

HOPEX IT Architecture

Guide d'utilisation



HOPEX V2

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière constituer un engagement de 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 - 2016

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.

INTRODUCTION À HOPEX IT ARCHITECTURE



HOPEX IT Architecture est un logiciel édité par **MEGA International** qui permet aux entreprises et autres organismes de représenter et de documenter leurs systèmes d'information.

HOPEX IT Architecture permet de décrire et d'analyser l'architecture des systèmes d'information :

- ✓ **La cartographie applicative** : description de l'architecture applicative.
La description de l'architecture applicative offre une vue détaillée des échanges d'information entre les applications, les services, les bases de données et les unités organisationnelles. **HOPEX IT Architecture** permet également de décrire des systèmes complexes, qui font intervenir des ressources autres que logicielles.
- ✓ **Le déploiement des applications** : description de l'infrastructure technique.
La description de l'infrastructure technique du système d'information permet 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 gestion et l'évolution du système d'information** : description de plans d'urbanisme et d'architectures de services.
Les 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 et l'analyse de scénarios d'évolution (avec **HOPEX Planning**).
- ✓ **La représentation des services rendus par l'architecture IT** : description des structures internes des applications et des services applicatifs.
HOPEX System Oriented IT Architecture propose les fonctionnalités dédiées à la représentation et à l'analyse d'architectures IT selon une approche orientée service.

HOPEX IT Architecture propose également une option 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 l'article technique "CMDB Import".

PRÉSENTATION DE CE GUIDE

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

☛ Les différences entre **HOPEX Web Front-End** et **HOPEX Windows Front-End** sont précisées le cas échéant pour chaque fonctionnalité.

Structure du guide

Le guide **HOPEX IT Architecture** contient les chapitres suivants :

- "Premiers pas avec HOPEX IT Architecture", page 9, décrit comment se connecter et créer les objets principaux d'un diagramme.
- "Modéliser l'architecture de l'entreprise", page 21, est une présentation d'utilisations plus spécifiques de **HOPEX IT Architecture**.
- "Modéliser des infrastructures complexes", page 59, présente comment modéliser les composants et les moyens de communication des infrastructures complexes.
- "Le déploiement d'applications", page 89, décrit comment déployer une application conformément à une infrastructure requise.
- "Urbanisation du système d'information", page 99, explique comment décrire l'urbanisme des systèmes d'information.
- "Architecture orientée services (SOA)", page 121, décrit comment organiser l'architecture orientée services de votre système d'information.
- "Détail des concepts", page 131, présente les diagrammes proposés et les concepts manipulés dans **HOPEX IT Architecture**.

Ressources complémentaires

Ce guide est complété par :

- le guide **HOPEX Common Features**, qui décrit les fonctionnalités de base communes aux produits et solutions **HOPEX**.
☛ Il peut être utile de consulter ce guide pour une présentation générale de l'interface.
- le guide d'administration **HOPEX Power Supervisor**.
- des fonctions techniques plus avancées sont décrites dans le guide **HOPEX Power Studio**.

Conventions utilisées dans le guide

Styles et mises en forme

👉 *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.**

PRÉSENTATION GÉNÉRALE DE HOPEX IT ARCHITECTURE

Pourquoi modéliser l'architecture d'un système d'information ?

Face aux exigences de compétitivité, de réactivité et d'efficacité, les entreprises ont besoin de systèmes d'information toujours plus performants.

La technologie offre aujourd'hui une grande richesse de possibilités qui ouvre la voie à la construction des architectures de systèmes d'information les plus complexes et diversifiés. Les besoins toujours plus importants de communication entre applications, notamment avec Internet, requièrent une réflexion plus globale et plus approfondie de l'organisation des composants logiciels.

Par une représentation schématique et structurée, détaillée ou synthétique, la modélisation aide à concevoir et décrire l'architecture des systèmes d'information, et facilite ainsi la maîtrise des évolutions ultérieures.

Il existe de multiples raisons de modéliser l'architecture d'un système d'information : l'architecte peut s'intéresser à la description de l'existant, en focalisant son analyse sur l'architecture logicielle et son intégration dans l'organisation, ou sur l'infrastructure technique et matérielle du système d'information ; il peut également étudier les évolutions nécessaires d'un système d'information, qu'il s'agisse d'intégrer une nouvelle application, un nouveau site Internet, ou un progiciel tel que SAP, ou d'optimiser une architecture client/serveur, en recherchant la meilleure répartition des données et des traitements.

L'architecte peut également être amené à représenter de manière précise les interactions entre les applications de l'entreprise et celles de ses partenaires, ou entre des entités autonomes à l'intérieur d'une même entreprise comme lors de la mise en place d'un moteur de workflow.

Par rapport à de simples représentations graphiques ou documents de présentation, la modélisation permet la constitution progressive d'une base d'information complète et parfaitement cohérente, facile à maintenir.

Présenter un système d'information complexe et étendu

Pour décrire l'architecture d'un système d'information complexe, la première étape consiste souvent à établir des représentations géographiques synthétiques et explicites qui mettent en évidence les principaux sites d'implantation de l'entreprise, ainsi que l'organisation des communications entre ceux-ci.

Selon le niveau de détail de ces cartes, la modélisation présentera le système d'information dans son ensemble, ou des sous-ensembles particuliers plus détaillés.

Par exemple, pour un système d'information réparti sur plusieurs continents, vous pouvez dresser une planisphère présentant les sites régionaux (ou par pays) qui composent le système, puis des cartes régionales spécifiques décrivant les implantations locales (siège national, agences départementales, etc.). Pour chaque carte, vous pouvez représenter les moyens de transmission entre les

différents sites ainsi que leur équipement applicatif et matériel respectifs.

Réaliser une cartographie des applications et des bases de données

Un système d'information se construit par étapes. Généralement, les entreprises ont développé ou acquis leurs applications au fur et à mesure des besoins, des progrès des technologies de l'information, des ressources financières disponibles, etc.

Après l'informatisation des fonctions comptables, les entreprises ont mis en place des outils informatiques spécialisés centrés sur la gestion de leur(s) métier(s). Ultérieurement sont arrivés les outils de management, et on constate actuellement la progression fulgurante des applications tirant profit d'Internet, couplant des technologies de consultation et de mise à jour de bases de données géographiquement éloignées.

Ces multiples applications informatiques, développées par étapes et parfois indépendamment les unes des autres, sont aujourd'hui appelées à communiquer entre elles et avec l'extérieur, à partager des données, à synchroniser leurs traitements et leur ergonomie.

Face à ces défis, l'entreprise qui souhaite maîtriser son système d'information et le faire évoluer doit être capable de dresser un inventaire de ses applications, des sites Internet/Extranet et des bases de données associées, ainsi que de les situer. Elle doit également cerner le périmètre fonctionnel et organisationnel des applications (fonctions et utilisateurs concernés, informations échangées), ainsi que les interdépendances entre elles.

Dans ce contexte, la réalisation d'une cartographie des applications, des sites Internet et des bases de données constitue une démarche extrêmement bénéfique pour l'entreprise.

Tout d'abord, l'analyse de l'existant informatique qu'elle inclut révélera les points faibles du système d'information : lacunes, applications concurrentes, saisies multiples des mêmes données, hétérogénéité. L'entreprise en déduira les actions d'amélioration de l'efficacité de l'organisation et de la productivité à mener.

Ensuite, la cartographie obtenue se révélera un outil de travail puissant dans le cadre de projets d'évolution du système d'information : études d'impact d'une nouvelle application ou d'un progiciel dans son futur environnement, formalisation des flux d'information dans un contexte d'échange de données informatisé (EDI), optimisation d'une architecture Client/Serveur.

Urbaniser un système d'information

Les systèmes d'information des entreprises se sont développés dans une grande disparité au fil du temps et au gré des projets. Dans ces conditions, il devient de plus en plus difficile d'intégrer de nouvelles applications dans les systèmes d'information existants ou de faire face à des changements technologiques. L'urbanisme informatique consiste à définir les différents composants d'un système d'information et leurs modalités d'assemblage de façon à faciliter l'intégration de ces nouvelles applications.

De la même façon que pour une ville, vous pouvez concevoir le système d'information d'une entreprise comme un plan d'urbanisme et définir un découpage durable du système d'information en quartiers indépendants.

Représenter le découpage fonctionnel d'une application

L'intérêt de fractionner une application en composants se manifeste tout au long du cycle de vie des applications. Cette démarche est essentielle dans la phase de spécification. Les différents besoins à couvrir sont étudiés dans le cadre de sous-projets fonctionnels complémentaires.

La décomposition en modules fonctionnels permet de maximiser les chances de succès de l'intégration d'une nouvelle application dans l'entreprise. Lors de cette phase cruciale de la vie d'une application, il est intéressant d'attribuer les fonctionnalités de l'application aux différents acteurs de l'entreprise pour étudier finement l'impact sur l'organisation et organiser les formations nécessaires.

Enfin, la représentation du découpage fonctionnel d'une application facilite la gestion de la réutilisation des composants logiciels dans le système d'information de l'entreprise.

Décrire les interactions entre applications

Les interactions définies au niveau international par des organismes tels que l'OAG (Open Applications Group) ou IFX (Interactive Financial Exchange), ainsi que la description des schémas XML associés (disponibles avec le produit **HOPEX System Blueprint**) permettent d'automatiser les échanges entre des applications appartenant à des entreprises différentes. L'utilisation d'interactions et de messages standard permet à ces entreprises de communiquer sans forcément se connaître à l'avance. C'est ce qui permet le fonctionnement des places de marché qui connaissent une expansion grandissante. Les mêmes mécanismes peuvent être utilisés à l'intérieur d'une entreprise comme lors de la mise en place de certains moteurs de workflow, ou pour automatiser les échanges entre une entreprise et ses filiales.

Pour plus de détails, voir ["Décrire les interactions"](#), page 51.

Optimiser une architecture client/serveur

Les architectures client/serveur se généralisent. Les possibilités offertes par ces approches en terme de réutilisabilité de composants logiciels élémentaires, de portabilité et d'évolutivité des applications permettent d'envisager des systèmes d'information efficaces et souples malgré la complexité des besoins à couvrir.

Cependant, toutes les architectures applicatives réalisables ne sont pas équivalentes sur le plan des flux d'information transportés, sur les délais, sur les risques, sur les coûts.

Face aux questions soulevées par ces approches, la modélisation détaillée des architectures client/serveur, incluant la décomposition des applications en modules client ou serveur et une réflexion sur l'implantation des données utilisées, permet d'optimiser la répartition des données et des traitements, à partir d'une appréciation qualitative du volume des flux d'information générés par les applications.

La modélisation introduit la possibilité d'étudier plusieurs scénarios d'organisation, dès la phase de conception, en offrant au concepteur les moyens de réaliser des simulations successives centrées sur des solutions alternatives d'implantation et de réplication des bases de données, ou de partage des ressources applicatives.

Décrire une infrastructure technique

Sur le plan technique, les systèmes d'information des entreprises héritent de l'hétérogénéité que l'on observe au niveau logiciel. Souvent, la coexistence de plusieurs générations d'équipements et de matériels informatiques vient encore en amplifier la complexité.

La modélisation de l'architecture du système d'information sous l'angle technique apportera donc souvent beaucoup à la connaissance et à la maîtrise qu'en ont les architectes et les responsables de l'entreprise.

Les diagrammes réalisés permettent d'identifier et de représenter les matériels existants (serveurs, postes de travail, etc.), les réseaux qui relient ces machines, et de décrire leurs caractéristiques. Ces diagrammes indiquent sur quelles machines sont installées les bases de données et les applications.

PREMIERS PAS AVEC HOPEX IT ARCHITECTURE



Ce chapitre a pour but de vous familiariser avec l'utilisation de **HOPEX IT Architecture** : il présente la connexion au logiciel ainsi que des fonctions de base liées à l'utilisation des diagrammes.

☛ *Pour plus de détails sur la gestion de votre bureau, des diagrammes et des objets voir le guide **HOPEX Common Features**.*

Les points suivants sont abordés ici :

- ✓ ["Conditions préalables à l'utilisation de HOPEX IT Architecture", page 10](#)
- ✓ ["Utiliser les diagrammes dans HOPEX IT Architecture", page 12](#)

CONDITIONS PRÉALABLES À L'UTILISATION DE HOPEX IT ARCHITECTURE

Certaines options permettent une meilleur utilisation de **HOPEX IT Architecture**.

Se connecter à HOPEX IT Architecture

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

Gérer les options relatives aux messages

Certaines fonctionnalités décrites ici s'appuient sur des fonctionnalités de gestion des flux entre les objets d'une architecture applicative basées sur la norme BPMN.

Pour pouvoir accéder aux fonctionnalités de gestion des flux au format BPMN :

1. Accédez à fenêtre des options :
 - (**HOPEX Windows Front-End**) depuis la barre de menus de votre bureau sélectionnez **Outils > Options**.
 - (**HOPEX Web Front-End**) depuis le groupe d'outils **Divers** de votre bureau **HOPEX IT Architecture** sélectionnez **Mon compte > Options**.
2. Dans l'arbre des options sélectionnez **Modélisation des processus et de l'architecture**.
3. Cochez l'option **Diagrammes d'architecture : Utilisation des flux de données BPMN**.

Cette option, cochée par défaut, vous permet d'utiliser le formalisme BPMN dans vos diagrammes.

Toutefois, si vous souhaitez continuer à utiliser des messages dans vos diagrammes, une option de comptabilité est disponible.

Pour activer l'option de compatibilité :

1. Accédez à fenêtre des options :
 - (**HOPEX Windows Front-End**) depuis la barre de menus de votre bureau sélectionnez **Outils > Options**.
 - (**HOPEX Web Front-End**) depuis le groupe d'outils **Divers** de votre bureau **HOPEX IT Architecture** sélectionnez **Mon compte > Options**.
2. Dans l'arbre des options sélectionnez **Modélisation des processus et de l'architecture**.
3. Cochez l'option **Diagrammes d'architecture : Utilisation des Messages**.

Si les deux options décrites ici sont cochées, vous pouvez facilement remplacer vos messages par des flux de format BPMN.

UTILISER LES DIAGRAMMES DANS HOPEX IT

ARCHITECTURE

Un projet de description de l'architecture applicative d'un système d'information consiste à recenser les applications existantes et leurs échanges. Les points suivants donnent quelques bases de la modélisation de l'architecture applicative.

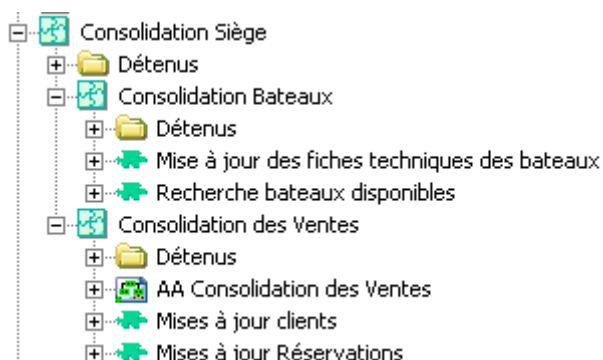
Créer une application

HOPEX IT Architecture recense les applications dans un arbre de navigation.

Pour créer une application (hors d'un diagramme) :

1. Accédez à l'arbre de navigation des Applications :
 - (**HOPEX Windows Front-End**), dans la barre de menus **HOPEX**, cliquez sur **Affichage > Fenêtres de navigation > Objets principaux**.
 - (**HOPEX Web Front-End**), dans les volets de navigation, sélectionnez **Objets principaux**.
2. Faites un clic droit sur le dossier **Applications** et sélectionnez **Nouveau > Application**.

A partir de l'application créée, vous pouvez notamment créer ses sous-applications et services ainsi que les diagrammes qui vont la décrire.



Pour plus de détails sur les applications, voir "[Décomposition du système d'information](#)", page 22.

Créer un diagramme décrivant une application

HOPEX IT Architecture offre différents diagrammes pour modéliser l'architecture applicative d'un système d'information. La liste des diagrammes disponibles est

rappelée dans le chapitre "Détail des concepts", page 131.

La procédure de création d'un diagramme varie légèrement suivant que vous êtes en Windows Front-End ou Web Front-End.

HOPEX Windows Front-End

Pour créer le diagramme d'une application dans **HOPEX Windows Front-End** :

1. Accédez à l'application.

Par exemple à partir de l'arbre des applications ou à partir d'un diagramme qui décrit l'application.

2. Faites un clic droit sur l'application et sélectionnez **Nouveau > Diagramme**.
3. Sélectionnez le type de diagramme.

Par exemple «Diagramme d'architecture applicative».

4. Conservez l'option **Initialiser le diagramme** sélectionnée (par défaut) pour initialiser le diagramme. Cette option permet de disposer d'objets posés automatiquement dans le diagramme qui est créé.
5. Cliquez sur **Créer**.
Le diagramme créé est nommé automatiquement par **HOPEX IT Architecture**. Il décrit l'application qui constitue l'objet décrit. Cette application est automatiquement placée au centre du diagramme (seulement si vous avez conservé l'option **Initialiser le diagramme**).

HOPEX Web Front-End

Pour créer le diagramme d'une application dans **HOPEX Web Front-End** :

1. Accédez à l'application.
2. Faites un clic droit sur l'application et sélectionnez **Nouveau** puis le type de diagramme voulu.

Par exemple «Diagramme d'architecture applicative».

Le diagramme créé est nommé automatiquement par **HOPEX IT Architecture**. Il décrit l'application qui constitue l'objet décrit. Cette application est automatiquement placée au centre du diagramme.

Créer un diagramme de vue d'ensemble des applications

Le diagramme de vue d'ensemble des applications présente les principales applications de l'entreprise et leurs interactions.

De la même façon, vous pouvez créer des vues d'ensemble pour les sites et l'infrastructure technique.

HOPEX Windows Front-End

Pour créer un diagramme de vue d'ensemble des applications dans **HOPEX Windows Front-End** :

1. Dans la barre de menus, cliquez sur **Affichage > Fenêtres de navigation > Objets principaux**.

2. Faites un clic droit sur le dossier **Vues d'ensemble** et sélectionnez **Nouveau > Diagramme**.
3. Dans la fenêtre qui apparaît, sélectionnez «Vue d'ensemble des applications».
4. Cliquez sur **Créer**.
Le diagramme apparaît.

HOPEX Web Front-End

Pour créer un diagramme de vue d'ensemble des applications dans **HOPEX Web Front-End** :

1. Affichez les Fenêtres de navigation.
2. Dans la fenêtre de navigation **Objets principaux**, faites un clic droit sur le dossier **Vues d'ensemble** et sélectionnez **Nouveau > Vue d'ensemble des applications**.
Le diagramme apparaît.

Créer un 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.

Pour créer un acteur dans **HOPEX IT Architecture** :

1. Dans la barre d'objets du diagramme, cliquez sur **Acteur** .
☛ Chaque fois que vous cliquez sur une icône dans la barre d'objets du diagramme (sauf l'icône de sélection) sa définition apparaît dans la fenêtre d'aide qui se trouve par défaut en bas du diagramme.
2. Cliquez dans le diagramme, à l'emplacement où vous désirez créer l'acteur.
La fenêtre de création d'un acteur apparaît.
3. Dans la fenêtre **Ajout d'un(e) Acteur**, saisissez le nom de l'acteur que vous souhaitez créer.
4. Cliquez sur **Créer**.
L'acteur apparaît dans votre diagramme.
5. Si nécessaire, retaillez la forme pour l'ajuster au texte.

Un acteur peut également être placé dans un couloir afin d'améliorer la présentation graphique du diagramme.

Pour plus de détails sur les couloirs, reportez-vous à la section "Utiliser les couloirs" du guide **HOPEX Common Features**.

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. Dans l'onglet **Caractéristiques**, cliquez dans le champ **Interne/Externe** et sélectionnez la valeur "Acteur externe".
3. Cliquez sur **OK** pour appliquer la modification et fermer la fenêtre de propriétés.

Cette caractéristique est représentée graphiquement et s'applique automatiquement dans le diagramme. Lorsque vous refermez la fenêtre de propriétés de l'Acteur "Agence de location de bateaux", sa couleur est modifiée.

☛ *HOPEX IT Architecture gère ainsi pour chaque objet des caractéristiques qui varient en fonction de son type. Ces caractéristiques peuvent être modifiées dans l'onglet **Caractéristiques** de la fenêtre de propriétés de chaque objet.*

☛ *Les caractéristiques des différents objets manipulés dans **HOPEX IT Architecture** sont décrites dans "Concepts gérés par HOPEX IT Architecture", page 144.*

Utiliser les flux

Il est possible de préciser le contenu des flux échangés entre les objets d'un diagramme.

📖 *Un flux représente la circulation d'information à l'intérieur de l'entreprise ou entre l'entreprise et son environnement. Un flux peut transporter un contenu.*

☛ *Nous recommandons d'utiliser les flux BPMN. Pour plus de détails sur les options à activer, voir : "Gérer les options relatives aux messages", page 10.*

Créer un flux avec un contenu

Vous pouvez préciser le contenu des **flux** échangés entre objets d'un diagramme directement lors de sa création.

📖 *Un flux représente la circulation d'information à l'intérieur de l'entreprise ou entre l'entreprise et son environnement. Un flux peut transporter un contenu.*

Pour créer le flux et son contenu :

1. Cliquez sur le bouton **Flux** et sélectionnez **Flux avec contenu**.



2. Cliquez sur le premier objet qui représente l'étape de départ et tirez un trait jusqu'à l'objet représentant l'étape suivante en maintenant le bouton de la souris enfoncé.
La fenêtre de **Création de Flux avec contenu** 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.

 Vous pouvez associer plusieurs contenus au flux. Pour plus de détails, voir "[Définir le contenu d'un flux](#)", page 16.

Définir le contenu d'un flux

Pour définir le contenu d'un flux :

1. Cliquez avec le bouton droit sur le flux et sélectionnez **Propriétés**.
La fenêtre de propriétés s'ouvre.
2. Cliquez sur **Caractéristiques**.
3. Cliquez sur la flèche à droite du champ **Contenu** et sélectionnez la commande **Relier Contenu**.
La fenêtre de recherche apparaît, avec une liste de contenus proposés pour le flux.
4. Sélectionnez le nom du contenu puis cliquez sur **OK**.

 Un contenu peut être utilisé par plusieurs flux puisqu'il n'est pas associé à un émetteur, ni à un destinataire.

Le nom du contenu apparaît dans le diagramme.

Créer un message

 Nous recommandons d'utiliser les flux BPMN, mais l'utilisation des messages reste disponible avec une option de compatibilité. Pour plus de détails voir : "[Gérer les options relatives aux messages](#)", page 10.

 Un message représente un flux circulant à l'intérieur de l'entreprise ou échangé entre l'entreprise et son environnement. C'est généralement un flux d'information comme une commande ou une facture. Par commodité, un flux financier comme le règlement du client, ou un flux de matière comme la livraison d'un produit est également représenté par un message.

Dans un diagramme, un **message** est décrit par :

- un trajet, c'est à dire l'indication de son émetteur et de son destinataire.
- un contenu, c'est à dire les objets ou les informations qui transitent de l'émetteur vers le destinataire du message. Ce contenu peut être réutilisé dans plusieurs messages.

Vous avez deux possibilités pour créer les messages. Vous pouvez :

- créer les messages en les reliant en même temps à leurs émetteurs et récepteurs.
- créer d'abord les messages et les relier ensuite à leurs émetteurs et récepteurs.

Créer les messages en même temps que les liens

Pour créer les messages en les reliant en même temps à leurs émetteurs et récepteurs :

1. Dans la barre d'objets du diagramme, cliquez sur **Message** .
2. Cliquez sur l'émetteur (acteur externe, application ou base de données), et faites glisser le curseur jusqu'au récepteur avant de relâcher votre pression.
La fenêtre d'ajout d'un **message** apparaît, vous invitant à saisir le nom du contenu utilisé par le message.
3. Saisissez le nom du contenu du message et cliquez sur **OK**.

☛ Par défaut, le message porte le même nom que son contenu.

☛ Lorsque le contenu du message existe déjà dans le référentiel **HOPEX**, **Ce Contenu existe déjà** s'affiche en bas de la fenêtre. Si vous souhaitez utiliser ce contenu, cliquez sur **OK**. Sinon, modifiez le nom du message dans le champ **Nom** et cliquez sur **OK**, un nouveau contenu sera créé pour ce message.

Le message se positionne automatiquement au milieu du trait qui relie l'émetteur au récepteur.

En procédant ainsi, il se peut que vous ayez à déplacer des messages : par exemple, si deux objets échangent plus d'un message, ceux-ci seront superposés et vous devrez déplacer un des messages pour que les deux soient visibles.

Créer les messages puis les liens

Pour créer les **messages**, et les relier ensuite à leurs émetteurs et récepteurs :

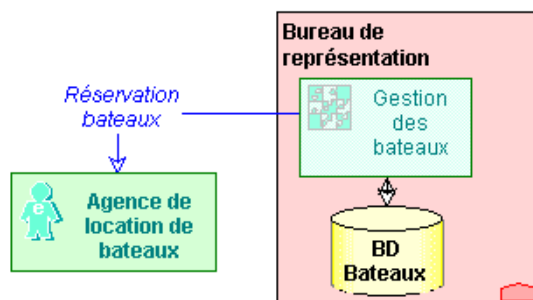
1. Dans la barre d'objets du diagramme, cliquez sur **Message** .
2. Dans le diagramme, cliquez là où vous voulez le positionner.
La fenêtre d'ajout d'un message apparaît, vous invitant à saisir le nom du contenu utilisé par le message.
3. Saisissez le nom du contenu du message, puis cliquez sur **OK**.

Pour créer chaque lien :

1. Dans la barre d'objets du diagramme, cliquez sur **Lien** .

2. Cliquez sur l'objet de départ (émetteur ou flux d'information) et faites glisser le curseur jusqu'à l'objet d'arrivée (flux d'information ou récepteur).

Ci-dessous le message "Réservation bateaux" est émis par le "Bureau de représentation" à destination de l'"Agence de location de bateaux".



Lorsqu'un flux d'information a plusieurs émetteurs ou plusieurs récepteurs, il peut être créé avec la première technique entre un émetteur et un récepteur, puis relié aux autres avec la seconde.

Un flux d'information peut être assorti de caractéristiques particulières relatives à la fréquence ou aux conditions d'émission. Dans ce cas, un **Temporisateur**  que vous liez au flux d'information correspondant vous permet d'indiquer cette caractéristique.

Gestion des vues spécialisées

HOPEX IT Architecture vous permet de restreindre ou d'étendre le champ des concepts que vous voulez voir figurer sur un diagramme.

Vous pouvez ainsi construire des diagrammes très complets, en superposant des niveaux spécialisés, comme vous le feriez avec des feuilles de calques :

- des calques qui concernent les aspects logiciels (sites, applications, bases de données, etc.)
- des calques qui présentent les acteurs qui utilisent les applications
- des calques qui complètent la représentation avec les équipements informatiques utilisés.

Pour choisir le mode de visualisation approprié:

- ☐ Dans la barre d'outils du diagramme, cliquez sur **Vues et détails**  . Par défaut, l'affichage est limité aux éléments graphiques (fond d'écran en arrière-plan) et à la vue standard, c'est-à-dire les concepts de base du diagramme et les liens entre eux.

Pour étendre le champ des concepts utilisables dans un diagramme :

- ☐ Dans la fenêtre **Vue et détails**, sélectionnez, les vues qui correspondent aux extensions souhaitées.

Pour restreindre la visualisation :

- Dans la fenêtre **Vue et détails**, désélectionnez la vue qui correspond à la vue spécialisée.
Cette action fait disparaître l’affichage des objets gérés dans le cadre de la vue spécialisée. Les objets disparus et leurs liens avec d’autres objets restent mémorisés par **HOPEX IT Architecture**. Pour les réafficher ultérieurement, à la même place, resélectionnez la vue correspondante.

Par exemple, pour représenter les acteurs d’une Agence de voyages et les processus de travail, ainsi que le serveur qui supporte les applications et les bases de données de l’agence, sélectionnez les vues **Acteurs**, **Serveurs** et **Postes de travail**.

Les icônes correspondantes apparaissent dans la barre d’objets.

Mettre en forme le diagramme

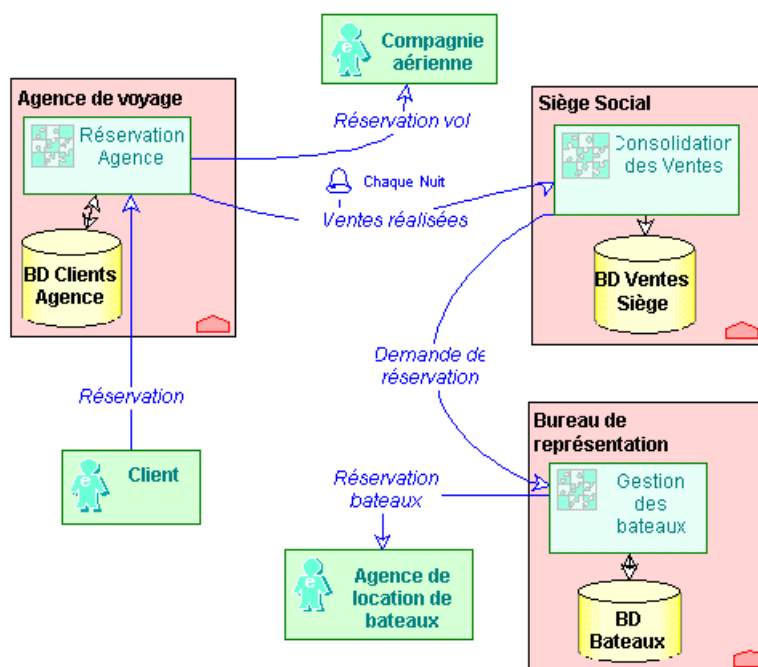
Une fois que vous avez créé tous les objets et liens, vous pouvez améliorer la présentation graphique de votre diagramme. Vous pouvez, par exemple :

- déplacer les objets
- aligner les objets (menu **Dessin > Aligner**)
- modifier la forme des liens
- ajouter des sommets sur les liens à l'aide de la touche <Ctrl> pour leur faire éviter des obstacles, ou les courber, etc.

☛ Pour plus de détails, voir le guide **HOPEX Common Features**.

☛ Pour plus de détails, voir "[Manipuler les diagrammes](#)", page 67.

Ci-dessous, pour faciliter la lisibilité du diagramme, les sites ont été agrandis pour pouvoir y déposer tous les objets qu'ils «contiennent».



MODÉLISER L'ARCHITECTURE DE L'ENTREPRISE



Ce chapitre décrit les concepts liés à la modélisation d'une architecture applicative et les diagrammes répondant aux divers besoins de l'architecte du système d'information.

Les points abordés dans ce chapitre sont :

- ✓ "Décrire une architecture applicative", page 22
- ✓ "Décrire l'environnement d'une application", page 25
- ✓ "Détailer l'architecture interne d'une application", page 30
- ✓ "Décrire les applications Internet", page 36
- ✓ "Créer des arbres applicatifs", page 38
- ✓ "Décrire l'infrastructure des télécommunications entre sites", page 41
- ✓ "Modéliser les infrastructures techniques", page 44
- ✓ "Rapports sur les architectures applicatives", page 47

DÉCRIRE UNE ARCHITECTURE APPLICATIVE

L'architecture applicative est une organisation logique des applications et de leurs échanges fournissant une représentation synthétique de tout ou partie du système d'information.

Les projets de cartographie applicative sont basés sur le recensement du patrimoine applicatif existant. La modélisation de ce patrimoine au sein d'un référentiel fournit un outil de maîtrise de l'existant, et sert de base à la mise en œuvre de projets d'évolution.

Décomposition du système d'information

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 ["Créer une application", page 12](#).

Système applicatif

Un système applicatif est un assemblage d'applications répondant à un ensemble cohérent de fonctionnalités, implémentées à l'aide de la coopération des applications qui composent ce système.

Les systèmes applicatifs peuvent :

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

Système applicatif et application peuvent faire l'objet de développements spécifiques (réalisés en interne ou bien achetés / sous-traités), ou être des produits du marché (progiciels).

C'est l'organisation logique et la structure des systèmes applicatifs et des applications, ainsi que la description de leurs échanges qui constituent les fondations de l'architecture applicative.

Dans **HOPEX IT Architecture**, les termes application ou système applicatif sont utilisés indistinctement pour nommer les composants d'une architecture applicative.

Nomenclature d'applications

La décomposition d'un système applicatif en sous-applications (ou sous-systèmes) s'envisage selon deux axes :

- d'une part, les éléments logiciels fournis par le système applicatif (le déploiement du système entraîne celui de ses composants, le développement du système est dépendant de celui de ses composants, etc.).
- d'autre part, l'application sous l'angle de son fonctionnement.
Dans ce cas il faut également inclure les sous-applications utilisées lors de l'exécution de l'application, mais non fournies.
Nous parlons ici d'applications utilisées de façon explicite, c'est-à-dire que l'application utilisatrice connaît l'application utilisée et fait directement appel à ses services.

Echanges entre applications

Voir ["Décrire les interactions"](#), page 51.

Gérer les versions d'applications

Le système d'information d'une entreprise est en perpétuelle évolution ; de nouvelles applications peuvent être intégrées, mais dans la plupart des cas, les modifications du SI consistent à adapter des applications existantes. Il est donc important de pouvoir suivre les évolutions de ces applications et d'être capable d'identifier les différences entre chaque version.

HOPEX IT Architecture permet de gérer les versions des applications et des architectures plus complexes :

- en s'appuyant sur une technique de modélisation : la variation, et
- en verrouillant l'application

Si vous créez une version d'une application, l'application est automatiquement verrouillée.

Pour apporter une modification à l'application, vous devez créer une nouvelle version.

☺ *Pour utiliser les variations (et par conséquent les versions), depuis la fenêtre des options, sélectionnez **Modélisation des processus et des l'architecture** puis sélectionnez l'option **Activation des variations**.*

☛ *Pour plus de détails sur les variations, voir le guide **HOPEX Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".*

DÉCRIRE L'ENVIRONNEMENT D'UNE APPLICATION

HOPEX IT Architecture permet d'affiner la description de l'architecture du système d'information. Dans une démarche de ce type, l'analyse se concentre souvent sur une partie seulement du système d'information.

Vous pouvez ainsi créer un diagramme d'environnement d'une application particulière pour une description des flux d'information échangés, des bases de données utilisées, des acteurs et des processus concernés.

Le diagramme d'environnement permet de placer l'application dans ses contextes d'utilisation. On y définit les échanges constatés ou attendus (contrats de service).

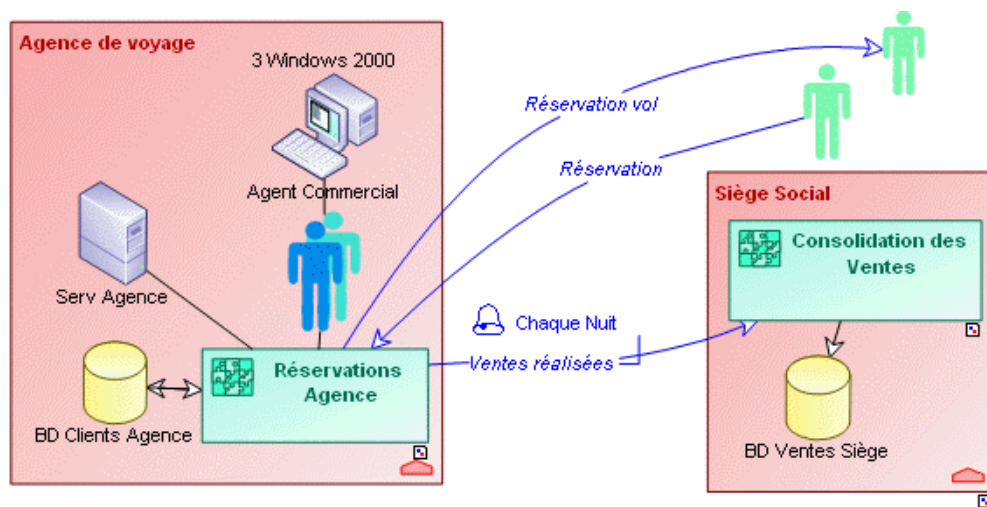
 Plutôt que d'utiliser des diagrammes d'environnement, **MEGA** recommande d'utiliser des diagrammes d'architecture applicative interne qui permettent de référencer les applications. Il faut alors créer des applications parentes de type "système applicatif" (ex : Système Applicatif comptable). Cette méthode permet d'améliorer notamment la description des flux inter-applicatifs, qui seront alors définis au sein d'un système applicatif.

☛ Le diagramme d'environnement ne devrait pas être utilisé pour décrire la décomposition en modules applicatifs ou en service (voir arbre applicatif). Il ne contient pas non plus les applications et service nécessaires à son fonctionnement (voir diagramme d'architecture interne).

☛ Ce cadre d'utilisation précis permet de proposer les comportements adéquats lors de l'introduction d'application ou de service dans le diagramme. Ainsi, toutes les applications ajoutées dans le diagramme sont considérées comme « applications voisines » et par conséquent aucun lien n'est ajouté automatiquement (lien de décomposition ou de participation à l'architecture interne).

