

# Architectures Orientées Services

## Module 6

Convergences à établir d'approches diverses

ISO IEC : RM/ODP

UN/CEFACT & OASIS : ebXML CCTS

OMG : Initiatives MDA, MOF, UML-OWL

Hégémonie à venir du Web sémantique

# Plan du module

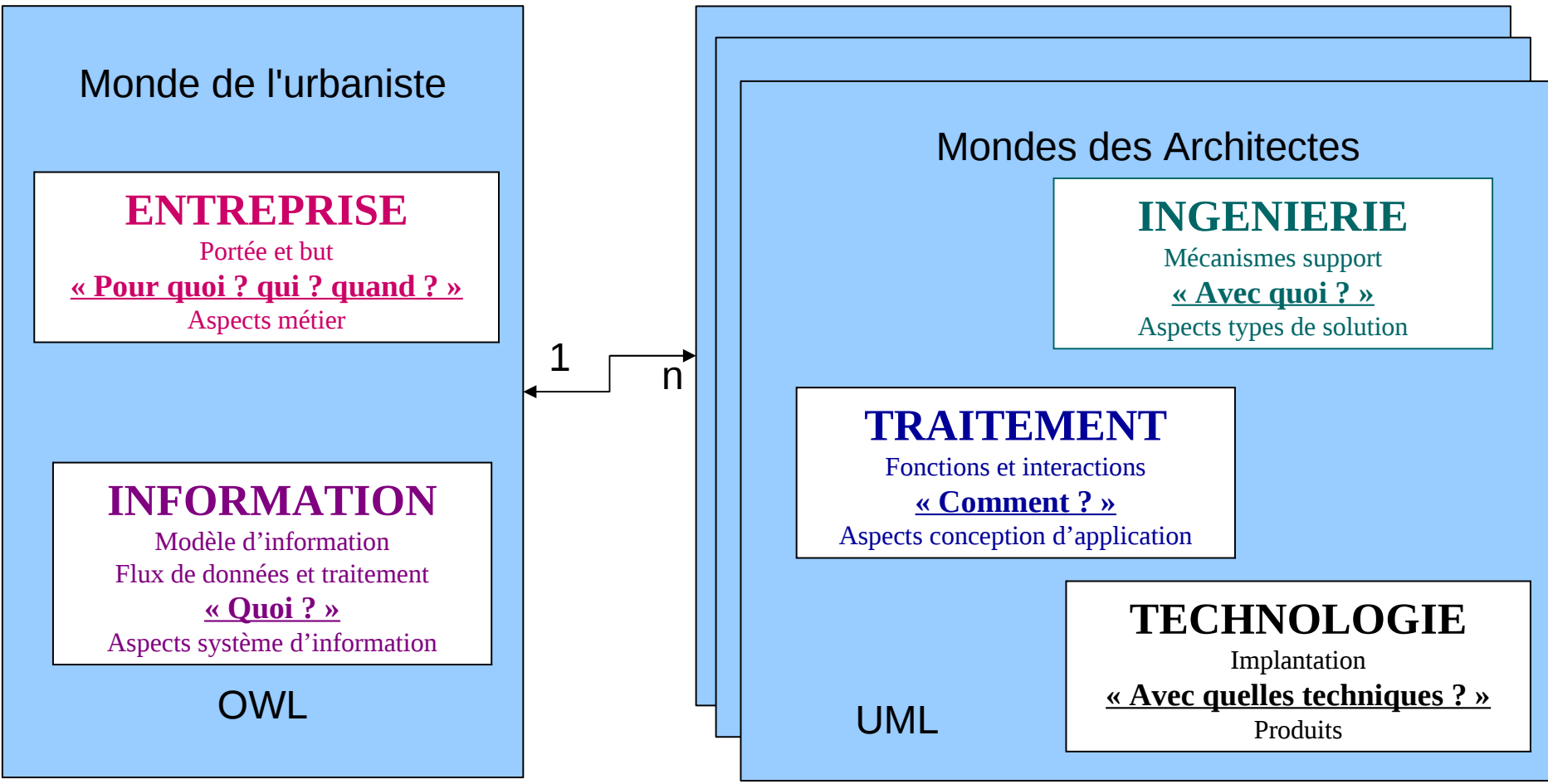
- Recherches de convergences d'initiatives de différentes origines
  - ISO-IEC et le standard RM/ODP Open Distributed Processing Reference Model
  - UN/CEFACT et OASIS : ebXML et les Core Component Types
  - l'OMG et ses initiatives de pour une approche formelle rigoureuse pour UML
    - Méta modélisation de UML : MOF, sérialisation XMI
    - Intégration des problématiques ontologiques dans UML : profil -OWL
    - La démarche MDA ; Model Driven Architecture, QVT Query/View/Transformation
- Apports du Web sémantique aux architectures orientés services
  - Évolution vers OWL-S des modalités de découverte et d'accès aux Web Services.
- Apports du Web sémantique à l'urbanisation de systèmes d'information
  - Cartographie ontologique

