

# *Architecture d'Entreprise : Guide Pratique de l'Architecture Logique*

Auteur : Equipe Conseil Modeliosoft  
Version: 1.0  
Copyright: Modeliosoft

## **Modeliosoft**

21 avenue Victor Hugo  
75016 Paris

[www.modeliosoft.com](http://www.modeliosoft.com)

## Introduction aux Guides Pratiques

Les Guides Pratiques sont issus de l'expérience des consultants de Modeliosoft et destinés à faciliter la construction de modèles en bénéficiant des capacités de l'atelier Modelio. Ils sont délibérément courts, pour fournir l'essentiel de la pratique en peu de pages. L'équipe conseil de Modeliosoft est à votre disposition pour vous assister dans vos travaux liés à la définition d'architecture d'entreprise, modélisation des processus métier, modélisation d'architectures logicielles, SOA, et assistance dans vos projets informatiques.

Modeliosoft vous propose une offre packagée conseil / outil. Pour plus d'informations, visitez [www.modeliosoft.com](http://www.modeliosoft.com).

Sous [www.modeliosoft.com](http://www.modeliosoft.com), vous pouvez librement télécharger Modelio Free Edition, ateliers gratuit, ergonomique et sans limitations pour la modélisation UML et la modélisation métier (Architecture d'Entreprise, BPM, architecture logique SOA, architecture logiciel).

Sous [www.modeliosoft.com](http://www.modeliosoft.com), vous pouvez également évaluer et acheter Modelio Enterprise Edition, pour bénéficier d'une grande richesse fonctionnelle : support du travail en équipe, analyse des objectifs, définition du dictionnaire et analyse des besoins, génération de code, génération documentaire sur l'ensemble du cycle de vie, et ainsi de suite.

Les Guides Pratiques disponibles sont les suivants :

- Guide Pratique des Cas d'Utilisation,
- Guide Pratique des Processus Métiers,
- Architecture d'Entreprise : Guide Pratique de l'Architecture Logique,
- Guide Pratique de la Modélisation de l'Organisation d'une Entreprise.

D'autres guides pratiques seront fournis prochainement – n'oubliez pas de consulter régulièrement nos sites.

## Qu'est-ce que l'architecture d'un système d'information ?

L'architecture logique s'adresse au système d'information, vu de manière macroscopique, en s'attachant à ses composants principaux, à leurs interconnexions et aux flux échangés, et en les structurant par regroupement dans des modules de plus grande taille.

*L'une des activités dans le domaine de l'architecture d'entreprise est l'organisation de la transformation progressive et continue du système d'information visant à le simplifier, à optimiser sa valeur ajoutée et à le rendre plus réactif et flexible vis à vis des évolutions stratégiques de l'entreprise, tout en s'appuyant sur les opportunités technologiques du marché.*

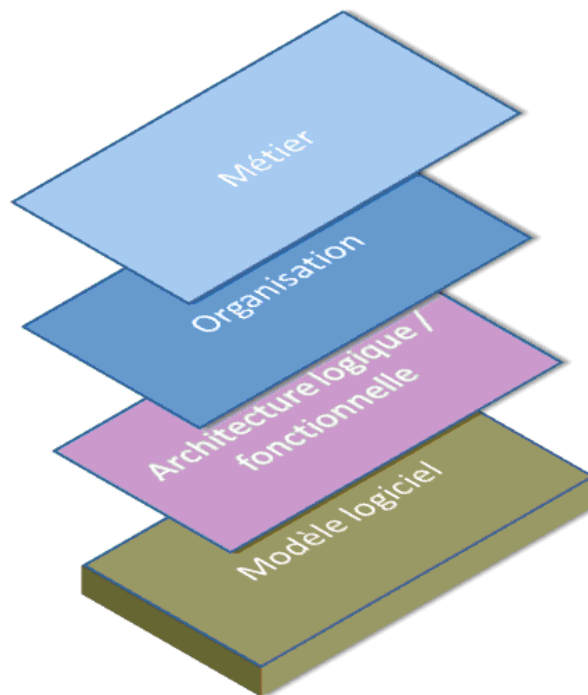
*La cartographie du système d'information fournit une vue globale des composants de celui-ci. Le résultat est un cadre cohérent, stable et modulaire, auquel les différentes parties prenantes se réfèrent pour toute décision d'investissement dans le système d'information.*