Il est possible de compléter le diagramme d'environnement en précisant quelques éléments de l'infrastructure technique, tels que les serveurs sur lesquels

fonctionnent les applications, les bases de données et le réseau qui relie les serveurs et les postes de travail de l'agence.



☞ L'icône  qui apparaît sous certains objets indique qu'ils sont décrits par un diagramme.

Pour plus de détails sur les diagrammes et les objets décrits, voir aussi :

- ✓ "Détail des concepts", page 131

Pré requis à l'utilisation des diagrammes d'environnement d'une application

Pour accéder aux diagrammes d'environnement d'une application :

- Accédez à fenêtre des options :
 - (**HOPEX Windows Front-End**) depuis la barre de menus de votre bureau sélectionnez **Outils > Options**.
 - (**HOPEX Web Front-End**) depuis le groupe d'outils **Divers** de votre bureau **HOPEX IT Architecture** sélectionnez **Mon compte > Options**.
- Dans l'arbre des options sélectionnez **Compatibilité > Diagrammes**.
- Cochez l'option **Application Création d'un diagramme d'environnement d'application**.

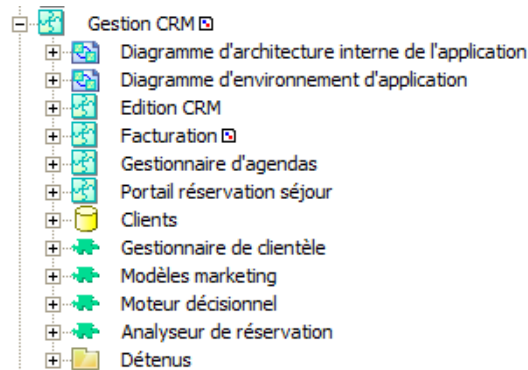
Ouvrir un diagramme d'environnement d'une application

Pour ouvrir le diagramme d'environnement d'une application :

- Dans la fenêtre de navigation **Objets principaux**, déployez le dossier **Applications**.

2. Dépliez le dossier de l'application voulue.

Ex. : l'application "Gestion CRM".



3. Double-cliquez sur **Diagramme d'environnement d'application**.

Créer un diagramme d'environnement d'une application

La procédure varie légèrement suivant que vous êtes en Windows Front-End ou Web Front-End.

HOPEX Windows Front-End

Pour créer un diagramme d'environnement d'une application depuis **HOPEX Windows Front-End** :

1. Accédez à l'application.
Par exemple à partir de l'arbre des applications ou à partir d'un diagramme qui décrit l'application.
2. Faites un clic droit sur l'application et sélectionnez **Nouveau > Diagramme**.
3. Sélectionnez le type de diagramme "Diagramme d'environnement d'application".
4. Conservez l'option **Initialiser le diagramme** sélectionnée (par défaut) pour initialiser le diagramme. Cette option permet de disposer d'objets posés automatiquement dans le diagramme qui est créé.
5. Cliquez sur **Créer**.
Le diagramme créé est nommé automatiquement par **HOPEX IT Architecture**. Il décrit l'application qui constitue l'objet décrit. Cette application est automatiquement placée au centre du diagramme (seulement si vous avez conservé l'option **Initialiser le diagramme**).

HOPEX Web Front-End

Pour créer un diagramme d'environnement d'une application dans **HOPEX Web Front-End** :

1. Accédez à l'application.

2. Faites un clic droit sur l'application et sélectionnez **Nouveau > Diagramme d'environnement d'application**.

Le diagramme créé est nommé automatiquement par **HOPEX IT Architecture**. Il décrit l'application qui constitue l'objet décrit. Cette application est automatiquement placée au centre du diagramme.

Vous pouvez accéder à ce diagramme depuis n'importe quel autre diagramme qui contient l'application décrite.

☞ Voir le guide **HOPEX Common Features** pour plus de détails sur la création, la manipulation et l'ajout d'objets dans un diagramme.

😊 Pour constituer ce nouveau diagramme d'après un diagramme existant, il est plus efficace de copier les objets existants puis de les coller, voir "[Copier les objets des diagrammes existants](#)", page 28.

Copier les objets des diagrammes existants

Dans un diagramme qui correspond à une vision partielle du système d'information, vous pouvez reprendre des éléments issus d'un diagramme plus général, utiles dans cette nouvelle perspective, et créer ensuite les objets complémentaires dont vous avez besoin pour obtenir le niveau de détail souhaité.

Vous pouvez par exemple copier les objets d'une vue d'ensemble préalablement créée par la direction informatique, et les intégrer dans le diagramme d'environnement d'une des applications décrites.

Sélectionnez, par exemple, les sites, les applications et les bases de données d'un diagramme ; copiez-les, allez dans le diagramme de destination et collez le tout.

Cette manipulation permet de récupérer aussi les liens et les mises en forme graphiques précédentes.

Vous pouvez également utiliser la sélection multiple d'objets :

- 】 Cliquez successivement sur les différents objets qui vous intéressent en maintenant la touche <Majuscules> enfoncée.

Réutiliser les objets

Certains objets existent déjà dans le référentiel **HOPEX**.

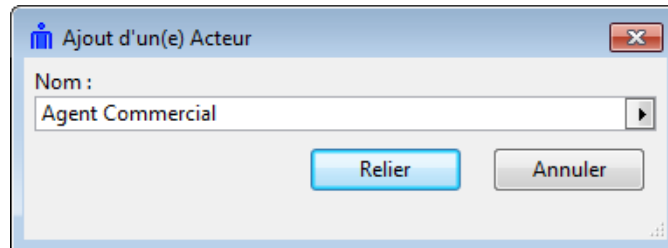
Pour réutiliser dans un diagramme un acteur déjà créé :

1. Accédez au diagramme.
2. Dans la barre d'objets, cliquez sur **Acteur** .
3. Dans le diagramme, cliquez là où vous voulez le voir apparaître. Une fenêtre de création s'ouvre.
4. Saisissez le nom de l'acteur.
Le bouton **Créer** se change en **Relier** dès que vous saisissez la dernière lettre du nom d'un acteur qui existe déjà dans le référentiel (Ex. : l'acteur

"Agent Commercial"). Ce changement indique que l'acteur (Ex. : "Agent Commercial") est connu du système.

☛ Si ce bouton affiche encore **Créer**, c'est qu'une faute de frappe empêche la reconnaissance du nom saisi : une lettre en trop ou en moins par exemple.

Les majuscules et les lettres accentuées ne sont pas discriminantes.



5. Cliquez sur **Relier**.
L'acteur apparaît dans votre diagramme avec les liens qui existent déjà dans le référentiel.

DÉTAILLER L'ARCHITECTURE INTERNE D'UNE APPLICATION

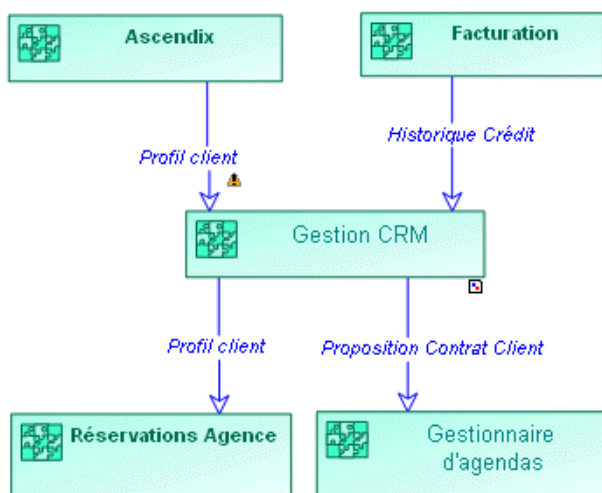
HOPEX IT Architecture permet d'affiner la description de l'architecture interne d'une application. Après avoir créé le diagramme d'environnement d'une application qui indique ses échanges avec d'autres applications, vous pouvez décrire en détail le fonctionnement interne de cette application.

Exemple de diagramme d'environnement de l'architecture de l'application "Gestion CRM".

La direction informatique de la société Alizés met en place une application de gestion de la relation client (Customer Relationship management ou CRM).

Un progiciel "Ascendix" mis en place à cette occasion transmet les profils des clients à l'application de gestion de la relation client qui elle-même met ces profils à la disposition de l'application de réservation des agences de voyages. L'historique des crédits accordés aux clients est transmis à l'application de gestion de la relation client par l'application de facturation et les propositions de contrat sont envoyées au gestionnaire d'agendas des commerciaux.

Gestion CRM - Diagramme d'environnement d'application



Dans la description de l'architecture interne d'une application, on montrera comment les contenus de messages reçus et émis par l'application sont traités par ses composants.

Pour assurer automatiquement la cohérence entre ces deux niveaux de description des applications, **HOPEX** vous propose de représenter les points de communication avec l'extérieur de l'application à l'aide de rôles.

Les composants de l'application décrite (sous-applications ou services) émettent et reçoivent ces messages vers ces rôles. **HOPEX** représente alors automatiquement les applications, acteurs externes qui émettent ou reçoivent des messages avec le même contenu dans la forme des rôles.

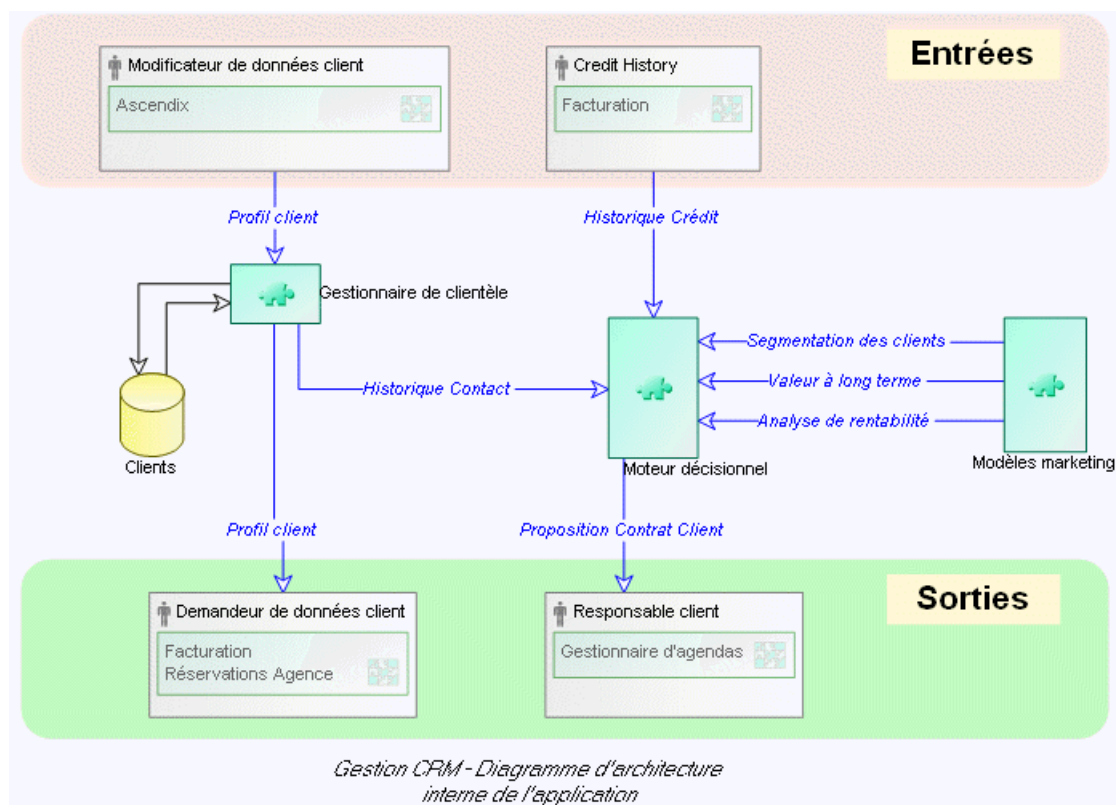
Pour faciliter la réutilisation des contenus de message, **HOPEX** propose une liste de contenus candidats lors du tracé d'un message dont le destinataire ou l'expéditeur est un rôle. L'ensemble des contenus de message émis ou reçus par l'application décrite, tracés par exemple dans un diagramme d'environnement, sont alors proposés dans la fenêtre de création.

Exemple de diagramme d'architecture interne de l'application "Gestion CRM".

Trois services sont gérés par l'application "Gestion CRM".

- Le gestionnaire de clientèle reçoit les profils client et les met à disposition des applications utilisatrices.
- Le moteur décisionnel se sert de l'historique des crédits du client pour élaborer une proposition de contrat.
- Le service des "Modèles marketing" fournit les données nécessaires au moteur décisionnel.

Dans les rôles apparaissent automatiquement les applications émettrices au destinataires des flux échangés avec l'extérieur de l'application "Gestion CRM".



Pour plus de détails sur les diagrammes et les objets décrits, voir aussi :

- ✓ ["Détail des concepts", page 131](#)

Ouvrir un Diagramme d'Architecture Interne d'une Application

Pour ouvrir le diagramme d'architecture interne d'une application :

1. Dans la fenêtre de navigation **Objets principaux**, sélectionnez le dossier **Applications**.
2. Dépliez le dossier de l'application.
3. Double-cliquez sur **Diagramme d'architecture interne de l'application**.

Décrire les flux de message

Vous pouvez définir dans les diagrammes d'architecture applicative les flux de données échangés entre l'application et l'extérieur.

 Pour plus de détails sur la création des flux et leur contenu, voir ["Utiliser les flux", page 15](#).

Lancer l'outil de contrôle (Windows Front-End)

Contrôler les objets d'un diagramme

HOPEX permet de contrôler les objets de votre diagramme.

Pour lancer l'outil de contrôle :

1. Depuis la barre de menus de votre bureau **HOPEX IT Architecture**, sélectionnez **Diagramme > Contrôler > Description de contrôle**. Une fenêtre vous demande de choisir un type de contrôle.

2. Sélectionnez "Contrôle d'un diagramme d'architecture applicative" et cliquez sur **OK**.
Une page HTML s'ouvre et détaille les contrôles effectués.



Les objets non commentés sont signalés. Lorsque vous cliquez sur le lien qui représente le nom de l'objet contrôlé, l'explorateur s'ouvre et vous permet de faire les modifications nécessaires.

Une fois le diagramme complété, vous pouvez à nouveau produire un rapport (MS Word).

☛ Pour plus de détails sur la production d'un rapport (MS Word), voir le guide **HOPEX Common Features**, chapitre "Générer la documentation avec HOPEX", sous-chapitre "Générer des rapports (MS Word)".

Dans **HOPEX IT Architecture**, l'outil de contrôle est disponible sur :

- les diagrammes d'architecture applicative (DAA) ou technique (DAT)
- les objets de type Application et Service (via le menu contextuel).

Appliquer des règles de cohérence

Vous pouvez contrôler votre diagramme en lui appliquant un règlement. Un règlement est un ensemble de règles qui définissent comment modéliser les objets.

Pour appliquer un règlement sur un diagramme :

1. Ouvrez le diagramme en question.
2. Depuis la barre de menus de votre bureau **HOPEX IT Architecture**, sélectionnez **Diagramme > Contrôler > Règlement avec propagation**.

La liste des règlements qui partitionnent les règles apparaît. Lorsqu'un règlement a été défini dans les options, il est proposé par défaut dans la liste.

3. Sélectionnez le règlement que vous souhaitez appliquer au diagramme.
4. Cliquez sur **OK**.

Un rapport s'ouvre sous forme de page HTML. Il contient les résultats du contrôle de cohérence sur les objets du diagramme. Des icônes apparaissent à côté de chaque objet pour indiquer si une règle a été respectée ou non.

 Pour plus de détails sur les règles de cohérence, voir le guide **HOPEX Common Features**, chapitre "Générer la documentation avec HOPEX", sous-chapitre "Contrôler des objets".

DÉCRIRE LES APPLICATIONS INTERNET

Les applications de type Internet/Intranet sont en pleine expansion, notamment en raison de la quasi-gratuité du déploiement de ces applicatifs. Les applications de consultation et de mise à jour de bases de données nécessitent, au-delà du développement, une modélisation fine de l'architecture.

L'important n'est pas de décrire de manière détaillée la structure physique des applications (évolutive par nature), mais l'architecture des grands composants de ces systèmes d'information :

1. Identifiez les applications (classiques ou Intranet), leur décomposition éventuelle en sous-applications, leurs interactions, et les bases de données.
2. Identifiez les sites tels que les hébergeurs, les sites de réplication et les acteurs mis en jeu.
3. Identifiez les messages essentiels pour la compréhension conceptuelle des applications et les principaux services auxquels il est fait appel (service de paiement, service de réservation, etc...).

La vision conceptuelle et synthétique de l'analyse applicative pourra être enrichie d'une description physique, à l'aide notamment de diagrammes d'infrastructure technique.

Exemple : Spécification d'un site marchand.

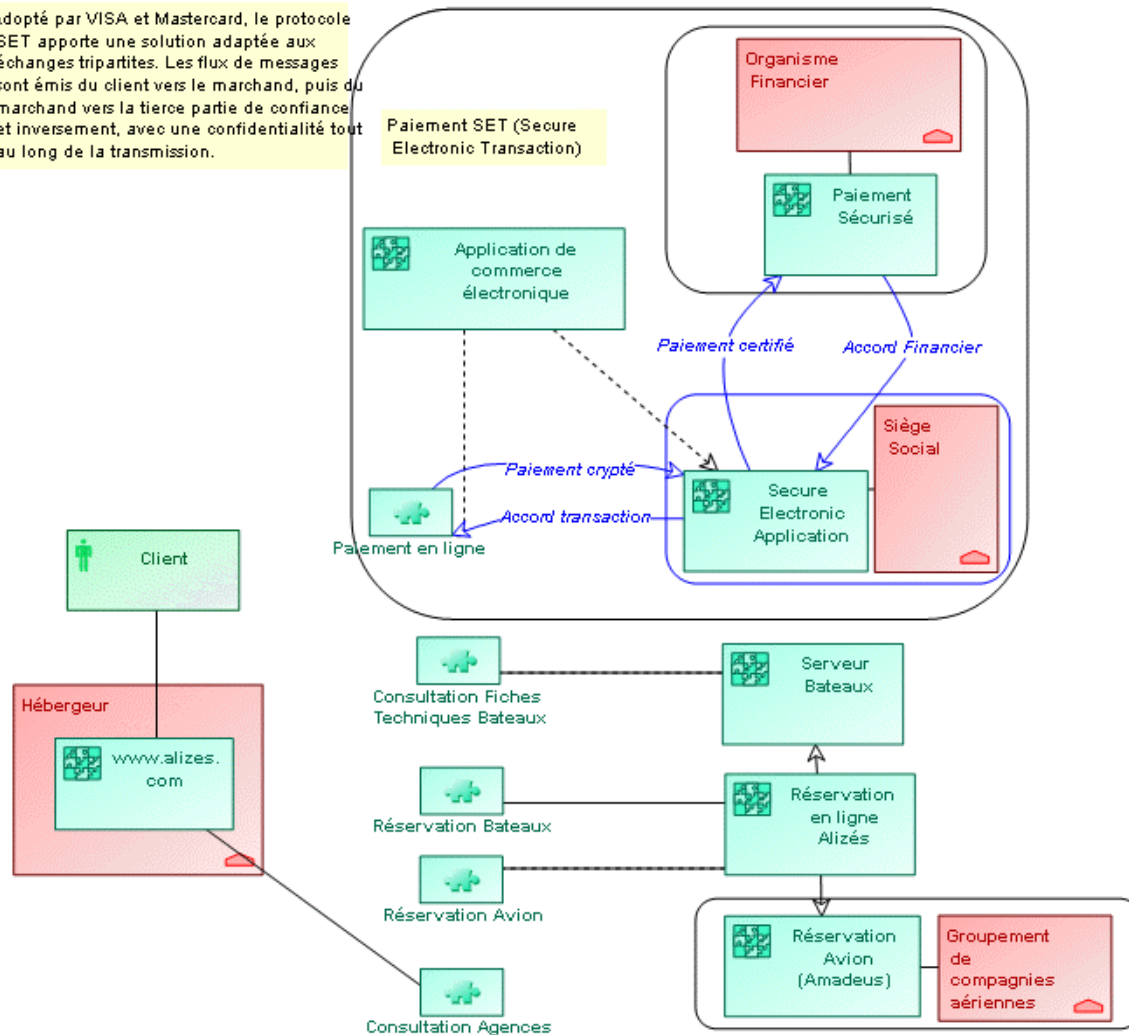
L'entreprise Alizés met à la disposition de ses clients un site Web qui héberge des applications et des données. Il est ainsi possible, à partir du site Web, de consulter le catalogue, de réserver et de procéder à un paiement sécurisé.

Le Directeur Informatique a créé un diagramme qui en décrit le fonctionnement. Il s'agit du Diagramme d'Architecture Applicative Interne SET.

SET (Secure Electronic Transaction) est un protocole permettant de sécuriser les échanges entre le client,

l'entreprise et l'organisme financier. Il a été adopté par Visa et Mastercard.

Adopté par VISA et Mastercard, le protocole SET apporte une solution adaptée aux échanges tripartites. Les flux de messages sont émis du client vers le marchand, puis du marchand vers la tierce partie de confiance et inversement, avec une confidentialité tout au long de la transmission.



Un service est un élément d'une application mis à la disposition de l'utilisateur final.

Dans l'exemple, les utilisateurs disposent, depuis le site Web, de services de consultation, de réservation et de paiement.

A l'aide de références externes, vous pouvez accéder au catalogue de l'entreprise Alizés et visualiser les fiches techniques qui correspondent à chaque type de bateau.

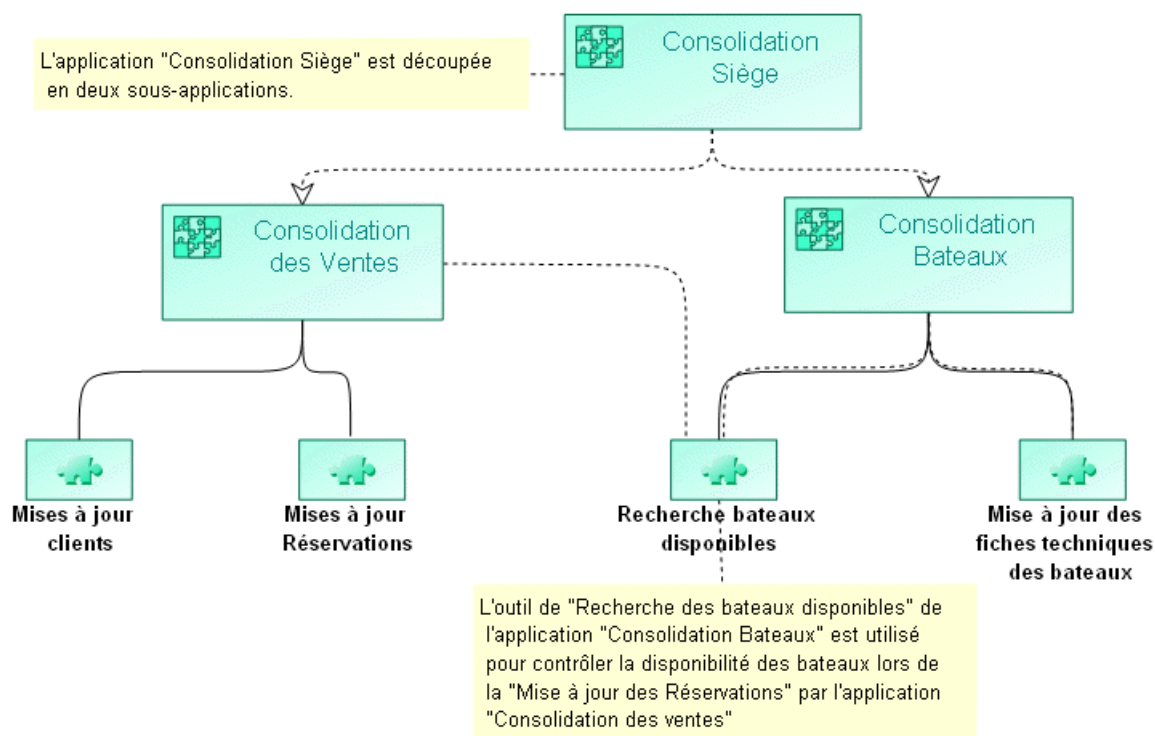
➤ Pour de plus amples informations sur la création des références externes, voir le guide **HOPEX Common Features**.

Pour plus de détails sur les diagrammes et les objets décrits, voir ["Détail des concepts", page 131](#).

CRÉER DES ARBRES APPLICATIFS

Les arbres applicatifs sont des diagrammes qui s'utilisent de la même manière que les diagrammes d'architecture applicative. En règle générale, ils sont construits a posteriori, et servent en quelque sorte de table des matières. Dans un arbre applicatif, vous pouvez distinguer les applications et leurs sous-applications.

Les arbres applicatifs permettent de faire apparaître (et de réparer) les lacunes et incohérences de la décomposition des applications.



Créer l'arbre applicatif d'une application

Pour créer l'arbre applicatif d'une application :

1. Depuis la fenêtre de navigation **Objets principaux**, faites un clic droit sur l'application en question et sélectionnez :
 - (**Windows Front-End**) **Nouveau > Diagramme** puis dans la fenêtre qui apparaît sélectionnez **Arbre applicatif**.
Conservez l'option "Initialiser le diagramme" sélectionnée (cette option permet de disposer d'objets posés automatiquement dans le diagramme qui est créé) et cliquez sur **Créer**.
 - (**Web Front-End**) **Nouveau > Arbre applicatif**.

2. Vous pouvez créer :
 - des applications, voir "Créer une application", page 39.
 - des services, voir "Créer un service", page 39.
 - des liens de composition, voir "Créer un lien de composition", page 39.
 - des liens d'architecture, voir "Créer un lien d'architecture", page 39.

Créer une application

Pour créer une application :

1. Depuis la barre d'objets du diagramme, cliquez sur **Application** .
2. Cliquez dans le diagramme.
La fenêtre de création d'une application apparaît.
3. Saisissez le nom de l'application que vous souhaitez créer.
4. Cliquez sur **Créer**.
L'application apparaît dans votre diagramme.

Créer un service

Un service se crée de la même façon qu'une application, dans la barre d'objets du diagramme à l'aide de **Service applicatif** .

Créer un lien de composition

Les liens de composition définissent les applications qui entrent dans le développement et le déploiement d'une application.

Pour créer un lien de composition entre deux applications :

1. Depuis la barre d'objets du diagramme, cliquez sur **Lien** .
2. Dans le diagramme, reliez la première application à la deuxième.
La liste des éléments déployés avec une application s'affiche dans la fenêtre de propriétés de l'application en question, sous la zone **Sous-applications** de l'onglet **Caractéristiques**.

Créer un lien d'architecture

Les liens d'architecture s'appliquent aux sous-élément impliqués dans le fonctionnement de l'application mais qui ne sont pas nécessairement déployés avec elle.

Les liens d'architecture ne sont pas visibles par défaut. Pour les afficher :

1. Depuis la barre d'outils du diagramme, cliquez sur **Vues et détails** .
2. Dans la fenêtre qui apparaît, sélectionnez la vue **Exécution**.
3. Cliquez sur **OK**.

Pour créer un lien d'architecture entre deux applications :

1. Depuis la barre d'objets du diagramme, cliquez sur **Lien** .
2. Dans le diagramme, reliez la première application à la deuxième.
La fenêtre de création d'un lien apparaît.

3. Sélectionnez le type de lien "Application de l'architecture interne" et cliquez sur **OK**.

Les liens d'architecture apparaissent en pointillés.

La liste des éléments impliqués dans le fonctionnement d'une application est disponible dans la fenêtre de propriétés de l'application en question, sous l'onglet **Architecture**, sous-onglet **Architecture applicative**.

DÉCRIRE L'INFRASTRUCTURE DES TÉLÉCOMMUNICATIONS ENTRE SITES

Pour décrire l'infrastructure des télécommunications entre les différents sites d'implantation de l'entreprise, il est intéressant de réaliser une série de diagrammes, d'échelle plus ou moins fine, présentant les connexions physiques ou virtuelles qui supportent les échanges d'informations.

Les descriptions peuvent être faites par raffinements successifs, jusqu'au niveau de détail souhaité.

Exemple

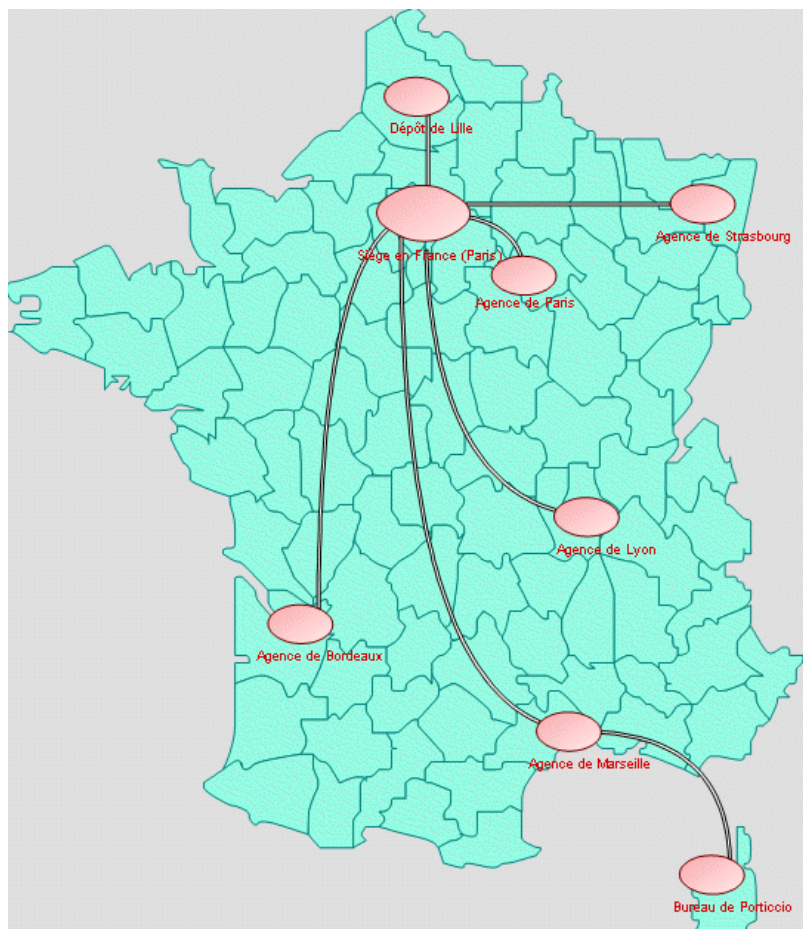
Infrastructure des télécommunications

Le Directeur Informatique d'Alizés a réalisé plusieurs diagrammes présentant l'infrastructure des télécommunications entre les différents sites d'implantation de la société.

Un premier diagramme présente le mode de communication entre le siège et les 3 bureaux de représentation d'Alizés situés hors de France.

Un second diagramme s'intéresse aux communications métropolitaines entre le siège social et les 5 agences de

voyages, ainsi qu'avec le bureau de représentation de Porticcio, en Corse.



Consulter un diagramme existant

Pour consulter la première carte, il faut retrouver ce diagramme parmi tous les travaux déjà réalisés par la Direction informatique d'Alizés et mémorisés dans le référentiel.

Pour consulter un diagramme existant :

1. Dans la fenêtre de navigation **Objets principaux**, déployez le dossier **Infrastructures techniques**.

2. Dans la liste, faites un clic droit sur l'infrastructure et sélectionnez **Diagramme d'infrastructure technique**.

☛ Alternative **Windows Front-End** : double-cliquez sur l'infrastructure ().

Le diagramme s'ouvre.

Insérer des éléments graphiques dans un Diagramme d'Infrastructure Technique (Windows Front-End)

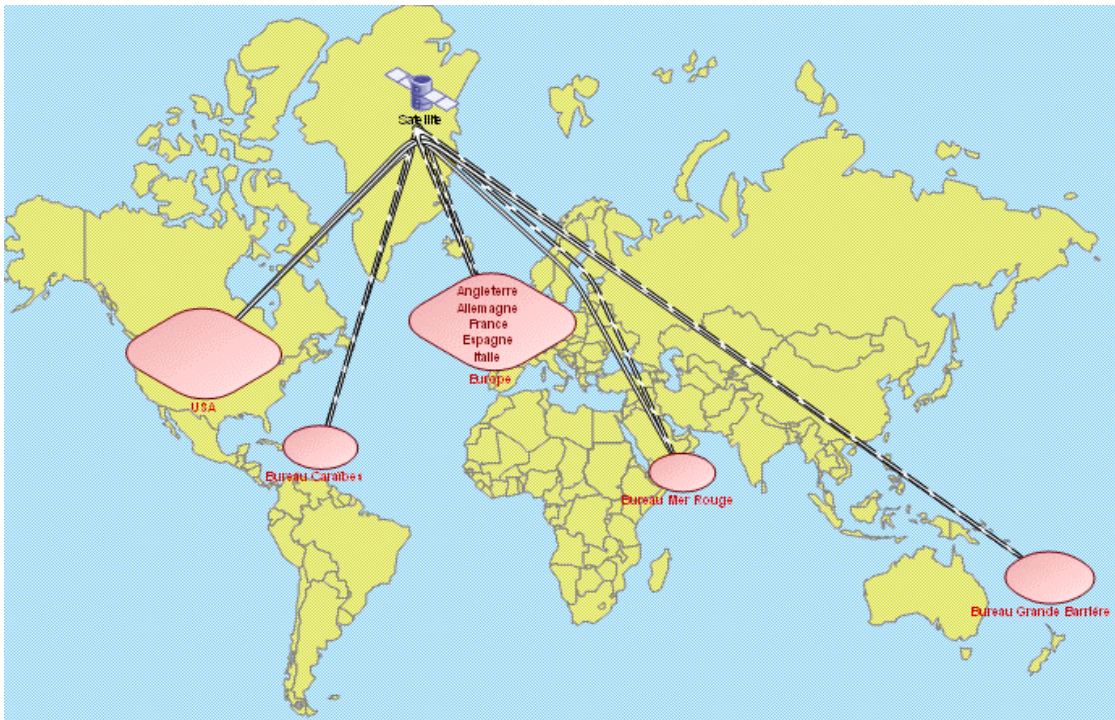
Un fond peut être utilisé pour améliorer la compréhension du diagramme.

Ex. : une carte du monde.

Pour afficher ou masquer ce fond :

1. Depuis la barre de menus **HOPEX (Windows Front-End)**, sélectionnez **Affichage > Arrière-plan**.

Exemple



HOPEX IT Architecture propose un certain nombre d'éléments graphiques usuels, que vous pouvez compléter, par exemple en utilisant une bibliothèque d'images.

☺ Vous pouvez aussi enrichir un diagramme avec des objets graphiques divers, importés par copier-coller.

MODÉLISER LES INFRASTRUCTURES TECHNIQUES

Les principes d'utilisation du diagramme d'infrastructure technique sont similaires à ceux décrits précédemment.

Les diagrammes d'infrastructure technique permettent d'identifier et de représenter les matériels -serveurs, postes de travail, etc. - et de décrire leurs caractéristiques, en termes de système d'exploitation, de puissance de calcul, de capacité mémoire ou de surface disque. Les réseaux qui relient ces machines sont documentés en termes de topologie, protocole et débit.

Ces diagrammes indiquent sur quelles machines sont installées les bases de données et les applications.

Exemple : Diagramme d'infrastructure technique du siège social.

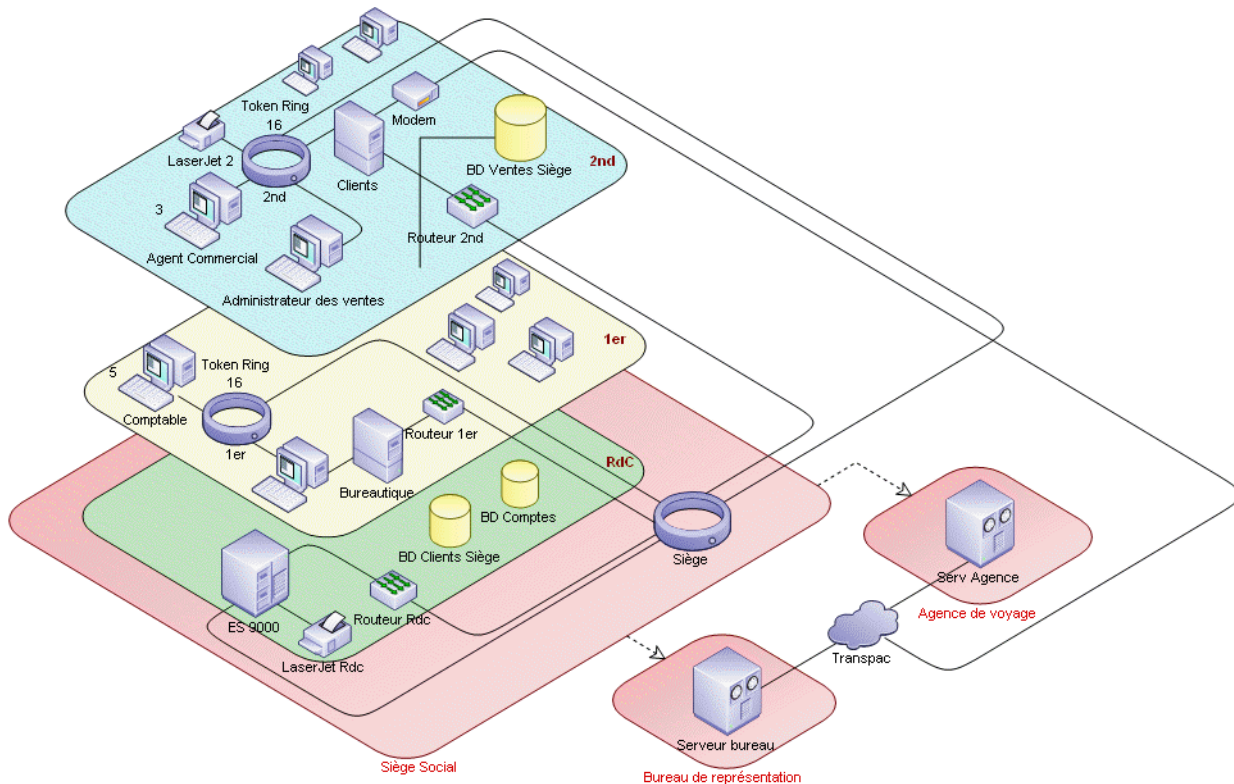
La Direction Informatique d'Alizés a réalisé un diagramme d'infrastructure technique du siège social (DIT Siège).