- Le modèle de référence RM-ODP est une norme de l'ISO et de la CEI qui définit les points de vue d'un réseau réparti de systèmes.
- Ces points de vue englobent :
  - le modèle de gestion (point de vue de l'entreprise),
  - le contenu de l'information et le comportement du système (point de vue de l'information),
  - les composantes, les interfaces et les contraintes (point de vue du traitement),
  - l'infrastructure et les mécanismes de distribution des composantes (point de vue de l'ingénierie) et
  - l'environnement de mise en œuvre et de déploiement utilisant les technologies, les normes et les produits actuels (point de vue de la technologie).

Points de vue du modèle RM-ODP	Champs d'intérêt
<p><u>Point de vue de l'entreprise</u> Présente un modèle de gestion que tous les intervenants doivent pouvoir comprendre, précise le but, la portée, les objectifs opérationnels, les politiques, les objets de l'entreprise, etc.</p>	<ul style="list-style-type: none"> <li>• But et portée</li> <li>• Politiques</li> <li>• Responsabilités</li> <li>• Processus de gestion et cas d'utilisation</li> </ul>
<p><u>Point de vue de l'information</u> Porte sur le contenu de l'information et le comportement du système (modèles de données, sémantique, schéma)</p>	<ul style="list-style-type: none"> <li>• Sémantique du traitement de l'information</li> <li>• Information du système</li> </ul>
<p><u>Point de vue du traitement</u> Capture les composantes, les interfaces, les interactions et les contraintes sans tenir compte de la distribution</p>	<ul style="list-style-type: none"> <li>• Description fonctionnelle</li> <li>• Interfaces</li> <li>• Opérations</li> <li>• Règles de liaison</li> </ul>
<p><u>Point de vue de l'ingénierie</u> Décrit l'infrastructure et les mécanismes pour la distribution des composantes, la transparence et les contraintes de la distribution, les liaisons et les interactions</p>	<ul style="list-style-type: none"> <li>• Infrastructure nécessaire pour soutenir la distribution</li> </ul>
<p><u>Point de vue de la technologie</u> Définit l'environnement de mise en œuvre et de déploiement utilisant les technologies, les normes et les meilleurs produits actuels</p>	<ul style="list-style-type: none"> <li>• Choix et pertinence des technologies pour soutenir le système de distribution</li> </ul>

# ISO-RM/ODP : une approche avec 5 points de vue

- RM-ODP : Open Distributed Processing Reference Model  
ISO 10746



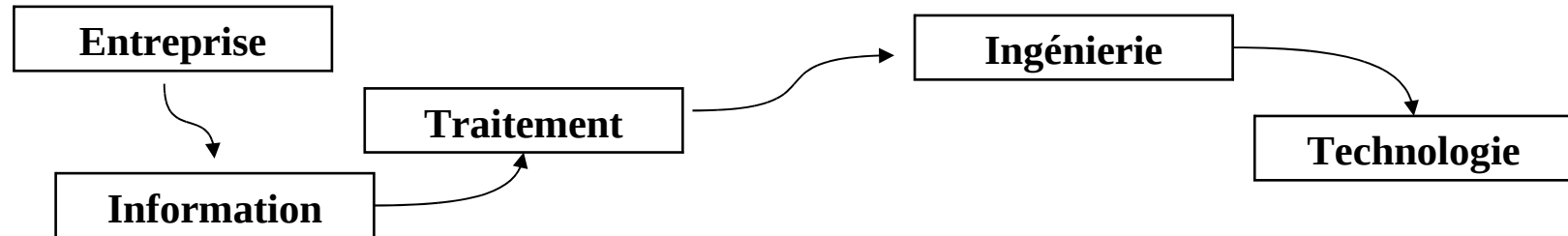
## ➤ **DASIBAO : une progression dans les points de vue ODP**

