractéristiques internes**.
2. Dépliez la section **Attributs**.
3. Cliquez dans le champ **Type expression** et sélectionnez le type de l'attribut à l'aide de la flèche.

Les classes proposées en standard sont :

Types alphanumériques		Compléments
M-Char	Chaîne de caractères alphanumériques de taille fixe	Longueur
M-Varchar	Chaîne de caractères alphanumériques de taille variable	
<b>Types numériques</b>		
M-Numeric	Numérique	Longueur, Décimale

M-Amount	Montant exprimé en monnaie	Longueur, Décimale
<b>Types dates</b>		
M-Date	Date	
M-Time	Heure	
M-Datetime	Date et heure	
<b>Types binaires</b>		
M-Timestamp	Identification générée automatiquement à partir de la date et de l'heure exprimée en millièmes de secondes après le 01 Janvier 1970	
M-Bool	Booléen valant 0 ou 1	
M-Multimedia	Chaîne binaire	

## PAQUETAGES ET TYPES ÉLÉMENTAIRES

---

### Paquetages



*Un paquetage partitionne le domaine d'étude et les travaux associés. Il permet de regrouper divers éléments, en particulier des cas d'utilisations et des classes. Un paquetage peut aussi contenir d'autres paquetages. Les paquetages sont liés entre eux à travers des rapports contractuels définissant leur interface.*

L'affectation des classes à des paquetages est extrêmement structurante. En effet, comme une classe ne peut être détenue que par un seul paquetage, il est nécessaire de définir des liens clients - fournisseurs entre les paquetages de façon à ce que les paquetages qui en ont besoin puissent utiliser les classes dont ils ne sont pas détenteurs.

Ceci est particulièrement important pour les classes de type élémentaire puisqu'elles vont être utilisées pour définir les attributs des autres classes.



*Règle : Une classe est détenue par un paquetage et un seul.*

Les types élémentaires disponibles pour les attributs d'une classe dépendent du paquetage qui la détient.

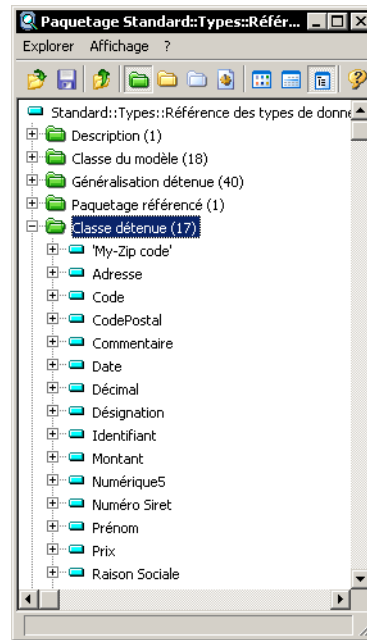
Pour typer les attributs d'une classe, on proposera les types élémentaires définis pour le paquetage qui contient cette classe.

Les types élémentaires disponibles sont les classes publiques de stéréotype "Type élémentaire" détenues ou utilisées par ce paquetage ou par les paquetages dont il est client.

On peut ainsi définir un paquetage de référence (ou plusieurs) détenant les types élémentaires utilisés dans l'entreprise. Chacun des autres paquetages sera déclaré client du paquetage de référence des types élémentaires.



Dans l'exemple ci-dessous, le paquetage "Référence des types de données" détient les classes "Adresse", "Code", "Date", etc.



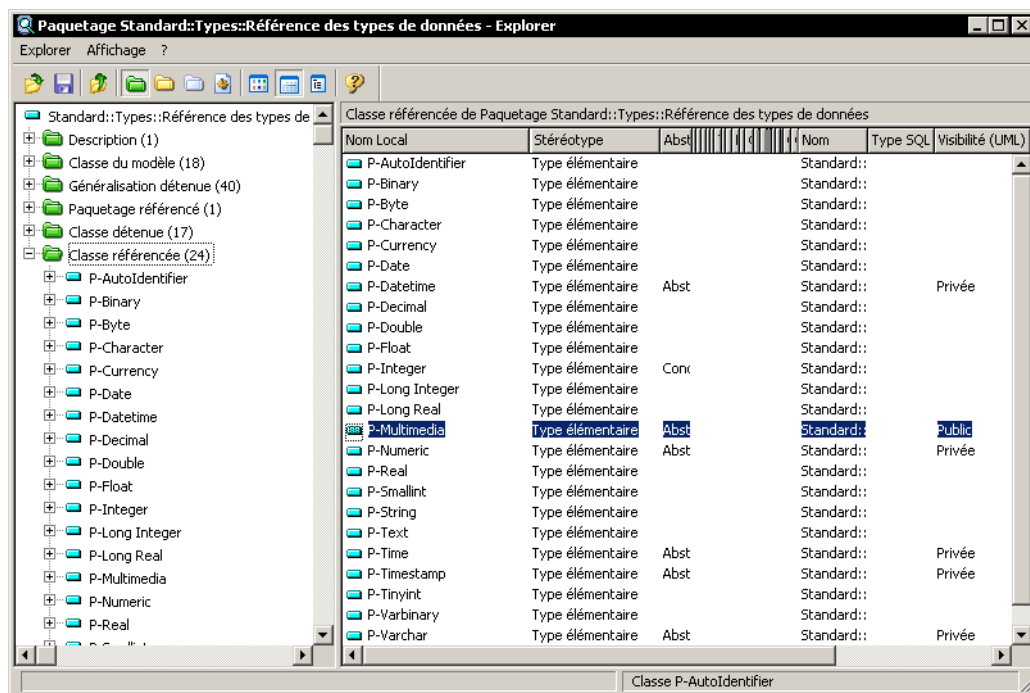
Il est référencé par les paquetages "Bibliothèque", "Gestion des commandes" etc.

Les attributs des classes de ces paquetages peuvent donc être typés à l'aide des types « Adresse », « Code », « Date », etc.

Il est également possible de préciser directement qu'un paquetage référence une classe détenue par un autre paquetage.

Dans l'exemple ci-dessous, les classes « P-Datetime », « P-Multimedia », « P-Numeric »c", etc., sont référencées par le paquetage « Référence des types de données » sans qu'il les détienne.

Parmi celles-ci, seule la classe « M-Multimedia » est rendue publique par ce paquetage.