Ce diagramme présente l'équipement informatique du siège, et en particulier les équipements de consolidation des

ventes (postes au 2ème étage) et les équipements "administratifs" (1er étage).

Les deux réseaux locaux sont interfacés.

Les communications avec les agences et les bureaux de représentation sont gérées sur le serveur du second étage.



Les objets manipulés dans un Diagramme d'Infrastructure Technique sont :

- les sites
- les serveurs
- les applications
- les bases de données
- les postes de travail
- les réseaux
- les noeuds de ces réseaux

☛ Voir *"Les Diagrammes d'Infrastructure Technique", page 137.*

Vous pouvez personnaliser votre diagramme. Voir le guide **HOPEX Common Features**, chapitre "Manipuler les diagrammes" pour :

- modifier la forme d'un objet

☛ La forme utilisée pour certains objets varie en fonction du type qui leur est attribué. Par exemple, à la création d'un nœud, vous pouvez choisir parmi les types suivants : "Satellite", "Imprimante", "Concentrateur", "Modem", "Routeur" et "Pont". De même, après avoir

*créé un réseau, vous pouvez préciser son type dans l'onglet **Caractéristiques** de sa fenêtre de propriétés.*

- afficher les caractéristiques des objets

Vous pouvez par exemple choisir de faire apparaître le nombre de postes disponibles pour un type d'acteur (Ex. : "Comptable").

☛ Ces valeurs peuvent être modifiées dans l'onglet **Caractéristiques** de la fenêtre de propriétés de l'objet.

Il peut être intéressant de paramétrer l'affichage des caractéristiques d'autres objets, par exemple des réseaux, pour lesquels vous pouvez afficher la topologie, le protocole et le débit de transmission.

- modifier la position des champs de texte (**Windows Front-End**).

RAPPORTS SUR LES ARCHITECTURES APPLICATIVES

Les rapports permettent d'analyser les données d'un référentiel afin d'obtenir une meilleure vue de l'information.

Il existe plusieurs rapports types, propres au domaine et à l'objet de l'analyse. Ici sont présentés les rapports types utilisés dans le cadre de **HOPEX IT Architecture**.

Les rapports types de HOPEX IT Architecture

Un rapport type définit les paramètres et les données sur lesquels se base un rapport.

Prérequis :

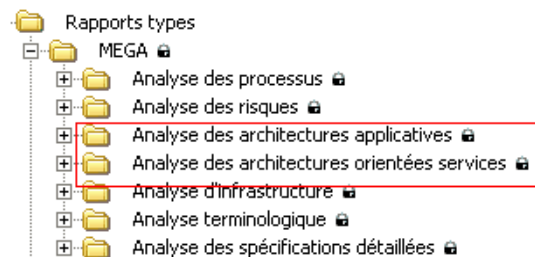
Pour afficher les rapports types vous devez avoir accès au métamodèle "avancé".

Pour configurer votre accès au métamodèle :

1. Accédez à la fenêtre de gestion des **Options** :
 - **(Windows Front-End)** dans la barre de menus de votre bureau **HOPEX IT Architecture** sélectionnez **Outils > Options**
 - **(Web Front-End)** dans le groupe d'outils **Divers** de votre bureau **HOPEX IT Architecture** sélectionnez **Mon compte > Options**.
2. Dans l'arbre des options sélectionnez **Référentiel** et dans le champ de l'option **Accès au métamodèle** sélectionnez "Avancé".
3. Cliquez sur **OK**.

Pour accéder aux rapports types ;

1. Accédez à la fenêtre de navigation **Utilitaires**.
2. Dépliez le dossier **Rapports Types**.
Les rapports types propres à **HOPEX IT Architecture** concernent différents domaines; les architectures applicatives, les architectures orientées services (SOA), les contraintes temporelles qui agissent sur le SI etc.



Rapports types liés aux architectures applicatives :

- Analyse fonctionnelle d'application : une analyse fonctionnelle d'application compare un groupe d'éléments fournissant des fonctionnalités vis-à-vis d'un ensemble de fonctionnalités attendues (le périmètre fonctionnel).
- Gestion de l'architecture applicative : ce type de rapport est dédié à l'étude de la gestion d'une architecture applicative. Il donne les détails des projets et des personnes impliquées dans cette gestion.
- Compatibilité des infrastructures : la compatibilité des infrastructures permet l'analyse des infrastructures techniques impliquées dans un système d'information. Par exemple, ce type de rapport permet de montrer le niveau d'utilisation d'un ensemble de technologies ou la compatibilité d'un ensemble de sous-systèmes d'information avec un groupe d'infrastructures techniques.
- Déploiement : ce type de rapport est dédié au problème de déploiement. Il permet de répondre aux questions de type "où sont déployées les applications ?", "quelles différences existe-t-il entre une architecture applicative et un de ses déploiements ?".
- Gestion de projet : ce type de rapport décrit l'impact d'un ensemble de projets sur le système d'information.
- Analyse de l'architecture As Is - To Be : ce type de rapport fournit une vue As Is - To Be d'architectures applicatives, en fonction des périodes de temps sélectionnées.

Rapports types liés aux architectures orientées services :

- Supervision des processus automatisés : ce type de rapport permet de retrouver les indicateurs et tableaux de bord dédiés à la supervision des processus automatisés.
- Analyse fonctionnelle de services : ce type de rapport compare un groupe de services informatiques fournissant des fonctionnalités vis-à-vis d'un ensemble de fonctionnalités attendues (le périmètre fonctionnel).
- Automatisation des processus : ce type de rapport détecte les potentialités d'amélioration du système d'information dans son support des activités humaines.
- Services It associés à un processus : ce type de rapport permet de retrouver les services en relation avec un ensemble de processus. Voir "[Architecture orientée services \(SOA\)](#)", page 121.
- Impacts des services IT sur les processus : les modifications opérées sur les services IT ont un impact sur les processus qui les utilisent dans le cadre de leur implémentation. Cet outil retrouve tous les processus touchés par un ensemble de services IT.
- Supervision des services informatiques : ce type de rapport permet de retrouver les indicateurs et tableaux de bords dédiés à la supervision des services informatiques et des applications.

Rapports d'analyse contextuelle :

- Analyse des contraintes temporelles : ce type de rapport permet de représenter et de modifier les contraintes temporelles entre les objets sélectionnés.
- Analyse des objets variés : ce type de rapport détaille les variations à partir d'objets variés donnés.
- Analyse des objets variants.
- Analyse des variations : ce type de rapport permet de détailler les objets variés et variants, ainsi que les variations liées, à partir de variations données.
Pour plus de détails sur les variations, voir le guide **HOPEX Common Features**.

Ces rapports types constituent des modèles et vous permettent de créer des rapports avec des paramètres prédéfinis.

☛ Pour plus de détails sur l'utilisation des rapports, voir le guide **HOPEX Common Features**, chapitre "Générer la documentation".

☛ Pour créer vos propres rapports types, voir le guide **HOPEX Power Studio**.

DÉCRIRE LES INTERACTIONS



Les points suivants présentent comment décrire les interactions entre les applications de l'entreprise et celles de ses partenaires.

- ✓ ["Contexte d'utilisation", page 52](#)
- ✓ ["Décrire une interaction", page 57](#)

CONTEXTE D'UTILISATION

L'utilisation des interactions permet de décrire en détail l'automatisation des échanges entre une entreprise et des acteurs externes.

Ces échanges peuvent être décrits en utilisant des standards définis par des groupements internationaux tels que l'OAG (Open Applications Group) ou par des fournisseurs de places de marché.

Une entreprise peut également définir ses propres modes d'interaction avec ses filiales, par exemple, ou avec ses sous-traitants.

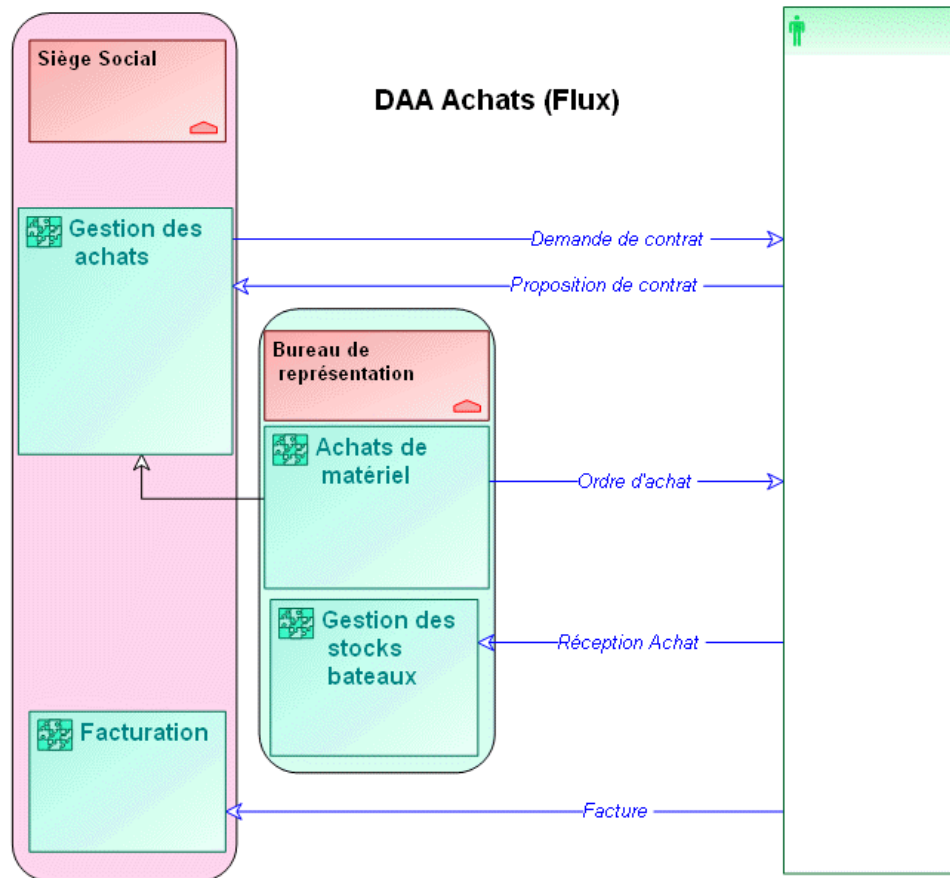
L'exemple présenté ici permet de décrire les échanges entre une entreprise et ses fournisseurs. Il s'appuie sur l'application "Système de Gestion des Achats".

Diagramme d'architecture interne du système de gestion des achats

L'application "Système de Gestion des Achats" est décrite par deux **Diagrammes d'architecture applicative interne**.

Le diagramme "Système de Gestion des Achats - DAA Achats (Flux)" présente :

- les principales applications qui interviennent dans le système de gestion des achats.
- les principaux flux échangés avec les fournisseurs dans le cadre du système de gestion des achats.



Les achats de matériel pour entretenir les bateaux sont effectués directement par le bureau de représentation en utilisant l'application "Achats de matériel". Celle-ci fait appel à une application centralisée de "Gestion des achats" au siège.

S'il s'agit d'un nouveau type de matériel ou d'un nouveau fournisseur, une demande de contrat est envoyée au

fournisseur à l'aide de l'application du siège. Celui-ci renvoie en échange une proposition de contrat.

Dans les autres cas, un agent du bureau de représentation envoie un ordre d'achat au fournisseur à l'aide de l'application "Achats de matériel".

La réception des achats livrés par le fournisseur est réalisée à l'aide de l'application "Gestion des stocks bateaux" du bureau de représentation.

La facture envoyée par le fournisseur est traitée par l'application "Facturation" du siège.

☛ Vous pouvez afficher les activités qui correspondent aux applications, voir ["Afficher les activités qui correspondent aux applications"](#), page 54

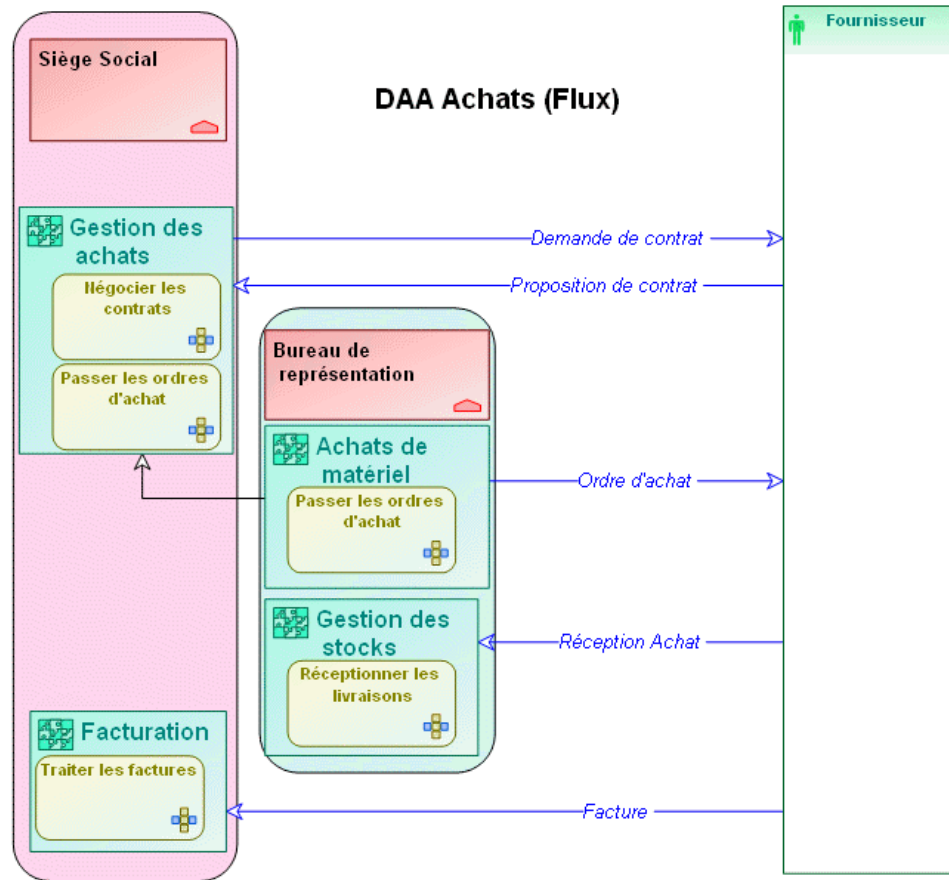
Afficher les activités qui correspondent aux applications

☛ Si vous disposez du produit **HOPEX Business Process Analysis**, vous pouvez afficher les activités qui correspondent aux applications.

Pour afficher les activités qui correspondent aux applications :

1. Dans la barre d'outils du diagramme, cliquez sur **Vues et détails**  .

2. Sélectionnez la vue **Métiers, Activités, Fonctionnalités, cas**.



L'activité "Passer les ordres d'achat" est répartie entre les applications "Gestion des achats" du siège et "Achats de matériels" du bureau de représentation.

Les activités "Négocier les contrats", "Réceptionner les livraisons" et "Traiter les factures" sont prises en compte respectivement par les applications "Gestion des achats", "Gestion des stocks bateaux" et "Facturation".

☛ La mise en œuvre des interactions par les activités d'un processus sont décrites dans la section ["Informatisation des échanges : les interactions"](#), page 55.

Informatisation des échanges : les interactions

📖 Une interaction 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

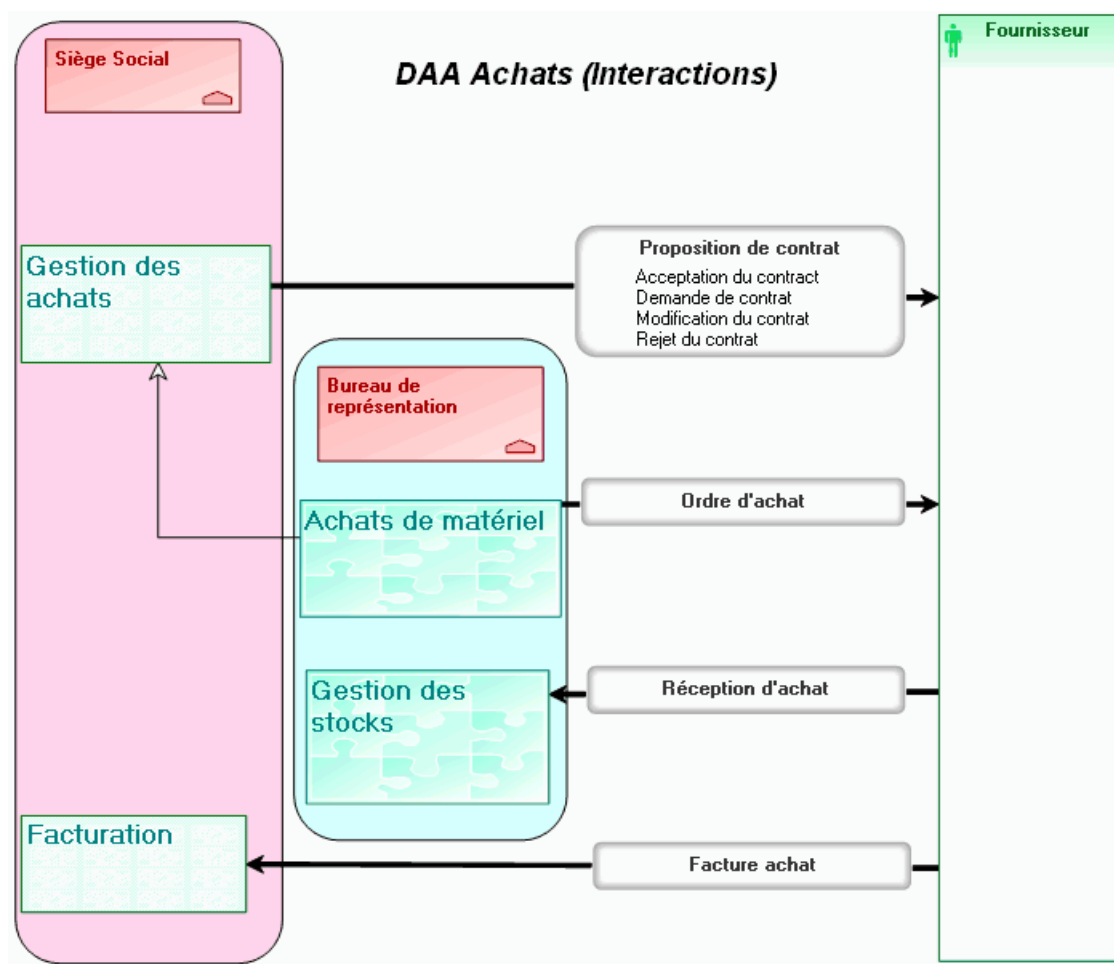
l'entreprise. Le contenu de ce contrat est décrit par un contrat d'échange.

 *Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.*

La représentation des flux échangés par les applications de l'entreprise avec ses fournisseurs peut être suffisante pour une compréhension globale des relations entre l'entreprise et ses fournisseurs.

Elle n'est cependant pas assez précise pour permettre l'informatisation des échanges avec ces fournisseurs.

Le **Diagramme d'architecture applicative interne** ci-dessous présente les échanges sous forme d'interactions. Les interactions permettent de décrire de manière précise l'interaction effective entre les partenaires.



DÉCRIRE UNE INTERACTION

Créer une interaction avec un nouveau contrat d'échange existant

Pour créer une interaction :

1. Dans la barre d'objets du diagramme, cliquez sur **Interaction** .
2. Tirez un lien entre les deux entités en communication
La fenêtre **Ajout d'une interaction** apparaît.
3. Cliquez sur la flèche à droite du champ **Contrat d'échange** et sélectionnez **Créer contrat d'échange**.
4. Saisissez le **Nom** de votre échange.
5. Cliquez sur **OK** pour fermer cette fenêtre.
L'échange est automatiquement créé.
6. Cliquez sur **OK**.
L'interaction apparaît dans le diagramme.

Décrire le contrat d'échange

Un **Contrat d'échange** peut être décrit par des **Echanges** ou des **Contrats d'échange** qui représentent les échanges d'informations entre des composants d'architecture.

 *Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.*

 *Un échange utilisé représente l'utilisation d'un échange dans un contrat d'échange.*

 *Un contrat d'échange utilisé représente l'utilisation d'un contrat d'échange dans un autre contrat d'échange.*

Pour décrire qu'un échange est utilisé par un contrat d'échange :

1. Ouvrez la fenêtre de propriétés du contrat d'échange
2. Cliquez sur l'onglet **Echange**
3. Cliquez sur le bouton **Nouveau**
Une fenêtre de choix s'ouvre.
4. Sélectionnez **Echange utilisé**, qui est le type d'échange que vous voulez utiliser et cliquez sur **OK**
La fenêtre de création s'ouvre.
5. Cliquez sur la flèche à droite du champ **Spécification**.
6. Sélectionnez **Lister** dans la liste déroulante et choisissez l'échange que vous voulez associer à l'échange que vous utilisez.
Le nom de l'échange apparaît dans le champ **Spécification**.
7. Sélectionnez dans le champ **A partir de** le rôle de l'échange décrit relié au rôle Consommateur de l'échange utilisé.

8. Sélectionnez dans le champ **Jusqu'à** le rôle de l'échange décrit relié au rôle Fournisseur de l'échange utilisé.
9. Cliquez sur **OK**.
 Vous pouvez associer plusieurs échanges au contrat d'échange.
10. Cliquez sur **OK**.

Décrire les flux d'un échange

Le contenu d'un échange est décrit par les flux et leur contenu qui sont échangés entre les deux rôles qui représentent les parties prenantes dans la conversation.

 Un flux représente la circulation d'information à l'intérieur de l'entreprise ou entre l'entreprise et son environnement. Un flux peut transporter 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.

Pour décrire les flux échangés :

1. Ouvrez la fenêtre de propriétés de l'échange.
2. Cliquez sur l'onglet **Flux**.
3. Cliquez sur le bouton **Nouveau**.
 La fenêtre de **Création de Flux avec contenu** s'ouvre.
4. A partir de la liste déroulante du champ **Contenu**, sélectionnez le contenu que vous souhaitez associer au flux.
 Le flux s'affiche avec son contenu dans la liste des contenus de la conversation.
 Vous pouvez associer plusieurs contenus au flux.
5. Précisez la direction de chacun des flux.
6. Cliquez sur **OK**.

MODÉLISER DES INFRASTRUCTURES COMPLEXES



HOPEX IT Architecture vous permet d'aller plus loin dans la description de l'architecture de systèmes et de prendre en compte des systèmes faisant intervenir des ressources autres que logicielles. Par exemple, un système peut être composé d'un radar, de pylônes, de différents logiciels, d'un réseau, de personnes etc. Afin de définir des systèmes hétérogènes, **HOPEX IT Architecture** fournit trois diagrammes : l'arbre d'architecture de ressources, le diagramme d'architecture de ressources et le diagramme d'assemblage d'objets manufacturés. Ils permettent de modéliser les composants du système ainsi que leur mode de communication.

Les points traités ici sont les suivants :

- ✓ "Contexte d'utilisation", page 60
- ✓ "Décrire un environnement d'architecture de ressources", page 69
- ✓ "Décrire un objet manufacturé", page 75
- ✓ "Décrire les communications d'une infrastructure complexe", page 78
- ✓ "Synthèse des termes utilisés", page 85

☛ L'ensemble des fonctionnalités de modélisation des infrastructures complexes sont accessibles si l'option **Modélisation d'infrastructure Post 2009 SP2** est sélectionnée (dans les **Options** de **HOPEX**, dossier **Modélisation des processus et de l'architecture**)

CONTEXTE D'UTILISATION

Les fonctionnalités proposées par **HOPEX IT Architecture** pour la modélisation des infrastructures complexes permettent de représenter les ressources matérielles et organisationnelles nécessaires au fonctionnement du système, les interactions entre les composants, les moyens de communication qui supportent ces interactions et les services offerts et utilisés par l'architecture modélisée.

Les composants décrits peuvent suivre une méthodologie de modélisation 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 de support présenté selon une approche descendante à partir de diagrammes d'architecture de ressources et d'un diagramme d'assemblage d'objets manufacturés.

Exemple d'environnement d'architecture de ressources

Diagramme d'environnement d'architecture de ressources

 *Un environnement d'architecture de ressources présente un contexte d'utilisation d'architectures de ressources. Il décrit les interactions, entre l'architecture de ressource et ses partenaires externes, qui lui permettent de remplir sa mission et d'assurer les fonctionnalités attendues.*

Le centre de support assure un service à ses clients. Il s'appuie sur un partenaire extérieur pour les questions qui requièrent une expertise technique.

Le diagramme est composé de :

- Un élément de type **utilisation d'architecture** qui représente une architecture de ressources interne à l'environnement.

 *Une utilisation d'architecture est la mise en place d'une architecture de ressources dans une autre architecture de ressources ou dans un environnement d'architecture de ressources.*

- Une **ressource d'architecture partenaire** qui représente une architecture de ressources externe utilisée dans le contexte de l'environnement d'architecture de ressources décrit.
- Un composant de type **utilisateur d'architecture de ressource** qui représente les catégories d'utilisateurs des services proposés par les architectures de ressources internes à l'environnement.

 *Un utilisateur d'architecture de ressources représente une unité organisationnelle interagissant aux frontières d'un environnement d'architecture de ressources.*

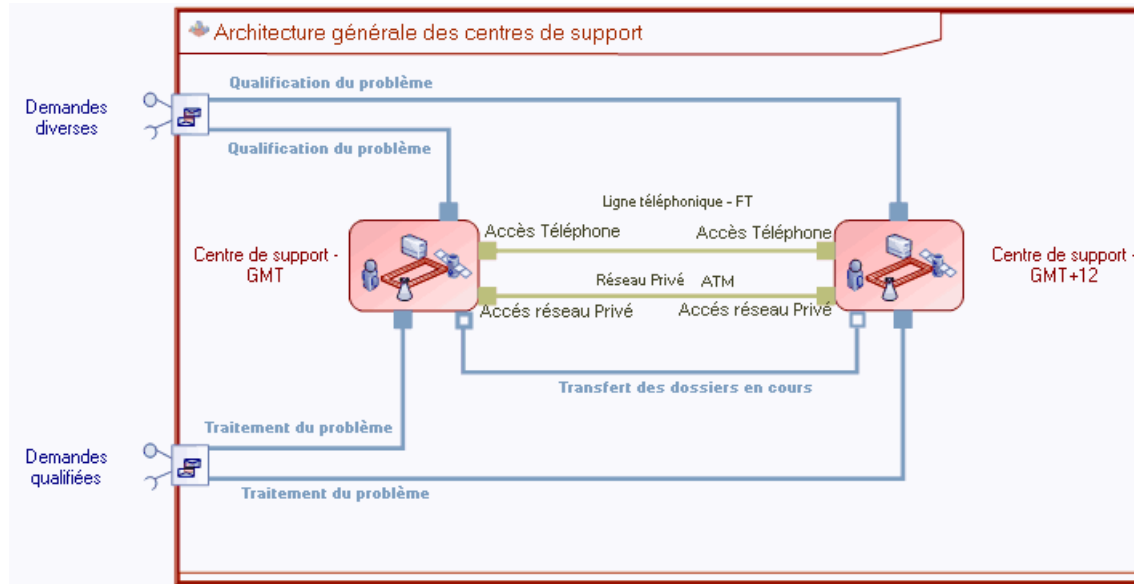
- Des **interactions** entre les composants qui représentent les demandes de services.

 *Une interaction 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 un contrat d'échange.

Diagramme d'architecture de ressources

Le diagramme suivant décrit l'environnement du centre de support.



La couverture 24/24 est assurée par deux équipes "Centres de support", une à Paris (Centre de support GMT) et l'autre à Singapour (Centre de support GMT+12). Chacune d'elle assure un service de support sur 12 heures.

A la fin de son service, chaque centre transfère les dossiers au centre qui prend le relais.

Un service de support est assuré pour prendre en charge en temps réel toutes les demandes client, un second service de support est assuré en temps différé pour des demandes précises déjà qualifiées.

Les composants de l'*architecture de ressources* sont décrits dans un diagramme d'architecture de ressources.

 Une architecture de ressources est la combinaison d'équipements matériels et d'organisation configuré pour fournir une capacité.

Le diagramme est composé de :

- deux éléments de type *Utilisation d'architecture* représentant les deux centres de support.

 Une utilisation d'architecture est la mise en place d'une architecture de ressources dans une autre architecture de ressources ou dans un

environnement d'architecture de ressources.

- deux **points de service** représentant les deux services de support offerts par chacun des centres.



Un point de service est un point à partir duquel le système reçoit une demande de service d'un autre système et fournit le service demandé.

- d'**interactions** entre les points de service et les centres représentant les demandes de services, d'une part, et le transfert des dossiers en cours, d'autre part.



Une interaction 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 un contrat d'échange.

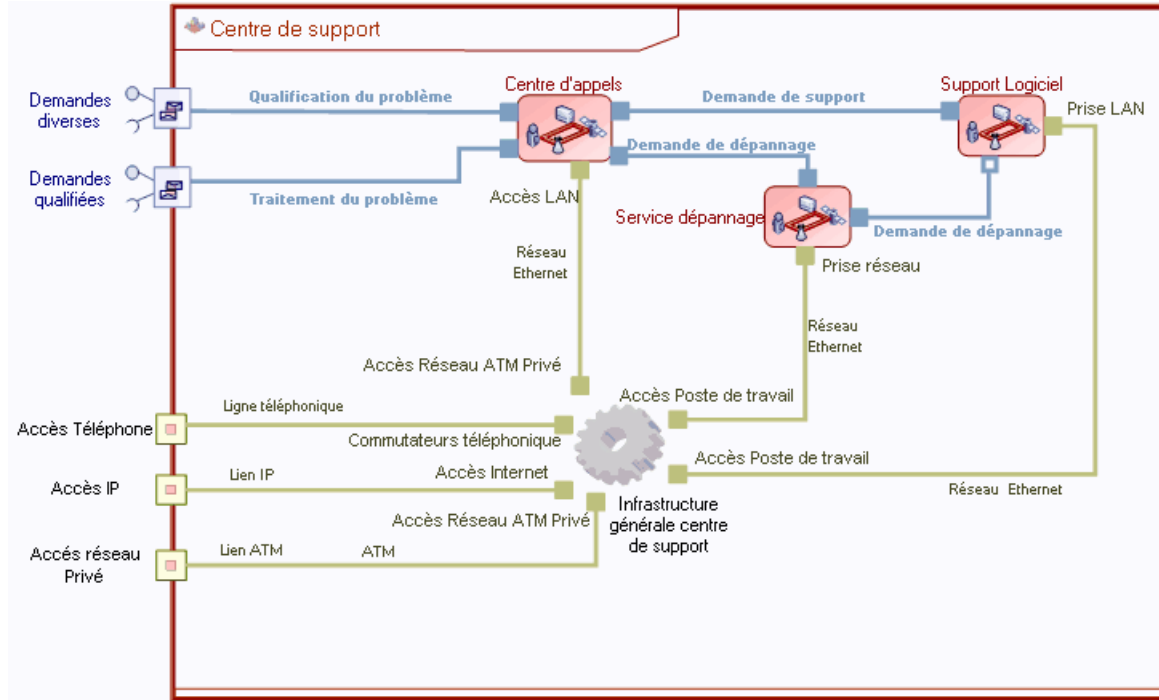
- des **canaux de communication** matériels entre les deux centres.



Un canal de communication permet d'établir une connexion physique entre deux ressources matérielles. Il supporte les interactions qui définissent les protocoles de communication entre ces ressources. Les canaux de communication connectent les ressources avec l'extérieur par le biais des ports de communication.

Architecture d'un centre de support

A partir de l'architecture générale du système, vous pouvez accéder au diagramme d'architecture de ressources qui décrit la structure commune aux deux centres de support.



Le centre d'appels traite l'ensemble des demandes, qualifiées ou non, les vérifie et les renvoie soit au support logiciel, soit au service dépannage.

Si le support logiciel détecte un problème de fonctionnement lié au matériel, il enregistre l'incident et avertit l'équipe de dépannage.

Les trois structures s'appuient sur une même infrastructure matérielle générale.

Ce diagramme est composé des ressources matérielles suivantes :

- des **équipements**.



Un équipement représente la mise en œuvre d'un objet manufacturé dans une architecture de ressources.

- des **ports de communication** qui représentent les points de communication physiques de l'architecture matérielle et qui implémentent les points de service.



Un port de communication est un point physique de communication avec une ressource. Ils se conforment à des protocoles de communication particuliers. Un port de communication implémente des

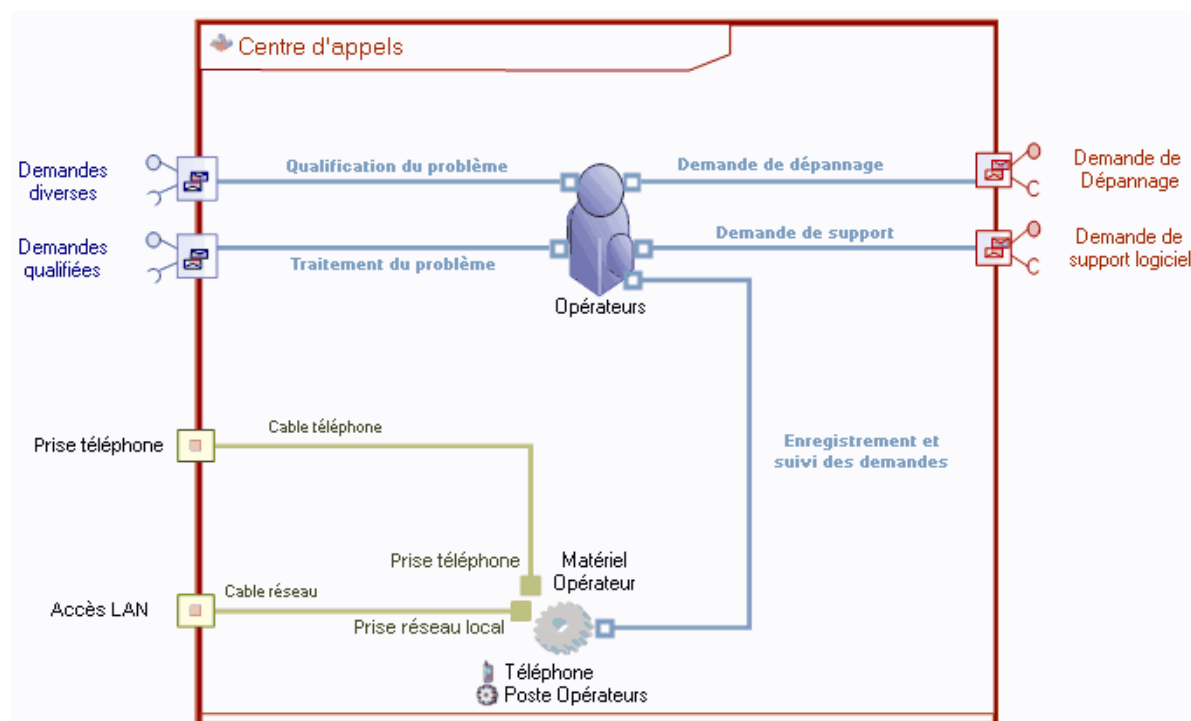
points de service et de requête.

- des **canaux de communication** matériels entre les deux centres.

 Un canal de communication permet d'établir une connexion physique entre deux ressources matérielles. Il supporte les interactions qui définissent les protocoles de communication entre ces ressources. Les canaux de communication connectent les ressources avec l'extérieur par le biais des ports de communication.

Architecture d'un centre d'appels

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



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 transmet l'appel à l'intervenant compétent (support logiciel ou service dépannage) via des points de requête.