Travaux de H.TRAN, L.DUCHIEN, P.BEDU, H.NGUYEN, J.PERRIN, J. LE DELLIU, B. TRAVERSON, A. PICAULT

Électricité de France

**Spécification fonctionnelle**

**Spécification technique**



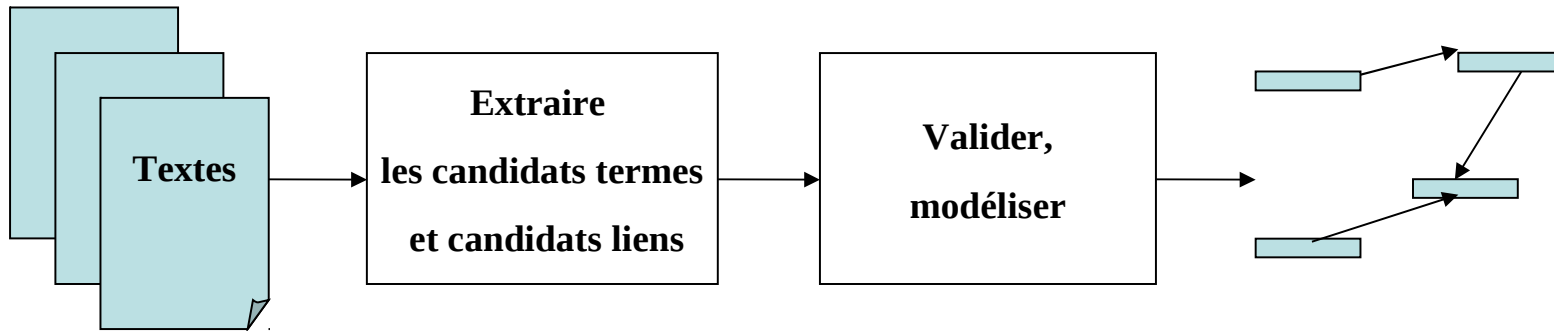
## Une gestion des relations avec les clients

### Suivi des contacts

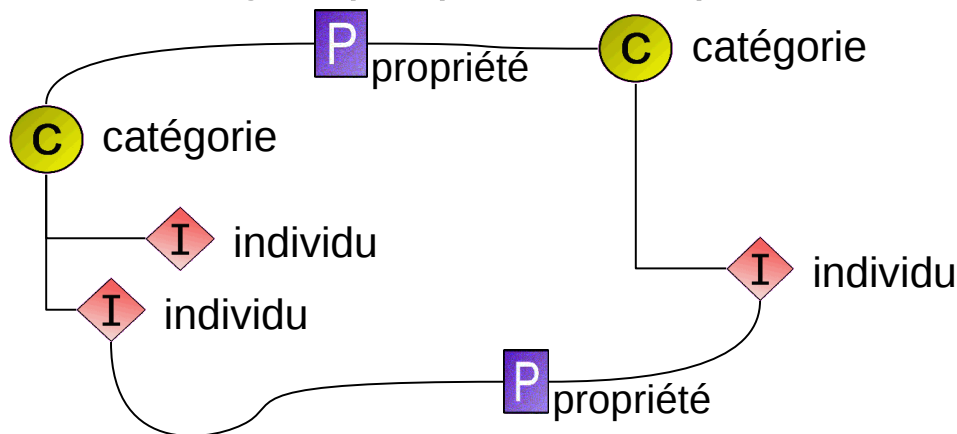
- **Contractualisation**
- Mise en œuvre des contrats

# Appréhender la terminologie du métier

- Identifier les termes et acronymes effectivement utilisés pour désigner les choses du métier