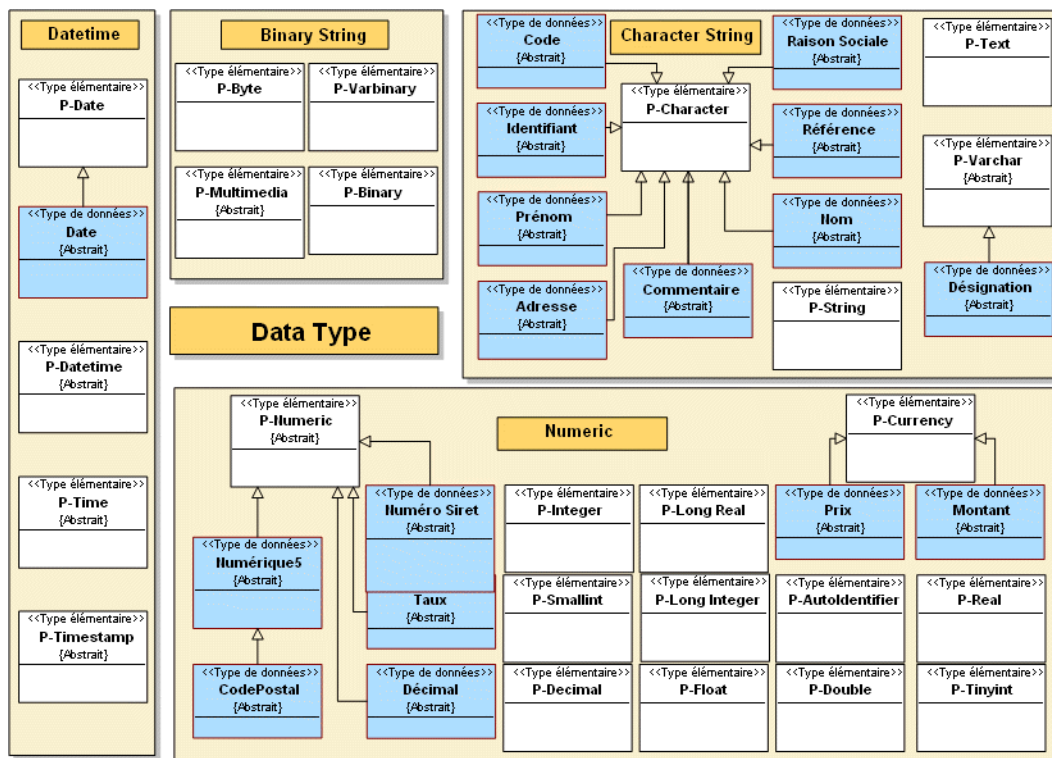
## DÉFINIR DE NOUVEAUX TYPES ÉLÉMENTAIRES

De nouveaux types élémentaires peuvent être définis à l'aide d'un diagramme de classes.

Ce diagramme de classes pourra décrire, selon que l'on aura choisi ou non la structuration des classes dans des paquetages :

- Une base de données de référence.
- Le paquetage des types de référence.

Vous pouvez définir vos propres types élémentaires en les déclarant sous-classes des types élémentaires proposés en standard comme dans l'exemple ci-dessous :

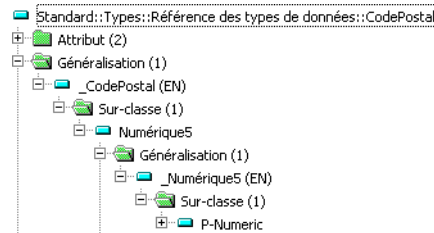


Les types élémentaires définis comme sous-classes vont hériter automatiquement des caractéristiques de leur super-classe. En particulier, la règle de transformation en datatype de la super-classe est appliquée à la sous-classe.

Il est possible de préciser sur la sous-classe une longueur et un nombre de décimales. Ceux-ci seront pris en compte pour la génération des datatypes s'ils n'ont pas déjà été définis pour la super-classe.

L'héritage peut se faire sur plusieurs niveaux.

Dans l'exemple suivant, le type élémentaire "CodePostal" est une spécialisation du type "Numérique5" de longueur 5, lui-même spécialisation du type standard "P-Numeric".

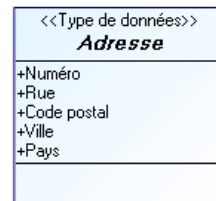


Si le nouveau type élémentaire n'est pas défini directement ou indirectement comme sous-classe d'un type élémentaire standard, il est nécessaire de mettre à jour le tableau de conversion des types élémentaires en datatypes de colonnes.

☛ Une correspondance peut également être définie directement entre un type et le datatype SQL généré pour chaque SGBD cible sans utiliser le mécanisme d'héritage (voir "Correspondances entre types pivots et datatypes" dans le guide **HOPEX Database Builder**).

## Type élémentaire composé

On peut définir un type élémentaire composé en lui précisant une liste d'attributs.

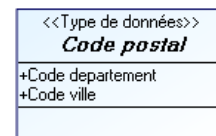
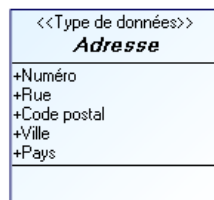


Ici le type Adresse est composé du numéro, de la rue, du code postal, de la ville et du pays.

Un attribut de type Adresse donnera lieu lors de la dérivation à ces cinq colonnes.

Il est possible de décomposer un type à plusieurs niveaux en affectant un type décomposé à l'un de ses attributs.

Par exemple, on peut décomposer le code postal en code ville et code département :



HOPEX XMI 2.1 Import for UML2

## XMI IMPORT OVERVIEW

---

The XML Metadata Interchange XMI is an OMG standard for exchanging UML Models between different UML products such as modeling tools and UML Design.

The XMI Import project aims at importing the content of .xmi and .uml files into HOPEX so that users can reproduce diagrams from other platforms. Only the data (the objects) are imported, not the drawings.

### Prerequisites

The XMI Import feature supports UML versions from 2.3 to 2.5. A file with a version lower than 2.3 or higher than 2.5 can be imported, without guarantee of full success.

### Scope of XMI Import

The purpose of the XMI Import tool is to import XMI data into HOPEX repository. The objects imported are those belonging to the Class Diagram, Use Case Diagram, Component Diagram, Composite Structure Diagram, Activity Diagram, Communication Diagram, Sequence Diagram, State Machine Diagram, Interaction Overview Diagram, Object Diagram, Deployment Diagram, etc.

See object details in [HOPEX/XMI Object Mapping](#).