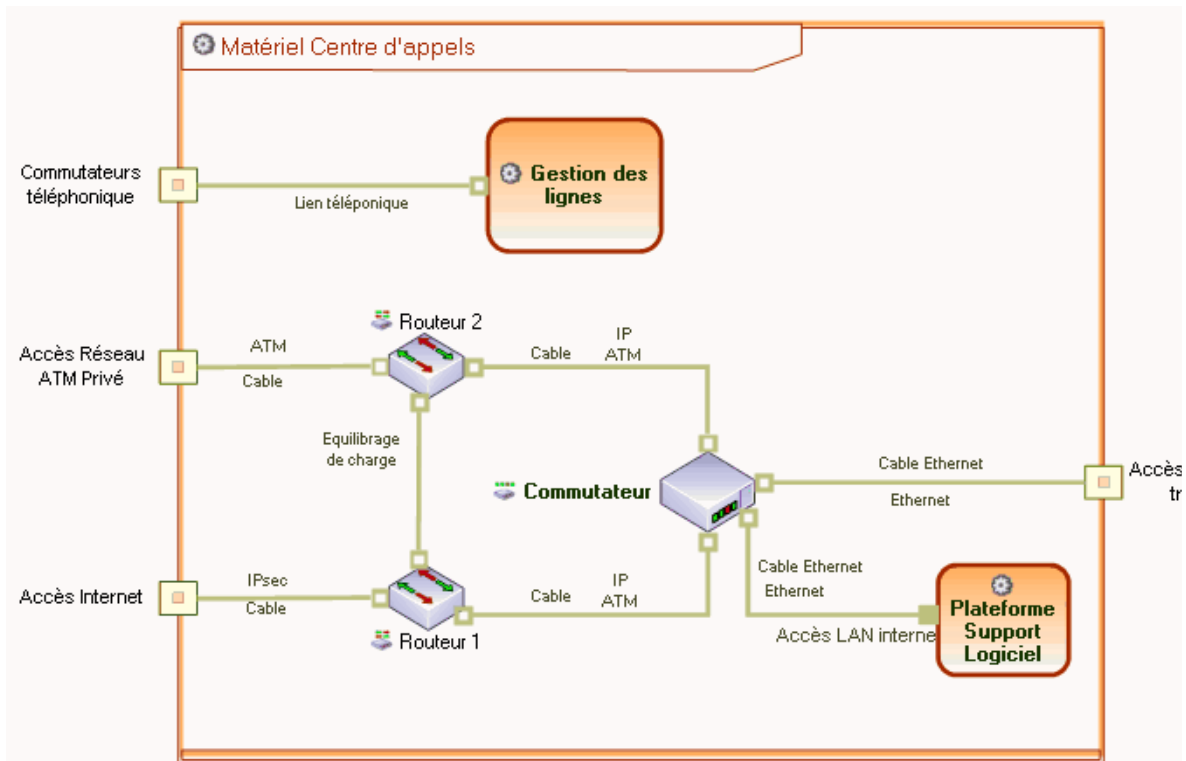
On suppose pour simplifier que chaque équipe est composée d'un responsable et de différents membres dont on ne décrira pas le niveau de spécialisation, ni la plage horaire couverte.

Ce diagramme contient deux *points de requête* à partir desquels les opérateurs effectuent des demandes de service aux autres architectures de ressources.

 Un point de requête est un point à partir duquel le système envoie une requête à un autre système et reçoit le service demandé.

Exemple d'objets manufacturés

Diagramme d'assemblage du matériel d'un centre de support



L'architecture matérielle de base d'un centre d'appels comprend trois points de liaison avec l'extérieur : un lien téléphonique, un lien vers un réseau privé ATM et un lien d'accès au réseau internet. Le point d'accès aux postes de travail permet de connecter les postes opérateur, support et dépannage.

Ce diagramme contient deux *composants manufacturés* représentant les équipements matériels d'un standard téléphonique et une plate-forme de tests utilisée pour le support logiciel.

 Un composant manufacturé représente la mise en place d'un produit manufacturé au sein d'un autre.

Notez que les *protocoles de communication* utilisés sont précisés sur les canaux de communication. Ils le sont également sur les ports de communication.

 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.

Les routeurs sont des objets manufacturés prédéfinis, proposés dans la bibliothèque de **HOPEX IT Architecture**.

Diagramme d'assemblage du matériel d'un PC standard

Le diagramme, plus précis, d'assemblage du matériel d'un PC standard est présenté ci-dessous.

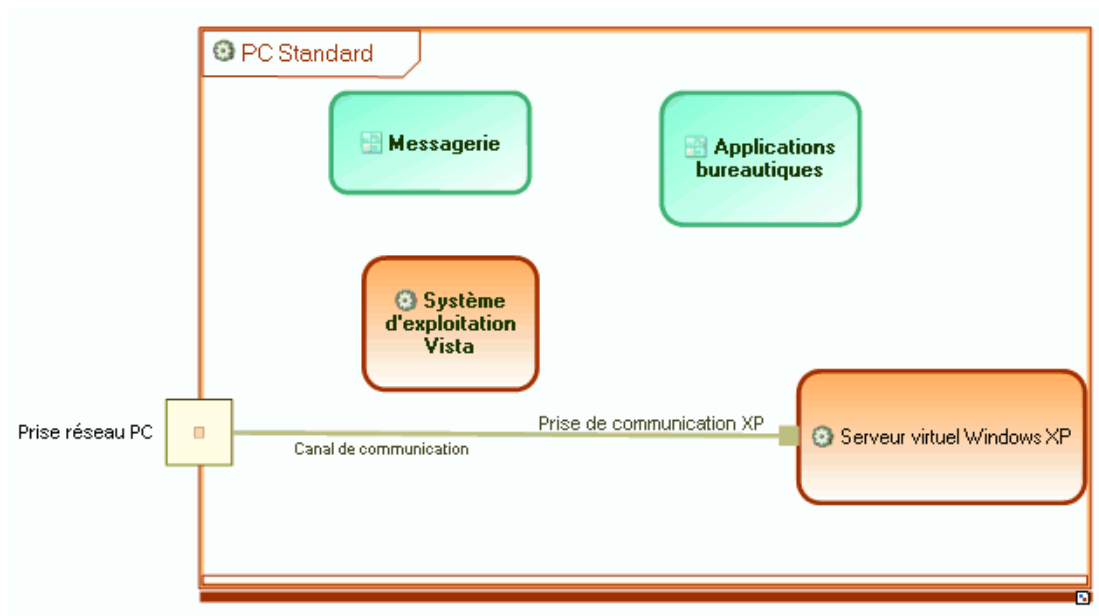


Diagramme d'assemblage de PC standard

Un PC standard est équipé d'un système d'exploitation Vista sur lequel tournent des applications bureautiques et de messagerie.

La migration vers Windows Vista pose un problème de compatibilité des applications Web avec le navigateur. Ainsi un PC standard dispose-t-il d'une connexion à un

serveur virtuel sous Windows XP et où fonctionnent nativement Internet Explorer 6 et ses applications Web.

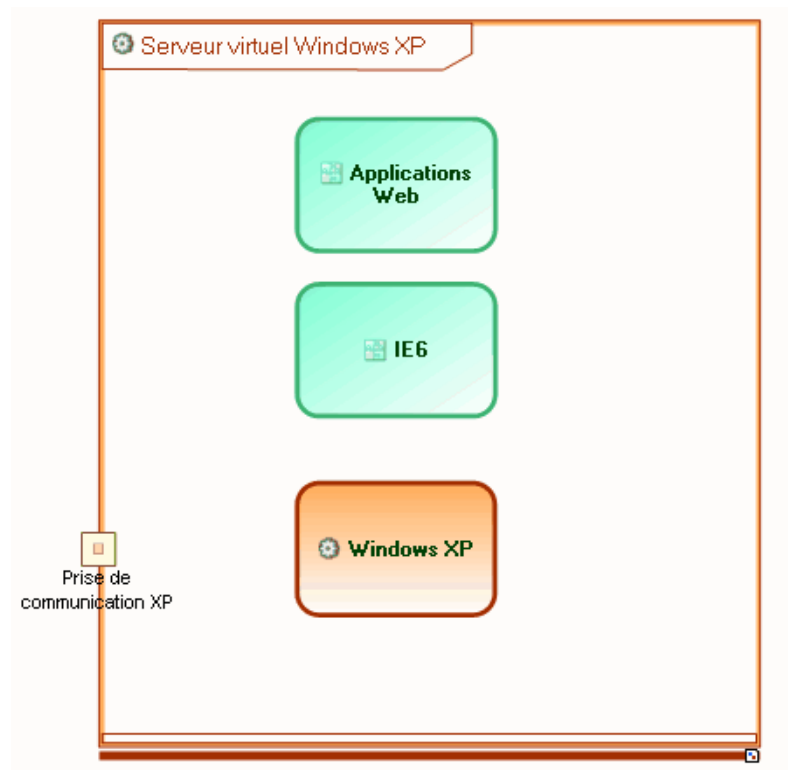


Diagramme d'assemblage du Serveur virtuel Windows XP (Centres d'appels)

Ces diagrammes présentant une architecture technique contiennent des composants de type *Hébergement d'application* et *hébergement de base de données* qui représentent les applications et les serveurs installés sur un PC standard.

 Un hébergement d'application représente la mise en œuvre d'une application dans un objet manufacturé.

 Un hébergement de base de données représente l'hébergement d'une base de données dans un objet manufacturé.

Les postes opérateurs et les postes support sont des variantes du poste standard. Tous les postes opérateurs et support ont ainsi la même architecture de base composée :

- d'un système d'exploitation, Windows Vista,
- d'une application de messagerie, Outlook,
- d'une suite d'applications bureautiques, la suite office,
- d'une prise réseau PC.

Un poste opérateur est basé sur cette architecture et héberge en plus des applications spécifiques pour la gestion de la CMDB, des appels et des incidents.

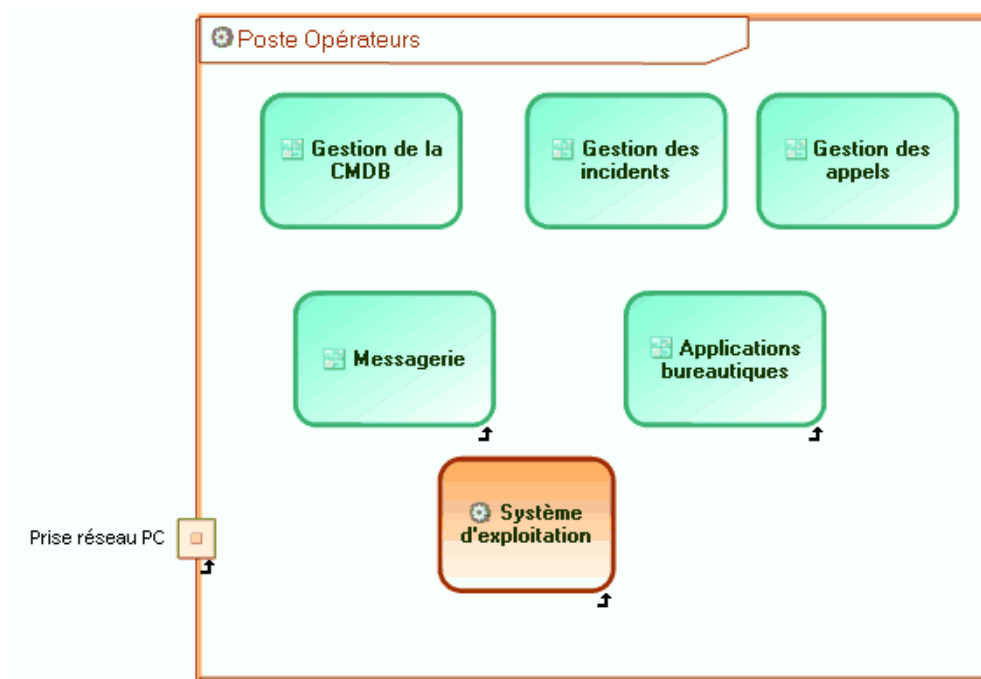


Diagramme d'assemblage d'un poste opérateur

Les objets manufacturés "Postes opérateurs" et "Poste support" étant définis à partir de l'objet manufacturé "PC Standard", si l'application de messagerie est remplacée par une autre dans "PC standard", elle est automatiquement modifiée dans les postes opérateurs et support.

➡ Pour plus de détails sur les variations, voir le guide **HOPEX Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".

DÉCRIRE UN ENVIRONNEMENT D'ARCHITECTURE DE RESSOURCES



Un environnement d'architecture de ressources présente un contexte d'utilisation d'architectures de ressources. Il décrit les interactions, entre l'architecture de ressource et ses partenaires externes, qui lui permettent de remplir sa mission et d'assurer les fonctionnalités attendues.

Une **architecture de ressources** est composée de l'ensemble des ressources matérielles et organisationnelles nécessaires au fonctionnement d'une infrastructure complexe. Les communications entre ces composants sont représentées par des interactions et les moyens matériels, supportant ces interactions, sont des canaux de communication.



Une architecture de ressources est la combinaison d'équipements matériels et d'organisation configuré 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.

Créer un environnement d'architecture de ressources

Pour créer un environnement d'architecture de ressources à partir d'une bibliothèque :

1. Vérifiez dans les options de **HOPEX**, dossier **Modélisation des processus et de l'architecture**, que l'option **Modélisation d'infrastructure Post 2009 SP2** est sélectionnée.
2. Faites un clic droit sur la bibliothèque qui vous intéresse et sélectionnez **Nouveau > Brique de modélisation**.
La fenêtre **Choix de la MetaClasse** apparaît.
3. Sélectionnez **Environnement d'architecture de ressources**.
La fenêtre **Création d'un environnement d'architecture de ressources** apparaît.
4. Saisissez le **Nom** de votre environnement et cliquez sur **OK**.
Le nouvel environnement d'architecture de ressources apparaît dans l'arbre de navigation.

La procédure qui permet de créer un diagramme d'environnement d'architecture de ressource varie légèrement suivant que vous êtes en Windows Front-End ou Web Front-End.

HOPEX Web Front-End

Pour créer un diagramme d'environnement d'architecture de ressources :

- 1. Faites un clic droit sur l'environnement d'architecture de ressources et sélectionnez **Nouveau > Diagramme > Diagramme d'environnement d'architecture de ressources**.
Le diagramme s'ouvre.

HOPEX Windows Front-End

Pour créer un diagramme d'architecture de ressources :

1. Faites un clic droit sur l'architecture de ressources et sélectionnez **Nouveau > Diagramme**.
Une fenêtre s'ouvre.
2. Sélectionnez le type de diagramme "Diagramme d'environnement d'architecture de ressources".
3. Conservez l'option **Initialiser le diagramme**.
4. Cliquez sur **Créer**.

Créer une architecture de ressources

Pour créer une architecture de ressources à partir d'un environnement d'architecture de ressources :

- 1. Faites un clic droit sur l'environnement d'architecture de ressources qui vous intéresse et sélectionnez **Nouveau > Architecture de ressources**.

²La procédure qui permet de créer un diagramme d'architecture de ressource varie légèrement suivant que vous êtes en Windows Front-End ou Web Front-End.

HOPEX Web Front-End

Pour créer un diagramme d'architecture de ressources :

- 1. Faites un clic droit sur l'architecture de ressources et sélectionnez **Nouveau > Diagramme > Diagramme d'architecture de ressources**.
Le diagramme s'ouvre.

HOPEX Windows Front-End

Pour créer un diagramme d'architecture de ressources :

1. Faites un clic droit sur l'architecture de ressources et sélectionnez **Nouveau > Diagramme**.
Une fenêtre s'ouvre.
2. Sélectionnez le type de diagramme "Diagramme d'architecture de ressources".
3. Conservez l'option **Initialiser le diagramme**.
4. Cliquez sur **Créer**.

Mettre en oeuvre une capacité

La création d'une architecture de ressources peut répondre au besoin d'une capacité.

Une capacité définit une aptitude escomptée, par exemple "Fournir un service support de niveau 3".

Vous pouvez relier une architecture de ressources à une capacité afin de décrire l'ensemble des moyens physiques et organisationnels mis en place pour fournir ce service.

Pour indiquer qu'une architecture de ressources est reliée à une capacité :

1. Ouvrez la fenêtre de propriétés de l'architecture de ressources.
2. Cliquez sur l'onglet **Caractéristiques**.
3. Dans le cadre **Capacité configurée**, cliquez sur le bouton **Relier**.
 Vous pouvez également créer une capacité depuis cette fenêtre.
4. Sélectionnez la capacité en question et cliquez sur **OK**.
Le nom de la capacité apparaît dans le cadre.

Décrire les ressources organisationnelles de l'architecture de ressources

Créer une Utilisation d'architecture

Pour décrire qu'une architecture, telle qu'un centre de support, met en oeuvre une autre architecture, telle qu'un service dépannage, vous allez créer dans l'architecture "Centre de support" un composant de type **Utilisation d'architecture** et lui associer l'architecture mise en oeuvre ("Service dépannage").

 Une utilisation d'architecture est la mise en place d'une architecture de ressources dans une autre architecture de ressources ou dans un environnement d'architecture de ressources.

Pour créer une **Utilisation d'architecture** :

1. Dans la barre d'objets du diagramme d'architecture de ressources, cliquez sur **Utilisation d'architecture** .
2. Cliquez dans le cadre de l'architecture décrite.
Une fenêtre de création vous demande de choisir la **Ressource déployée**.
Il s'agit de l'architecture de ressources mise en oeuvre (par exemple "Service dépannage"). Vous pouvez sélectionner une architecture de ressource 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, vous pouvez différer la création de cette ressource. Si vous cliquez sur **OK**, le composant **Utilisation d'architecture** est créé sans ressource.
3. Cliquez sur **Terminer**.

Créer une ressource humaine

Pour décrire qu'une architecture, telle qu'un centre d'appels, s'appuie sur des opérateurs pour prendre les appels et traiter les demandes, vous allez créer un composant de type **Ressource humaine** et lui associer l'acteur "Opérateur".

 Une ressource humaine est une ressource organisationnelle configurée pour supporter certaines capacités dans le cadre d'une architecture de ressources.

Pour créer une ressource humaine :

1. Dans la barre d'objets du diagramme d'architecture de ressources, cliquez sur **Ressource humaine** .
2. Cliquez dans le cadre du diagramme d'architecture de ressources. Une fenêtre de création vous demande de choisir la ressource à déployer.
3. Sélectionnez l'acteur que vous souhaitez affecter à la ressource et cliquez sur **Terminer**.

 Il est possible de créer une Ressource humaine sans qu'aucun acteur ne lui soit attaché. Si l'Acteur n'existe pas encore dans le référentiel, ou s'il n'est pas encore identifié, vous pouvez différer l'affectation de l'Acteur à la ressource.

Décrire les ressources matérielles

Créer un équipement

Pour décrire qu'une architecture, telle qu'un centre de support logiciel, s'appuie sur des ressources matérielles telles que des postes de travail hébergeant des applications, vous allez créer sur l'architecture des composants de type **Équipement** et leur associer les postes de travail souhaités.

 Un équipement représente la mise en œuvre d'un objet manufacturé dans une architecture de ressources.

Pour créer un équipement :

1. Dans la barre d'objets du diagramme, cliquez sur **Équipement** .
2. Cliquez dans le cadre du diagramme. Une fenêtre de création vous demande de choisir la ressource à déployer.
3. Sélectionnez l'objet manufacturé que vous souhaitez affecter à l'équipement (ici les postes opérateurs) et cliquez sur **OK**.

 Il est possible de créer un Équipement sans qu'aucun objet manufacturé ne lui soit attaché. Si l'Objet manufacturé n'existe pas encore dans le référentiel, ou s'il n'est pas encore identifié, vous pouvez différer l'affectation de la ressource à l'équipement.

 Vous pouvez également créer un équipement en sélectionnant un objet manufacturé dans la fenêtre de navigation de **HOPEX** et en le glissant dans le diagramme avec la souris.

Les canaux et les ports de communication

Dans une architecture de ressources, les Canaux de communication supportent le transfert des informations d'un équipement à un autre. Pour plus de détails sur la création de ces canaux et des protocoles de communication associés, voir "[Les](#)

[canaux de communication", page 82.](#)

Les Ports de communication permettent de connecter les équipements de l'architecture de ressources avec des éléments matériels extérieurs. Pour plus de détails sur la manière de connecter les ports de communication entre différents composants, voir ["Relier les points d'élément de communication", page 83.](#)

Décrire les services et les requêtes dans une architecture de ressources

Une architecture de ressources est créée pour assurer un ou plusieurs services.

☞ Pour plus de détails, voir ["Les points de service", page 78](#) et ["Les points de requête", page 79.](#)

Décrire les interactions dans un diagramme d'architecture de ressources

Dans un diagramme d'architecture de ressources, les *interactions* permettent de représenter les échanges prévus entre les entités organisationnelles.

📖 Une interaction 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 un contrat d'échange.

Les modalités d'échange sont définies par un *contrat d'échange* qui est affecté à l'interaction.

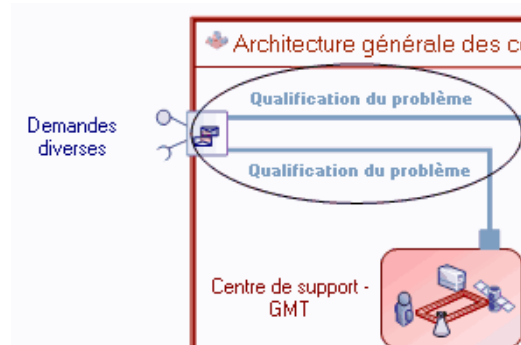
📖 Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

Vous pouvez définir des interactions entre :

- deux composants de type "Utilisation des ressources" pour représenter les échanges prévus entre ces entités,
- un composant de type "Utilisation des ressources" et un équipement pour représenter les modalités d'utilisation de la ressource matérielle par la ressource organisationnelle. Par exemple, vous pouvez représenter que l'utilisation de la plate-forme de tests logiciels est accordée sur réservation.
- deux composants de type équipement pour représenter les modalités d'utilisation d'une ressource par une autre dans le contexte de l'architecture de ressources modélisée. Par exemple, les modalités

d'utilisation d'un matériel de remplacement (spare) dépendent du niveau de priorité de l'organisation qui le demande.

- un point de service et un ou plusieurs composants de type "Utilisation de ressources" pour représenter la mise en oeuvre du service au sein de l'architecture de ressources,
- un composants de type "Utilisation de ressources" et un point de requête pour représenter que l'entité fait appel à une ressource organisationnelle extérieure.



Dans l'exemple du système général du support, deux interactions sont utilisées pour indiquer que les services de support sont pris en charge soit par un centre soit par un autre en fonction du créneau horaire. Les deux interactions sont exécutées selon le même contrat d'échange.

Pour plus de détails sur les modalités de gestion des interactions, voir ["Gérer les interactions", page 80](#).

DÉCRIRE UN OBJET MANUFACTURÉ

Un objet manufacturé est tout type d'élément physique en dehors du domaine applicatif ou organisationnel (organisationnel incluant les personnes). Un objet manufacturé peut représenter un système matériel, un sous-système, une plateforme, un composant ou simplement un élément physique ayant des caractéristiques spécifiques.

Vous pouvez décrire les composants d'un *objet manufacturé* dans un diagramme d'assemblage d'objets manufacturés.

Vous pouvez insérer dans ce diagramme :

- des composants manufacturés,
- des hébergements d'application,
- des ports et des canaux de communication,
- des points de service et des points de requête,
- des interactions.

Créer un objet manufacturé

Pour créer un objet manufacturé à partir d'une bibliothèque :

1. Vérifiez dans les options de **HOPEX**, dossier **Modélisation des processus et de l'architecture**, que l'option **Modélisation d'infrastructure Post 2009 SP2** est sélectionnée.
2. Faites un clic droit sur une bibliothèque et sélectionnez **Nouveau > Objet manufacturé**.

Pour créer le diagramme d'un objet manufacturé **HOPEX Windows Front-End** :

1. Faites un clic droit sur l'objet manufacturé et sélectionnez **Nouveau > Diagramme**.
2. Dans la fenêtre qui apparaît, sélectionnez **Diagramme d'assemblage d'objets facturés** et cliquez sur **Créer**.

Le diagramme s'ouvre.

☛ Pour créer un diagramme d'objet manufacturé avec **HOPEX Web Front-End**, faites un clic droit sur l'objet manufacturé et sélectionnez **Nouveau > Diagramme > Diagramme d'assemblage d'objets facturés**.

Décrire les composants d'un objet manufacturé

Créer un composant manufacturé

Pour décrire qu'un objet manufacturé, tel que l'architecture matérielle d'un opérateur, met en œuvre un autre objet manufacturé, tel qu'un poste opérateur,

vous allez créer dans le cadre de l'architecture matérielle un **composant manufacturé** et lui associer un poste opérateur.



Un composant manufacturé représente la mise en place d'un produit manufacturé au sein d'un autre.

Pour créer un composant manufacturé :

1. Dans la barre d'objets du diagramme d'assemblage d'objets manufacturés, cliquez sur **Composant manufacturé** .
2. Cliquez dans le cadre de l'objet décrit.
Une fenêtre de création vous demande de choisir l'**Objet manufacturé utilisé**, soit à partir des objets manufacturés existants, soit à partir d'un nouvel objet manufacturé.

Exemple : sélectionnez "Poste opérateur".



Dans le cas où l'objet manufacturé que vous souhaitez utiliser n'existe pas encore dans le référentiel, vous pouvez différer sa création. Vous pouvez aussi le modifier ultérieurement.

3. Cliquez sur **OK**.

Créer un hébergement d'application

Pour décrire qu'un objet manufacturé, tel qu'un poste de travail standard, est un PC standard hébergeant des applications de base telles qu'une application de messagerie, vous allez créer un composant de type Hébergement d'application et lui associer l'application voulue.

Pour créer un hébergement d'application :

1. Dans la barre d'objets du diagramme d'assemblage d'objets manufacturés, cliquez sur **Hébergement d'application** .
2. Cliquez dans le cadre du diagramme.
Une fenêtre de création vous demande de choisir l'**Application hébergée**.
3. Sélectionnez l'application (dans notre exemple l'application de messagerie) et cliquez sur **OK**.

Les canaux et les ports de communication

Les Canaux de communication supportent le transfert des informations d'une ressource matérielle à une autre. Pour plus de détails sur la création de ces canaux et des protocoles de communication associés, voir ["Les canaux de communication", page 82](#).

Les Ports de communication permettent de connecter les ressources matérielles de l'objet manufacturé avec des éléments matériels extérieurs. Pour plus de détails sur la manière de connecter les ports de communication entre différents composants, voir ["Relier les points d'élément de communication", page 83](#).

Décrire les services et les requêtes d'un objet manufacturé

Une architecture matérielle est créée pour assurer un ou plusieurs services. Par exemple, la plate-forme de tests logiciels est utilisée pour comprendre des dysfonctionnements et pour faire des tests d'installation.

☛ Pour plus de détails, voir ["Les points de service", page 78](#) et ["Les points de requête", page 79](#).

Décrire les interactions dans un diagramme d'assemblage d'objets manufacturés

Dans un diagramme d'assemblage d'objets manufacturés, les *interactions* permettent de représenter les échanges prévus entre les ressources matérielles.

📖 Une interaction 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 un contrat d'échange.

☛ Pour plus de détails sur les interactions, voir ["Gérer les interactions", page 80](#).

Vous pouvez définir des interactions entre :

- deux ressources matérielles pour représenter les modalités d'utilisation d'une ressource par une autre dans le contexte de l'architecture modélisée. Par exemple, une application métier peut s'appuyer sur des services rendus par l'application de messagerie.
- un point de service et un ou plusieurs composants manufacturés pour représenter la mise en œuvre du service au sein de l'architecture matérielle. Par exemple, plusieurs applications logicielles couvrent le service attendu.
- un composant de type composant manufacturé et un point de requête pour représenter qu'une ressource matérielle fait appel à un service extérieur.

DÉCRIRE LES COMMUNICATIONS D'UNE INFRASTRUCTURE COMPLEXE

Dans une infrastructure complexe, les communications s'appuient sur :

- des points de service et des points de requête,
- des interactions,
- des ports et des canaux de communication.

HOPEX IT Architecture offre des facilités cohérentes et homogènes pour gérer chacun de ces composants et vérifier la pertinence des informations spécifiées.

Décrire les services et les requêtes

Les points de service

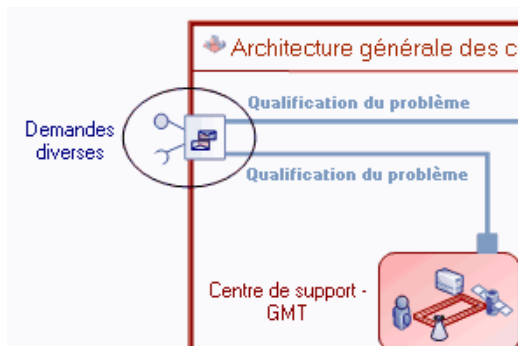
Les services assurés par les éléments d'une architecture de ressource sont représentés par des *points de service* 

 *Un point de service est un point à partir duquel le système reçoit une demande de service d'un autre système et fournit le service demandé.*

Le service est demandé selon des modalités précises définies par un *contrat d'échange* affecté au point de service.

 *Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.*

Les ressources activées pour assurer un service sont liées au point de service par des interactions. Si l'activation de plusieurs ressources est nécessaire, alors plusieurs interactions doivent être créées entre le point de service et les ressources de l'architecture.



Dans l'exemple présenté ici, le service de support est lié à deux interactions, basées sur le même contrat, qui

représentent l'activation d'un centre de support ou de l'autre en fonction du créneau horaire.

Pour créer un point de service, voir "[Créer un point de service ou un point de requête](#)", page 79.

Les points de requête

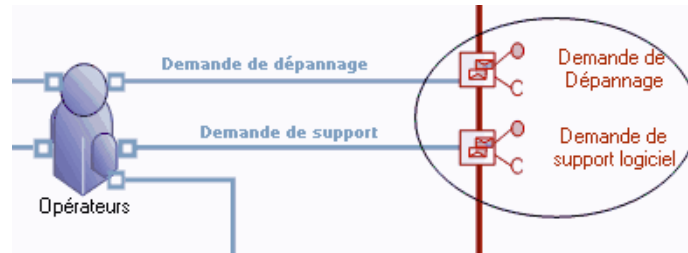
Un *point de requête*  permet de représenter l'utilisation d'un service externe.

 *Un point de requête est un point à partir duquel le système envoie une requête à un autre système et reçoit le service demandé.*

Le service est demandé selon des modalités précises définies par un *contrat d'échange* affecté au point de requête.

 *Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.*

Les ressources qui émettent une requête sont liées au point de requête par une interaction.



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 "[Créer un point de service ou un point de requête](#)", page 79.

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 service est un point à partir duquel le système reçoit une demande de service d'un autre système et fournit le service demandé.*

 *Un point de requête est un point à partir duquel le système envoie une requête à un autre système et reçoit le service demandé.*

Pour créer un point de service :

1. Dans la barre d'objets du diagramme d'architecture de ressources, cliquez sur **Point de service** .

2. Positionnez l'objet sur le bord du cadre de l'architecture.
Une fenêtre de création s'ouvre.
3. Cliquez sur **Choix du service Rendu** pour définir le protocole qui permet l'activation de ce point de service.
La fenêtre de recherche s'ouvre.
4. Sélectionnez le protocole qui permet de relier ce point de service à l'extérieur.
Le rôle se met automatiquement à jour.
5. Cliquez sur **OK**.

Gérer les interactions

Une *interaction* représente l'échange d'informations entre des entités organisationnelles du système.

 Une interaction 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 un contrat d'échange.

Le contenu d'une interaction est décrit par un *contrat d'échange*.

 Un contrat d'échange est un modèle de contrat entre des entités organisationnelles. Ce contrat est décrit par les échanges effectués entre un rôle initiateur et un ou plusieurs rôles contributeurs.

 Pour plus de détails, voir le chapitre "[Décrire les interactions](#)", page 51.

Dans le contexte des infrastructures complexes, les interactions sont supportées physiquement par des canaux de communication. Voir "[Les canaux de communication](#)", page 82.

Créer une interaction

Pour créer une interaction :

1. Dans la barre d'objets du diagramme d'architecture de ressources, cliquez sur **Interaction** .
2. Tirez un lien entre les deux entités en communication.
3. Dans la fenêtre d'ajout d'interaction, précisez le protocole que vous souhaitez utiliser.
4. Cliquez sur **OK**.

Relier les points d'interaction

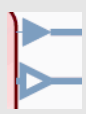
Le point d'interaction relie une interaction à l'un des composants en communication. Il permet de spécifier :

- le point de service, ou le point de requête, qui intervient dans la communication
- le rôle, consommateur ou fournisseur, joué par le point d'interaction dans le protocole d'échange.

Pour relier le point d'interaction :

- 1. Faites un clic droit sur l'interaction à proximité de l'entité en communication et sélectionnez **Relier > Point d'interaction**.
Une fenêtre s'ouvre, elle présente la liste des points de service et de requête du composant en communication.
- 2. Sélectionnez un point de service ou un point de requête.
- 3. Cliquez sur **OK**.

La représentation graphique du point d'interaction donne des indications sur la nature et la cohérence des informations apportées.

	Le point d'interaction n'est pas défini
	Un point d'interaction a été défini, mais aucune information n'a été donnée sur le rôle joué par le point d'interaction dans le protocole
	Le point d'interaction, défini ou non, joue le rôle du consommateur dans le protocole
	Le point d'interaction, défini ou non, joue le rôle du fournisseur dans le protocole
	Il y a une erreur de modélisation : les deux points d'interaction jouent le rôle du fournisseur dans le protocole
	Il y a une erreur de modélisation : les deux points d'interaction jouent le rôle du consommateur dans le protocole
	La modélisation des points d'interaction est correcte

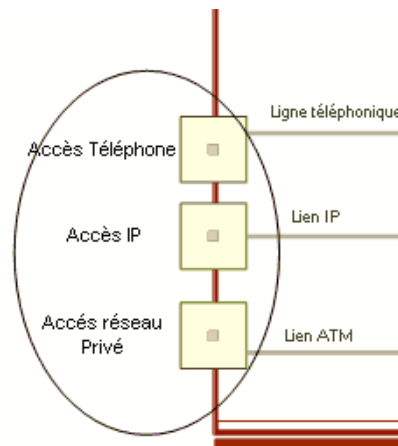
Décrire les communications au niveau matériel

Les ports de communication

Les *ports de communication*  sont des points physiques de communication qui peuvent être définis dans les objets manufacturés et dans les architectures de ressources.

 *Un port de communication est un point physique de communication avec une ressource. Ils se conforment à des protocoles de communication particuliers. Un port de communication implémente des points de service et de requête.*

Ils assurent le transfert physique des informations échangées sur les 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](#)", page 83.

Les canaux de communication

Les **canaux de communication** permettent de connecter les ressources matérielles entre elles, ou à des ressources organisationnelles, ou encore à des ports de communication.



Un canal de communication permet d'établir une connexion physique entre deux ressources matérielles. Il supporte les interactions qui définissent les protocoles de communication entre ces ressources. Les canaux de communication connectent les ressources avec l'extérieur par le biais des ports de communication.

Créer un canal de communication

Pour créer un canal de communication :

1. Dans la barre d'objets du diagramme d'architecture de ressources, cliquez sur **Canal de communication** .
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. Faites un clic droit sur le canal et sélectionnez **Relier > Protocole de communication**.
2. Dans la fenêtre de recherche qui apparaît, double cliquez sur le protocole de communication. Le nom du protocole apparaît à côté du nom du canal.

Les protocoles de communication réseau

Un *protocole de communication* est supporté par un canal de communication.

 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, un protocole SOAP pourra être basé sur un protocole HTTP SMTP ou FTP pour le transport, eux-mêmes basés sur TCP qui est lui-même basé sur Ethernet.

Un utilisateur peut être amené à construire sa propre couche de protocoles communication et à les assigner aux ports et canaux de communication.

☛ Les protocoles de communication supportés par un port de communication doivent être compatibles avec les ports de communication auxquels il est connecté.

Relier les points d'élément de communication

Comme les interactions, les canaux de communication sont représentés par un lien dont les extrémités ont une signification graphique. Par exemple, lorsque l'extrémité du canal affiche un carré vide, cela indique que le point d'élément de communication (situé à l'extrémité du canal) n'est connecté à aucun port de l'équipement relié.



Pour relier un point d'élément de communication au port d'un équipement, ou d'une architecture de ressource :

1. Faites un clic droit sur le point d'élément de communication, à l'extrémité du canal de communication, et sélectionnez **Relier > Port de communication**
Une fenêtre présentant la liste des ports de communication de l'objet s'ouvre
2. Sélectionnez un port de communication
3. Cliquez sur **OK**.