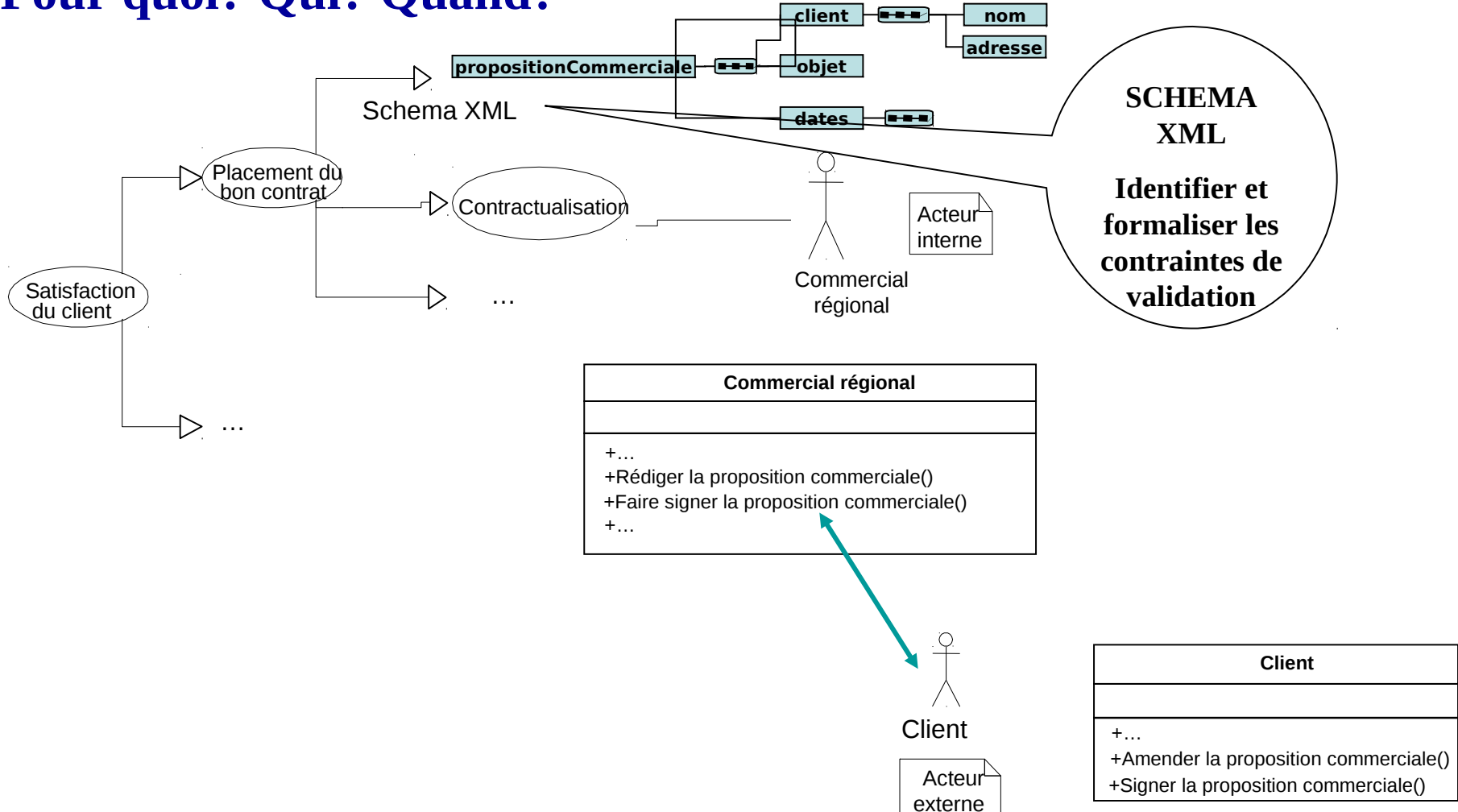


- Faire un modèle ontologique explicite formel du domaine concerné et de ses objets peuplé d'exemples