🚫\* **Only objects are imported. The XMI Import tool does not take into account UML profiles, extensions and graphical diagram drawings.**

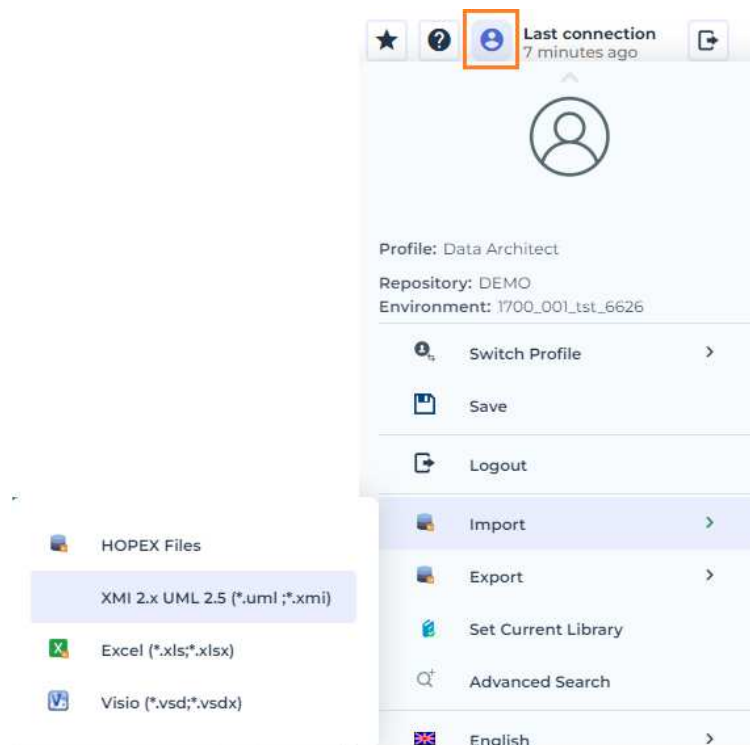
## IMPORTING XMI AND UML FILES

---

Depending on the source tool, the XMI import tool can import .xmi or .uml files.

To import a file:

1. In HOPEX, select **Main Menu > Import > XMI 2.x UML 2.5 (\*.uml; \*.xmi)**.



The import wizard appears.

2. In **File Location**, select the file to be imported.
3. Select the library in which you want to import the data (optional).
4. Click **Next**.

The wizard shows the import process progression.

Then it shows the report of imported data.

## HOPEX/XMI OBJECT MAPPING

---

The following paragraph indicates what kinds of objects are imported by UML2 diagram types. Only objects belonging to the selected package are imported.

All objects that do not belong to a package are attached to a package called “Default Package”.

A package “UML Primitive Types” is imported by default if not already.

### **Class Diagram**

Classes, attributes, associations, association ends, generalizations, generalization sets, operations, parameters, data types, primitive types, interfaces, enumerations, etc.

### **Use Case Diagram**

Use cases, actors, packages, constraints, extension points (text), participations, extensions (link), inclusions (link), generalization, dependencies, etc.

### **State Machine Diagram**

State machines, regions, states, pseudo states, transitions, constraints, etc.

### **Protocol State Machine Diagram**

Protocol state machines, regions, states, pseudo states, transitions, constraints, etc.

### **Activity UML Diagram**

Actions, control nodes, Input Pins, Output Pins, Exchange Pins, central buffer nodes, data store nodes, activity partitions, control flows, object flows, exception handlers, activities uml, activity parameter nodes, structured activity nodes, expansion regions, expansion nodes, interruptible activity regions, etc.

### **Component Diagram**

Components, ports, packages, interfaces, required interfaces, provided interfaces, classes, Connectors, realized elements, etc.

### **Composite Structure Diagram**

Collaborations UML, collaboration uses, parts, dependencies, connectors, interfaces, classes, provided interfaces, required interfaces, etc.



## Sequence Diagram

Life lines, combined fragments, interaction uses, gates, states invariant, UML messages, constraints, etc.

---

Messages include those exchanged directly between lifelines as well as messages exchanged through execution specification.

---

## Communication Diagram

Life lines, connectors, UML messages, etc.

## Deployment Diagram

Packages, components, artifacts UML, nodes UML, devices, execution environments, interfaces, deployment specifications, deployments, manifestations, deployment configurations, component instances, device instances, node instances, execution environment instances, communication paths, etc.

🔗\* Only objects owned by the selected package or its sub-packages are exported.

Objects that are linked to objects contained in the selected export package but owned by another package are also exported in order to ensure links. However, they will be owned by the exported package.

The following table indicates concepts managed by the export tool:

## Class Diagram

MEGA Concepts	MetaAttribute	MetaAssociation(End)
Class	Name	Class Target Dependency
	xmi_id	Realization Class
	Visibility	Nested Class
	Comment	Association
	Abstract	Connector
	IsLeaf	Association Class
	IsActive	Attribute
	Client Dependency	Operation (UML)
		Generalization
		Required Interface
		Provided Interface
		Constraint
		Port
		AssociationEnd
		Owned Part
		Behavior: State Machine
		Behavior: Activity Uml
		Behavior: Interaction Uml
		Behavior: Collaboration Uml
		Protocol: ProtocolStateMachineDiagram
		Source Dependency
Data Type	Name	Class Target Dependency

MEGA Concepts	MetaAttribute	MetaAssociation(End)
	xmi_id Visibility Abstract IsLeaf IsActive Client Dependency	Nested Class
Interface	xmi_id Name Visibility Comment Abstract IsLeaf Client Dependency	Class Target Dependency Nested Class Association Attribute Operation (UML) Generalization RequiredInterface SpecificationInterface
Enumeration	_Hexaidabs Name Visibility Comment Client Dependency	Class Target Dependency Attribute Operation (UML) Literal Value RequiredInterface Specification Interface
LiteralValue	_Hexaidabs Name	Value Slot

MEGA Concepts	MetaAttribute	MetaAssociation(End)
Expression	xmi_id Name Visibility Comment Client Dependency	Class Target Dependency Specification Interface
Primitive Type	Name xmi_id Visibility Abstract IsLeaf IsActive	Class Target Dependency Nested Class
Association	xmi_id Name Visibility Comment IsAssociationDerived IsNavigable	Connection Dependency (Target Association) Class via AssociationEnd
Association End	xmi_id Name Aggregation: Composite/Shared	Association  Dependency
Association Class	xmi_id	Class Target Dependency