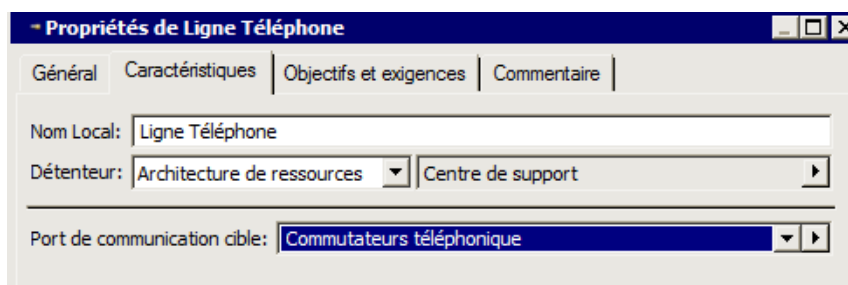
Exemple

Par exemple, dans le cas du centre de support, le canal de communication "Ligne téléphonique" relie le port de

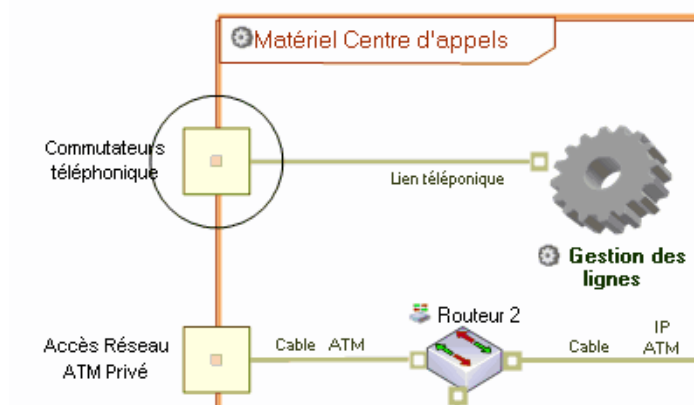
communication "Accès téléphone" à l'Infrastructure générale du centre de support.



Depuis la fenêtre de propriétés du point d'élément de communication, vous pouvez voir que la "Ligne téléphonique" est connectée à l'objet manufacturé via le port de communication "Commutateur téléphonique".



Le "Commutateur téléphonique" est un des ports disponibles dans l'objet manufacturé.



SYNTHÈSE DES TERMES UTILISÉS

Les diagrammes et leurs objets

Le tableau ci-dessous constitue une synthèse des termes utilisés pour les différents éléments d'une infrastructure complexe, en fonction du contexte dans lequel ce composant est mis en oeuvre..

Diagramme	Élément décrit	Objet associé	Commentaire
Diagramme d'une architecture de ressources	Équipement	Objet manufacturé	
Diagramme d'une architecture de ressources	Ressource humaine	Acteur	
Diagramme d'une architecture de ressources	Utilisation d'architecture	Architecture de ressources	Utilisation d'une architecture par une autre
Diagramme d'assemblage d'objets manufacturé	Composant manufacturé	Objet manufacturé	Utilisation d'un objet manufacturé dans un autre
Diagramme d'assemblage d'objets manufacturé	Hébergement d'application	Application	L'application est déployée avec l'objet

Les infrastructures complexes sont basées sur deux types de concepts :

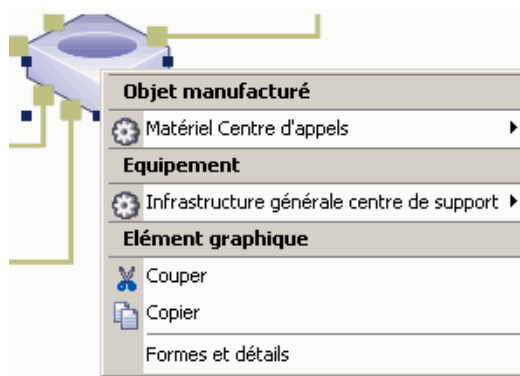
- Les éléments décrits dans le diagramme, matériels ou organisationnels, qui sont détenus par une architecture, tels que : Équipement, Utilisation d'architecture, Ressource humaine ou Hébergement d'application. Les objets associés à ces éléments peuvent être modifiés sans conséquence sur les éléments eux-mêmes ou leur environnement.
- Les objets associés à ces éléments. Il s'agit des Architectures de ressources, des Objets manufacturés, des Acteurs ou des Applications.

Dans l'exemple de l'architecture générale du centre de support, le Centre de support GMT et le Centre de support GMT+12 sont des utilisations d'architecture. Le "Centre de Support" décrit l'architecture de ressources générique mise en oeuvre dans les deux centres. Cette architecture se compose elle-même de différentes ressources : équipements et utilisation d'architecture.

Le menu contextuel des objets

Dans un diagramme d'architecture de ressources, le menu contextuel d'un élément d'architecture (par exemple un Equipement) présente les commandes propres au type d'objet déployé pour l'élément (Objet manufacturé), suivies des commandes relatives à l'équipement lui-même. Le troisième niveau du menu contextuel concerne le graphisme.

☛ Si le type d'élément associé à la ressource n'est pas défini, la première partie du menu n'apparaît pas.



Propriétés des éléments décrits

Pour accéder à l'objet qui définit un élément d'une architecture de ressources :

1. Ouvrez la fenêtre de propriétés de l'élément (par exemple une utilisation d'architecture) et sélectionnez l'onglet **Caractéristiques**.
L'objet qui définit l'élément (ici, une architecture de ressources) apparaît dans le champ **Ressource Déployée**.
2. Cliquez sur la flèche à droite du champ **Ressource Déployée**, pour accéder au menu contextuel de la ressource déployée ou la remplacer sans changer l'élément courant.

Propriétés des objets associés

Vous pouvez spécifier des propriétés sur un objet manufacturé, une architecture de ressources, un acteur ou une application.

Vous pouvez attribuer par exemple un mode de transmission aux téléphones d'un centre d'appel, un numéro de série ou toute autre caractéristique particulière.

Pour définir une propriété :

1. Ouvrez la fenêtre de propriétés de l'objet et cliquez sur l'onglet **Propriétés**.

Exemple : Objet manufacturé "Téléphone".

2. Cliquez sur **Nouveau**  .
La fenêtre de création d'une valeur de propriété apparaît.

3. Renseignez le nom de la propriété, son type et sa valeur.

Responsables		Propriétés		Points de service et de requête		Risques		Utilisation	
+ ↓ ✓ ✗ ↔ 📄 📄 📄									
Nom Local		Type de propriété		Valeur de propriété		Unité			
Mode de transmission		Mode de transmission		DECT					

Une fois un type de propriété créé, vous pouvez lui attribuer différentes valeurs sur différents objets. Ainsi, selon les objets, le "Mode de transmission" pourra prendre les valeurs "DECT", "Wi-Fi", etc.

Pour spécifier une nouvelle valeur :

- 1 Lors de l'ajout d'une propriété sur un objet, recherchez à l'aide de la flèche le **Type de propriété** en question (ici, Mode de transmission) et définissez une nouvelle valeur pour l'objet auquel s'applique la propriété (par exemple Wi-Fi).

Héritage des propriétés

Lorsqu'un objet hérite des propriétés d'un autre, l'héritage porte sur la valeur de la propriété et non du type. Ainsi, si vous avez défini sur un objet une propriété de type "Mode de transmission" ayant pour valeur "DECT", les objets héritant de cet objet auront tous le mode de transmission "DECT".

Pour modifier la valeur d'une propriété héritée, vous devez créer une nouvelle propriété du même type avec une nouvelle valeur (par exemple, le mode de transmission "Wi-fi") et remplacer la propriété héritée par la nouvelle.

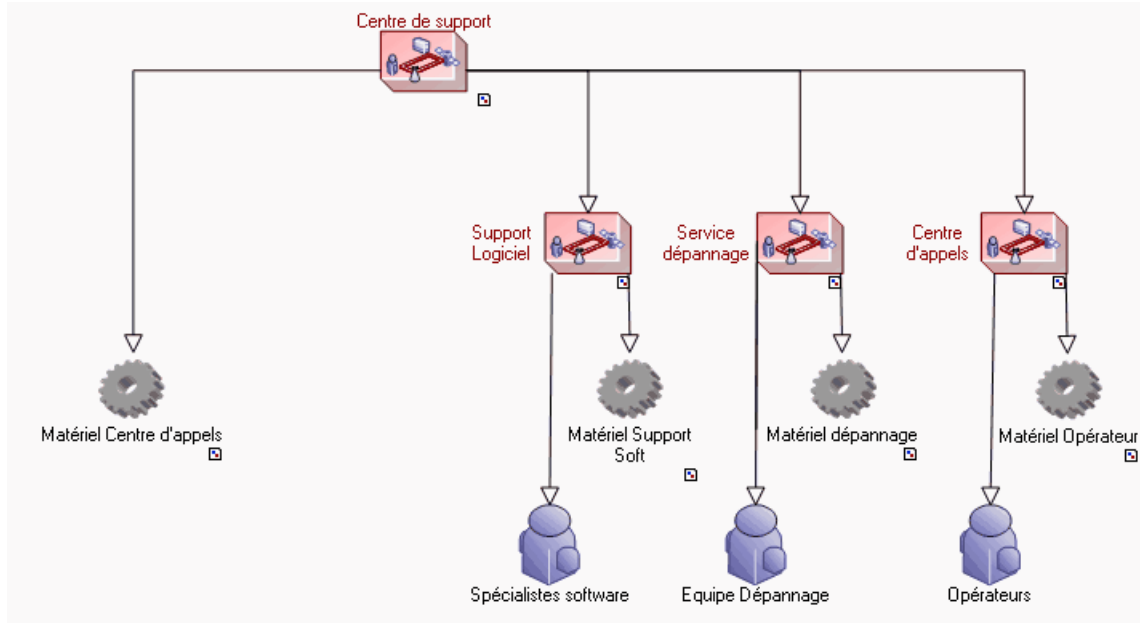
☛ Pour plus de détails sur les variations, voir le guide **HOPEX Common Features**, chapitre "Manipuler les objets du référentiel", "Les variations d'objets".

L'arbre d'architecture de ressources

L'arbre d'architecture de ressources permet de représenter les types de ressources utilisées dans une architecture de ressources.

Les ressources représentées sont :

- des ressources humaines
- des architectures de ressources
- des objets manufacturés



Dans l'exemple du centre de support, le diagramme d'arbre d'architecture de ressources présente le matériel du centre et les architectures de ressources qui permettent de traiter les demandes. Chacune de ces architectures est elle-même composée d'un acteur et d'une architecture matérielle.

LE DÉPLOIEMENT D'APPLICATIONS



HOPEX permet d'associer à une application un ou plusieurs déploiements. Un déploiement est supporté par un site. Sur un même site, une application est déployée pour offrir différents services à différents utilisateurs.

Par ailleurs, la problématique du déploiement d'application consiste aussi à déterminer la conformité entre l'infrastructure requise pour déployer une application et l'infrastructure existante. L'outil de modélisation doit aider à répondre aux questions suivantes :

- Quelles sont les infrastructures nécessaires pour déployer cette application ?
- L'infrastructure existante permet-elle de déployer cette application ?
- Quels sont les éléments manquants ?

Pour répondre à ces questions, deux modèles sont nécessaires : la description de l'infrastructure requise d'une part et la description de l'infrastructure effective d'autre part. Mettre en correspondance ces deux infrastructures aide à déterminer le niveau de conformité.

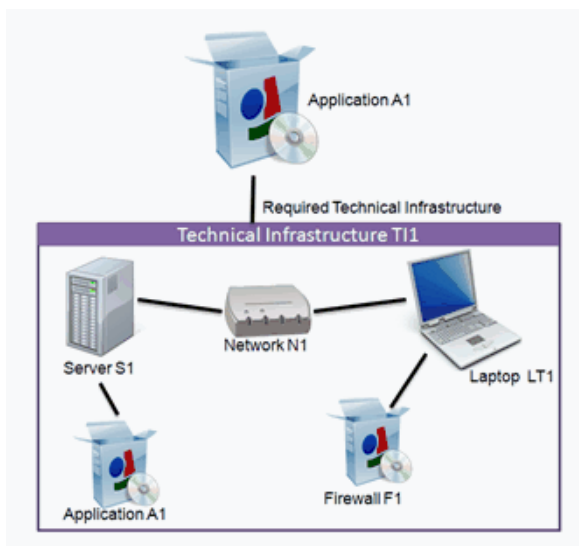
Les points traités ici sont :

- ✓ ["Définir l'architecture requise", page 90](#)
- ✓ ["Déployer l'infrastructure requise", page 91](#)
- ✓ ["Analyser le déploiement d'applications", page 95](#)

DÉFINIR L'ARCHITECTURE REQUISE

La figure suivante décrit l'infrastructure technique requise pour une application donnée. Elle montre également où l'application est déployée dans cette infrastructure, par quel serveur elle est hébergée.

Spécifier une infrastructure requise permet donc de répertorier les éléments nécessaires à une application mais peut également servir à positionner l'application au sein de cette architecture.



☛ Pour préciser comment une application doit être déployée, il est également possible d'utiliser le concept d'Architecture de ressources. Pour afficher les objets liés à ce modèle, vous devez cocher l'option *Modélisation d'infrastructure post 2009 SP2* dans les options de modélisation de **HOPEX**.

Pour créer l'infrastructure technique d'une application :

- Faites un clic droit sur l'application et sélectionnez **Nouveau > Infrastructure technique**.
Vous pouvez ensuite créer le diagramme correspondant.

Voir aussi :

- ✓ ["Modéliser les infrastructures techniques", page 44.](#)

DÉPLOYER L'INFRASTRUCTURE REQUISE

Pour qu'une infrastructure existante soit conforme à une infrastructure requise, vous devez mettre en correspondance les éléments des deux infrastructures.

L'éditeur de correspondance offre une cartographie des deux modèles et vous permet de tracer les correspondances entre les éléments de chacun.

Un élément de l'infrastructure requise peut avoir une correspondance dans plusieurs architectures déployées.

Créer un arbre de correspondance

L'arbre de correspondance vous permet de déployer l'infrastructure requise d'une application.

Dans l'exemple suivant, vous allez déployer l'infrastructure de l'application "Bureautique" dans l'agence de Bordeaux.

L'utilisation de l'éditeur de correspondances varie légèrement selon que vous êtes en Windows Front-End ou Web Front-End.

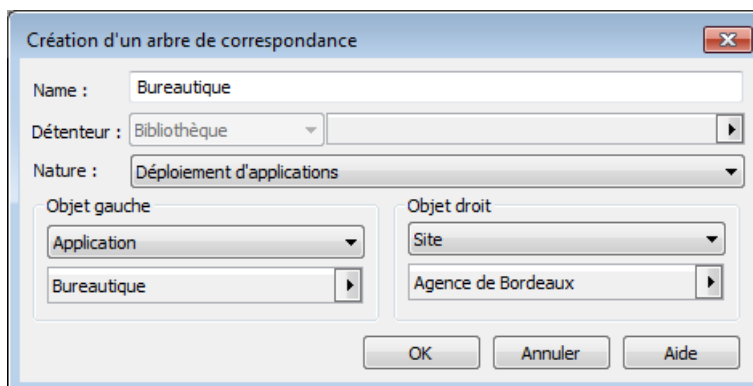
HOPEX Windows Front-End

Pour créer un arbre de correspondance :

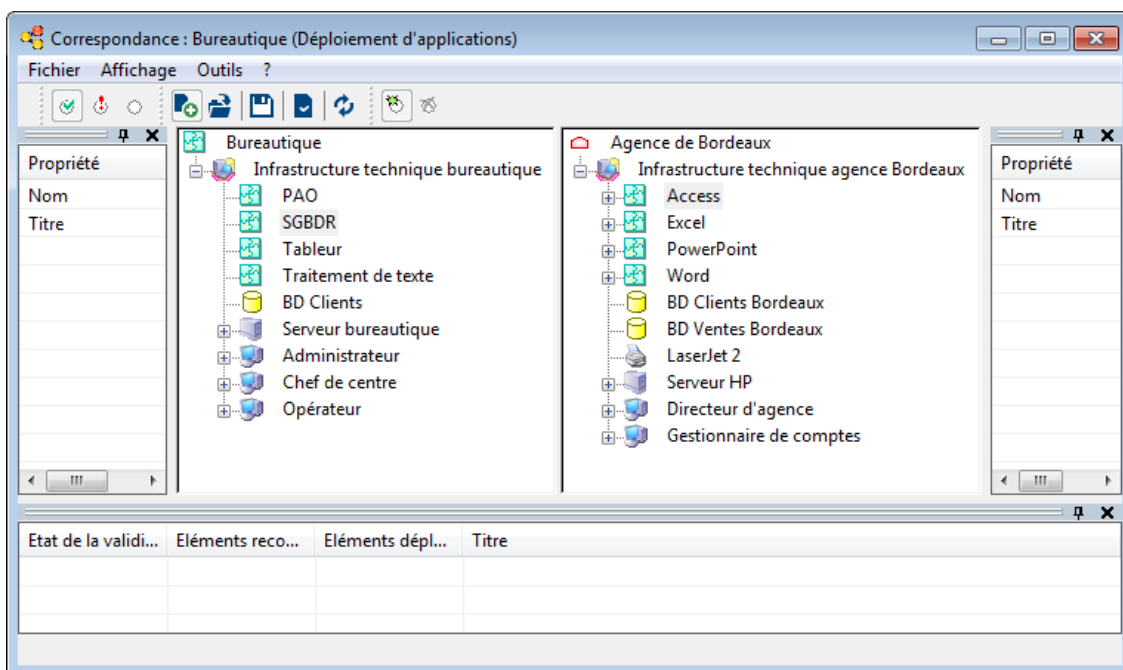
1. Dans la barre de menus **HOPEX**, cliquez sur **Outils > Editeur de correspondance**.
2. Dans la barre de menus de l'éditeur de correspondance, cliquez sur **Fichier > Créer un arbre de correspondance**.
3. Indiquez le nom de l'arbre créé.
4. Dans le champ **Nature**, sélectionnez la nature de l'arbre.
Ex. : "Déploiement d'application".
5. Dans le cadre **Objet gauche**, sélectionnez l'application à déployer.
Ex. : "Bureautique".

6. Dans le cadre **Objet droit**, sélectionnez le site sur lequel l'application va être déployée.

Ex. : "Agence de Bordeaux".



7. Cliquez sur **OK**.
L'arbre de correspondance apparaît.
L'éditeur affiche à gauche l'infrastructure technique requise pour l'application "Bureautique" et à droite celle pour l'"Agence de Bordeaux".



8. Cliquez sur le menu **Outils > Éditeur de correspondances**. Une fenêtre apparaît.
9. Laissez l'option cochée par défaut **Créer un arbre de correspondance**, et cliquez sur **Suivant**.
10. Indiquez le nom du nouvel arbre de correspondance.

11. Dans la champ **Nature**, précisez le type de correspondance effectué. Ici il s'agit de mettre en correspondance des modèles de données.
12. Dans les champs **Objet gauche** et **Objet droit**, à partir des types d'objet concernés (modèle de données), sélectionnez les modèles que vous souhaitez aligner.
13. Cliquez sur **OK**.
L'éditeur affiche l'arbre de correspondance qui juxtapose les deux modèles.

HOPEX Web Front-End

Pour créer un arbre de correspondance :

1. Cliquez sur le menu **Outils > Éditeur de correspondances**.
Une fenêtre apparaît.
2. Laissez l'option cochée par défaut **Créer un arbre de correspondance**, et cliquez sur **Suivant**.
3. Indiquez le nom du nouvel arbre de correspondance.
4. Dans le champ **Nature**, précisez le type de correspondance effectué.
Ex. : "Déploiement d'application".
5. Dans le cadre **Objet gauche**, sélectionnez l'application à déployer.
Ex. : "Bureautique".
6. Dans le cadre **Objet droit**, sélectionnez le site sur lequel l'application va être déployée.
Ex. : "Agence de Bordeaux".
7. Cliquez sur **OK**.
L'éditeur affiche l'arbre de correspondance qui juxtapose les deux modèles.

Établir les correspondances

Pour créer une correspondance entre un objet de l'infrastructure requise -à gauche de l'éditeur- et un objet de l'infrastructure effective :

1. Dans l'éditeur de correspondance, sélectionnez un objet dans l'infrastructure de gauche.
2. Sélectionnez l'objet équivalent dans l'infrastructure de droite.
3. Cliquez sur le bouton **Créer un élément de correspondance**.
 *En Windows Front-End, vous pouvez également créer la correspondance à partir du menu contextuel du dernier objet sélectionné, en utilisant la commande **Établir une correspondance**.*

La correspondance se crée à partir du dernier objet sélectionné.

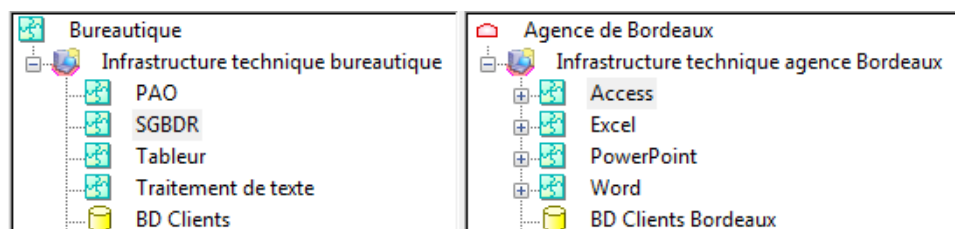
Les objets mis en correspondance sont marqués d'une coche verte.

Si une correspondance ne peut pas être créée, un message d'erreur apparaît.

Exemple

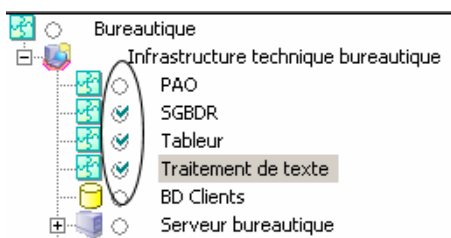
Dans l'exemple ci-dessous, pour indiquer que le SGBDR requis correspond à Access dans l'agence, vous devez

sélectionner le SGBDR à gauche puis celui de l'agence à droite, et créer la correspondance depuis ce dernier.



Etat des objets

Des indicateurs permettent d'indiquer l'état des objets mis en correspondance.



Une barre de filtrage vous permet d'afficher l'ensemble ou uniquement certains de ces indicateurs. Cette barre est disponible sous le menu **Affichage > Barre d'outils** de l'éditeur.

Les objets peuvent être caractérisés par les états suivants :

-  Valide
-  Invalide (Lorsqu'un objet a conservé une correspondance vers un objet qui n'existe plus)
-  Sans correspondance

ANALYSER LE DÉPLOIEMENT D'APPLICATIONS

Des rapports permettent d'évaluer le niveau de conformité d'une architecture existante avec un modèle d'architecture requise.

Afin de vérifier que le déploiement d'une application est conforme à l'infrastructure requise :

1. Faites un clic droit sur l'application à analyser et sélectionnez **Recherche de rapport**.

La liste des rapports disponibles pour l'objet apparaît.

2. Cliquez sur **Analyses de conformité du déploiement d'architectures**.

3. Dans **Conformité du déploiement des architectures**, cliquez sur **Lancer un nouveau rapport**.

Trois rapports sont disponibles :

- Vue générale de la conformité du déploiement des architectures
- Analyse de conformité du déploiement des architectures
- Détails de la conformité du déploiement des architectures niveau 2

Si aucun déploiement n'a encore été réalisé sur l'application sélectionnée, un message vous invite à en créer un. Dans ce cas, vous devez sélectionner l'application à déployer puis l'architecture ou le site ciblés. Voir "[Créer un arbre de correspondance](#)", page 91.

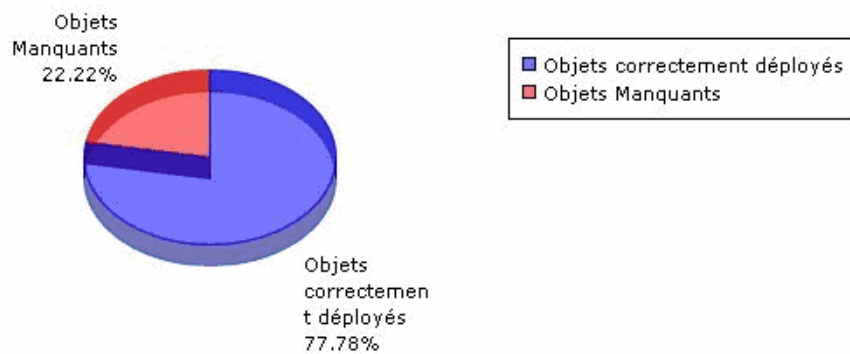
Exemple de rapport d'analyse

Dans notre exemple, nous voulons vérifier que l'application "Bureautique" commune aux différentes agences a été correctement déployée dans l'agence de Bordeaux.

Après lancement de l'analyse de conformité, un premier chapitre de rapport affiche le résultat global du déploiement :

Vue globale de déploiement des architectures

Déploiement d'Architecture



Le dernier chapitre affiche plus de détails ; il donne le pourcentage des objets de l'infrastructure "Bureautique" qui ont été correctement déployés dans l'agence de Bordeaux et précise les objets manquants.

Nombre de Applications correctement déployés : 3/4 (75%).

Recommandé	Deployé
 Tableur	 Excel
 SGBDR	 Access
 PAO	-
 Traitement de texte	 Word

Agence de Bordeaux - Bureautique - Base-de-données Déploiement

Nombre de Base-de-données correctement déployés : 1/1 (100%).

Recommandé	Deployé
 BD Clients	 BD Clients Bordeaux

Agence de Bordeaux - Bureautique - Poste de Travail Déploiement

Nombre de Postes de Travail correctement déployés : 2/3 (66,67%).

Recommandé	Deployé
 Opérateur	-
 Administrateur	 Gestionnaire de comptes
 Chef de centre	 Directeur d'agence

Les indicateurs de conformité

Les indicateurs qui signalent la qualité du déploiement sont :

- le nombre d'objets de l'infrastructure requise correctement déployés.
- le pourcentage des objets correctement déployés.
- le nombre d'éléments manquants.
- le pourcentage des objets manquants.

URBANISATION DU SYSTÈME D'INFORMATION



Les systèmes d'information des entreprises se sont développés dans une grande disparité au fil du temps et au gré des projets. Différents types d'applications se sont accumulés au hasard des fusions/acquisitions ou des centralisations/décentralisations. Il existe généralement dans les grandes entreprises une redondance importante au niveau des données, des services et de leur cas d'utilisation.

Dans ces conditions, il devient de plus en plus difficile d'intégrer de nouvelles applications dans les systèmes d'information existants ou de faire face à des changements technologiques.

La finalité de l'urbanisme des systèmes d'information est de définir les grands principes d'implémentation et de construction des applications informatiques, en assurant une cohérence d'ensemble, pour diminuer les coûts de construction et d'intégration des nouvelles applications et de maintenance des anciennes. Ceci doit permettre d'améliorer la réactivité de l'entreprise face à un environnement en constante évolution.

Ce chapitre explique comment décrire l'urbanisme des systèmes d'information à l'aide des diagrammes d'urbanisme.

Les points suivants sont abordés ci-après :

- ✓ "Contexte d'utilisation", page 100
- ✓ "Le plan d'urbanisme", page 102
- ✓ "Niveau d'urbanisme", page 104
- ✓ "Les canaux de communication", page 110
- ✓ "Correspondances entre zones d'urbanisme", page 113
- ✓ "Conformité d'un système applicatif au plan d'urbanisme", page 115

CONTEXTE D'UTILISATION

Les objectifs de l'urbanisme

L'objectif principal de l'urbanisme est de permettre au système d'information d'évoluer progressivement, sans remise en cause d'ensemble, tout en acceptant la cohabitation harmonieuse de programmes ou logiciels d'origine et d'époques différentes.

Il est pour cela nécessaire de définir des règles de conception des systèmes d'information qui soient valables pour de longues années et donc indépendantes des changements de technologies.

L'urbanisation présuppose que l'inventaire du patrimoine applicatif ait été dressé, mais il n'est cependant pas nécessaire d'avoir documenté l'architecture applicative en détail ; seule la nomenclature des applications et services est indispensable.

Un deuxième objectif de l'urbanisme informatique est d'identifier les redondances fonctionnelles pour :

- éviter d'en créer de nouvelles grâce à la réutilisation des ressources logicielles existantes lors du développement de nouvelles applications.
- réduire les coûts de maintenance en effectuant une rénovation bloc par bloc, qui consiste à définir une nouvelle ressource qui se substitue aux anciennes et réponde aux divers cas d'utilisation.
- consolider deux Systèmes d'Information comme dans le cas d'une fusion/acquisition.
- préparer le déploiement d'un logiciel d'EAI à grande échelle.

Il faut cependant noter que certaines redondances sont justifiées. En effet, le fait que deux cas d'utilisation soient semblables n'implique pas nécessairement qu'il faille leur donner une solution unique. D'autres contraintes (exploitation, performance, etc.) peuvent conduire à les maintenir séparés.

Justification de l'urbanisme

L'urbanisation se justifie pour les entreprises :

- qui ont un patrimoine applicatif important
- qui ont une longue histoire informatique
- pour lesquelles l'informatique est stratégique.

Elle peut se justifier également en cas de :

- Fusion/Acquisition
Elle entraîne immédiatement un très fort niveau de redondance, notamment sur les ressources d'exploitation. L'urbanisation est un préalable à la consolidation des moyens informatiques.
- Entreprises très « progicielisées »
Là encore, le niveau de redondance peut être important, notamment sur les données. L'objectif de l'urbanisation est de préparer la synchronisation des données.

En dehors de ces cas, les projets d'urbanisation sont rarement des projets phares. Ils doivent être menés « a minima » sans perturbation pour les autres projets.

Urbanisme et architecture applicative

L'architecture applicative vise à définir ou documenter (cartographie) la structure des applications et de leurs services en mettant en évidence leurs coopérations. Elle explique **comment** fonctionne le système d'information.

L'urbanisme du système d'information explique **ce que font** les applications de manière à documenter les redondances fonctionnelles dans le but de les réduire.

LE PLAN D'URBANISME

L'urbanisation consiste à regrouper les différentes applications utilisées dans le système d'information dans des blocs d'urbanisme sur le critère de la similitude fonctionnelle.

Plusieurs stratégies de regroupement peuvent être définies. Il est ainsi possible de définir plusieurs *plans d'urbanisme* décrivant le système d'information avec des points de vue différents.



Un plan d'urbanisme est un découpage du système d'information suivant un critère particulier. Ce peut être un regroupement par grandes fonctions de l'entreprise, ou bien suivant l'origine des applications, comme à la suite d'une fusion, voire par type d'environnement système, ou tout autre critère pertinent dans le contexte de l'entreprise.

Créer un plan d'urbanisme

Pour créer un plan d'urbanisme :

1. Dans la fenêtre de navigation **Objets principaux**, faites un clic droit sur le dossier **Plan d'urbanisme** et sélectionnez **Nouveau > Plan d'urbanisme**.
La fenêtre de création d'un plan d'urbanisme apparaît.
2. Saisissez son nom.
3. Cliquez sur **OK**.

Le plan d'urbanisme est créé et ajouté dans la liste des plans d'urbanisme.

Ouvrir un diagramme d'urbanisme

Pour ouvrir un diagramme d'urbanisme :

1. Dans la fenêtre de navigation **Objets principaux**, déployez le dossier **Plan d'urbanisme**.
2. Faites un clic droit sur le plan d'urbanisme qui vous intéresse et sélectionnez le diagramme qui vous intéresse.

Dupliquer un plan d'urbanisme

Il est possible de dupliquer un plan d'urbanisme dans le but de lister les étapes à prévoir dans son élaboration.



*Pour plus de détails sur la duplication des objets, voir le guide **HOPEX Common Features**, chapitre "Manipuler les objets HOPEX", paragraphe "Dupliquer les objets".*

Nom des plans d'urbanisme en environnement multilingue

Lorsque vous dupliquez un plan d'urbanisme, vous devez être dans la langue du référentiel pour que les traductions dans les différentes langues soient correctement reportées sur le(s) duplicata(s).

☛ Pour connaître la langue de votre référentiel, consultez les propriétés du référentiel.

NIVEAU D'URBANISME

Les traitements du système d'information sont découpés selon différents niveaux :

- **Zone**
C'est le plus haut niveau de découpage du système d'information de l'entreprise. On trouvera par exemple, dans la plupart des entreprises, une zone d'acquisition des données, une zone opérationnelle de traitement de l'information, une zone de pilotage, une zone "référentiel", etc.
- **Quartier**
Le quartier est un regroupement de traitements pouvant par exemple correspondre à une activité ou à un métier, c'est-à-dire à une nature d'information.
- **Bloc**
Le bloc correspond au composant de base de l'urbanisme. Un bloc est un ensemble de données et de traitements homogènes. Les applications informatiques sont regroupées à l'intérieur des blocs d'urbanisme.
- etc.

☛ Des niveaux supplémentaires peuvent être spécifiés le cas échéant.

Zones et sous-zones

Il est possible de créer des "sous-zones" à partir d'une zone, quel que soit le niveau d'urbanisme ("zone", "quartier" ou "bloc").

- Dans un Plan d'urbanisme, lorsque vous créez une **Sous-zone** à partir d'une "Zone" (clic droit sur la zone d'urbanisme **Nouveau > Sous-zone d'urbanisme**), la sous-zone devient automatiquement zone de niveau "Quartier".
- Dans un Plan d'urbanisme, lorsque vous créez une **Sous-zone** à partir d'un "Quartier", la sous-zone devient automatiquement "Bloc".

☛ Le visuel de la zone change suivant le niveau d'urbanisme.

Propriétés des zones, quartiers et blocs

Pour visualiser le **Niveau d'urbanisme** de la zone :

- 1 Ouvrez la fenêtre de propriétés de la zone.

L'attribut **Couleur** permet de donner une autre dimension à vos diagrammes d'urbanisme. La valeur donnée à cet attribut modifie la couleur de la forme de la zone, du quartier ou du bloc affichée dans le diagramme.

L'onglet **Éléments urbanisés** permet de préciser les applications, bases de données, services ou cas d'utilisations concernés.

Exemple de découpage en zones, quartiers et blocs

Voici une description d'une entreprise fictive appelée "Alizés" :

La société Alizés était à l'origine un constructeur de voiliers.

Pour vendre ou louer ses voiliers, la société a développé un réseau d'agences de voyages et de bureaux de représentation dans les îles à vocation touristique, telles que les Caraïbes ou Hawaï, ainsi que dans les principales grandes villes du monde. Elle a également développé un réseau d'agences bancaires pour financer l'achat de ses voiliers par des particuliers.

A la suite de changements technologiques, l'activité d'origine de construction de voiliers a périclité, et il n'en est resté que l'activité de fabrication de pièces détachées pour les bateaux.

En revanche, l'activité liée à la location et à la vente de bateaux s'est énormément développée. L'entreprise propose la location de bateaux de plaisance de toutes sortes, à voile ou à moteur, et des prestations annexes, telles que la location de matériel de plongée, la location de voitures, la réservation d'hôtels et de billets d'avion. Elle possède également maintenant plusieurs navires de croisières et sa propre chaîne de restauration.

L'entreprise est organisée en trois filiales principales :

- La société Alizés qui gère la location de bateaux de plaisance ou de croisière et les activités annexes telles que la réservation d'hôtels et de billets d'avions.
- La société MyBanque qui regroupe les activités bancaire du groupe.
- La société HBC qui fabrique et vend les pièces détachées pour les bateaux.

Voici une liste plus détaillée des diverses activités du groupe :

- Location de bateaux de toutes tailles
- Réservation de billets d'avion et d'hôtels
- Location de voitures
- Organisation de croisières
- Organisation de séminaires pour les entreprises
- Gestion des fiches techniques descriptives des bateaux
- Gestion de pièces détachées et réparation de bateaux
- Achats de pièces détachées pour les bateaux
- Fabrication et commercialisation de pièces détachées pour les bateaux
- Gestion de bibliothèques pour mettre des livres à la disposition des passagers
- Gestion de distributeurs de billets mis à la disposition des passagers sur les navires de croisière
- Gestion des ordres de bourses des passagers sur les navires de croisière
- Autres services bancaires
- Services de restauration sur place ou à domicile

Pour contribuer à l'introduction de nouvelles applications dans son système d'information, et pour faciliter l'évolution des applications existantes vers Internet, le directeur informatique de l'entreprise a commencé un plan d'urbanisation de son système d'information.

Le diagramme suivant présente un diagramme d'urbanisme fonctionnel de ce système d'information.