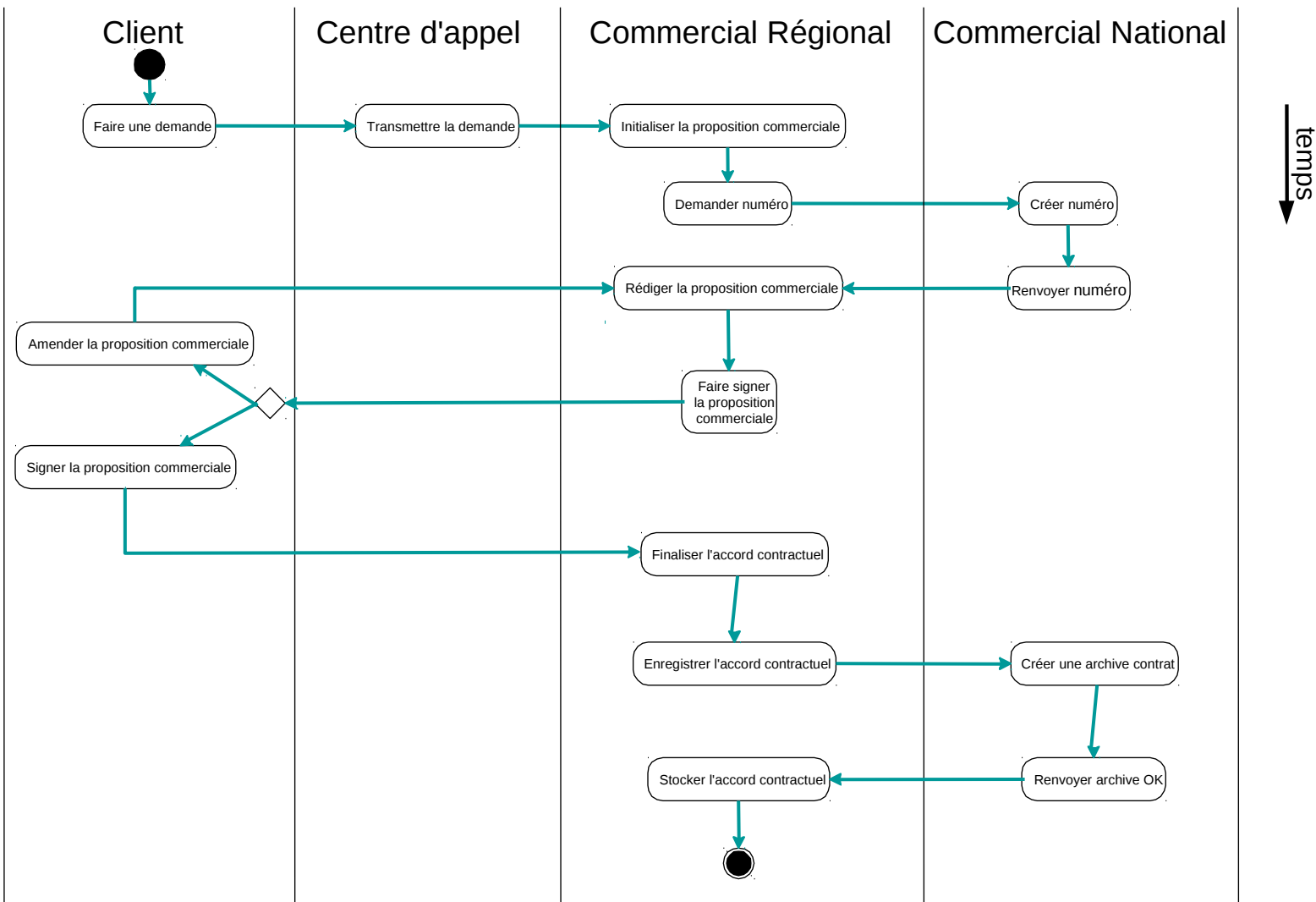




## Pour quoi? Qui? Quand?



# Point de vue Entreprise : Exemple



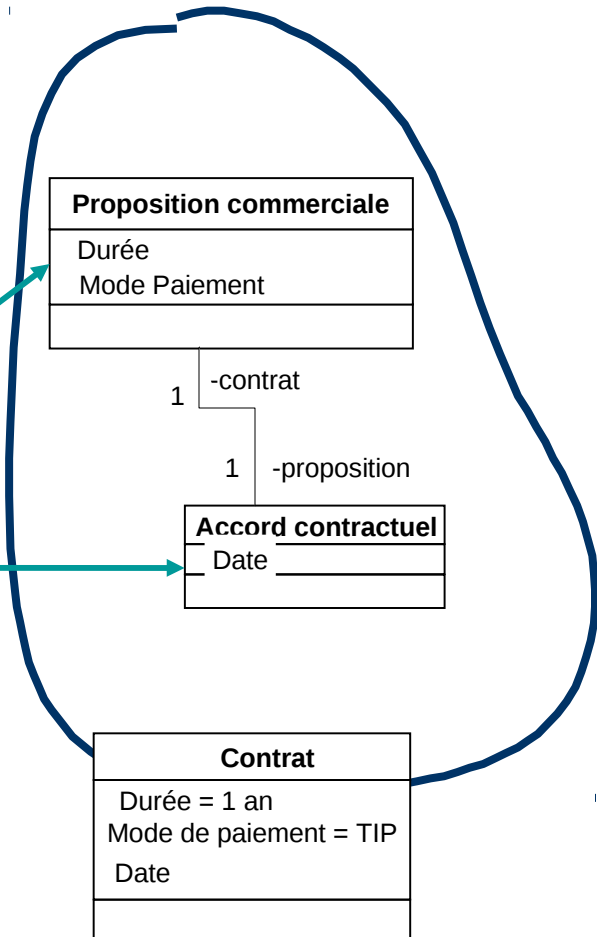
temps ↓

◇ Décision