MEGA Concepts	MetaAttribute	MetaAssociation(End)
	Name Visibility Comment IsLeaf Abstract IsActive IsAssociationDerived IsNavigable	Nested Class AssociationEnd Association Association Class Class via AssociationEnd Attribute Operation (UML)
Attribute	xmi_id Name Visibility Comment IsLeaf IsOrdred Uniqueness ReadOnly IsDerived InitialValue Multiplicity: UpperValue, LowerValue	Dependency (Target Attribute) AttributType
Operation (UML)	xmi_id Name Visibility Comment	Precondition Postcondition Parameter ReturnType

MEGA Concepts	MetaAttribute	MetaAssociation(End)
Dependency	Abstract	Target Dependency
	IsQuery	
Dependency	_Hexaidabs	Class Source
	Name	Class Target
	Visibility	Stereotype
	Comment	
Generalization	xmi_id	Super Class
	Name	UML constraint
	Comment	
GeneralizationSet	xmi_id	Generalization
	Name	
	Comment	
	IsComplete	
	IsDisjoint	
Constraint	xmi_id	ConstrainedClass
	Name	ConstrainedGeneralization
	Comment	ConstrainedElement
	MaxInt	Actor (UML)
	MinInt	Package
	Specification	UseCase
		UseCaseParticipation

MEGA Concepts	MetaAttribute	MetaAssociation(End)
Parameter	_Hexaidabs Name Comment	Parameter Type
Behavior (UML)	_Hexaidabs Name	

## Use Case Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
UseCase	xmi_id Name Visibility Comment	UsesUseCase OwnedExtension ExtensionPoint Behavior: State Machine Behavior: Protocol State Machine Behavior: Interaction UML Behavior: Activity UML Behavior: Collaboration UML Constraint Generalization
Actor (UML)	xmi_id Name Visibility Comment	Participation Constraint Generalization
Participation	xmi_id	UseCase

	Name	Actor (UML)
	Comment	Constraint
Extension	Multiplicity	
	xmi_id	Extended Use Case
	Name	Extension Location
	Comment	

## Composite Structure and Communication Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Collaboration uml	xmi_id	CollaborationRole
	Name	OwnedConnector
	Comment	OwnedCollaborationUse
	IsAbstract	
	IsLeaf	
Collaboration use	xmi_id	Type
	Name	
	Comment	
Part	xmi_id	ConnectorEnd (of the LifeLine who represents the part)
	Name	Dependency
	Visibility	
	Client Dependency	
	IsLeaf	
	IsUnique	



	IsOrdered Multiplicity Aggregation: Composite/Shared Comment	
Connector	xmi_id Name Connector Kind IsLeaf	OwnedConnectorEnd
ConnectorEnd	xmi_id Name Multiplicity	Connector

## State Machine

MEGA Concepts	MetaAttribut	MetaAssociation(End)
State Machine	xmi_id Name Comment Reentrant	DetailedState Region
Region	xmi_id Name Comment	State PseudoState Transition
State (UML)	xmi_id	Detailing Behavior

	Name Comment	Outgoing Incoming OwnedRegion DoActivity ExitActivity EntryActivity ConnectionPoint (Entry Point/ Exit Point)
Pseudo State	xmi_id Name Comment PseudoStateKind	Outgoing Transition Incoming Transition
Transition (UML)	xmi_id Name Comment	Source Target Source Pseudo State Target Pseudo State Trigger Effect (Behavior) Constraint
Event (UML)	xmi_id Name Comment	EventKind
Protocol State Machine	xmi_id Name Visibility	Region

	Comment	
Trigger (UML)	xmi_id	Event (UML)
	Name	
	Comment	

### Sequence, Communication and Interaction Overview Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Interaction UML	xmi_id Name Comment	Gate Fragment (Combined Fragment, State Invariant, ...) LifeLine Message Action Parameter Operation (UML) OwnedInteraction (UML) OwnedInteractionOperand
Interaction Operand	xmi_id Name Comment	Fragment (Combined Fragment, State Invariant, ...) OwnedInteraction (UML) OwnedInteractionOperand
OccurrenceSpecification	xmi_id Name Comment	Event (UML) Message
ExecutionSpecification	xmi_id Name	start finish

	Comment	
LifeLine	xmi_id Name Comment	ElementRepresentedByALifeline ElementCoveringLifeline OwnedSelector
Combined Fragment	xmi_id Name  InteractionOperatorKind Comment	InteractionOperand CoveredLifeLine
Interaction Use	xmi_id Name Comment	RefersTo CoveredLifeLine
State Invariant	xmi_id Name Comment	InvariantConstraint CoveredLifeLine
Gate	xmi_id Name Comment	
Message UML	xmi_id Name MessageKind MessageSort Comment	Receiver Sender Connector

## Activity and Interaction Overview Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Action	xmi_id Name IsLeaf ActionKind Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow OwnerGroup RequestOperation CalledBehavior StructuralFeatureElementManagedByAnAction RequestSignal Association ClassManagedByAnAction Variable InputPin OutputPin ProtectingExceptionHandler Trigger LocalPostCondition LocalPreCondition Constraint
Control Node	xmi_id Name IsLeaf Comment ControlNodeType	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow OwnerGroup

Input Pin	xmi_id Name IsLeaf InputPinKind ControlType OrderingKind Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow Constraint
Output Pin	xmi_id Name IsLeaf OutputPinKind ControlType OrderingKind Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow Constraint
Exchange Pin	xmi_id Name IsLeaf ControlType OrderingKind Comment	OutgoingObjectFlow IncomingObjectFlow OwnerGroup
Central Buffer Node	xmi_id Name IsLeaf	OutgoingObjectFlow IncomingObjectFlow OwnerGroup

	ControlType OrderingKind Comment	
Data Store Node	xmi_id Name IsLeaf ControlType OrderingKind Comment	OutgoingObjectFlow IncomingObjectFlow OwnerGroup Constraint
Activity Partition	xmi_id Name IsDimension IsExternal Comment	ContainedElement Constraint
Object Flow	xmi_id Name IsLeaf Comment	Guard Weight SourceElement TargetElement Constraint
Control Flow	xmi_id Name IsLeaf Comment	Guard Weight SourceElement TargetElement