*A travers la cartographie du système d'information, on représente la cartographie applicative de l'existant, et la cible, c'est-à-dire l'architecture visée à l'issue d'une évolution programmée du SI.*

*D'autre part, l'architecture logique supportée par Modelio SOA Solution permet de représenter une architecture idéale, basée sur les principes architecturaux des architectures orientées service (SOA).*

On peut ainsi représenter l'existant, en répertoriant des composants applicatifs ne relevant en général d'aucune forme architecturale globale, et modéliser la cible en s'appuyant sur une démarche SOA, ou adopter une démarche hybride face à cette approche.

## Quand et comment l'articuler ?



**Figure 1 - Différents modèles pour l'entreprise et son SI**

La Figure 1 fournit le positionnement du modèle de l'architecture logique et fonctionnelle par rapport aux autres modèles et travaux autour d'un SI. Chacun de ces modèles fait l'objet d'un guide spécifique (voir le Guide du modèle de l'organisation). L'architecture fonctionnelle ou logique détermine les constituants du SI nécessaires pour assister le fonctionnement de l'entreprise. Le modèle de l'architecture logique/fonctionnelle est établi quand on veut comprendre, documenter, adapter ou améliorer l'organisation du SI. Typiquement, il permet de bâtir une compréhension commune du SI par le biais d'une carte générale, et de définir la cible des évolutions du SI par le biais d'une architecture souhaitée à terme. Une architecture logique idéale est indépendante de la technologie. On peut toutefois lui accrocher des indications techniques pour définir quelles technologies sont utilisées ou retenues pour chaque composant.

Un scénario idéal et théorique détermine ainsi la séquence :

1. Construire le modèle métier (Pour des conseils pratiques, voir [www.praxeme.org](http://www.praxeme.org) – guides de l'aspect sémantique).
2. Elaborer le modèle de l'organisation (voir le guide pratique du modèle d'organisation).
3. Cartographier et urbaniser le SI (c'est à dire, l'organiser et le rendre cohérent).
4. Bâtir l'architecture logique de la cible.
5. Réaliser, acquérir et assembler les composants applicatifs.

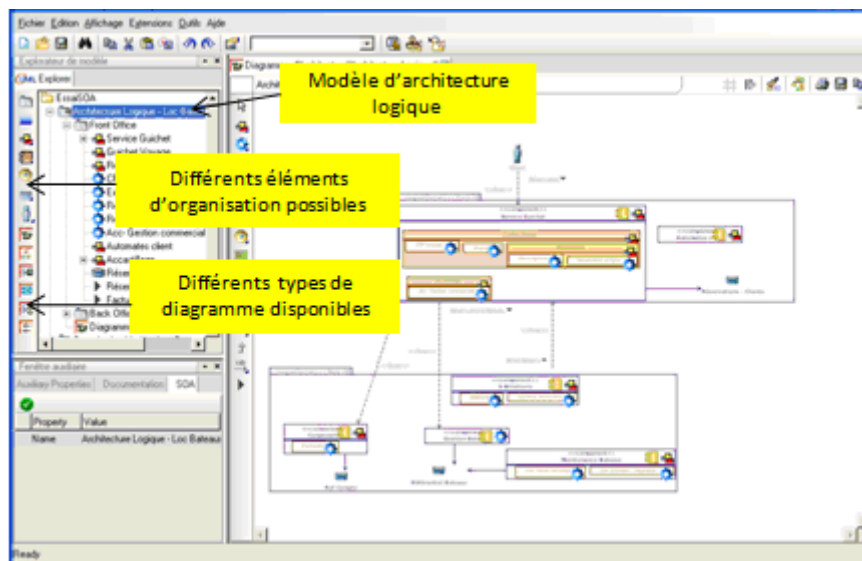
Les opérations 3 et 4 qui nous intéressent peuvent toutefois être entreprises en parallèle, voire en l'absence des modèles amont.

Enfin viennent les étapes de réalisation logicielle, où la MOE mettra en œuvre des modèles UML centrés sur le SI et sa réalisation, conformément à l'architecture logique cible.