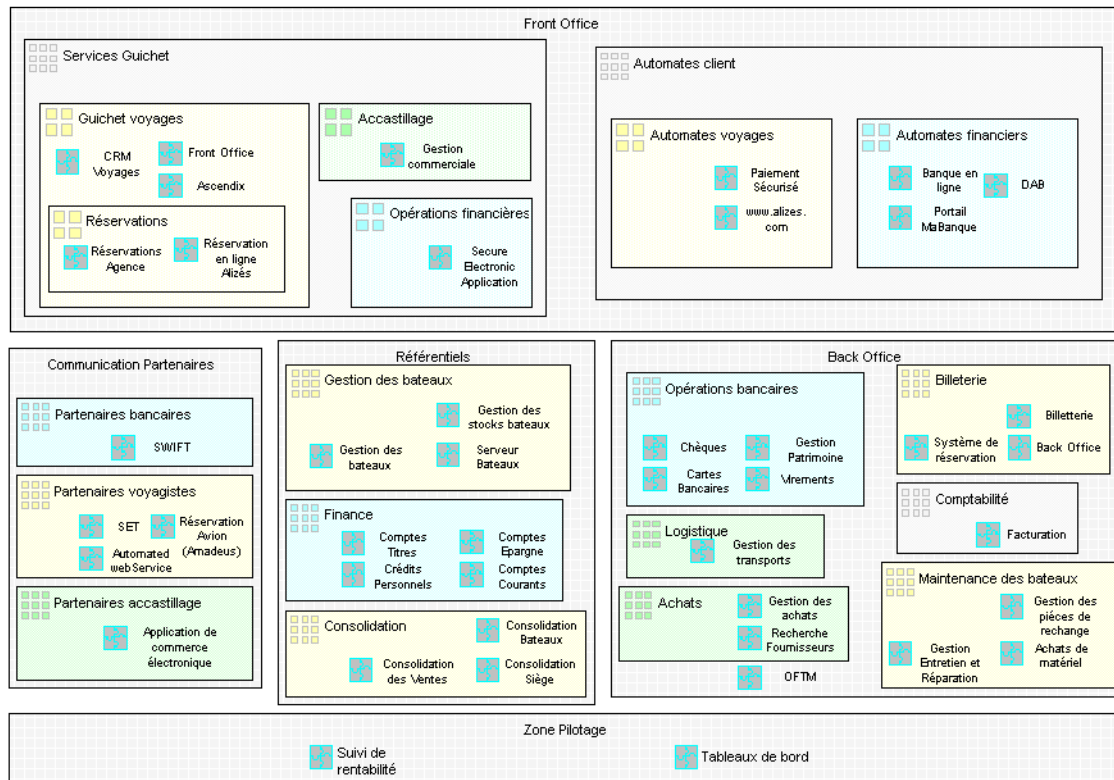


Diagramme d'urbanisme fonctionnel du système d'information

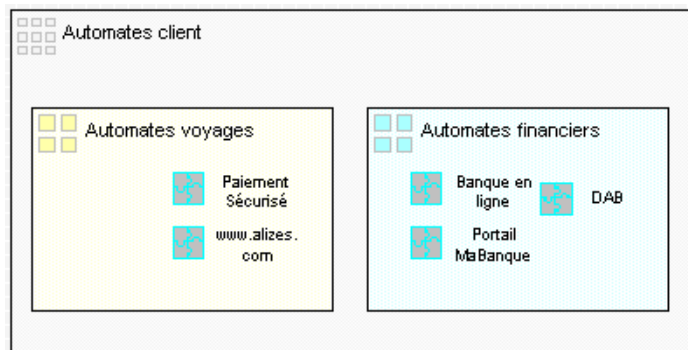
Le système d'information est découpé en zones en fonction de leur interaction avec les clients de l'entreprise et ses autres partenaires.

La couleur des zones, quartiers et blocs représente ici la filiale d'origine de ces zones, quartiers et blocs : Alizés en jaune, HBC en vert et MyBanque en bleu.

Zone d'acquisition des données

Une première zone appelée "Front Office" contient les applications qui sont utilisées directement au contact du client. Cette zone est découpée en deux quartiers :

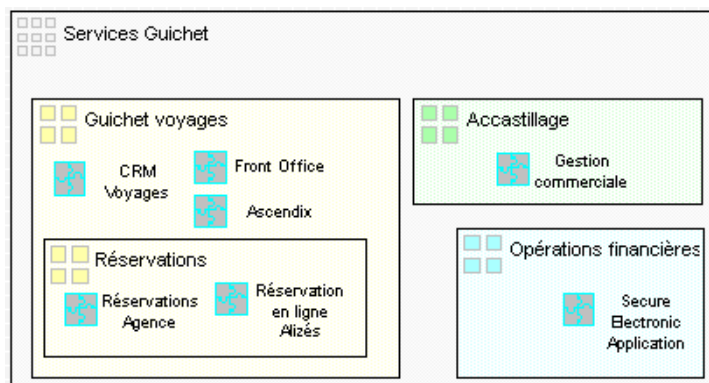
- les "Automates client" qui sont utilisés directement par le client sans intervention du personnel de l'entreprise, comme les systèmes de réservation par Internet ou les automates bancaires.



Quartier "Automates client"

Ce quartier est découpé en deux blocs, l'un contenant les automates utilisés pour la gestion des voyages, l'autre les automates financiers.

- les "Services au guichet" comme la réservation d'un voyage qui sont mis en oeuvre par des acteurs de l'entreprise en présence du client.

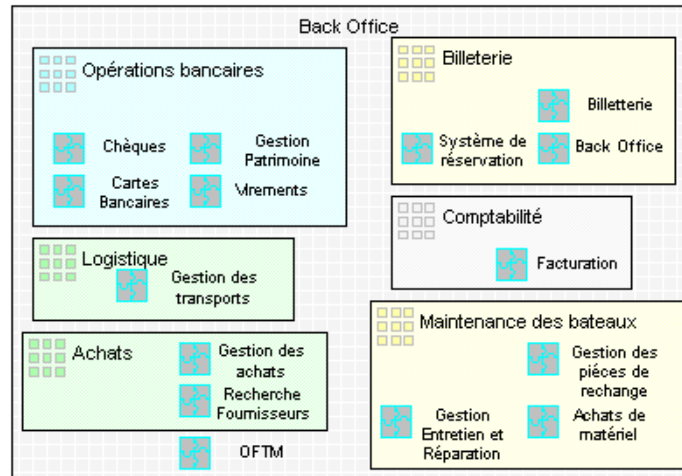


Quartier "Services Guichet"

Ce quartier est découpé en trois blocs, l'un contenant les applications utilisées pour la gestion des voyages, un autre les applications financières et le troisième, les applications utilisées pour la vente de pièces détachées pour les bateaux. Un sous-bloc spécifique a été identifié pour la gestion des réservations.

Zone de traitement des données

Une deuxième zone, appelée "Back office" contient les applications utilisées pour le service du client, mais sans sa présence.



Zone "Back Office"

Elle contient divers quartiers, consacrés aux "Opérations bancaires", à la "Billetterie", à la "Logistique", aux "Achats", à la "Maintenance des bateaux" et à la "Comptabilité".

Autres zones

La zone "Référentiel" contient les applications utilisées pour l'accès aux données persistantes de l'entreprise.

La zone de "Pilotage" contient les applications utilisées par la direction de l'entreprise pour orienter l'évolution de la stratégie de l'entreprise. Ce sont généralement des statistiques ou des tableaux de bord issus de la compilation des données persistantes du référentiel de l'entreprise.

Il est parfois également utile de définir une zone de communication avec les partenaires de l'entreprise, en particulier ses fournisseurs.

LES CANAUX DE COMMUNICATION

Un des principaux objectifs de l'urbanisation est de faciliter l'évolution du système d'information en permettant de modifier un des blocs applicatifs ou d'en ajouter un nouveau avec un minimum d'impact sur les autres blocs. Pour cela, il est indispensable d'identifier les *canaux* de communication entre les applications, ainsi que la nature des flux échangés entre elles.



Un canal est une voie de communication par laquelle transite les flux d'information (les messages et leurs contenus) entre un émetteur et récepteur. Ces derniers peuvent être des zones d'urbanisme, des métiers ou des acteurs de l'entreprise, etc.

Le diagramme d'urbanisme permet de mettre en évidence ces canaux de communications.

Prenons l'exemple du diagramme d'urbanisme "DU Canaux" qui décrit le plan d'urbanisme "SI Fonctionnel"

Ce diagramme présente les canaux de communications identifiés entre différentes zones et quartiers de l'entreprise.

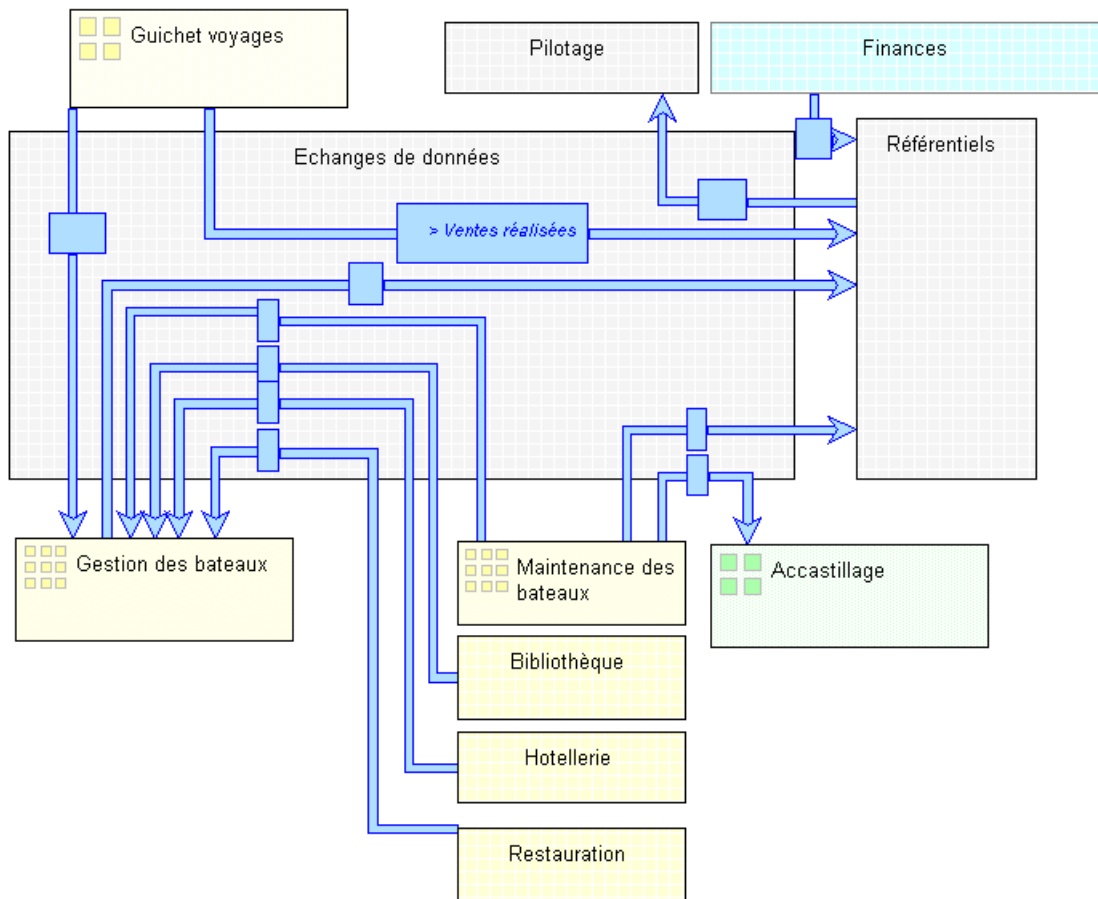


Diagramme d'urbanisme "Canaux"

Vous pouvez préciser directement les contenus d'information sur chaque canal dans l'onglet **Contenus échangés** de sa fenêtre de propriétés. Ces contenus correspondent aux informations autorisées à transiter entre deux zones. Les contenus peuvent être ajoutés dans le sens du canal (descendant) ou dans le sens inverse. Vous pouvez choisir d'utiliser un seul canal en positionnant les contenus autorisés dans les deux sens ou d'utiliser deux canaux différents pour lesquels seuls les contenus dans le sens descendant sont indiqués. La deuxième solution est plus lisible graphiquement mais génère plus de canaux entre les zones.

Les contenus indiqués pourront être comparés avec ceux effectivement véhiculés par les applications résidant dans la zone source et cible du canal. C'est ce que réalise le rapport de conformité. Voir "[Conformité d'un système applicatif au plan d'urbanisme](#)", page 115.

Ces contenus peuvent ensuite être affichés dans le diagramme.

Vous pouvez afficher les messages échangés entre les applications contenues dans chacune des zones, quartiers ou blocs d'urbanisme mises en correspondance par l'intermédiaire d'un canal de communication.

Pour afficher les messages échangés entre les applications :

1. Faites un clic droit sur un canal et sélectionnez **Formes et détails**.
2. Dans la fenêtre **Affichage**, sélectionnez le dossier **Messages descendants (Applications)**.
3. Dans l'onglet **Messages descendants (Applications)** précisez les messages échangés par ce canal de communications que vous désirez voir apparaître dans le diagramme.

☛ Les messages proposés sont tous les messages échangés par les applications appartenant aux zones, quartiers ou blocs mis en communication par le canal.

Vous pouvez faire de même pour les messages ascendants.

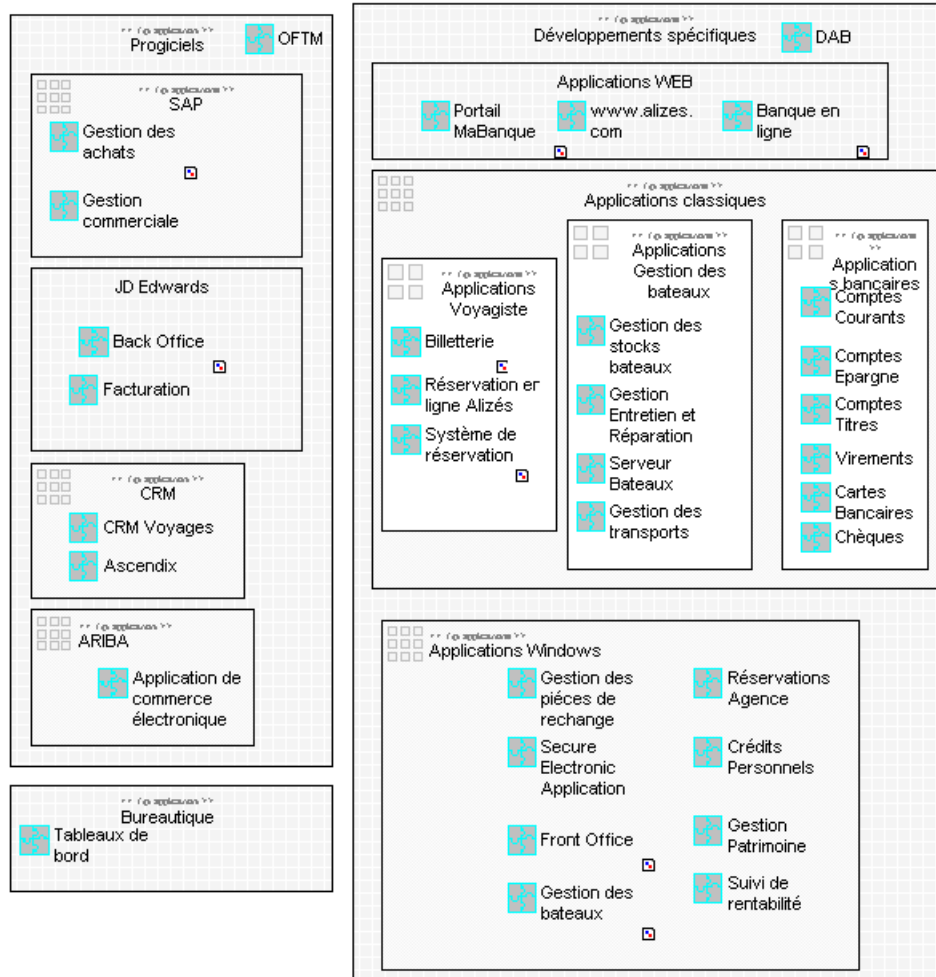
Les messages sélectionnés sont alors affichés dans le diagramme, précédés d'une flèche indiquant la direction du flux, ascendant ou descendant.



CORRESPONDANCES ENTRE ZONES D'URBANISME

Vous pouvez faire plusieurs plans d'urbanisme qui décrivent un même système d'information en fonction de critères différents.

Le diagramme suivant présente un découpage du système d'information de l'entreprise en fonction de critères purement techniques :

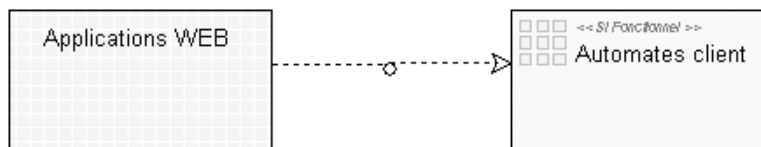


Vous pouvez préciser les **correspondances** entre les zones définies dans le cadre de plans d'urbanisme différents.

Une correspondance d'urbanisme permet de préciser les relations qui existent entre des zones, quartiers ou blocs définies dans le cadre de plans d'urbanisme différents.

Pour cela :

1. Dans la barre de menus **HOPEX**, sélectionnez **Affichage > Vues et détails** et assurez-vous que la vue **Correspondances** est sélectionnée.
2. Dans la barre d'objets du diagramme, cliquez sur **Correspondance d'urbanisme** .
3. Créez la correspondance en allant d'une zone à la deuxième.
4. Indiquez son nom dans la fenêtre qui s'affiche.
La correspondance s'affiche alors dans le diagramme.



Vous pouvez ensuite relier cette correspondance à d'autres zones, quartiers ou blocs.

CONFORMITÉ D'UN SYSTÈME APPLICATIF AU PLAN D'URBANISME

Parmi les rapports types sur les architectures applicatives (voir "[Rapports sur les architectures applicatives](#)", page 47), **HOPEX IT Architecture** fournit le rapport type "Conformité du SI par rapport à un plan d'urbanisme" qui constitue un outil de contrôle du plan d'urbanisme.

Ce rapport type a pour but de garantir une cohérence du système lors de l'ajout d'applications.

Spécifier les règles de conformité d'un plan d'urbanisme

Un système applicatif doit suivre les règles édictées par l'urbaniste du système d'information. Ces règles sont définies au niveau de chaque zone et les éléments hébergés dans ces zones doivent s'y conformer.

Deux types de règles peuvent être spécifiés sur une zone d'urbanisme :

- des **critères de zone d'urbanisme**. Il s'agit d'objets HOPEX que vous pouvez relier à la zone afin d'indiquer que ces objets doivent y être implémentés. Ce peut être :
 - une application
 - une fonctionnalité
 - une infrastructure technique
 - un objet manufacturé
 - un service applicatif
 - un standard.
- et des **exigences**.

Vous pouvez définir ces règles dans la fenêtre de propriétés d'une zone d'urbanisme.

Pour indiquer, par exemple, qu'une fonctionnalité existante doit être implémentée dans une zone d'urbanisme :

1. Ouvrez la fenêtre de propriétés de la zone.
2. Cliquez sur l'onglet **Éléments urbanisés**.
3. Sous le dossier **Critère des zones d'urbanisme**, faites un clic droit sur le dossier **Fonctionnalité** et sélectionnez **Relier**.
4. Recherchez la fonctionnalité à relier à la zone et cliquez sur **OK**.

Pour que le système applicatif soit conforme au plan d'urbanisme, il faut qu'un des éléments hébergés de la zone implémente cette fonctionnalité. Voir "[Indiquer les éléments hébergés par la zone](#)", page 116.

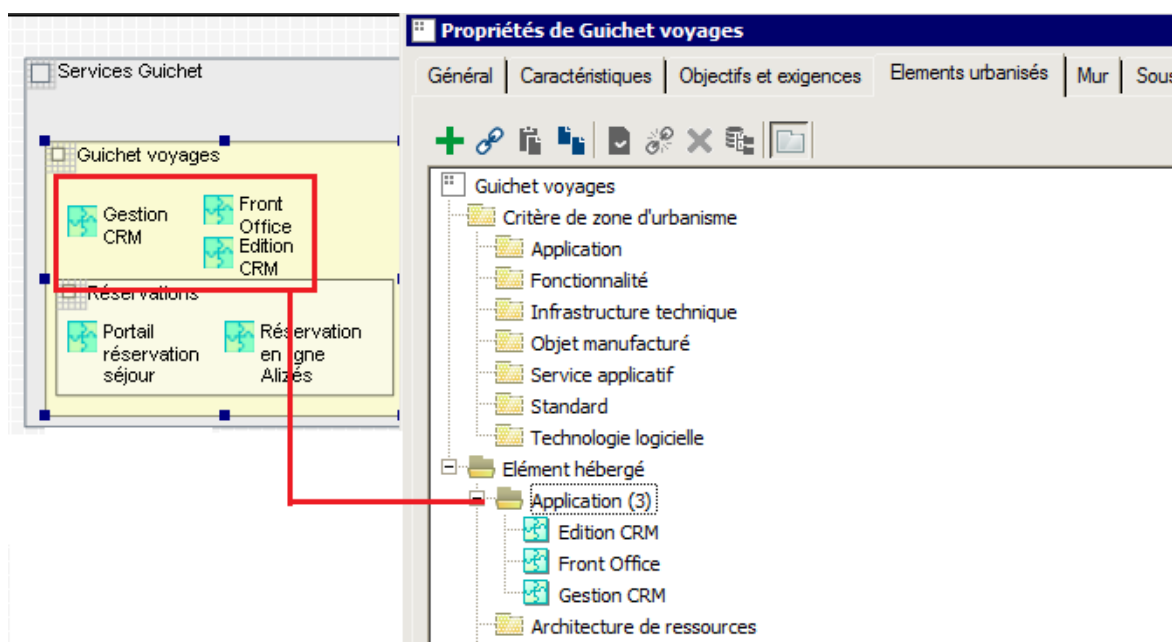
Les exigences à respecter pour la zone d'urbanisme peuvent être définies sous l'onglet **Objectif et Exigences**.

😊 Vous avez la possibilité d'analyser la structure hiérarchique d'un plan d'urbanisme. Le rapport de structure hiérarchique permet d'identifier l'organisation des zones d'un plan d'urbanisme et plus particulièrement de définir les zones touchées par de nouvelles fonctionnalités.

Indiquer les éléments hébergés par la zone

Pour déterminer le meilleur positionnement d'applications dans le plan d'urbanisme, vous pouvez lancer un rapport de "Conformité du SI par rapport à un plan d'urbanisme". Celui-ci place les applications dans les zones appropriées en fonction des règles que vous avez définies précédemment. Voir "[Fonctionnement du rapport de conformité](#)", page 117.

Selon les résultats du rapport, ou dans le cas d'exigences qui vous sont propres, vous pouvez préciser manuellement que telle application est hébergée par telle zone du plan.



Pour indiquer qu'une application est hébergée dans une zone d'urbanisme :

1. Ouvrez la fenêtre de propriétés de la zone.
2. Cliquez sur l'onglet **Eléments urbanisés**.
3. Dépliez le dossier **Elément hébergé**.
4. Faites un clic droit sur **Application** et sélectionnez **Relier**. La fenêtre de recherche apparaît.
5. Sélectionnez l'application et cliquez sur **OK**. L'application apparaît dans les éléments hébergés.

Lier les éléments hébergés aux règles d'urbanisme

Pour être évalués conformes au plan d'urbanisme, les éléments hébergés sur une zone devront suivre les règles définies au niveau de cette zone, autrement dit être reliés aux éléments spécifiés sur la zone (par exemple les fonctionnalités implémentées).

Fonctionnement du rapport de conformité

Le rapport type "Conformité du SI par rapport à un plan d'urbanisme" met en relation les informations des zones et des éléments hébergés pour déterminer le niveau de conformité de chaque système applicatif.

Le rapport généré est capable de positionner automatiquement les applications du système étudié dans les zones qui permettent d'obtenir le meilleur niveau de conformité. Ce positionnement automatique est basé sur la comparaison des applications aux règles mais prend également en compte l'interdépendance des applications. Ainsi, le déplacement d'une application d'une zone vers une autre peut engendrer une baisse de conformité d'une autre application (par exemple quand le déplacement d'une application bouleverse l'échange des données avec une autre).

Si l'utilisateur a positionné lui-même l'application dans une zone, l'outil respecte la décision de l'utilisateur et ne cherche pas à la repositionner.

Évaluer la conformité du SI au plan d'urbanisme

Pour lancer un rapport de conformité du SI par rapport à un plan d'urbanisme :

1. Cliquez sur la fenêtre de navigation **Documentation**.
2. Dépliez le dossier **Rapport**.
3. Faites un clic droit sur le rapport type "Conformité du SI par rapport à un plan d'urbanisme" et sélectionnez **Nouveau > Rapport**.

☺ *En règle générale, vous pouvez afficher les types de rapports disponibles sur un objet (faites un clic droit sur l'objet et sélectionnez "Recherche de rapport").*

Un assistant apparaît.

4. Indiquez le nom du rapport et cliquez sur **Suivant**.
L'assistant rappelle le rapport type avec son commentaire.
5. Cliquez sur **Suivant**.
Vous allez définir les paramètres du rapport, c'est à dire les données en entrée du rapport.

Le rapport type "Conformité du SI par rapport à un plan d'urbanisme" comprend deux paramètres à intégrer dans le plan d'urbanisme : les Eléments du SI et les Plans d'urbanisme.

6. Dans le cadre **Eléments du SI**, cliquez sur **Relier** .
7. Sélectionnez les éléments du SI à analyser, par exemple des applications.
Une fois sélectionnés, les éléments du SI apparaissent dans l'assistant.

8. De la même façon, dans le cadre **Plans d'Urbanisme**, cliquez sur **Relier**  et définissez le second paramètre "Plan d'urbanisme".
9. Cliquez sur **Suivant**.
Vous allez définir les chapitres de rapport.
10. Sélectionnez les chapitres de rapport.
 - Conformité au plan d'urbanisme : indique les applications qui correspondent le mieux aux zones d'urbanisme.
 - Paramètres du rapport : détaille les données entrées et la façon dont elles sont interprétées dans les différents rapports.
11. Cliquez sur **OK** (Web Front-End) ou **Terminer** (Windows Front-End).
Le rapport apparaît dans la liste des rapports.

Pour ouvrir un rapport :

1. Faites un clic droit sur le rapport souhaité et sélectionnez **Ouvrir**.
Les résultats du rapport s'ouvrent dans la page centrale.

Résultats du rapport

Le chapitre de rapport "Conformité au plan d'urbanisme" affiche la liste des critères de conformité :

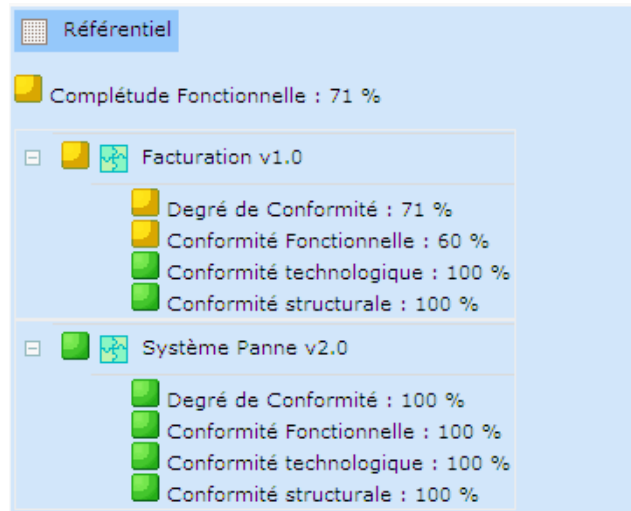
- Complétude fonctionnelle : vérifie que l'ensemble des éléments hébergés implémentent toutes les fonctionnalités requises. Pour une note maximale, aucune fonctionnalité ne doit être oubliée.
- Conformité fonctionnelle : chaque élément hébergé doit implémenter des fonctionnalités au sein de sa zone d'urbanisme et dans aucune autre.
- Transfert de données inter-zones : entre deux zones d'urbanisme sont parfois définis des canaux de communication transportant des contenus. L'analyse vérifie que ces flux d'information sont conformes aux exigences prévues entre les zones.
- Conformité technologique : l'élément hébergé doit être implémenté dans une zone conformément aux technologies définies dans la zone d'urbanisme.
- Conformité des exigences : l'élément hébergé dans une zone doit être contraint aux exigences de cette zone, autrement dit l'élément doit être relié à ces exigences.
- Conformité structurale : il convient de respecter la structure des éléments ; par exemple les sous-applications d'une application doivent être hébergées dans la même zone qui contient l'application.
- Conformité des standards : un élément conforme est un élément qui respecte les standards définis dans une zone.

Sur ces critères, le rapport compare les données définies sur les zones d'urbanisme (fonctionnalités, exigences etc.) et celles des éléments hébergés.

 *Il s'agit de critères prédéfinis, fournis par **HOPEX** pour répondre à des attentes courantes, mais vous pouvez ajouter vos propres critères de rapport. Pour plus de détails sur l'ajout de critère, voir l'article technique "City Planning Conformity - Adding an IT Criterion.pdf".*

Un premier tableau définit dans quelles zones d'urbanisme les éléments analysés - ici les applications Facturation v1.0 et Système Panne v2.0 - s'insèrent le mieux.

Vous pouvez voir les résultats de chaque critère en cliquant sur le signe + devant le nom de l'élément.



Le tableau suivant présente les niveaux de conformité par élément hébergé (applications) avec la valeur obtenue pour chaque critère.

Tableau des niveaux de conformité des éléments hébergés					
▼	Zone d'urbanisme ▼	 86 % Note globale ▼	 80 % Conformité Fonctionnelle ▼	 100 % Conformité technologique ▼	 100 % Conformité structurale ▼
 Facturation v1.0	 Référentiel	 71 %	  60 %	  100 %	 100 %
 Système Panne v2.0	 Référentiel	 100 %	 100 %	  100 %	 100 %

Le dernier tableau affiche les niveaux de conformité par zone d'urbanisme. Dans l'exemple ci-dessous, la zone "Référentiel" affiche une complétude fonctionnelle de 71%, qui s'explique par le fait que deux des fonctionnalités requises pour la zone ne sont pas implémentées par les applications de la zone.

Tableau des niveaux de conformité des zones d'urbanisme	
Zone d'urbanisme	71 % Complétude Fonctionnelle
Référentiel	71 % Ces fonctionnalités ne sont implémentées par aucun élément hébergé de la zone : - Connaître situation d'un client - Afficher le contenu du catalogue

Le rapport place automatiquement les applications hébergées dans les zones en fonction des éléments qui la définissent (ex. fonctionnalité, exigence, flux de

données...). Le choix se fait de telle sorte que la note de conformité globale soit la plus élevée possible.

Certaines applications peuvent être suivies d'une icône représentant une épingle . Cette épingle indique que l'application a été positionnée dans la zone par l'utilisateur et non par le rapport.

 Les rapports sont dynamiques; cliquez sur **Recharger**  (en haut à droite du rapport) pour mettre à jour l'analyse en fonction des dernières modifications apportées au référentiel.

ARCHITECTURE ORIENTÉE SERVICES (SOA)



Surfant sur la vague technologique des services Web, en grande partie à l'origine de l'engouement qu'elle provoque, l'architecture orientée service (SOA) fait peu à peu son chemin comme principe d'organisation des systèmes informatiques.

Cette nouvelle approche cherche à séparer le service rendu du mode de communication. Les appels de fonctions sont remplacés par des échanges d'informations portées par des messages. Avec cette approche dite orientée service, chaque service est en mesure de gérer ses données et son état.

Les points qui suivent expliquent comment organiser l'architecture orientée services de votre système d'information.

- ✓ ["Le service, charnière entre les métiers et l'informatique", page 122](#)
- ✓ ["Principes de l'architecture orientée service", page 123](#)
- ✓ ["Ajustement de l'architecture applicative", page 129](#)

HOPEX System Oriented IT Architecture propose les fonctionnalités dédiées à la représentation et à la documentation d'architectures IT selon une approche orientée service. Pour plus d'informations, voir le guide **HOPEX System Oriented IT Architecture**.

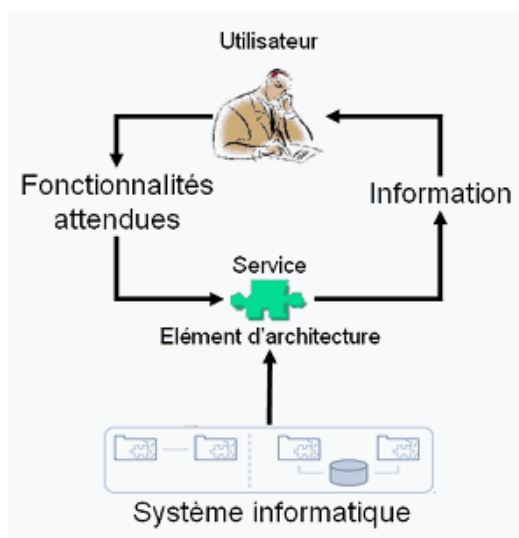
LE SERVICE, CHARNIÈRE ENTRE LES MÉTIERS ET L'INFORMATIQUE

Sur le plan des spécifications fonctionnelles, les départements métiers peuvent exprimer leurs besoins - pour schématiser - sous forme de listes d'informations qu'ils attendent du système. Ce faisant, ils décrivent les fonctionnalités qui doivent à terme être supportées par un système.

Les équipes informatiques ont, quant à elles, pour tâche de grouper sous forme de service les messages délivrant les informations demandées.

Dans un contexte de SOA, un service est une unité de traitements cohérents et indivisibles qui coordonne un ensemble de messages et d'événements, dans le but de réaliser une ou plusieurs fonctionnalités.

Les services peuvent être mis en correspondance avec les opérations réalisées à chaque étape des processus métiers de l'entreprise. Les services se présentent ainsi comme point d'articulation entre les métiers et l'informatique. Les métiers les voient par le biais des fonctionnalités rendues, les informaticiens les voient en tant qu'éléments d'architecture du système.



Le service, à l'articulation des métiers et de l'informatique

PRINCIPES DE L'ARCHITECTURE ORIENTÉE SERVICE

L'architecture orientée service est une organisation du système informatique fondée sur l'identification et l'agencement des services et des messages assurant leur communication. Le service est ainsi l'unité d'œuvre de l'architecture du système, tant dans les phases de spécification que de développement et de déploiement. L'approche SOA intègre nativement les principes de modularité, d'interfaçage, de contractualisation, et d'interopérabilité. Elle assure ainsi une adaptation rapide du système d'information au regard des évolutions des besoins de l'entreprise.

Parallèlement, elle permet de capitaliser la mise en place de bonnes pratiques par l'élaboration d'une architecture de référence. Cette architecture de référence pourra également être utilisée pour répondre à des problématiques de convergence de systèmes.

Concevoir un système orienté service

La notion de service répond à quatre principes qui sont les fondements de la conception des systèmes d'information :

- La modularité : les services constituant un agencement cohérent de fonctionnalités permettent de concevoir le système d'information comme un assemblage d'éléments préfabriqués autonomes.
- L'interfaçage : la frontière d'un service permet de distinguer les données échangées par le service des traitements internes réalisés pour produire ces données. Les données requises et produites par un service définissent l'interface de communication du service.
- La contractualisation : l'interface d'un service est conçue comme un ensemble d'échanges réciproques entre le service et ses utilisateurs. Cette réciprocité définit les rôles du service et de ses utilisateurs dans le cadre d'un contrat d'échange.
- L'interopérabilité : seules les données échangées sont à partager. Ainsi, un programme COBOL peut-il être mis en relation avec un programme java dès lors que tous deux interagissent en tant que services échangeant des messages. On passe ainsi d'une intégration technique (cobol/java) à une intégration sémantique (de quoi parle-t-on dans l'échange ?).
En outre, les standards de communication (HTTP, SOAP) et de description de données (XML) ou d'interfaçage (WSDL) ont permis l'accès à l'information au meilleur coût et pour toutes les organisations, ce qui démultiplie les capacités d'interopérabilité.

Concevoir une architecture de référence

Parmi les avantages apportés par la mise en place d'une architecture orientée service, il faut considérer le découplage possible de ce type d'architecture vis-à-vis

des applications qui les mettent en œuvre. Ainsi, on peut envisager la conception d'architectures dédiées à des problématiques restreintes mais ne faisant pas intervenir d'applications. Dans ce type d'architecture, les éléments d'étude sont réduits aux services, à leurs échanges et aux fonctionnalités rendues.

La conception de ce type d'architecture de service peut être motivée par des projets de convergence ou de mise en place de bonnes pratiques.

L'architecture orientée service trouve également d'autres débouchés naturels tels que la réutilisation des composants, la fourniture de service de support et l'amélioration de la gestion de capacité.

Réutilisation

Une description des applications allant jusqu'au détail des services qu'elles regroupent permet d'identifier les services élémentaires qui pourront être publiés pour réutilisation lors de la spécification d'une nouvelle application.

Fourniture de services de support

L'architecture de service facilite la mise en place d'architectures de support dédiées à la gestion des services fonctionnels : gestion de la sécurité, gestion de la communication, gestion de la performance. Les architectures de support sont un outillage clé pour la gouvernance des systèmes d'information.

Amélioration de la gestion de capacité

L'architecture de service facilite l'administration et le routage des messages permettant la mise en place de gestion de capacité. Ainsi, un service introduisant des chutes de performance lorsque le taux de charge augmente est dupliqué sur plusieurs serveurs et le message d'appel du service est routé vers le serveur ayant la charge la plus faible (principe de Load Balancing).