Exception Handler	xmi_id	Constraint
	Name	ProtectedNode
	Comment	ExceptionInput Constraint
Activity UML	xmi_id	ElementOwnedByAnActivityUML
	Name	Constraint
	Reentrant	
	SingleExecution	
	IsLeaf	
	Comment	
Activity Parameter Node	xmi_id	OutgoingObjectFlow
	Name	IncomingObjectFlow
	IsLeaf	Constraint
	ControlType	
	OrderingKind	
	Comment	
Structured Activity Node	xmi_id	OutgoingControlFlow
	Name	OutgoingObjectFlow
	IsLeaf	IncomingControlFlow
	MustIsolate	IncomingObjectFlow
	Comment	OwnerGroup
		ContainedElement InputPin



		OutputPin Constraint
Expansion Node	xmi_id Name IsLeaf ControlType OrderingKind Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow Region Constraint
Expansion Region	xmi_id Name IsLeaf MustIsolate ExpansionKind Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow OwnerGroup ContainedElement ExpansionNode InputElement OutputElement
Interruptible Activity Region	xmi_id Name Comment	OwnerGroup ContainedElement Constraint
Loop Node	xmi_id Name IsLeaf	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow

	MustIsolate TestedFirst Comment	IncomingObjectFlow OwnerGroup ContainedElement InputPin OutputPin Constraint Test
Conditional Node	xmi_id Name IsLeaf MustIsolate Assured Determinate Comment	OwnerGroup ContainedElement InputPin OutputPin Constraint OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow
Sequence Node	xmi_id Name IsLeaf MustIsolate Comment	OutgoingControlFlow OutgoingObjectFlow IncomingControlFlow IncomingObjectFlow OwnerGroup ContainedElement InputPin OutputPin Constraint

## Package Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Package	Name xmi_id Visibility Comment	Client Dependency Package Target Dependency Owned Class Owned Package Owned Association Association Class Owned Dependency Owned Element (UML): Generalization Owned Use Case Owned Actor (UML) Constraint Behavior: State Machine Behavior: Protocol State Machine Behavior: Activity Uml Behavior: Interaction Uml Behavior: Collaboration Uml Owned Component Owned Event

## Component Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Component	xmi_id Name Comment	required Interface provided Port

	Isleaf Visibility Client Dependency	OwnedPart
Port	xmi_id Name Comment Client Dependency	
Interface	xmi_id Name Visibility Comment Abstract IsLeaf Client Dependency	Class Target Dependency Nested Class Association Association Class Attribute Operation (UML) Generalization RequiredInterface SpecificationInterface

## Deployment Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Artifact UML	xmi_id Name Comment Client Dependency	Target Dependency OwnedAttribute OwnedOperation NestedArtifact
Node UML	xmi_id	Deployment

	Name Comment Client Dependency	
Device	xmi_id Name Comment Client Dependency	Deployment
Execution Environment	xmi_id Name Comment Client Dependency	Deployment
Deployment Specification	xmi_id Name Comment Client Dependency	Target Dependency
Instance ( Node UML/Device/Execution Specification/Component)	xmi_id Name Comment Client Dependency	Instantiated Element Target Dependency
Communication Path	xmi_id Name Comment	Communication Path End

Deployment	xmi_id Name Comment	Deployed Element Deployment Configuration
Manifestation	xmi_id Name Comment	Multiplicity Deployed Element Deployment Configuration

## Object Diagram

MEGA Concepts	MetaAttribut	MetaAssociation(End)
Instance	xmi_id Name Comment	
Link	xmi_id Name Comment	LinkEnd
LinkEnd	xmi_id Name Comment	Instance

HOPEX XMI 2.1 Export for UML2

## XMI EXPORT OVERVIEW

---

The XML Metadata Interchange XMI is an OMG standard for exchanging UML Models between different UML products such as modeling tools and UML Design.

The XMI 2.1 Export project aims at exporting the content of HOPEX Diagrams as .xmi files so that models modeled in HOPEX can be imported by UML tools such as Eclipse EMF.

### Prerequisites

The XMI 2.1 export feature is available with HOPEX UML, and supports XMI version 2.1 with UML 2.3.

### Scope of XMI Export

The purpose of XMI export is to translate the specification of HOPEX Class Diagrams, Use Case Diagrams, Component Diagram, Composite Structure Diagram, Activity Diagram, Communication Diagram, Sequence Diagram and State Machine Diagrams into XMI. Diagrams and diagrams drawings are not considered except with UML2 plugin for Eclipse.

The tool handles translation of the concepts of the above HOPEX diagrams that have a correspondence in UML 2.0. The list of supported mappings is detailed below.



## EXPORTING XMI FILES

---

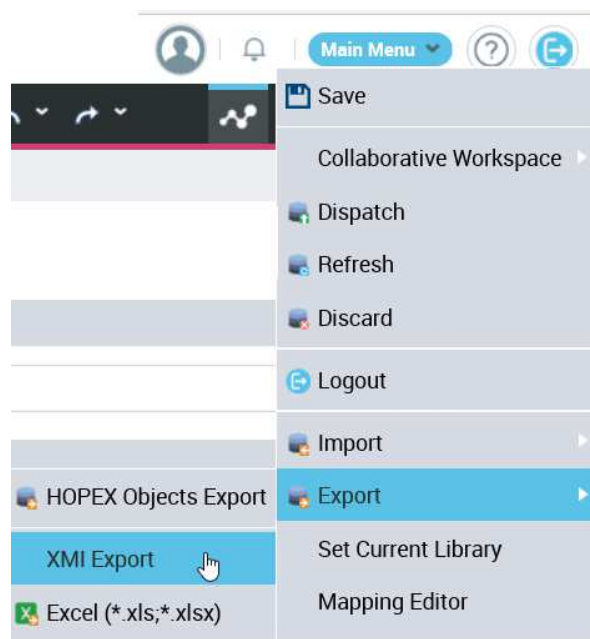
Depending on the destination tool, the XMI export tool produces two types of file. For Eclipse with UML plugin, the export will produce one .uml file for data and some .umlclass, .umlusc and/or .umlstm files for diagrams (each file represents an HOPEX diagram).

For other modeling tools, as Enterprise Architect or MagicDraw UML, the export will produce one .xmi file for data but no diagram description files will be generated.

### Export for Eclipse with UML2 plugin

To export HOPEX data for Eclipse with UML2 plugin:

1. In HOPEX, select **Main Menu > Export > XMI Export**.



The export dialog box appears.

2. Select the **Package** to be exported.
3. Specify the name and path of the file to be exported.
4. Under **Options**, check the **Export for Eclipse** parameter.
5. Click **Next**.