## Bonnes pratiques

### Démarrer sous Modelio SOA Solution

Modelio SOA Solution permet de modéliser l'urbanisation et l'architecture logique. Dans l'explorateur, sélectionner le bouton "Créer Architecture Logique", et la racine du modèle d'organisation est créée. Sous cet élément, apparaissent dans la palette les éléments d'organisation que l'on peut créer.



**Figure 2 – Démarrer le modèle d'organisation sous Modelio SOA Solution**

Nous recommandons de réaliser la cartographie de l'existant, l'urbanisation de la cible et l'architecture logique sous trois modèles logiques différents (trois racines sous un même projet Modelio typiquement).

La technique de composition et d'assemblage des composants présentée Figure 7 permet de présenter différentes occurrences (instances) d'un même élément (composant, quartier, application, ...) sous différents modèles : ainsi, la cible peut reprendre des éléments de l'existant sans les redéfinir, l'architecture logique peut décomposer des applications identifiées dans la cartographie, etc.

## Construire la carte d'un SI

Construire une carte du SI (Système d'Information) consiste dans un premier temps à étudier les différents secteurs fonctionnels d'une entreprise (production, administration, ventes, etc.), afin d'être en mesure d'en réaliser une cartographie, puis d'étudier de la même manière son système d'information.

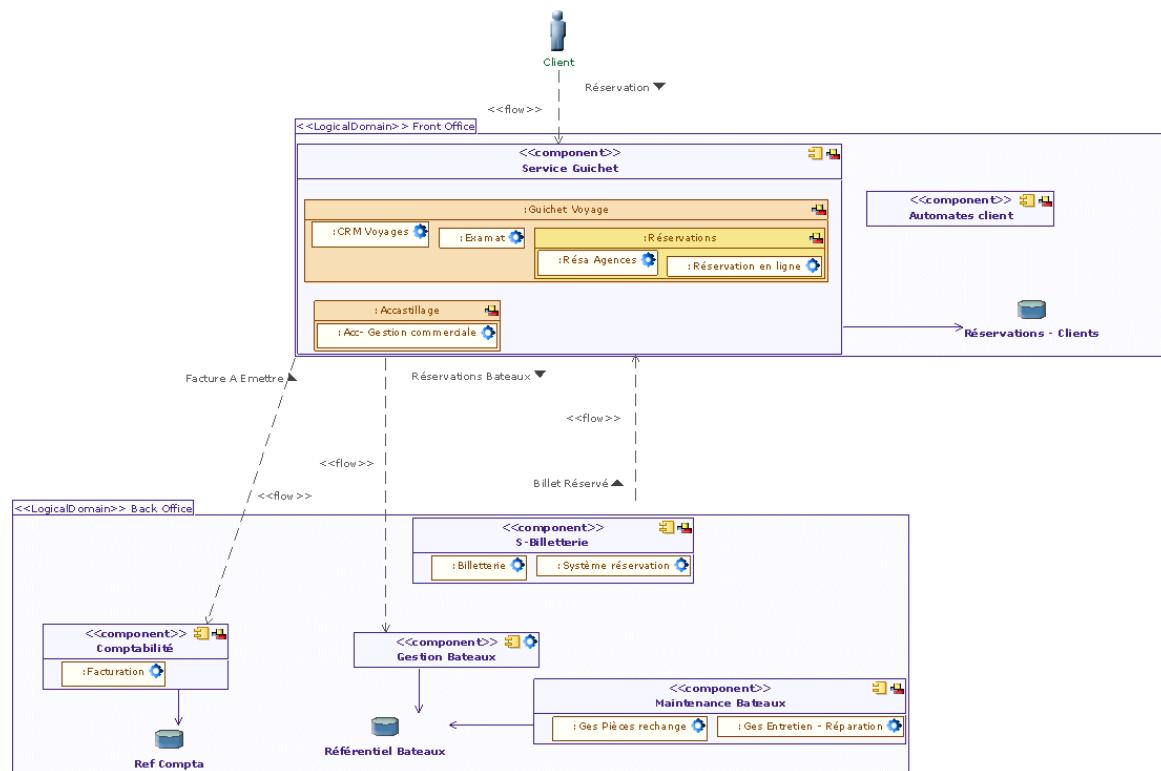
Une telle démarche commence par le recensement et la capitalisation de l'ensemble des informations sur le système d'information de l'entreprise (bases de données, applications, services, etc.), en relation avec leur fonction, afin de les rationaliser et de permettre de valoriser le capital informationnel de l'entreprise.