Caractérisation des services

Que ce soit lors de la conception d'un système orienté service, ou de la création d'une architecture de référence, l'identification des services est une étape incontournable.

On peut s'aider de la typologie ci-dessous pour identifier les services :

- les services fonctionnels
- les services déduits des modèles de données
- les services applicatifs
- les services d'orchestration de processus
- les services techniques et d'infrastructure

Ces services peuvent être déduits de l'analyse des processus de l'entreprise. L'analyse d'un processus aboutit à une décomposition en opérations réalisées par les acteurs de l'entreprise (voir le guide de **HOPEX Business Process Analysis** pour plus de détails). Pour chaque opération, il est possible de déterminer dans

quelle mesure elle doit être outillée à l'aide d'un service informatisé. On obtient alors un ensemble de services utilisés lors de la réalisation du processus.

Les services déduits des modèles de données

Les données utilisées dans le contexte des processus de l'entreprise constituent également des indicateurs pour la détection des services. Dans l'exemple précédent, l'opération "Finaliser la proposition de crédit" met en évidence la nécessité de disposer d'un ou plusieurs services de gestion de l'entité "Offre de crédit". Le crédit est lié lui-même à l'entité client dont on voudra consulter les propriétés.

Les services de lecture, de mise à jour, de création et de destruction de chacune de ces entités peuvent ainsi être définis. Ces quatre services de base sont généralement référencés par l'acronyme CRUD (Create, Retrieve, Update, Delete) et aboutissent le plus souvent à une implémentation dans une base de données relationnelle. De ces services atomiques découlent également les services d'interface homme-machine les mettant en œuvre.

Les services applicatifs

Les services applicatifs encapsulent des algorithmes réalisant des calculs complexes ou mettant en œuvre des règles métiers. Dans l'exemple de crédit, un service de simulation de prêt immobilier est utilisé lors de l'entretien commercial avec le client. Ce service met en œuvre des algorithmes de calculs centralisés et peut-être partagé par d'autres processus.

Les services d'orchestration de processus

Les services d'orchestration de processus permettent d'enchaîner des tâches implémentées par des services élémentaires dans le but d'automatiser tout ou partie de la chaîne de valeur d'un processus métier.

Par exemple, un système de réservation de billets d'avion prend en charge la mise en attente d'un client au cas où un vol est complet. Dès qu'un désistement survient, le service gérant le processus prend automatiquement en charge le rappel du client (envoi du courrier électronique) et opère la réservation. Dans ce cas de figure, le service gère l'état courant de la transaction de réservation (en attente, réservé, annulé) pendant une période qui peut aller de quelques heures à plusieurs semaines.

Les moteurs d'exécution de processus du marché (BPMS : Business Process Management System) reposent sur la description formelle des processus par le biais d'un langage. Le plus populaire est BPEL (Business Process Execution Language). Il existe également une notation graphique pour représenter ces définitions de processus : BPMN (Business Process Management Notation).

Voir le guide **HOPEX System Blueprint** pour plus de détails.

Les services de communication

Une architecture orientée service est basée sur les services et leurs échanges. La gestion des échanges donne lieu elle-même à la mise en place de services dédiés.

Dans une architecture de ce type, il est impératif d'utiliser des protocoles et des services standard capables de transporter les messages entre les différents services fonctionnels. Ces derniers étant de nature hétérogène, le système de communication repose sur des protocoles indépendants des messages transférés. Il est alors nécessaire de disposer de services dédiés à la "traduction" des messages depuis le format natif vers le format indépendant et inversement.

SOAP est un bon exemple de protocole mis en place pour répondre à cette problématique. On trouve bon nombre de services capables de prendre en charge ou de transférer un message SOAP que ce soit sur des plates-formes Unix, Windows ou autres.

Les services d'administration

Le système d'information est un des leviers majeurs de la mise œuvre des processus. En tant que tel, il nécessite une administration qui doit permettre de prendre des décisions relatives à son évolution. Il est donc opportun de définir une architecture spécifique pour son administration. L'objectif est généralement de déterminer :

- si l'ensemble des services mis en place répondent aux besoins
- s'ils sont correctement dimensionnés du point de vue de la performance
- s'ils sont continuellement accessibles
- s'ils sont régulièrement et facilement mis à jour, etc.

Pour répondre à ces problématiques internes, on pourra avoir recours aux services définis ci-dessous (liste non exhaustive) :

- Services de recensement des incidents
- Services d'analyse du trafic réseau
- Services de mise à jour des applications
- Services d'administration de la sécurité
- Services d'identification des ressources matérielles, etc.

Les services de sécurité

Tout comme celle d'administration, la notion de sécurité du système d'information peut être vue en dehors de l'aspect fonctionnel de l'entreprise. C'est une problématique à part entière qui traverse toutes les couches de services quel que soit le métier adressé. Ici, l'objectif est de protéger le système d'information contre toutes sortes d'agressions qu'elles soient logicielles ou matérielles, intentionnelles ou non.

Parmi les services correspondant à cette problématique, on trouve :

- Les services d'accès : l'objectif est de contrôler l'accès aux applications, aux serveurs ou à certaines parties du réseau. Les applications de type pare-feu sont dédiées à cette problématique.
- Les services d'authentification : il s'agit de contrôler l'identité des personnes accédant à des ressources logicielles. Des protocoles comme SSO (Single Sign On) permettent d'unifier l'authentification pour plusieurs applications utilisées sur le même poste.
- Les services d'intégrité : le but est d'assurer que les données transmises n'ont pas été altérées pendant leur transport. Ces services sont généralement basés sur le calcul d'une clé dont la valeur dépend du

contenu transféré. A l'arrivée, la clé et le contenu sont comparés pour détecter une éventuelle faille d'intégrité.

- Les services de confidentialité : il s'agit de contrôler que les données transférées à une personne ne peuvent pas être lues par une autre. Des services d'encryptage sont généralement la solution.

Un certain nombre d'organismes se sont donnés pour tâche de recenser les risques envisageables en termes de sécurité afin de mieux les appréhender. On trouve parmi eux les recommandations ITIL (Information Technology Infrastructure Library), ou celle du CERT (Computer Emergency Response Team).

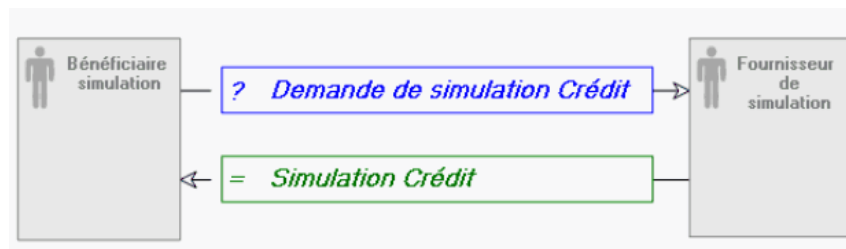
Identification des échanges

L'identification des échanges est réalisée conjointement à l'identification des services. De la découverte d'un service découlent les échanges qui s'y rattachent. Par exemple, à partir d'un service "Simulateur de crédit" on identifie un message entrant "Demande de simulation crédit" et un message sortant "Simulation crédit".

Cet exemple montre qu'un message seul ne suffit pas à représenter l'ensemble des interactions nécessaires à l'obtention d'un résultat. A cette fin, il est nécessaire de regrouper ces messages au sein d'une interaction.

Une interaction est un contrat d'échange entre deux partenaires. Chaque partenaire est représenté par le rôle qu'il joue dans le contrat d'échange. Chaque échange élémentaire est spécifié par un message.

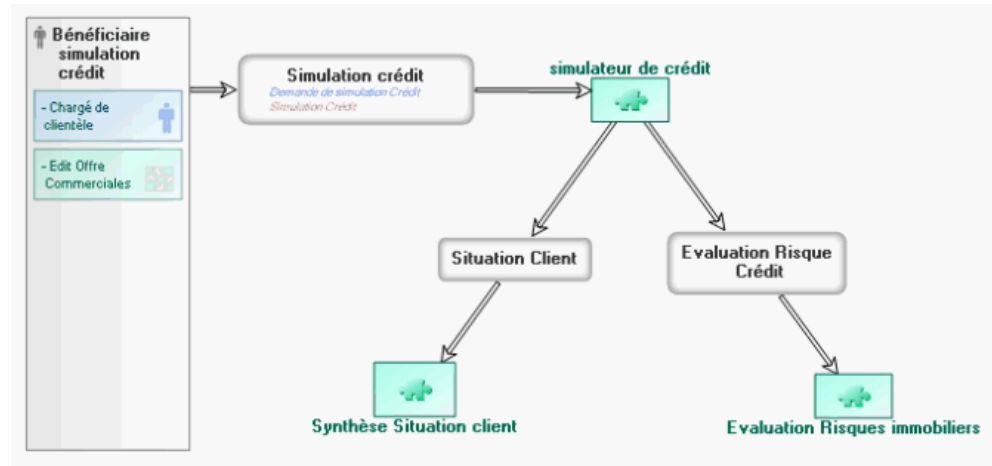
Les interactions permettent de décrire les interfaces des services, c'est à dire l'ensemble des messages en entrée et en sortie qu'ils doivent prendre en compte. Dans notre exemple l'interaction Simulation Crédit comporte les deux messages Demande de simulation crédit et Simulation crédit.



Exemple d'interaction

L'application "Simulation de crédit" offre un service "Simulateur de crédit." Celui-ci joue le rôle de "Fournisseur de simulation". Il doit donc pouvoir répondre au message "Demande de simulation crédit" par le message "Simulation crédit". Le diagramme d'architecture interne de l'application "Simulation de crédit" montre comment le service "Simulateur de crédit" est sollicité au moyen de l'interaction "Simulation crédit". Ce même service doit lui-même collaborer avec les services

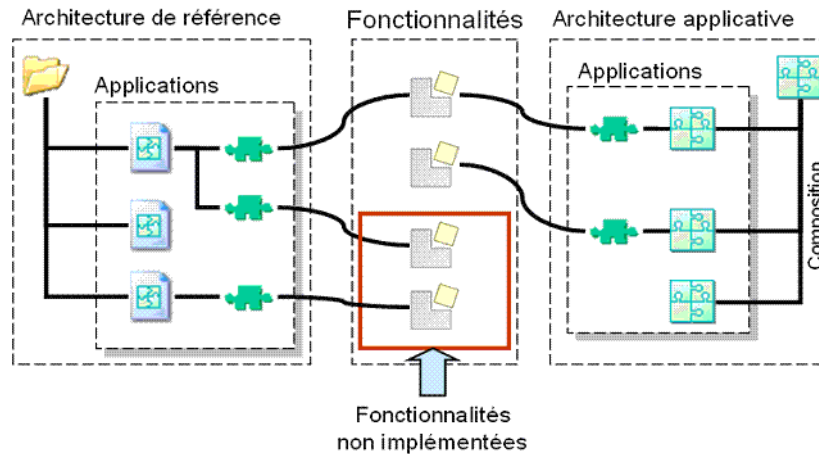
"Consultation situation client" et "Evaluation risques immobiliers" afin de pouvoir effectuer ses traitements.



Exemple de services collaborant - Application Simulation de crédit

AJUSTEMENT DE L'ARCHITECTURE APPLICATIVE

L'analyse des écarts permet de recenser les fonctionnalités non implémentées.



Adéquation de l'architecture applicative avec l'architecture de référence

Il est possible d'évaluer :

- quelles fonctionnalités ne sont fournies par aucune application.
- combien d'applications existantes ne peuvent pas être intégrées à l'architecture de services (problème d'interface, de séparation des services).
- pour les applications intégrées, quels services doivent être revus. Par exemple, parce qu'ils intègrent trop de fonctionnalités et donc rigidifient l'architecture de services.

Intégration de nouvelles applications

Lorsque vous intégrez de nouvelles applications dans un parc applicatif, différents critères sont à considérer comme :

- la pérennité de l'application : une étude du fournisseur doit être envisagée afin d'anticiper la longévité du produit. On considère généralement des critères liés au fournisseur comme les conditions de maintenance et de support, les références client, la surface financière de l'entreprise, etc.
- les contraintes techniques : bien que l'architecture de services ait été définie de manière à s'abstraire des contraintes d'implémentation, il y a souvent des contraintes techniques qui sont imposées dès le début de l'étude. Ces contraintes sont à considérer lors des choix applicatifs (et souvent il vaut mieux les garder à l'esprit pendant la conception de l'architecture orientée service pour éviter des études dont on sait par

avance qu'elles sont non réalisables). Par exemple, si le système d'exploitation est imposé (à cause du parc matériel existant), le catalogue d'applications s'en trouve inévitablement restreint.

- l'utilisation de standards : l'application choisie repose-t-elle sur des standards, que ce soit pour l'export de ces données (format XML), pour le stockage des informations (base relationnelle), pour l'interface homme-machine (réduction des temps d'apprentissage), etc. Ces considérations conditionnent les potentialités d'évolutions futures.

Accessibilité des services

Les applications rendent accessibles certains de leurs services pour d'autres applications en vue de leur réutilisation. Pour ce faire, elles doivent disposer de points de connexion avec l'extérieur. Seuls les services reliés à un point de connexion sont accessibles pour une application donnée.

Déploiement

Il s'agit de définir sur quelle infrastructure reposent les applications choisies pour réaliser l'architecture orientée service.

Si les contraintes techniques n'ont pas été imposées au départ, le choix des applications va certainement en apporter de nouvelles. Par exemple, le système d'exploitation, la taille des serveurs d'hébergement, la puissance de ces serveurs sont autant de paramètres à considérer lors de la mise en place de l'infrastructure technique.

Rapports d'analyse SOA

HOPEX IT Architecture fournit des rapports permettant d'analyser une architecture orientée services :

- Analyse fonctionnelle de services : ce type de rapport compare un groupe de services informatiques fournissant des fonctionnalités vis-à-vis d'un ensemble de fonctionnalités attendues (le périmètre fonctionnel).
- Services It associés à un processus : ce type de rapport permet de retrouver les services en relation avec un ensemble de processus.

Voir ["Rapports sur les architectures applicatives"](#), page 47.

DÉTAIL DES CONCEPTS



Ce chapitre présente de manière détaillée les concepts utilisés dans **HOPEX IT Architecture**.

Les diagrammes utilisés par **HOPEX IT Architecture** permettent de supporter les réflexions qui concernent l'architecture des systèmes d'information.

Selon l'objectif visé, la modélisation présente une approche globale du système d'information ou s'intéresse à un sous-système particulier.

De même, elle peut privilégier une vision logicielle du système d'information centrée sur les applications, les flux d'information échangés et les structures de mémorisation des données ou une vision technique mettant au premier plan les éléments matériels.

Pour répondre à ces différents besoins, **HOPEX IT Architecture** propose plusieurs types de représentation :

- les Diagrammes d'Architecture Applicative, pour une vision logicielle du système d'information.
- les Diagrammes d'Infrastructure Technique, qui privilégient les aspects techniques et matériels du système d'information.
- l'Arbre Applicatif, qui présente la décomposition logicielle d'une Application.

Les points suivants sont abordés ici :

- ✓ ["Les Diagrammes d'Architecture Applicative", page 132](#)
- ✓ ["Les Diagrammes d'Infrastructure Technique", page 137](#)
- ✓ ["L'Arbre Applicatif", page 141](#)
- ✓ ["Concepts gérés par HOPEX IT Architecture", page 144](#)

LES DIAGRAMMES D'ARCHITECTURE APPLICATIVE

Les diagrammes d'architecture applicative permettent de décrire l'environnement logiciel d'une *application*, d'un système applicatif ou de l'entreprise elle-même. Ils permettent également de décrire l'architecture interne d'un système applicatif, d'une application ou d'un *service*.

Pour répondre à ces différents besoins, il en existe plusieurs spécialisations :

- **Vue d'ensemble des applications**
Ce diagramme présente les principales applications de l'entreprise et leurs interactions.
- **Diagramme d'environnement d'une application**
Ce diagramme présente les échanges entre l'application et ses principales applications utilisatrices. Il permet de placer l'application dans son environnement sans aborder la question de son architecture interne. Il définit les échanges attendus ou constatés (contrats d'interfaçage, « contrats de services » ... attendus). Mais il ne décrit pas sa décomposition en modules applicatifs, services... et ne contient pas les applications, services ou bases de données nécessaires à son fonctionnement.
- **Diagramme d'architecture applicative interne**
Ce diagramme détaille les échanges entre les composants (applications et services) impliqués dans le comportement d'une application. Il permet de définir l'architecture et les échanges internes à l'application nécessaires à son bon fonctionnement et d'expliquer comment sont traités ou produits les échanges avec l'environnement de l'application.
- **Diagramme d'architecture de service**
Ce diagramme détaille les échanges entre les composants impliqués dans le comportement d'un service.

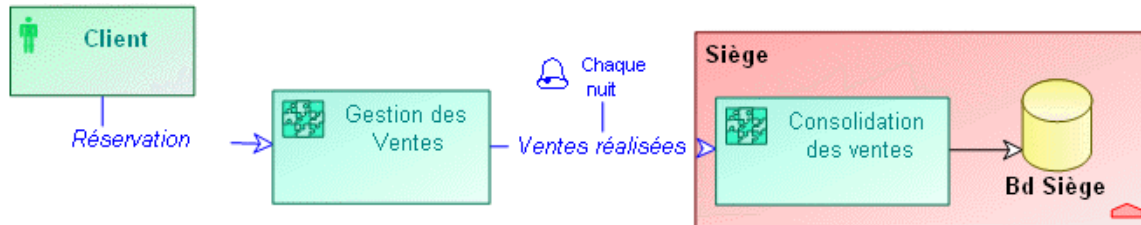
Les principaux concepts de ces diagrammes sont les applications et les communications qui existent entre elles.

Les concepts de base des DAA

Sur le périmètre décrit, les diagrammes d'architecture applicative montrent :

- l'*application* décrite, ou les applications principales dont on étudie les interdépendances.
- les *flux d'Information* échangés par les applications, ainsi que leurs éventuelles temporisations.
- les *sites* d'exécution des applications.
Une application peut concerner plusieurs sites (application client/serveur dont les traitements sont répartis, application installée dans plusieurs sites de l'entreprise). Une application peut être subdivisée en sous-applications.

- les **bases de données** consultées ou mises à jour par les applications, en précisant les sites d'implantation et les machines qui les supportent.
- les **acteurs** externes à l'entreprise (clients, fournisseurs) qui émettent des messages (flux d'information) pris en compte par les applications ou en reçoivent.



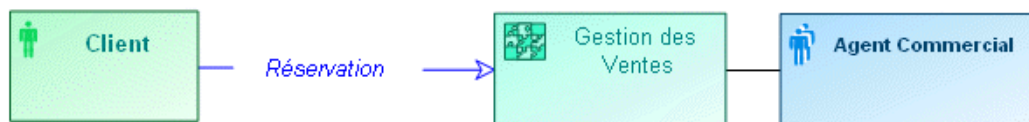
- les interfaces entre les applications, ainsi que les flux d'Information et les bases de données consultées ou mises à jour.
- les échanges d'informations entre bases de données lorsque les applications qui les effectuent ne sont pas significatives (transferts de fichiers, réplication d'une base de données).



- les **services** utilisés par les applications pour faire par exemple les transferts d'une base de données vers une autre.

Eléments organisationnels : les acteurs

Les diagrammes d'architecture applicative peuvent également présenter des éléments d'organisation tels que les acteurs internes ou externes à l'entreprise qui utilisent les applications et les messages qu'ils reçoivent ou émettent.

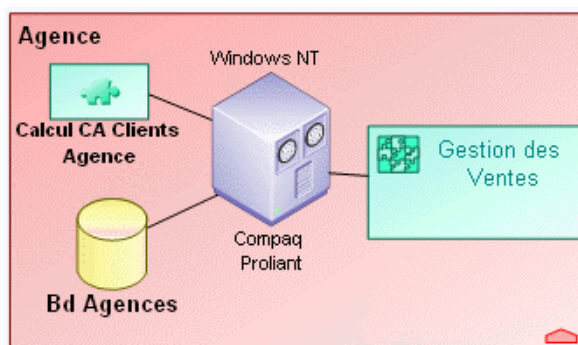


Eléments techniques et matériels : les serveurs et postes de travail

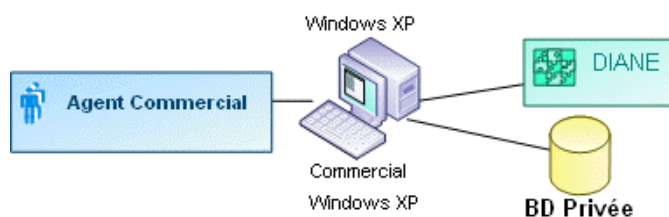
Les diagrammes d'architecture applicative peuvent être enrichis par des éléments techniques qui permettent le dimensionnement adéquat des machines et des réseaux.

Vous pouvez préciser sur quel serveur une application ou un service tourne, ainsi que l'implantation géographique de ce serveur sur un site, et la ou les bases de données implémentées sur ce serveur. Connaissant les différentes applications qui tournent sur ce serveur, vous êtes mieux à même de dimensionner celui-ci. Vous

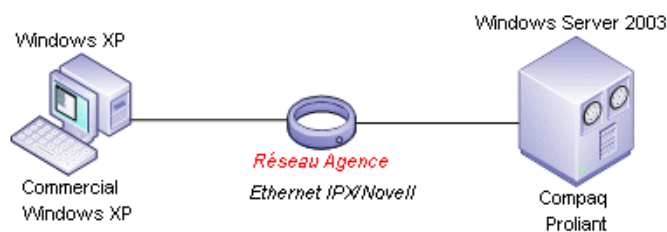
pouvez alors en déduire le type de machine à utiliser, ainsi que le système d'exploitation, et la capacité mémoire et disque nécessaires.



La définition des postes de travail nécessite de décrire les machines qui seront affectées aux acteurs de l'entreprise. Vous pouvez indiquer les applications et les bases de données qui fonctionnent dessus. Ces informations permettront de dimensionner de manière adéquate ces machines.



Vous pouvez décrire le réseau qui va permettre de relier entre elles les différentes machines.

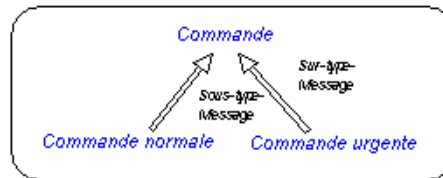


Spécialisation

Vous pouvez préciser les différents cas particuliers d'un même message. Certains traitements seront valables dans le cas général (appelé sur-type), par exemple toutes les commandes sont émises par un client. D'autres ne seront utilisés que dans un cas particulier (appelé sous-type) par exemple, les commandes passées par

Internet ont un traitement particulier différent de celui pour les commandes passées par Fax.

La commande émise par un client sera traitée en suivant deux processus différents selon qu'il s'agit d'une commande normale ou d'une commande urgente.

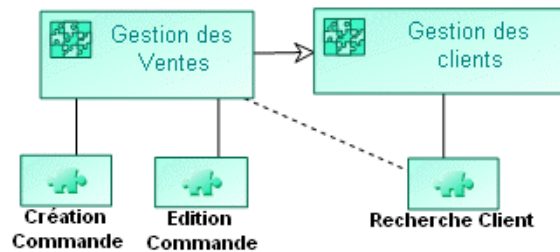


Spécialisation des messages

Il est possible de définir des sous-types d'acteur, site, et message.

Répartition des traitements

Répartition des traitements entre une application cliente et une application serveuse. Il est également possible de préciser quels services définis dans l'application serveuse sont utilisés par l'application cliente.



Répartition des traitements

Analyse inter-sites

Liaisons nécessaires ou existantes entre les sites.

Les principales utilisations des DAA

Selon l'étendue du champ de l'étude, l'approche du système d'information sera générale ou détaillée : étude globale du SI et des interactions entre les sous-systèmes (diagramme général des applications de l'entreprise), ou focalisation sur un sous-système pour mettre en évidence les constituants élémentaires (Diagramme d'une application particulière et de son environnement).

L'approche globale correspond à une cartographie des applications du système d'information et vise ainsi à positionner chacune d'elles par rapport à l'architecture

générale du système d'information, en mettant en évidence les sites principaux, les flux d'information et les données majeures.

L'approche détaillée orientée sur la vision "logicielle" du système d'information permet d'étudier et de représenter la décomposition d'une application en sous-ensembles livrables ou substituables par morceau. Elle fait souvent apparaître les principaux acteurs et, si besoin, les réseaux et les équipements matériels requis.

LES DIAGRAMMES D'INFRASTRUCTURE TECHNIQUE

L'infrastructure technique concrétise matériellement les objectifs stratégiques de l'architecture logicielle.

Ce type de diagramme (Diagramme d'Infrastructure Technique ou DIT) est donc orienté vers la description d'architectures techniques détaillées : gestion des Réseaux, des flux d'Information, du routage.

Il permet également d'étudier le dimensionnement adéquat des équipements qui supportent les applications et les bases de données, ainsi que des réseaux qui véhiculent les informations échangées.

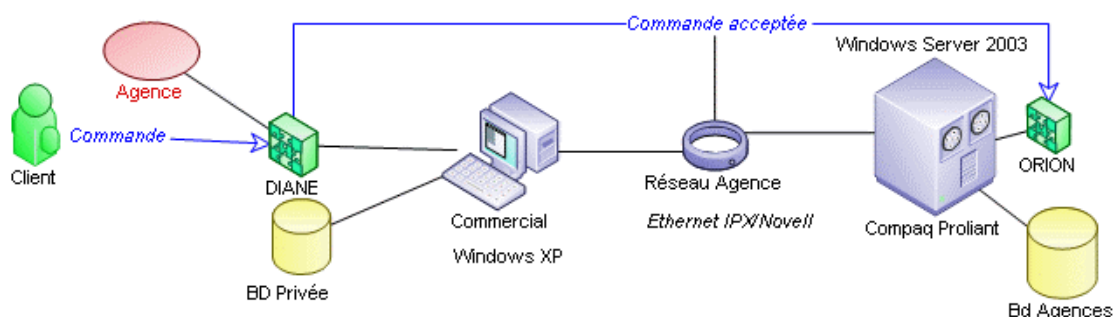
Les principaux concepts décrits sont les réseaux et les serveurs.

On peut utiliser un diagramme d'infrastructure technique pour définir une "Vue d'ensemble de l'infrastructure technique" de l'entreprise ou pour concevoir le "Diagramme d'une infrastructure technique" particulière.

Les concepts de base du DIT

Sur le périmètre décrit, un diagramme d'infrastructure technique montre :

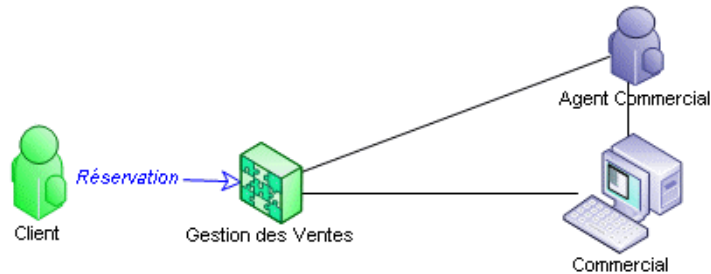
- les *réseaux* qui relient les postes clients aux serveurs et qui véhiculent les informations.
- les *serveurs* de chaque site sur lesquels les applications et les bases de données sont installées.
- les *postes de travail*, ainsi que les applications qui fonctionnent dessus, et les bases de données qui y sont installées.
- les autres *nœuds* du réseau, tels que les imprimantes, modems, etc., qui peuvent être connectés entre eux ou à des serveurs.
- les sites d'implantation des réseaux, des serveurs, des postes de travail et des autres matériels.
- les flux d'information échangés par les applications, les bases de données, les acteurs externes. Ces messages peuvent circuler sur les réseaux.
- les *acteurs* externes qui émettent des messages pris en compte par les applications ou en reçoivent.



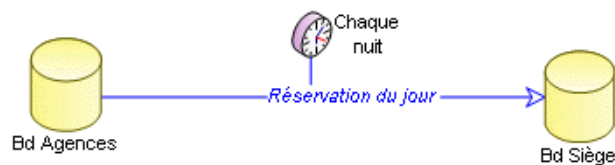
Éléments organisationnels : acteurs, etc.

Les diagrammes d'infrastructure technique peuvent également être enrichis par des éléments d'organisation :

- Vous pouvez préciser, pour une application disponible sur le réseau, les acteurs qui l'utilisent. Le poste de travail mis à disposition d'un acteur devra comporter l'accès à cette application.

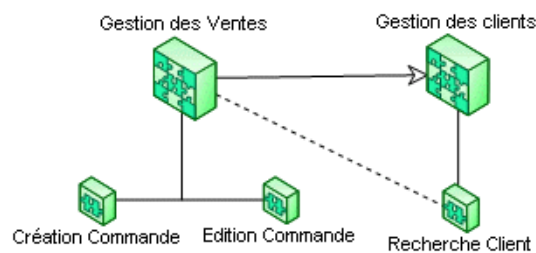


- Vous pouvez temporiser les messages dans le cas de transfert de fichier.



Répartition des traitements

Répartition des traitements entre une application cliente et une application serveuse. Vous pouvez préciser quels services définis dans l'application serveuse sont utilisés par l'application cliente.



Spécialisation

Vous pouvez définir des sous-types d'acteur, site, et message.

Les principales utilisations des DIT

Selon l'étendue du champ de l'étude, l'approche du système d'information sera générale ou détaillée : étude globale de l'architecture du SI, des modes de communications entre les différents sites de l'entreprise, ou focalisation sur un site particulier pour décrire l'infrastructure technique de cette partie du système d'information.

L'approche globale correspond à une cartographie des implantations de l'entreprise et présente les modes de communication intersite, en mettant en évidence les réseaux, les serveurs utilisés ainsi que les applications et les bases de données émettrices ou réceptrices de ces échanges d'informations.

L'approche détaillée centrée sur la description d'un site particulier permet d'étudier et de représenter l'infrastructure technique du système d'information local, et les principales liaisons avec l'extérieur. Elle fait apparaître les réseaux locaux, les serveurs supportant les applications et les bases de données, les postes de travail connectés et les différents matériels divers utilisés (imprimantes, etc.).

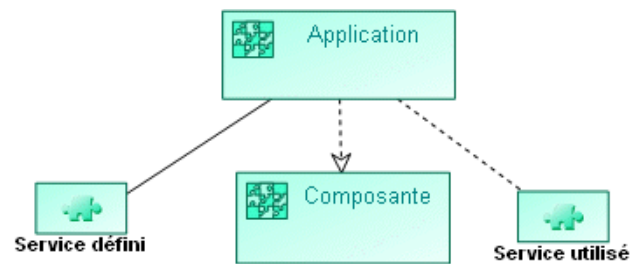
L'approche détaillée permet également de s'intéresser à la description d'un réseau local particulier et l'ensemble des équipements connectés et des applications et bases de données installées.

L'ARBRE APPLICATIF

L'Arbre Applicatif est un diagramme de **HOPEX IT Architecture** qui permet la description de l'architecture logicielle d'une application. Focalisé sur une *application* particulière, ce type de diagramme permet une représentation synthétique de sa décomposition en modules (sous-applications) et en services élémentaires.

L'Arbre Applicatif montre :

- l'application principale dont on étudie la décomposition.
- les modules applicatifs (sous-applications) qui composent l'application principale.
- les services définis dans le cadre de cette application, ainsi que ceux utilisés par cette application mais gérés par une autre application.



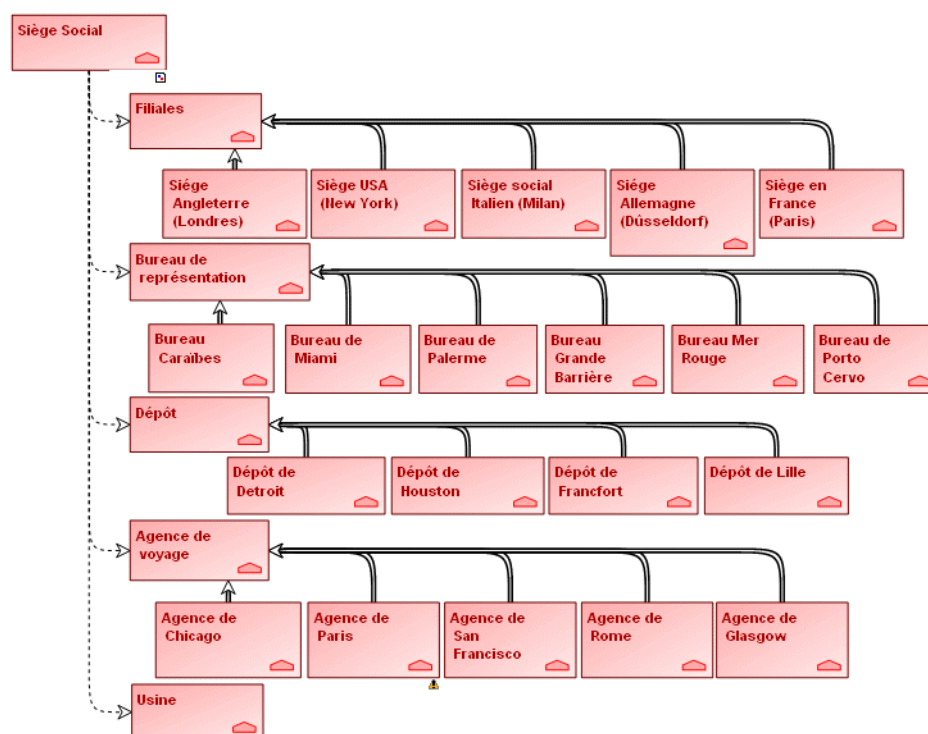
Arbre Applicatif

L'ARBRE DE SITES

L'arbre de sites est un diagramme de **HOPEX IT Architecture** qui permet une description arborescente d'un site. Focalisé sur un *site* particulier, ce type de diagramme permet une représentation synthétique de sa décomposition en sous-sites.

L'Arbre de site montre :

- le site principal dont on étudie la décomposition
- les sous-sites qui composent le site principal
- les spécialisations géographiques de chaque type de site



Arbre de sites

VUE D'ENSEMBLE DES SITES ET BILAN DES FLUX SITES

La vue d'ensemble des sites présente le bilan des flux entre les principaux sites de l'entreprise.

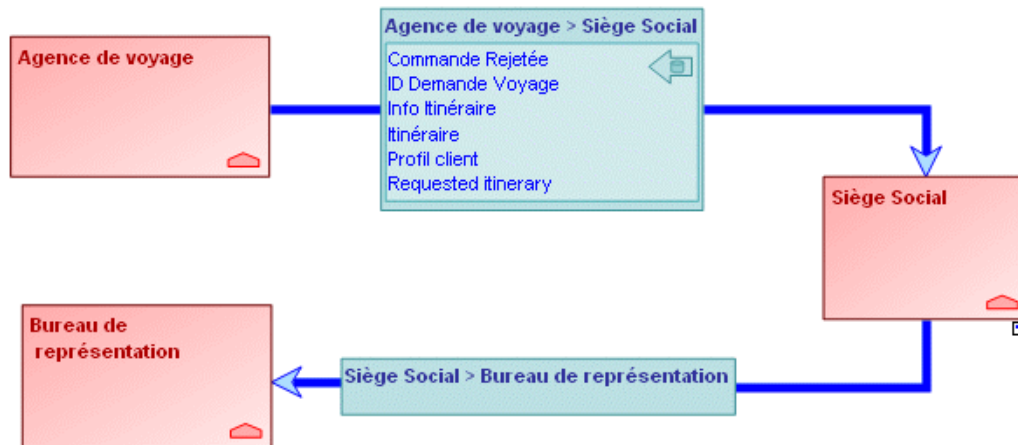
Pour un site donné, le bilan des flux entre sites présente un bilan des flux échangés entre ses composants et/ou d'autres sites.

Ce bilan est réalisé au moyen de canaux qui recensent les flux échangés entre les éléments (applications, services, ...) hébergés par chaque site.

Ces diagrammes montrent :

- les sites qui échangent des données
- les canaux d'échanges entre ces sites

Les canaux sont remplis automatiquement avec les contenus des messages échangés entre les applications, les services, etc. hébergés par chacun des sites.



Bilan des flux entre sites

CONCEPTS GÉRÉS PAR HOPEX IT ARCHITECTURE

Les schémas suivants présentent le formalisme d'utilisation de chaque objet (concept) géré par **HOPEX IT Architecture**.

Ils mettent en évidence l'exhaustivité des liens avec différents autres objets qu'il est possible de représenter dans des diagrammes.

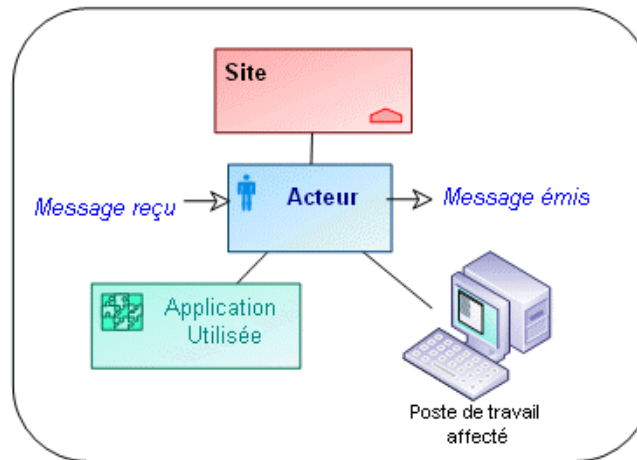
Comme vous l'avez vu dans le paragraphe précédent, l'utilisation de certains objets est limitée à un type de diagramme particulier (Noeud dans le DIT).