The window shows the export process progression.

Then the window showing the report of all exported data appears.

## Export for other tools

Because many modeling tools do not support the UML Diagram Interchange Specification, the MEGA XMI 2.1 Export feature exports .xmi file for data but no diagram description files for other tools than Eclipse.

To export HOPEX data for modeling tools such as Enterprise Architect or MagicDraw UML:

1. In HOPEX, select **Main Menu > Export > XMI Export**.

The export dialog box appears.

2. Select the **Package** to be exported.
3. Specify the name and path of the file to be exported.
4. **Uncheck the Export for Eclipse** parameter.
5. Click **Next >**.

The window that appears shows the export process progression.

Then the window showing the report of all exported data appears.

## HOPEX/XMI OBJECT MAPPING

---

The XMI export feature translates a class diagram or use case diagram or state machine diagram or even protocol state machine diagram specified in HOPEX into an XMI compliant output file.

The following paragraph indicates what kinds of objects are exported by UML2 diagram types. Only objects belonging to the selected package are exported.

### **Class Diagram**

Packages, classes, interfaces, enumerations, literal strings (expression text), associations, association roles, generalizations, constraints, required interfaces (link), provided interfaces (supported interface link), data types (class stereotype), primitive types (class stereotype), attributes, operations.

### **Use Case Diagram**

Use cases, actors, packages, constraints, extension points (text), participations, extensions (link), inclusions (link), generalization, dependencies.

### **State Machine Diagram**

State machines, regions, states, pseudo states, transitions, constraints

### **Protocol State Machine Diagram**

Protocol state machines, regions, states, pseudo states, transitions, constraints

### **Activity UML Diagram**

Actions, control nodes, Input Pins, Output Pins, Exchange Pins, central buffer nodes, data store nodes, activity partitions, control flows, object flows, exception handlers, activities uml, activity parameter nodes, structured activity nodes, expansion regions, expansion nodes, interruptible activity regions.

### **Component Diagram**

Components, ports, packages, interfaces, required interfaces, provided interfaces, classes, Connectors, realized elements.

### **Composite Structure Diagram**

Collaborations UML, collaboration uses, parts, dependencies, connectors, interfaces, classes, provided interfaces, required interfaces.

## Sequence Diagram

Life lines, combined fragments, interaction uses, gates, states invariant, messages UML, constraints.

---

Messages include those exchanged directly between lifelines as well as messages exchanged through execution specification.

---

## Communication Diagram

Life lines, connectors, messages UML.





















## Deployment Diagram

Packages, components, artifacts UML, nodes UML, devices, execution environments, interfaces, deployment specifications, deployments, manifestations, deployment configurations, component instances, device instances, node instances, execution environment instances, communication paths.

🔗\* Only objects owned by the selected package or its sub-packages are exported.

Objects that are linked to objects contained in the selected export package but owned by another package are also exported in order to ensure links. However, they will be owned by the exported package.

The following table indicates concepts managed by the export tool:

HOPEX Concepts	
	Package
	Name
	_Hexaidabs
	Visibility
	Comment
	Client Dependency
	Package Target Dependency
	Owned Class
	Owned Package
	Owned Association
	Association Class
	Owned Dependency
	Owned Element (UML): GeneralizationSet
	Owned Use Case
	Owned Actor (UML)
	Constraint
	Behavior: State Machine
	Behavior: Protocol State Machine
	Behavior: Activity Uml
	Behavior: Interaction Uml

## HOPEX Concepts

## 👤 Behavior: Collaboration Uml

Owned Component

 Owned Event

#### Class

■ Name

 **\_Hexaidabs**

 Visibility

 Comment

## Abstract

IsLeaf

IsActive

■ Client Dependency

### Class Target Dependency

## Realization Class

## Nested Class

## Association

## Connector
















### Association Class

 Attribute

## 🔗 Operation (UML)

## Generalization

### Required Interface

HOPEX Concepts	
	Provided Interface
	Constraint
	Method
	Port
	AssociationEnd
	Owned Part
	Behavior: State Machine
	Behavior: Activity Uml
	Behavior: Interaction Uml
	Behavior: Collaboration Uml
	<b>Data Type</b>
	Name
	_Hexaidabs
	Visibility
	Abstract
	IsLeaf
	IsActive
	Client Dependency
	Class Target Dependency
	Nested Class
	<b>Interface</b>
	_Hexaidabs

## HOPEX Concepts

- ☐ Name
- ☐ Visibility
- ☐ Comment
- ☐ Abstract
- ☐ IsLeaf
- ☐ Client Dependency
- ☒ Class Target Dependency
- ☒ Nested Class
- ☒ Association
- ☒ Association Class
- ☒ Attribute
- ☒ Operation (UML)
- ☒ Generalization
- ☒ RequiredInterface
- ☒ SpecificationInterface























## ☒ Enumeration













- ☐ \_Hexaidabs
- ☐ Name
- ☐ Visibility
- ☐ Comment
- ☐ Client Dependency
- ☒ Class Target Dependency















































HOPEX Concepts
<ul style="list-style-type: none"> <li>Attribute</li> <li>Operation (UML)</li> <li>Literal Value</li> <li>RequiredInterface</li> <li>Specification Interface</li> </ul>
<b>Expression</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Visibility</li> <li>Comment</li> <li>Client Dependency</li> <li>Class Target Dependency</li> <li>Specification Interface</li> </ul>
<b>Primitive Type</b> <ul style="list-style-type: none"> <li>Name</li> <li>_Hexaidabs</li> <li>Visibility</li> <li>Abstract</li> <li>IsLeaf</li> <li>IsActive</li> <li>Class Target Dependency</li> <li>Nested Class</li> </ul>























HOPEX Concepts	
<div> <div> <div></div> <div>Association</div> </div> <div> <div></div> <div>_Hexaidabs</div> </div> <div> <div></div> <div>Name</div> </div> <div> <div></div> <div>Visibility</div> </div> <div> <div></div> <div>Comment</div> </div> <div> <div></div> <div>IsAssociationDerived</div> </div> <div> <div></div> <div>IsNavigable</div> </div> <div> <div></div> <div>Connection</div> </div> <div> <div></div> <div>Dependency (Target Association)</div> </div> <div> <div></div> <div>Class via AssociationEnd</div> </div> </div>	
<div> <div> <div></div> <div>Association Class</div> </div> <div> <div></div> <div>_Hexaidabs</div> </div> <div> <div></div> <div>Name</div> </div> <div> <div></div> <div>Visibility</div> </div> <div> <div></div> <div>Comment</div> </div> <div> <div></div> <div>IsLeaf</div> </div> <div> <div></div> <div>Abstract</div> </div> <div> <div></div> <div>IsActive</div> </div> <div> <div></div> <div>IsAssociationDerived</div> </div> <div> <div></div> <div>IsNavigable</div> </div> <div> <div></div> <div>Class Target Dependency</div> </div> <div> <div></div> <div>Nested Class</div> </div> </div>	