L'objectif d'une démarche de carte du SI est donc d'aboutir à une structuration du système d'information permettant d'en améliorer ses performances et son évolutivité. Elle permet ainsi de donner les moyens à l'entreprise de faire évoluer son système d'information en connaissance de cause.

Cette démarche s'appuie sur des niveaux de granularité de composition différents, tels que des systèmes, des sous-systèmes et des applications :

- Les systèmes, qui sont les grands domaines fonctionnels supportés par le SI,
- Les sous-systèmes, qui décomposent les grands systèmes en parties cohérentes,
- Les applications, qui correspondent aux éléments autonomes fournissant des services bien identifiés. La plupart du temps, celles-ci sont déjà identifiées.

Dans une approche plus générale permettant une transition vers une architecture SOA, Modelio SOA Solution supporte ces notions à travers celles de système (systèmes ou sous-systèmes), d'application et de composant de service à un niveau plus fin de granularité. L'imbrication entre composants (ex : l'application "Résa Agences" dans le système "Réservations", "Réservations" dans le système "Guichet Voyage", "Guichet Voyage" dans le système "Service Guichet") s'effectue en déployant une occurrence de chaque composant dans un composant de plus haut niveau. Sous Modelio, ceci fonctionne en créant une instance, selon le principe du déploiement décrit ci-dessous).



**Figure 3 - Exemple d'urbanisation avec Modelio SOA Solution  
(Diagramme d'architecture Modelio)**



## L'architecture logique

L'architecture logique occupe une place centrale : Instrument privilégié pour la construction et la maintenance du système, pivot sur lequel s'articulent le métier et sa traduction logicielle, elle constitue la référence pour l'organisation des projets, la construction technique et le plan de progression. Toutefois, l'architecture logique n'est pas un modèle abstrait, ni une cartographie fonctionnelle cible lointaine. Il s'agit plus pragmatiquement de la description des constituants du système et de leurs relations.

L'architecture logique décrit le système à un niveau plus détaillé que l'urbanisation : elle décompose les applications pour aboutir aux "composants de service".

L'architecture organisera les composants du système en couches, en s'appuyant sur une typologie liée à leurs niveaux de stabilité. Elle utilisera la notion de service pour garantir l'autonomie et l'interchangeabilité des composants. Il existe un consensus aujourd'hui pour bâtir les architectures de systèmes à partir d'une typologie de services bien établie, organisée en couches logiques. Schématiquement, les processus s'appuient sur un ensemble de services de plus bas niveau et d'accès aux données.

Les couches logiques de stabilité croissante établissent la règle de base de dépendance : un composant ne peut pas utiliser un composant d'une couche d'un niveau supérieur (par exemple, un composant Entité ne doit pas utiliser un composant Fonction ou Processus).