Par ailleurs, certains objets ne sont accessibles dans un diagramme qu'à travers des vues spécialisées privilégiant, selon les cas, une vision logicielle, organisationnelle, conceptuelle ou technique du système d'information.

Enfin, **HOPEX IT Architecture** est capable de gérer une hiérarchie d'objets de même nature, pour prendre en compte des notions de :

- **Composition** : un objet se définit comme la somme de plusieurs autres objets de même nature, mais dont le périmètre est plus réduit. Par exemple, l'application "Réservation Agence" est composée de N sous-applications (x, y, z,...).
- **Spécialisation** : un objet générique peut être décrit plus finement à travers plusieurs objets de même nature, mais plus spécialisés (des cas particuliers ou des sous-types). Par exemple, l'agence de voyage de Lyon est une spécialisation (un cas particulier) de l'agence-type décrite dans la plupart des diagrammes généraux ; une "grosse agence de voyage" est une spécialisation (un sous-type) de l'agence-type décrite dans la plupart des diagrammes généraux.

Acteur



Environnement d'un acteur

Un **acteur** correspond généralement à un élément de la structure de l'entreprise.

Un acteur :

- peut être localisé sur un ou plusieurs sites.
- émet et reçoit des messages.
- utilise une ou plusieurs applications.
- peut se composer de sous-acteurs et avoir des sous-types.

Exemples

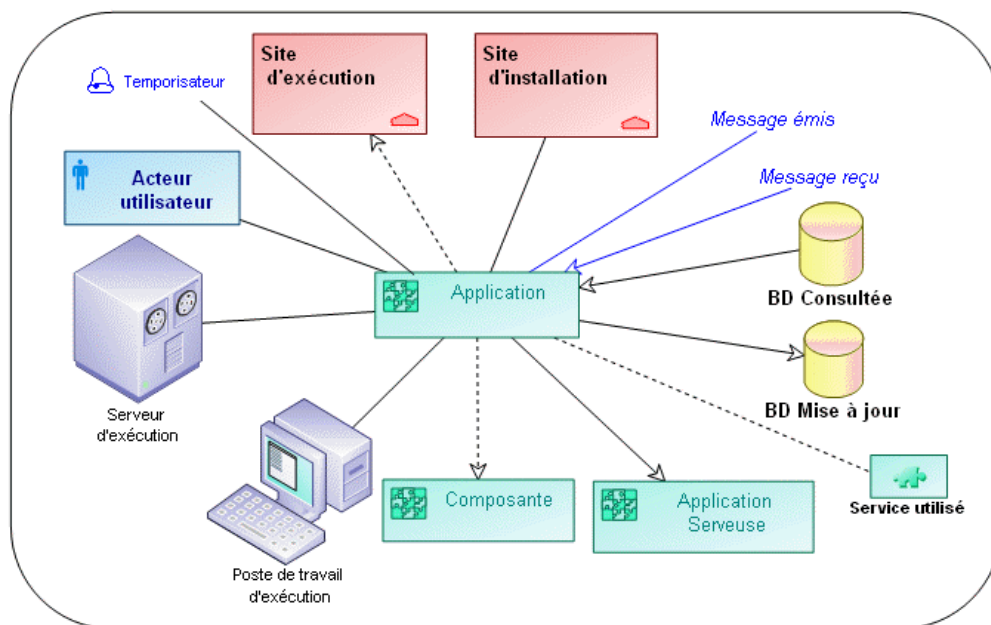
- Vendeur, secrétaire commerciale, chef de rayon (postes de travail).
- Comptabilité, facturation (services).

L'utilisation de cet objet est possible dans la vue "Acteur" du DAA et du DIT.

➡ *Un acteur peut être de type externe. Dans ce cas, il s'agit d'un organisme qui effectue des échanges de flux (Messages) avec l'entreprise.*

Exemples : "Client", "Fournisseur", "Banque".

Application



Environnement d'une application

☛ Une application est un ensemble de composants logiciels qui constituent un tout cohérent au regard des développements informatiques.

Une **application** émet et reçoit des messages, en principe en provenance et à destination d'autres applications. Ces échanges sont spécifiés dans les Diagrammes d'Architecture Applicative.

Une application :

- est utilisée par des acteurs sur les sites où elle est installée.
- s'exécute sur un ou plusieurs sites, postes de travail, serveurs, ou nœuds.
- peut être découpée en plusieurs applications composantes.
- peut déclencher l'exécution d'une application serveuse.
- consulte ou met à jour une base de données.
- définit un certain nombre de services.
- utilise des services définis dans le cadre d'autres applications.

Vous pouvez associer à une application :

- un Diagramme d'Architecture Applicative
- une Infrastructure Technique
- un Arbre Applicatif

Cet objet est accessible dans la vue standard du DAA, du DIT et de l'AA.

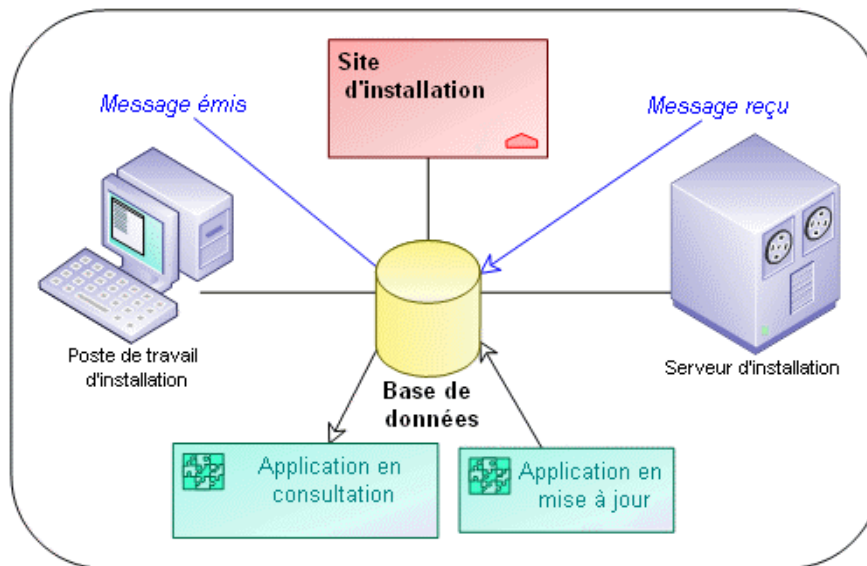
La description d'une application peut être précisée par son type :

- Type-application : il permet d'indiquer s'il s'agit d'un progiciel, d'un développement spécifique, d'un logiciel système (Windows NT, etc.), d'un middleware ou d'un logiciel de bureautique (Word, Excel, etc.).

Exemples

- Application "Crédit" dans le secteur bancaire.
- Application "Gestion des stocks" dans la distribution.

Base de données



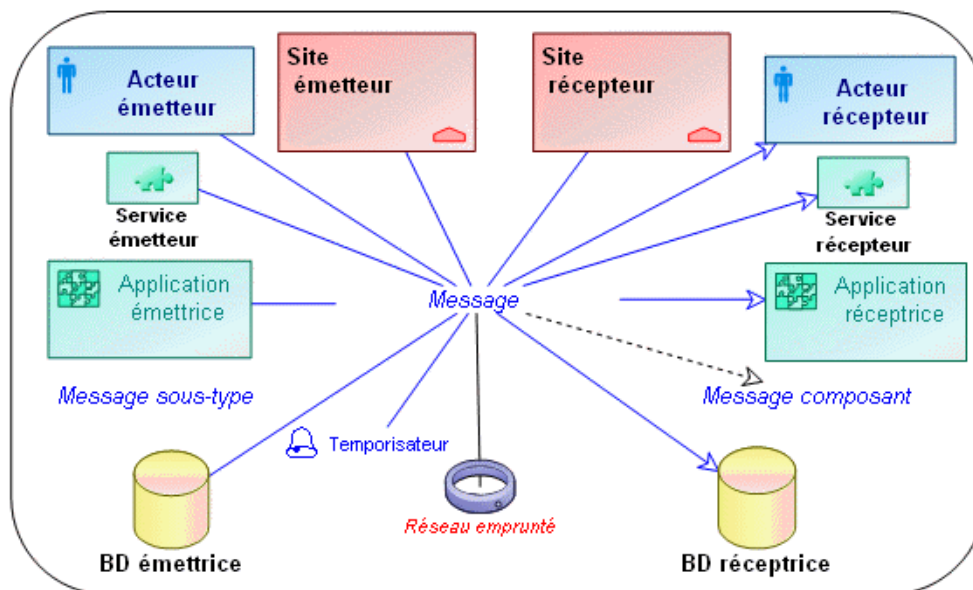
Environnement d'une base de données

Une **base de données** est localisée dans un site et installée sur un poste de travail ou un serveur ; elle est consultée et mise à jour par des applications ou des services. Elle peut être mise à jour par des flux d'information (message reçu) et on peut en extraire des données (message émis).

Cet objet est accessible dans la vue standard du DAA et du DIT.

☛ Une base de données permet de spécifier la structure de stockage logique ou physique des données.

Flux d'information (Message)



Environnement d'un message

Un message représente un flux circulant à l'intérieur de l'entreprise ou échangé entre l'entreprise et son environnement. C'est généralement un flux d'information comme une commande ou une facture. Par commodité, un flux financier comme le règlement du client, ou un flux de matière comme la livraison d'un produit est également représenté par un message.

Un **message** peut être émis ou reçu par :

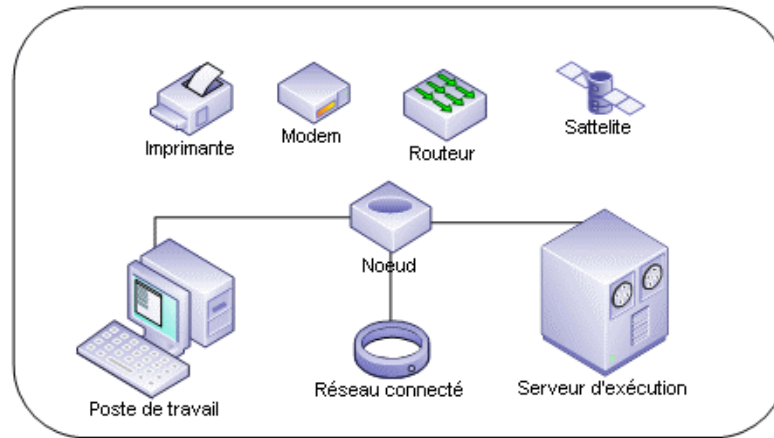
- un acteur
- un site
- une application
- un service
- une base de données

Il peut, en outre, être relié à un temporisateur, c'est-à-dire avoir une fréquence de circulation prédéterminée. On peut aussi préciser quel réseau il emprunte.

L'émission d'un message peut être conditionnée par la vérification d'un prédicat.

Cet objet est accessible dans la vue standard du DAA et du DIT.

Noeud



Environnement d'un noeud du réseau

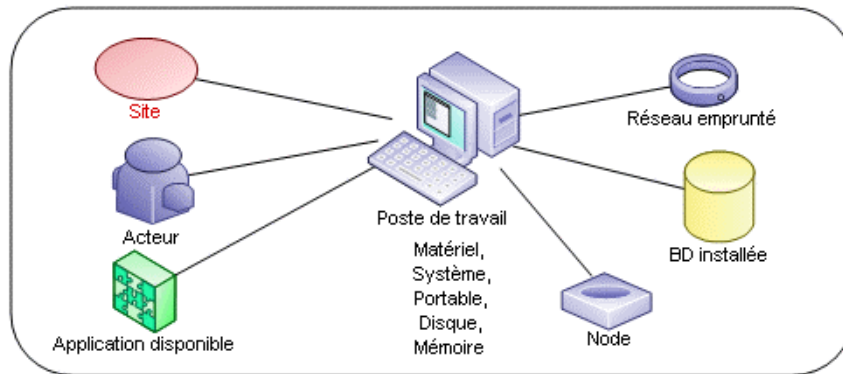
Un noeud est un élément du réseau tel qu'une imprimante, un modem, un routeur, etc.

Un *noeud* peut être connecté au réseau, ou bien directement à un poste de travail ou un serveur.

L'utilisation de cet objet est limitée au DIT, dans la vision standard du système d'information.

Un nœud peut être caractérisé par le type d'équipement dont il s'agit : routeur, imprimante, pont filtrant, modem, etc. La liste prédéfinie des types de nœuds peut être étendue par l'utilisateur.

Poste de travail



Environnement d'un poste de travail

Poste type correspondant à un profil d'utilisateur.

Dans le DAA ou le DIT, la définition des postes de travail nécessite de décrire les machines qui seront affectées aux acteurs de l'entreprise. Il est possible d'indiquer les applications et les bases de données qui fonctionnent sur un poste. Ces informations permettront de dimensionner de manière adéquate ces machines.

Le réseau qui permettra de connecter ensemble les différents postes de travail et les serveurs est également indiqué.

On peut aussi préciser les nœuds du réseau tels que imprimantes, modems, qui seront directement connectés à ce poste de travail.

Cet objet est accessible dans la vue standard du DIT.

Dans un DAA, l'utilisation de cet objet est possible dans la vue technique du système d'information.

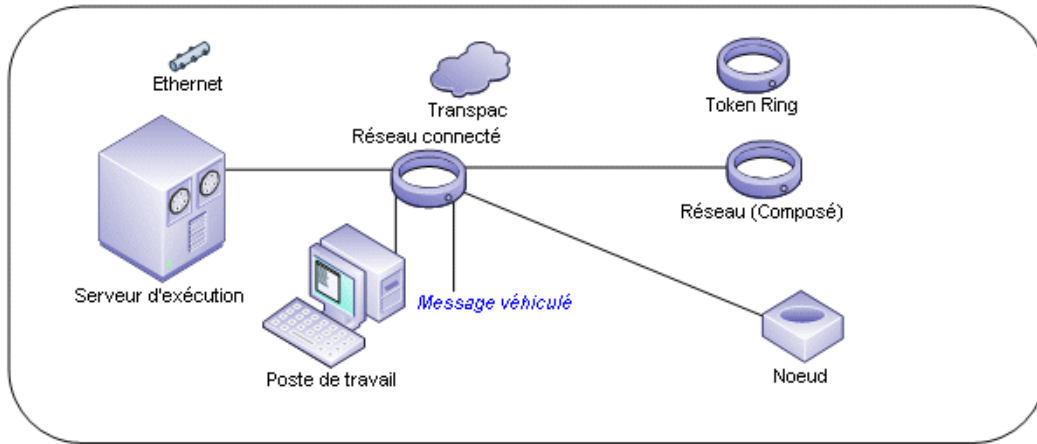
Projet

Un projet est une partie d'un système dont l'étude est confiée à une même équipe.

Un *projet* peut être décomposé en projets composants.

Un projet peut avoir pour objet la modélisation d'une application.

Réseau



Environnement d'un réseau

Un réseau est un système d'ordinateurs géographiquement éloignés les uns des autres, interconnectés par des télécommunications, généralement permanentes.

Un **réseau** peut être local à un site de l'entreprise. Il permet de connecter les serveurs et les postes de travail, dans le DAA ou le DIT.



Les messages qui circulent sur le réseau entre les différentes applications ou qui sont émis par des acteurs externes à l'entreprise sont également précisés.

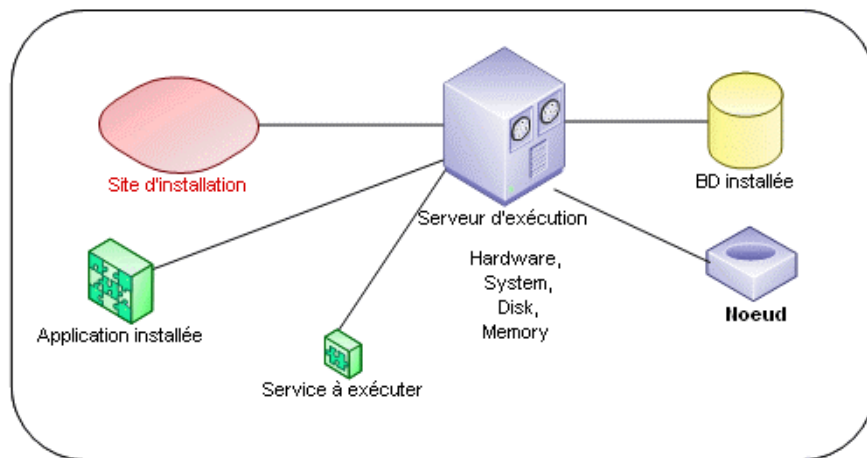
Les différents noeuds du réseau (modems, imprimantes, etc.) sont décrits.

Il est possible de définir la composition du réseau logique (c'est-à-dire le protocole) en réseaux physiques (topologie).

Cet objet est accessible dans la vue standard du DIT.

Dans un DAA, l'utilisation de cet objet est possible dans la vue technique.

Serveur



Environnement d'un serveur

Un serveur est une ressource informatique matérielle, pouvant disposer d'une Base de données et sur laquelle des Applications peuvent s'exécuter.

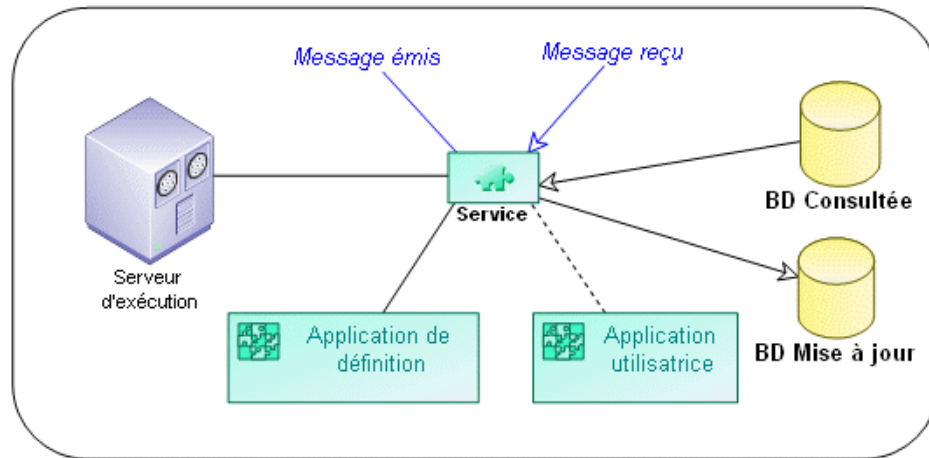
Dans le DAA ou le DIT, il est possible de préciser sur quel *serveur* une application tourne, ainsi que l'implantation géographique de ce serveur sur un site, et la ou les bases de données implémentées sur ce serveur. On peut préciser le type de machine à utiliser, ainsi que le système d'exploitation, et la capacité mémoire et disque nécessaires, ainsi que les nœuds du réseau (modem, routeur, etc.) directement connectés au serveur.

Il est également possible d'indiquer explicitement qu'un service s'exécute sur un serveur différent de celui où s'exécute l'application à laquelle il appartient.

Cet objet est accessible dans la vue standard du DIT.

Dans un DAA, l'utilisation de cet objet est possible dans une vue spécifique.

Service



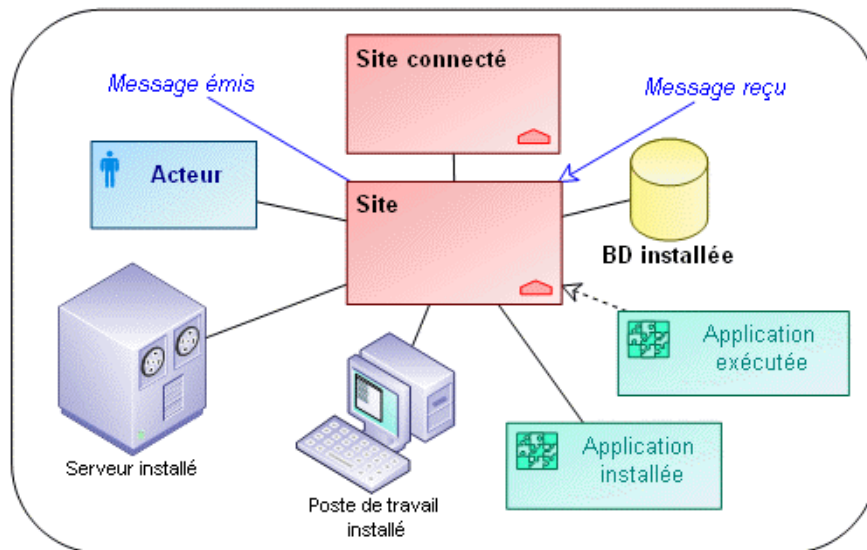
Environnement d'un service

Un service 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.

On peut préciser dans chacun des diagrammes de **HOPEX IT Architecture** quels **services** sont définis à l'intérieur de chaque application et ceux qui sont réutilisés dans le cadre d'une autre application.

On peut indiquer les bases de données consultées ou mises à jour par ces services, les messages qu'ils émettent ou qu'ils reçoivent ainsi que le serveur sur lequel ils s'exécutent s'ils ne s'exécutent pas sur le même poste de travail que l'application dont ils font partie.

Site



Environnement d'un site

Un **site** correspond à un lieu géographique d'intérêt pour l'organisation :

- Parce qu'une partie de l'activité de l'entreprise y est effectuée, ou
- Parce que des informations y sont stockées : données informatiques, mais aussi archives papier ou autres supports d'information.

Un site peut être composé de sites et avoir des sous-types. Son organisation est fournie par les acteurs qui y sont rattachés. Il peut recevoir et émettre des messages. Un site peut être connecté à d'autres sites. Une application peut être utilisée sur un site et être exécutée sur un autre.

Il est possible d'associer à un site :

- Une Infrastructure Technique

Cet objet est accessible dans la vue standard du DAA et du DIT.

Temporisateur



Environnement d'un temporisateur

Un *temporisateur* décrit la fréquence :

- de déclenchement d'une application batch, ou
- de circulation d'un message.

Exemples :

Hebdomadaire

Deux fois par mois

Toutes les 100 factures

Premier jour ouvré du mois

Virtualization and Clustering Modeling

This paper proposes an approach to model virtualization and clustering based on the infrastructure metamodel supplied from the 2009 SP2 release of the Mega modeling tool.

Introduction

What is Virtualization?

Virtualization refers to the abstraction of computer resources. This technique allows the virtually creation of servers, the installation of software packages on these virtual servers and the running of the installed software packages as if they were located on an actual machine. The technique was primarily used to minimize space allocation necessary to store servers, to reduce electricity power the servers consume and so to save money. Other perspectives are the ability to balance from one configuration to another depending on the current load in a completely transparent manner for the end-user.

Need for Virtualization Modeling

The virtualization has progressed so that complete technical environments can be virtually created (servers, networks, sets of application...). On the other hand it becomes more and more difficult to understand where applications and servers are actually located and then to evaluate risks for the business if any technical item fails.

This is the reason why there are more and more requests for virtualization modeling and the ability to deduce impact of virtualized item to the business.

What is Clustering?

A computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks and share common storage means. Clusters are usually deployed to improve performance and/or availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Since the group of clustered computers is viewed as a single one by the end-user the clustering technique can be considered similar to the virtualization technique. But the slight difference here is all the machines may exist ... or be virtual. In this context, the performance improvement is due to the grouped computation capabilities while the virtualization objective is to transport end-user machines to the most appropriate location in real-time.

Need for Clustering Modeling

Clustering leads to the same questions as virtualization does: What are the impacts of a server failure on my business? Is this actual server really useful for my tasks? And so on. To answer such questions, the modeling of clustering is also necessary because the business features is attached to the apparent server while the supporting applications are hosted by the clustered servers.

Prerequisites

The purpose of this article is to explain what can be some practices to model both virtualization and clustering. They are two examples of technical infrastructures that can be supported thanks to the resource architecture metamodel introduced in the 2009 SP2 release. The two examples will be developed based on this metamodel and so the sole prerequisite for such implementation is the use of the MEGA 2009 SP2 release. However, the use of a MEGA 2009 SP3 release is recommended because a larger number of items has been included in the MEGA Architecture add-on and among these items the “Virtual Server” artifact that will be used later in this paper.

In order to benefit from this new metamodel, the option “Business Process and Architecture Modeling / Infrastructure Modeling post-2009 SP2” must be activated (Refer to the “Tool / Option” menu command).

For a better understanding of this article, the reader must have some knowledge regarding architecture modeling and the practice of the MEGA Modeling tool. More information can be retrieved on the topics described below in the MEGA Architecture user guide.

The Resource Architecture Metamodel

This section explores the main concepts of the resource architecture metamodel introduced in the MEGA 2009 SP2. The following table lists some concepts of these metamodel. For more explanations, refer to the MEGA Architecture guide.

Concept	Comment
Resource Architecture	A resource architecture is a combination of physical assets and organization configured to provide a capability.
Artifact	An artifact is any element in the physical domain that is not an application or an organizational element (where organizational includes people). An Artifact can represent a physical system, sub-system, platform, component or simply a physical item that has specific attributes.
Application	An application is a set of software tools as seen from the software development viewpoint.
Org-Unit	An org-unit represents a person or a group of persons that intervenes in the enterprise business processes or information system. An org-unit can be internal or external to the enterprise. - An internal org-unit is an organizational element of the enterprise structure such as a department, a service, or a workstation. An internal org-unit is defined based on how detailed you require your view of the enterprise to be (cf org-unit-type). Example: financial management, sales management, marketing department, account manager. - An external entity is an organization that exchanges flows with the enterprise. Example: Customer, Supplier, Government Office.

These four concepts help to model resource architectures that describe solutions designed to support a given capability. An architecture design explains how components are grouped together and communicate each other. The designed solutions include both the technical aspect (equipment and software) and the human resources. In this context, the architecture is composed of items

exposed above and is more focused on the infrastructure part so only the artifact/application part is detailed.

In order to enable the description of interaction in the context of a solution (resource architecture) each inserted item is represented by an intermediate concept referencing it. The figure hereafter shows these intermediate items. This mechanism separates the interactions that equipments or applications perform in more than one context (refer to the *“Artifact Type and Instance”* article for more details on this topic).

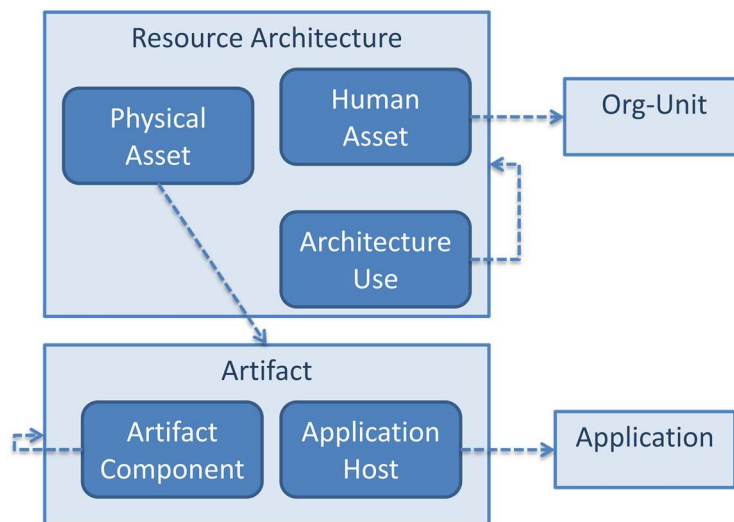


Figure 1: The Concepts Introduced for the Modeling of Technical Architecture Items.

Modeling Virtualization

A first attempt to enable virtualization modeling has been initiated in the MEGA 2007 release. The objective was to enable the identification of virtual servers and the application “generating” such servers. So, an association has been added that links applications to servers (instance of the metaclass Server). This link can be drawn in a technical infrastructure diagram and so belong to the technical infrastructure metamodel introduced prior to the 2009 SP2 release.

The link was sufficient to express facts such as: “the VMWare application creates this virtual server”. Servers linked to an application can be identified as virtual servers since they are generated by a virtual machine application. Since the virtual server is a server regarding the type, it can be used as any other servers to host applications.

This principle has not been reintroduced in the metamodel proposed from the 2009 SP2 release for several reasons:

- The simple link between applications and servers is not enough to model all virtualization configurations. Depending on the context, virtualization can be handled by an application, an operating system, a dedicated server and even a cluster of servers. In these cases, numerous associations should be necessary and even an intermediate object to group the potential virtual server generators.
- The new infrastructure metamodel is not defined at the same level. Servers are not described as instances of a dedicated metaclass but as artifacts inheriting from a reference

Server artifact. This new design principle induces that a link between artifacts is available for any artifact even if virtualization has no meaning for some of them.

So, the question now is how to model virtualization based on the Artifact metamodel? Two facts are relevant in such modeling:

- This server is virtual
- This virtual server is generated by this technology (operating system, dedicated servers...).

Identifying Virtual Servers

To model the first topic, one way consists of starting from an existing artifact. The MEGA Architecture add-on contains some artifacts that can be used as starting points. Among them there is the *Server* artifact. From the 2009 SP3 release, the add-on contains also a *Virtual Server* artifact that inherits from the *Server* artifact. Every new virtual server is then a new artifact inheriting from the *Virtual Server* artifact. The variation mechanism can be used to create a family of virtual servers and so refine their definitions.

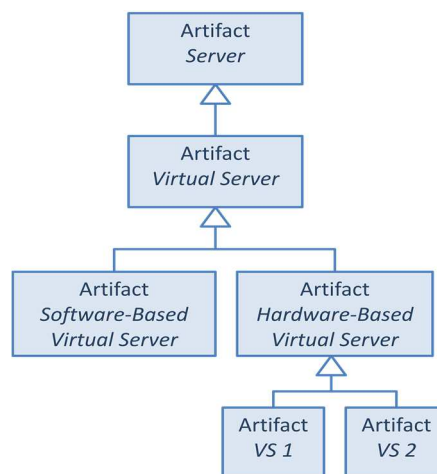


Figure 2: Example of Virtual Server Taxonomy.

This first step allows the identification of the virtual server. Once created, they can be retrieved using a query on variable objects (or more precisely on artifacts). Such a query, named “*Variants of an Object*”, is supplied from the MEGA 2009 SP3 release. The query looks like:

Select [Variable Object] Where [Variant Side Variation].[Varied Object] Deeply = &"Variant"

Processing the query will ask the user for a variable object. Then the *Virtual Server* artifact is selected and the query lists all the virtual servers in the repository. If a more specialized virtual server is selected, the query lists the corresponding subset of virtual servers.

The same identification principle can be applied to any artifact. Indeed, the virtualization technique is available for servers but also for more complex architectures. For example, a virtual network can be created or a complete virtual architecture including a network and a set of servers. For these two examples, a *Virtual Network* artifact is designed as a starting point for all virtual servers while a *Virtual Architecture* instance of the Resource Architecture concept matches the starting point of all

virtual architecture items. Then, variants of the virtual architecture are described with their own components.

Using Properties to Model Virtual Servers

Another way to identify virtual servers is simply by the addition of a customer property on the *Server* artifact. For example, the property type can be named *Virtual* and the available values for this property can be *yes* or *no*.

The advantage of this simple proposal is that the type can be reused for different artifacts (server, network ...) or resource architecture. The research of virtual servers is deduced from the previous query but adding the property value condition. Such a query looks like:

```
Select [Artifact] Where ([Variant Side Variation].[Varied Object] Deeply Like "Server") And  
([owned property].([Property Value]="yes") And  
([Property Type].ShortName Like "Virtual"))
```

There are at least two drawbacks with this proposal:

- The query to search for virtual item is more complicated and the ERQL request written above does not manage all the possible cases. Indeed, since the property values can be inherited from varied ancestors, the value is not always connected directly to the artifact. In consequence the query must be updated to take this inheritance into account. This is not the purpose of this article to expose how to go through inherited properties but the fact is some code must be used instead of the ERQL syntax to perform a complete search (vb script, javascript or java code, refer to the MEGA API).
- The second point is that the proposal does not allow the definition of taxonomy of virtual servers has the variation does and then to simply ask for servers, virtual servers, hardware virtual servers... Instead, each sub-family should be implemented thanks to additional properties making the search tool again more complex.

For the reasons exposed here, the use of a dedicated artifact is recommended compared with a solution based on properties.

Including Virtual Servers in an Architecture

Whatever the way the virtual servers are created, they result to some artifacts that can be used in any solution architecture or embedding artifacts describing what are the environments letting the virtual servers available. The common example is some virtual servers hosted by a physical server. An approach can be described by the following steps:

1. Create the virtual servers as previously explained adding the applications exposed by these virtual servers.
2. Create the physical server from the *Server* artifact.
3. Insert the virtual servers in the physical one as Artifact Components.
4. Establish the links between the virtual servers to explain how they interact - if they do so - in the context of the physical server. Such links can be communication channels with communication protocols.
5. Detail the applications that the physical server hosts. Among these applications there is probably some virtualization tools (e.g. VMWare). At this point, if the virtualizing tool is

considered as a platform technology it is more appropriate to model it based on an artifact as are operating systems. One can create the corresponding application later and insert it in the platform technology as an *Application Host*.

Figure 3: An Example of Some Virtual Servers.

In the example above, two servers are available for the end user: *VS 1* and *VS 2*. They are both virtual servers hosting business application (e.g. Invoice). The infrastructure supporting the end-user servers is composed of a physical server *Server XYZ* that generates the virtual server and hosts an instance of the *VMWare* application. The implementation of such design in the MEGA 2009 SP3 Release is based on the *Artifact Assembly Diagrams*. These diagrams are very simple from a graphical representation perspective and look like the following:

Figure 4: The Artifact Assembly Diagram of the *Server XYZ*.

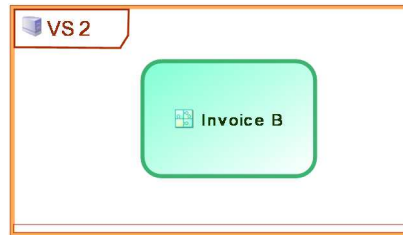


Figure 5: The Artifact Assembly Diagram of the Server VS 2.

The principle exposed here can be replicated to any virtual structure. A complex virtual architecture composed of a network, n servers and some hosted applications is described the same way: the physical server contains all the virtual items including the network, the virtual servers and the applications that are all artifacts or applications.

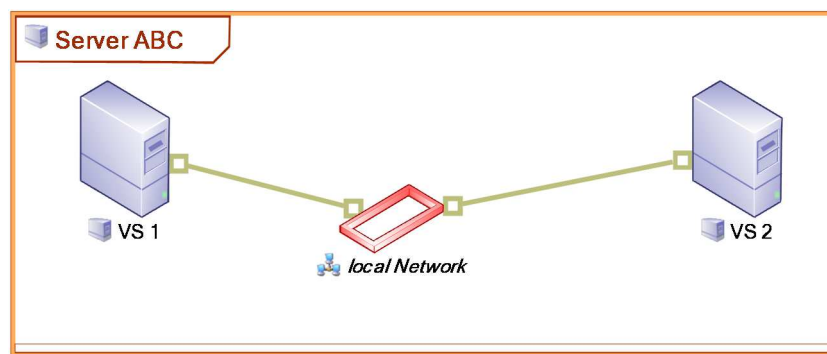


Figure 6: An Example of a Complex Virtual Configuration Hosted by the Server ABC.

Modeling Clustering

The modeling of clustering is based on the same principle proposed for the virtualization. A cluster is a group of servers that share common resources to run an expected application faster than each individual server is able to do. For the end-user, there is only one server and only one application he wants to use.

In this context, the modeling approach is the following:

1. Create all the physical servers that are grouped into a cluster. Each server inherits from the *Server* artifact.
2. Create the cluster as an artifact inheriting from the *Server* artifact.
3. Include all the primary physical servers in the cluster as Artifact Components.
4. Add the applications hosted by the cluster that are available for the end user.
5. Add the potential applications of each individual server that are useful for the clustering.
6. Create the links between the physical servers that explain what the communication means are and what is exchanged in the context of the cluster.

Figure 7: Example of Clustering Model.

In the example above, the cluster server *XYZ* hosts the application *Invoice* from the end-user perspective. Depending on the current load, the application is balanced either to *Server A* or *Server B* but this is totally transparent for the end-user. At least the model explains the architecture and then it is possible to deduce that the application *Invoice B* can be impacted by a failure of the physical server if the architecture describes some communications channel between the application and the servers.

Mixing Virtualization and Clustering

Finally, it is possible to have configuration including virtual items and clusters. For example, one can create a cluster of virtual servers ... or some virtual cluster of servers! Based on the proposed approach there is no problem to design both these combinations. Virtual servers are simply grouped in a cluster server and in a physical server. The reusability of the artifacts is powerful in this context and it is not mandatory that the virtual and cluster group contains the same items.

Figure 8: A Cluster of Virtual Servers.