HOPEX Concepts	
	AssociationEnd
	Association
	Association Class
	Class via AssociationEnd
	Attribute
	Operation (UML)
	<b>Attribute</b>
	_Hexaidabs
	Name
	Visibility
	Comment
	IsOrdred
	Uniqueness
	ReadOnly
	IsDerived
	InitialValue
	Multiplicity : UpperValue, LowerValue
	Dependency (Target Attribute)
	AttributType
	OverloadedAttribute
	<b>Operation (UML)</b>
	_Hexaidabs

HOPEX Concepts	
	Name
	Visibility
	Comment
	Abstract
	IsQuery
	Precondition
	Postcondition
	Method
	Parameter
	ReturnType
	Target Dependency
	<b>Dependency</b>
	_Hexaidabs
	Name
	Visibility
	Comment
	Class Source
	Class Target
	<b>Generalization</b>
	_Hexaidabs
	Name
	Comment























HOPEX Concepts	
	Super Class
	UML constraint
	<b>GeneralizationSet</b>
	_Hexaidabs
	Name
	Comment
	IsComplete
	IsDisjoint
	Target Dependency
	Generalization
	<b>Constraint</b>
	_Hexaidabs
	Name
	Comment
	MaxInt
	MinInt
	ConstrainedClass
	ConstrainedGeneralization
	ConstrainedElement
	Actor (UML)
	Package
	UseCase

















HOPEX Concepts	
	UseCaseParticipation
	<b>Parameter</b>
	_Hexaidabs
	Name
	Comment
	Parameter Type
	<b>Behavior (UML)</b>
	_Hexaidabs
	Name
	Specification
	<b>UseCase</b>
	_Hexaidabs
	Name
	Visibility
	Comment
	UsesUseCase
	OwnedExtension
	ExtensionPoint
	Behavior: State Machine
	Behavior: Protocol State Machine
	Behavior: Interaction UML
	Behavior: Activity UML

HOPEX Concepts	
 Constraint   Generalization	
 <b>Actor (UML)</b>	
 _Hexaidabs   Name   Visibility   Comment	
 Participation   OwnedExtension   Constraint   Generalization	
 <b>Participation</b>	
 _Hexaidabs   Name   Comment   Multiplicity	
 UseCase   Actor (UML)   Constraint	
 <b>Extension</b>	
 _Hexaidabs   Name	












HOPEX Concepts
<ul style="list-style-type: none"> <li>Comment</li> <li>Extended Use Case</li> <li>Extension Location</li> </ul>
<b>State Machine</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Reentrant</li> <li>DetailedState</li> <li>Region</li> </ul>
<b>Region</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>State</li> <li>PseudoState</li> <li>Transition</li> </ul>
<b>State (UML)</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Detailing Behavior</li> </ul>















HOPEX Concepts	
	Outgoing
	Incoming
	OwnedRegion
	OwnedRegion
	DoActivity
	ExitActivity
	EntryActivity
	<b>Pseudo State</b>
	_Hexaidabs
	Name
	Comment
	PseudoStateKind
	Outgoing Transition
	Incoming Transition
	<b>Transition (UML)</b>
	_Hexaidabs
	Name
	Comment
	Source
	Target
	Source Pseudo State
	Target Pseudo State























HOPEX Concepts	
	Trigger
	Effect (Behavior)
	Constraint
<b>Event (UML)</b>	
	_Hexaidabs
	Name
	Comment
<b>Protocol State Machine</b>	
	_Hexaidabs
	Name
	Reentrant
	Comment
	Region
<b>Action</b>	
	_Hexaidabs
	Name
	IsLeaf
	ActionKind
	Comment
	OutgoingControlFlow
	OutgoingObjectFlow























HOPEX Concepts	
	IncomingControlFlow
	IncomingObjectFlow
	OwnerGroup
	RequestOperation
	CalledBehavior
	StructuralFeatureElementManagedByAnAction
	RequestSignal
	Association
	ClassManagedByAnAction
	Variable
	InputPin
	OutputPin
	ProtectingExceptionHandler
	Trigger
	LocalPostCondition
	LocalPreCondition
	<b>Control Node</b>
	_Hexaidabs
	Name
	IsLeaf
	Comment
	OutgoingControlFlow

HOPEX Concepts
<ul style="list-style-type: none"> <li> OutgoingObjectFlow</li> <li> IncomingControlFlow</li> <li> IncomingObjectFlow</li> <li> OwnerGroup</li> </ul>
<p> <b>Input Pin</b></p> <ul style="list-style-type: none"> <li> _Hexaidabs</li> <li> Name</li> <li> IsLeaf</li> <li> ControlType</li> <li> OrderingKind</li> <li> Comment</li> <li> OutgoingControlFlow</li> <li> OutgoingObjectFlow</li> <li> IncomingControlFlow</li> <li> IncomingObjectFlow</li> </ul>
<p> <b>Output Pin</b></p> <ul style="list-style-type: none"> <li> _Hexaidabs</li> <li> Name</li> <li> IsLeaf</li> <li> ControlType</li> <li> OrderingKind</li> <li> Comment</li> </ul>























HOPEX Concepts
<ul style="list-style-type: none"> <li>✎ OutgoingControlFlow</li> <li>✎ OutgoingObjectFlow</li> <li>✎ IncomingControlFlow</li> <li>✎ IncomingObjectFlow</li> </ul>
<div data-bbox="239 683 446 716">  <b>Exchange Pin</b> </div> <ul style="list-style-type: none"> <li> _Hexaidabs</li> <li> Name</li> <li> IsLeaf</li> <li> ControlType</li> <li> OrderingKind</li> <li> Comment</li> <li>✎ OutgoingControlFlow</li> <li>✎ OutgoingObjectFlow</li> <li>✎ IncomingControlFlow</li> <li>✎ IncomingObjectFlow</li> <li>✎ OwnerGroup</li> </ul>
<div data-bbox="239 1527 526 1561">  <b>Central Buffer Node</b> </div> <ul style="list-style-type: none"> <li> _Hexaidabs</li> <li> Name</li> <li> IsLeaf</li> <li> ControlType</li> <li> OrderingKind</li> </ul>