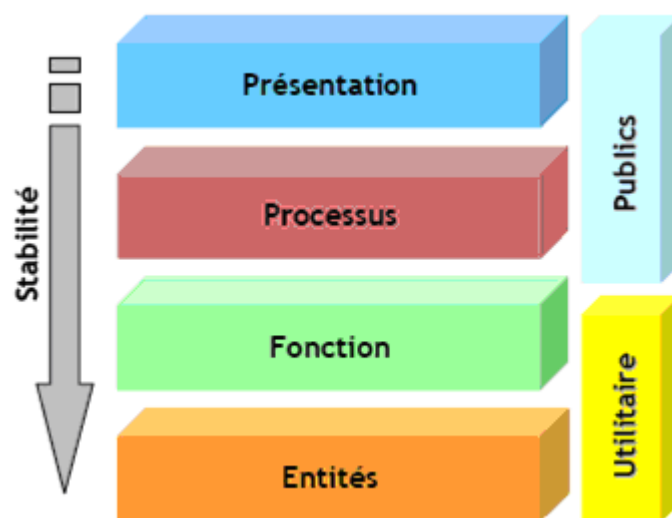
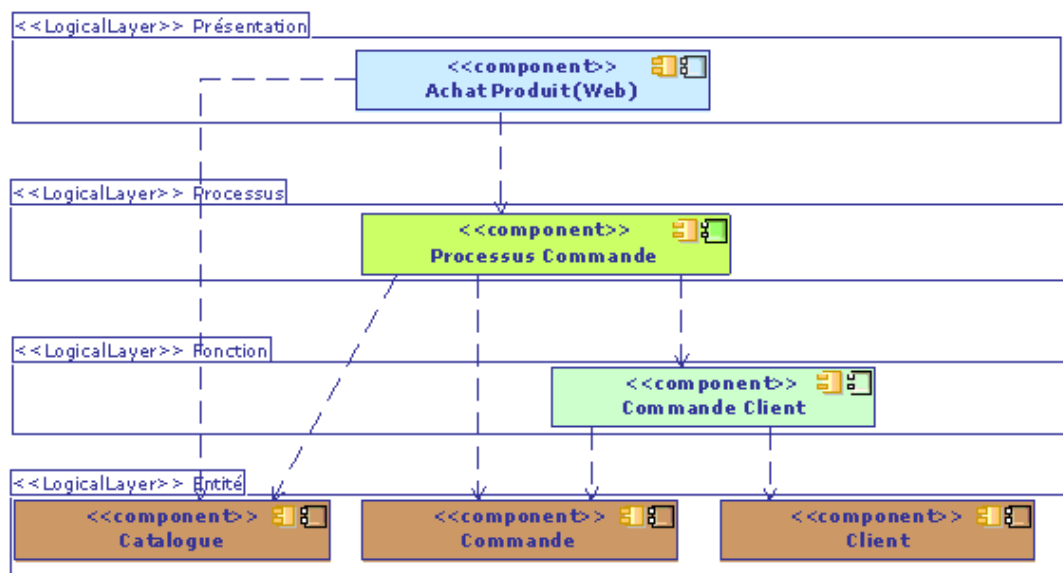


Figure 4 - Modèle en couches logiques




Chaque type de composant joue un rôle spécifique:

- Composant "*Présentation*" : Mise en œuvre du dialogue avec l'utilisateur : IHM, gestion de la session utilisateur (ceci n'est pas un composant de service à proprement parler).
- Composant "*Processus*" : Support de processus métiers complets (rôle d'orchestration); s'appuie notamment sur des composants de type "Fonction" et "Entité".
- Composant "*Fonction*" : Composition de services. Adaptations fonctionnelles ou traitements localisés.
- Composant "*Entité*" : Service d'accès aux données persistantes (CRUD12), aux bases de données et référentiels.
- Composant "*Utilitaire*" : Fournisseur de services d'infrastructure ou transversaux (messagerie, tableau de bord, éditique, annuaire).
- Composant "*Public*" : Dédié aux services accessibles à l'extérieur du SI (B2B, partenaires).



**Figure 5 – Architecture de composants de services: exemple (diagramme d'architecture)**

Le tableau ci-dessous résume les critères pour les trois catégories essentielles de composants :

Type	Rôle	Type de participant	Granularité
	Processus métier transverse, Orchestration de service	Fournisseur et consommateur de service	Granularité élevée. Transverse par nature
	Processus de traitement, composition de services, adaptation	Fournisseur et consommateur de service	Granularité moyenne
	Accès à un objet métier clé	Fournisseur de service	Granularité fine, focalisé sur un objet métier clé

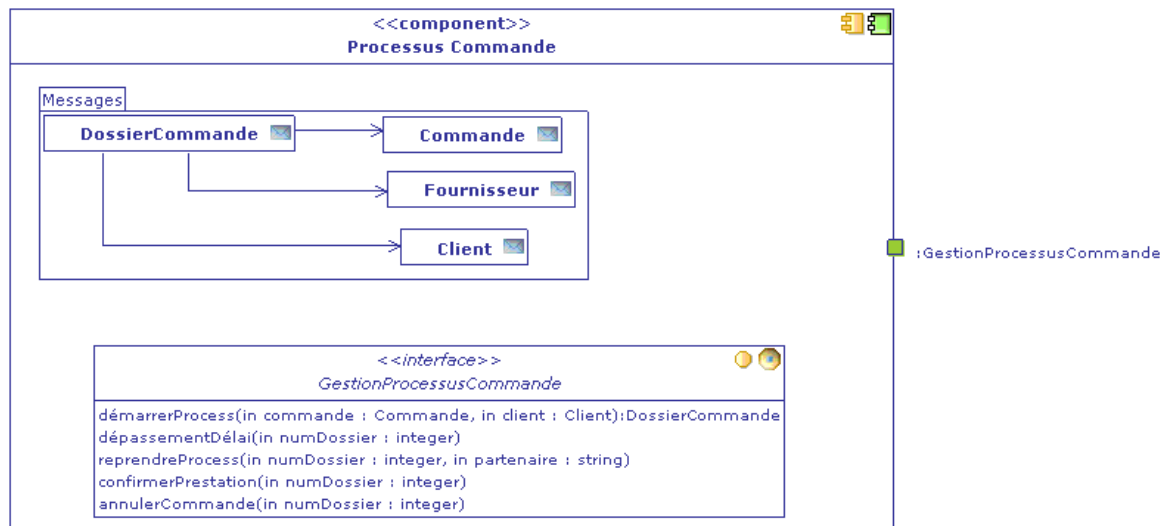
**Règle :** Pour chaque objet métier clé, on doit trouver un composant Entité correspondant.

**Règle :** Les composants Processus proviennent directement des processus métier de l'entreprise.

Les composants "Fonction" sont mis en place graduellement par consolidations successives du système. Ils fournissent les services proches de la vision utilisateur, par composition de services de type Entité. (Exemple : "Contrat Client" sera produit en agrégeant "Contrat" et "Client").

Les composants de service de type Entité sont focalisés sur un objet métier clé du système (par exemple Client, Contrat, Commande, ...). Leur rôle est de permettre un accès aux informations relatives à cet objet métier, le plus souvent associé à une base de données. On y trouve typiquement les opérations de lecture, écriture ou de requête. On impose que tout accès à un objet métier clé passe par le composant Entité correspondant qui est unique.

Les composants de type Processus automatisent une partie des processus métier : ce sont des processus de traitement. Les opérations de service Processus sont liées aux événements du processus : démarrage, arrêt, ou spécifiques au métier.



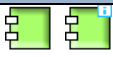
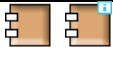
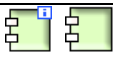
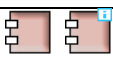





**Figure 6 - Composant processus: interface et type de donnée d'échange (diagramme de composant)**







La démarche pour l'identification et la construction des services dépendra du contexte de l'entreprise, des méthodes ou des modèles d'urbanisation utilisés et combinera les démarches types suivantes :

- Démarche par processus métiers : Détermination des processus métiers à automatiser, en s'appuyant sur la modélisation de l'organisation et des processus (voir guide de modélisation des processus, et guide de la modélisation de l'organisation).
- Démarche orientée données : objets métier clés du système.
- Démarche orientée applications : restructuration de certaines applications par mutualisation de services.

## Eléments essentiels pour modéliser l'organisation

Le tableau ci-dessous présente les éléments principaux fournis par Modelio SOA Solution pour modéliser l'architecture logique, et l'urbanisation sur SI :




Icône	Nom	Définition
	Composant de service processus ou instance de composant processus	Support de processus métiers complet (rôle d'orchestration).
	Composant de service entité ou instance	Représentation logique d'un concept métier autonome.
	Composant de service fonction, ou instances	Composant intermédiaire assemblant divers composants Entité ou fonction pour offrir un service de plus haut niveau.
	Composant de service utilitaire ou instance	Fournisseur de services d'infrastructure ou transversaux.
	Composant de service public ou instance	Fournisseur de services public.
	Composant d'interaction	Implémentation du dialogue et de l'interface avec l'utilisateur.
	Système ou Fédération de systèmes : composant et instance	Représentation de systèmes ou sous-systèmes, de leur constitution en sous-systèmes/composants, et de leur assemblage.
	Application, Instance d'application	Fédération d'un ensemble de composants qui concourent à fournir des services dédiés à une ligne métier ou une utilisation spécifique du système.
	Composant "base de données"	Représente un référentiel de données partagé. On l'utilise pour les applications existantes (cartographie), où le découpage en composants de service n'est pas fait. Une architecture de service ne présente pas ce type de composant, la gestion des données étant assurée via les composant "Entité".