HOPEX Concepts
<ul style="list-style-type: none"> <li>Comment</li> <li>OutgoingObjectFlow</li> <li>IncomingObjectFlow</li> <li>OwnerGroup</li> </ul>
<b>Data Store Node</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>IsLeaf</li> <li>ControlType</li> <li>OrderingKind</li> <li>Comment</li> <li>OutgoingObjectFlow</li> <li>IncomingObjectFlow</li> <li>OwnerGroup</li> </ul>
<b>Activity Partition</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>IsDimension</li> <li>IsExternal</li> <li>Comment</li> <li>ContainedElement</li> </ul>
<b>Object Flow</b>

HOPEX Concepts	
	_Hexaidabs
	Name
	IsLeaf
	Comment
	Guard
	Weight
	SourceElement
	TargetElement
 <b>Control Flow</b>	
	_Hexaidabs
	Name
	IsLeaf
	Comment
	Guard
	Weight
	SourceElement
	TargetElement
 <b>Exception Handler</b>	
	_Hexaidabs
	Name
	Comment
	ProtectedNode







HOPEX Concepts	
	ExceptionInput
	<b>Activity UML</b>
	_Hexaidabs
	Name
	Reentrant
	SingleExecution
	IsLeaf
	Comment
	ElementOwnedByAnActivityUML
	<b>Activity Parameter Node</b>
	_Hexaidabs
	Name
	IsLeaf
	ControlType
	OrderingKind
	Comment
	OutgoingObjectFlow
	IncomingObjectFlow
	<b>Structured Activity Node</b>
	_Hexaidabs
	Name
	IsLeaf















































HOPEX Concepts	
	MustIsolate
	Comment
	OutgoingControlFlow
	OutgoingObjectFlow
	IncomingControlFlow
	IncomingObjectFlow
	OwnerGroup
	ContainedElement
 <b>Expansion Node</b>	
	_Hexaidabs
	Name
	IsLeaf
	ControlType
	OrderingKind
	Comment
	OutgoingControlFlow
	OutgoingObjectFlow
	IncomingControlFlow
	IncomingObjectFlow
	Region
 <b>Expansion Region</b>	
	_Hexaidabs

HOPEX Concepts	
<ul style="list-style-type: none"> <li>Name</li> <li>IsLeaf</li> <li>MustIsolate</li> <li>ExpansionKind</li> <li>Comment</li> <li>OutgoingControlFlow</li> <li>OutgoingObjectFlow</li> <li>IncomingControlFlow</li> <li>IncomingObjectFlow</li> <li>OwnerGroup</li> <li>ContainedElement</li> <li>ExpansionNode</li> </ul>	
<b>Interruptible Activity Region</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>ContainedElement</li> </ul>	
<b>Collaboration uml</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>IsAbstract</li> </ul>	

HOPEX Concepts
<ul style="list-style-type: none"> <li>IsLeaf</li> <li>CollaborationRole</li> <li>OwnedConnector</li> <li>OwnedCollaborationUse</li> </ul>
<b>Collaboration use</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Type</li> <li>Dependency</li> </ul>
<b>Part</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Visibility</li> <li>Client Dependency</li> <li>IsUnique</li> <li>IsOrdered</li> <li>Multiplicity</li> <li>Comment</li> <li>ConnectorEnd (of the LifeLine who represents the part)</li> <li>Dependency</li> </ul>
<b>Connector</b>

HOPEX Concepts	
	_Hexaidabs
	Name
	Connector Kind
	IsLeaf
	Comment
	 OwnedConnectorEnd
 <b>LifeLine</b>	
	_Hexaidabs
	Name
	Comment
	 ElementRepresentedByALifeline
	 ElementCoveringLifeline
	 OwnedSelector
 <b>Combined Fragment</b>	
	_Hexaidabs
	Name
	InteractionOperatorKind
	Comment
	 InteractionOperand
	 CoveredLifeLine
 <b>Interaction Use</b>	
	_Hexaidabs

HOPEX Concepts	
	Name
	InteractionOperatorKind
	Comment
	RefersTo
	CoveredLifeLine
	<b>State Invariant</b>
	_Hexaidabs
	Name
	Comment
	InvariantConstraint
	CoveredLifeLine
	<b>Gate</b>
	_Hexaidabs
	Name
	Comment
	<b>Message UML</b>
	_Hexaidabs
	Name
	MessageKind
	Comment
	Receiver
	Sender

HOPEX Concepts	
	Connector
	<b>Artifact UML</b>
	_Hexaidabs
	Name
	Comment
	Client Dependency
	Target Dependency
	<b>Node UML</b>
	_Hexaidabs
	Name
	Comment
	Client Dependency
	Deployment
	<b>Device</b>
	_Hexaidabs
	Name
	Comment
	Client Dependency
	Deployment
	<b>Execution Environment</b>
	_Hexaidabs
	Name

HOPEX Concepts
<ul style="list-style-type: none"> <li>Comment</li> <li>Client Dependency</li> <li>Deployment</li> </ul>
<b>Deployment Specification</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Client Dependency</li> <li>Target Dependency</li> </ul>
<b>Instance ( Node UML/Device/Execution Specification/Component)</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Client Dependency</li> <li>Instantiated Element</li> <li>Target Dependency</li> </ul>
<b>Communication Path</b> <ul style="list-style-type: none"> <li>_Hexaidabs</li> <li>Name</li> <li>Comment</li> <li>Communication Path End</li> <li>Target Dependency</li> </ul>

HOPEX Concepts	
<div> <div></div> <b>Communication Path</b> </div>	<div> <div></div>           _Hexaidabs         </div>
	<div> <div></div>           Name         </div>
	<div> <div></div>           Comment         </div>
	<div> <div></div>           Multiplicity         </div>
	<div> <div></div>           Deployment Target         </div>
<div> <div></div> <b>Deployment</b> </div>	<div> <div></div>           _Hexaidabs         </div>
	<div> <div></div>           Name         </div>
	<div> <div></div>           Comment         </div>
	<div> <div></div>           Multiplicity         </div>
	<div> <div></div>           Deployed Element         </div>
	<div> <div></div>           Deployment Configuration         </div>