	Message	Les messages ou "types de donnée d'échange" représentent la structure des flux échangés via les opérations de service.
	Service, Opération de service	Fonction fournie par le SI et rendue publique, disponible et invocable par une interface (contrat) pour être mutualisée ou orchestrée.
	Service fourni	Point d'accès à un service fourni par un composant. Ce point d'accès est typé par le service fourni
	Service requis	Point d'accès à un service requis par un composant. Les services fournis et requis sont reliés par des connecteurs.
	Package, Package d'implémentation, Package modèle de données	Ces packages décomposent les composants ou systèmes. Le package d'implémentation permet d'exprimer l'implémentation en UML d'un composant. Le Package structure les éléments du modèle logique (comme typiquement les interfaces ou messages) sous les composants. Les modèles données sont restreints au modèle statique pour modéliser les schémas de données (en général sous les systèmes).
	Unité d'architecture logique	Unité le plus en amont de structuration. Permet de structurer en domaines fonctionnels par exemple. Peut être utilisé pour représenter des zones en urbanisation.

## Vision matérielle et géographique – déploiement des composants du SI

Le modèle logique sera encore plus explicite et mieux compris si son déploiement matériel géographique est défini. On positionne ainsi les différents composants du système, en se limitant aux éléments de granularité supérieure, sur les matériels sur lesquels ils s'exécuteront (serveurs, poste de travail, etc.). On peut également localiser géographiquement ces éléments matériels en déployant les matériels sur des localisations géographiques. Ce travail consolide la manière dont le déploiement des actuelles et futures applications est effectué. Il aide à valider l'architecture et ses impacts.

Pour avoir ce modèle, il faut sélectionner la racine de tout le modèle dans l'explorateur, et créer un modèle de déploiement métier, ou d'implémentation métier.

Icône	Nom	Définition
	Serveur	Ordinateur central faisant office de serveur d'application ou de données. On déploie des composants, systèmes, application ou bases de données sur le serveur.
	Station de travail	Poste de travail d'un utilisateur du système. Sert de point d'accès au système. On peut également déployer des composants sur une station. Il est déconseillé de déployer des bases de données sur des stations.
	Instance d'application, ...	Permet de déployer une occurrence d'un élément du modèle logique (système, application, composant, etc.) dans un site. (créer l'instance graphiquement dans le lieu de déploiement (par exemple un serveur), typer l'instance par le site).

Le modèle se construit ensuite simplement en définissant les éléments de l'architecture matérielle (serveurs, stations), en les rattachant par des associations représentant les canaux de communication essentiels. Ensuite, les composants logiques sont déployés sur l'architecture matérielle en créant des instances typées par ces unités (voir Figure 7). Il est ainsi possible de déployer un même composant logique plusieurs fois dans des matériels différents.

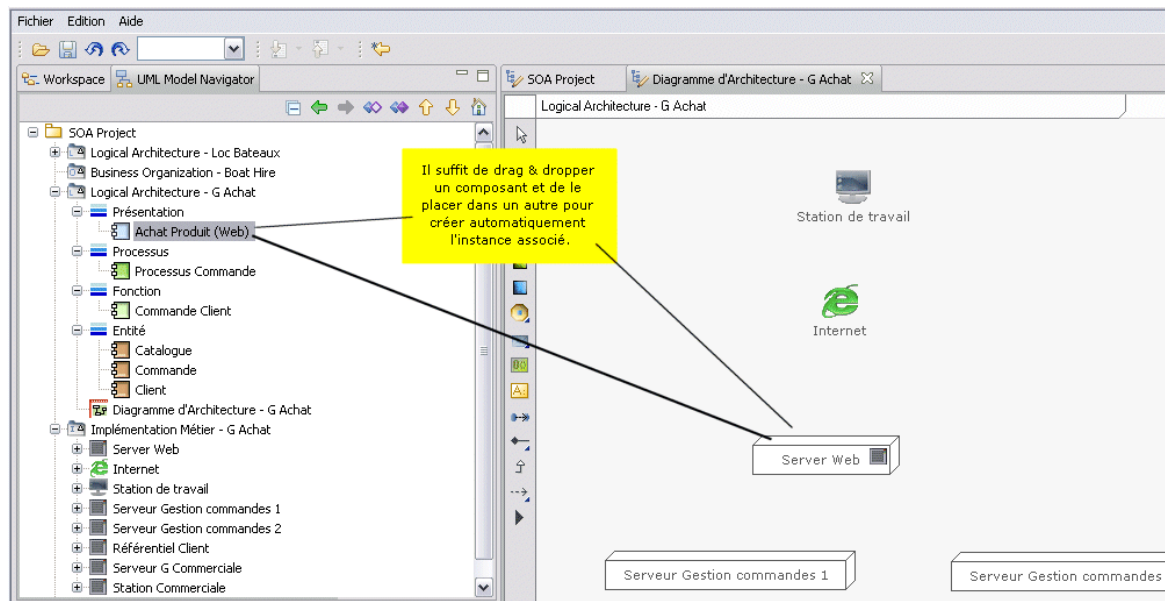


Figure 7 - Déploiement d'un composant logique sur un serveur

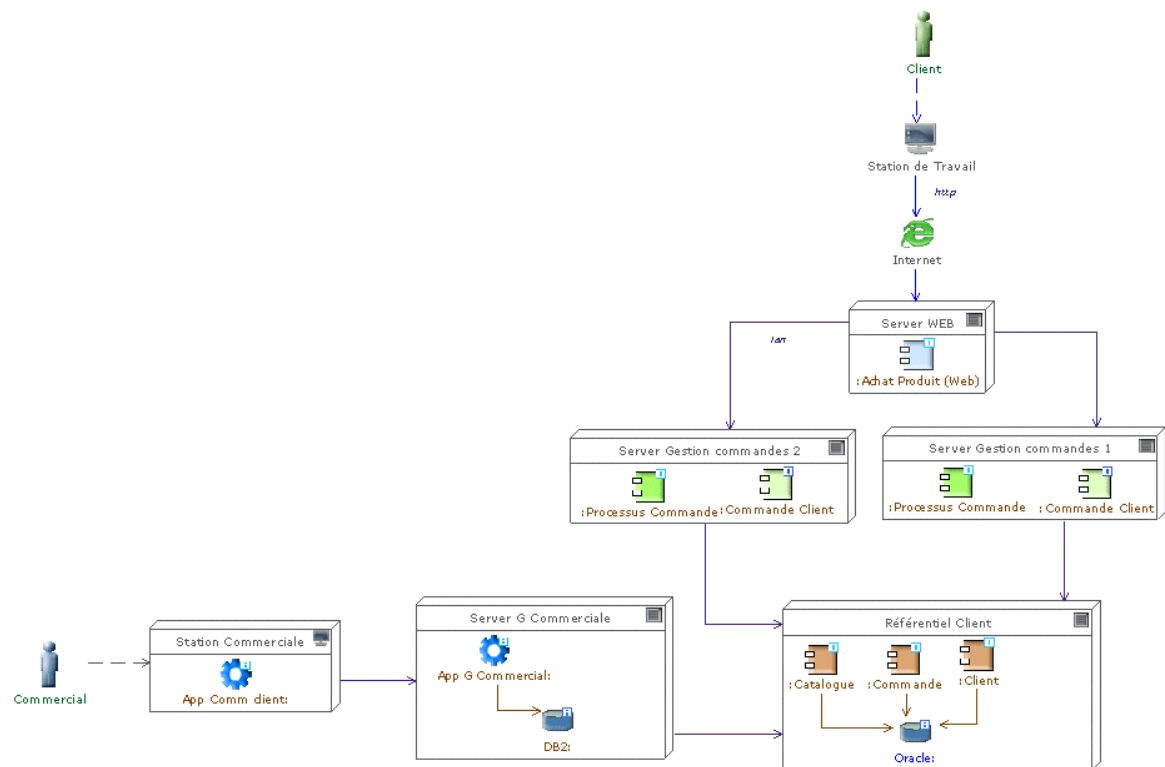


Figure 8 - Exemple de déploiement d'une architecture logique sur du matériel (diagramme physique)