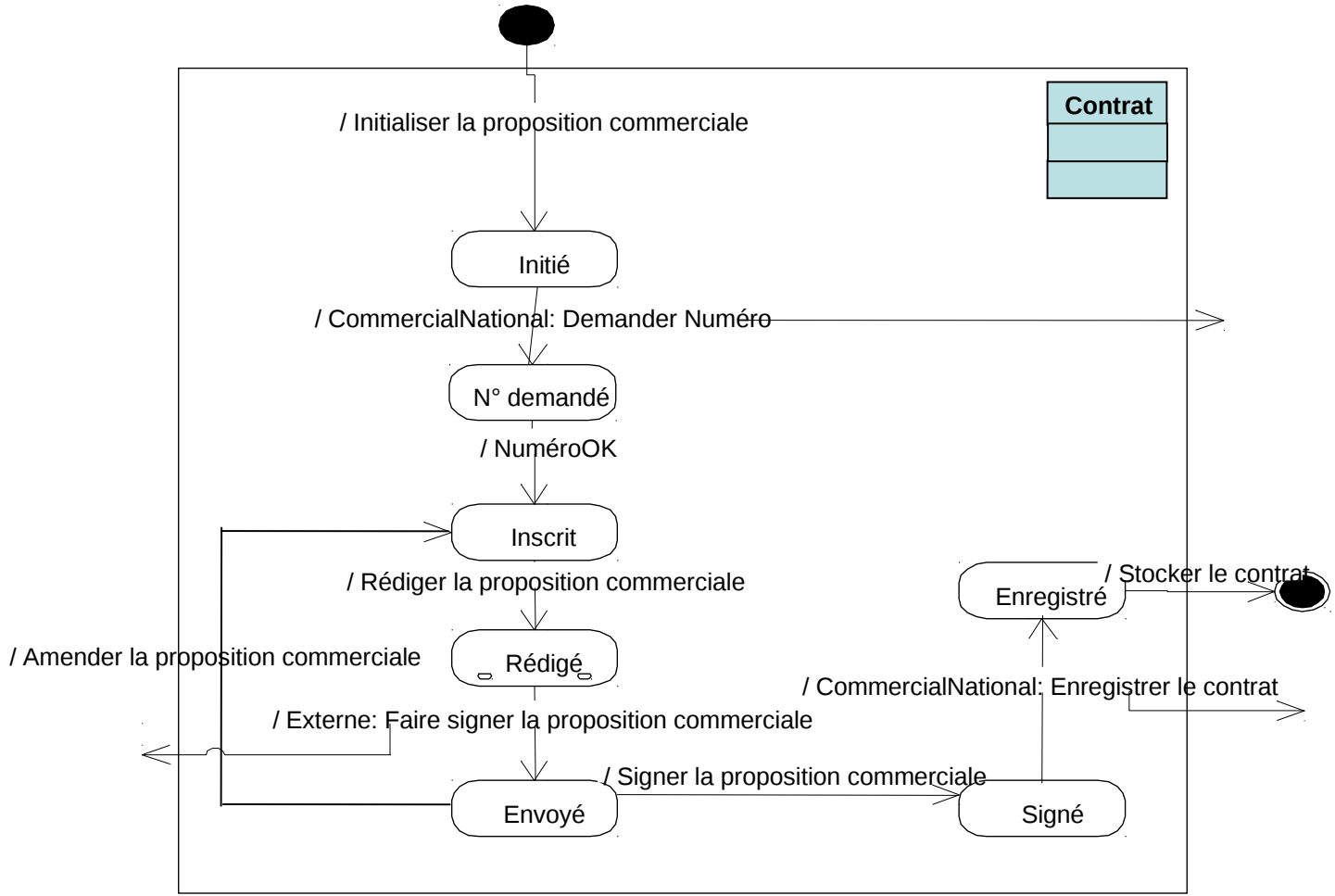
# Point de vue Information : Exemple

## ➤ Quoi?

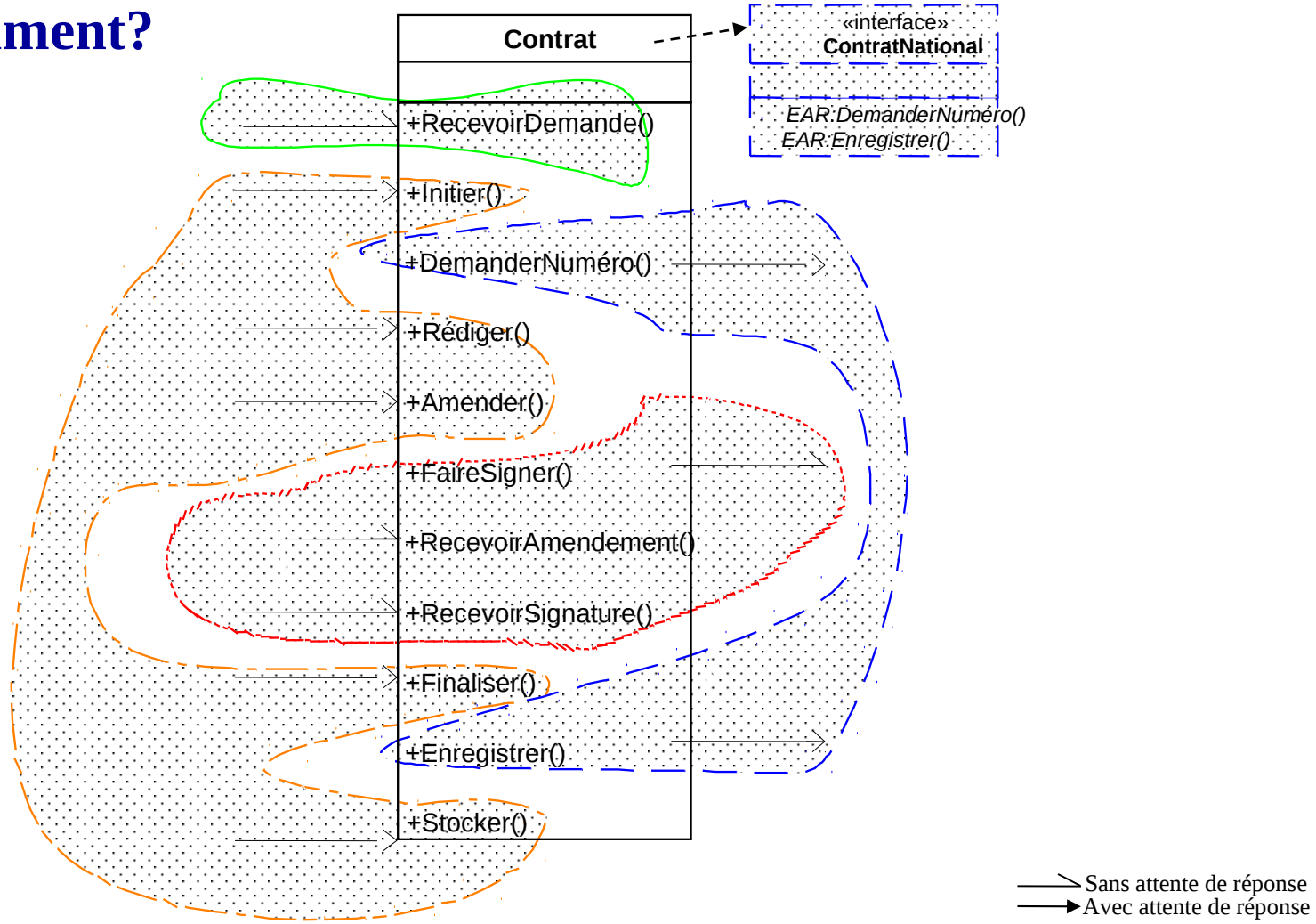
<u>Objet d'entreprise :</u>	<u>Action interne :</u>
Commercial régional	Initier la <b>proposition commerciale</b> Rédiger la <b>proposition commerciale</b>
	Finaliser <b>l'accord contractuel</b> Stocker <b>l'accord contractuel</b>



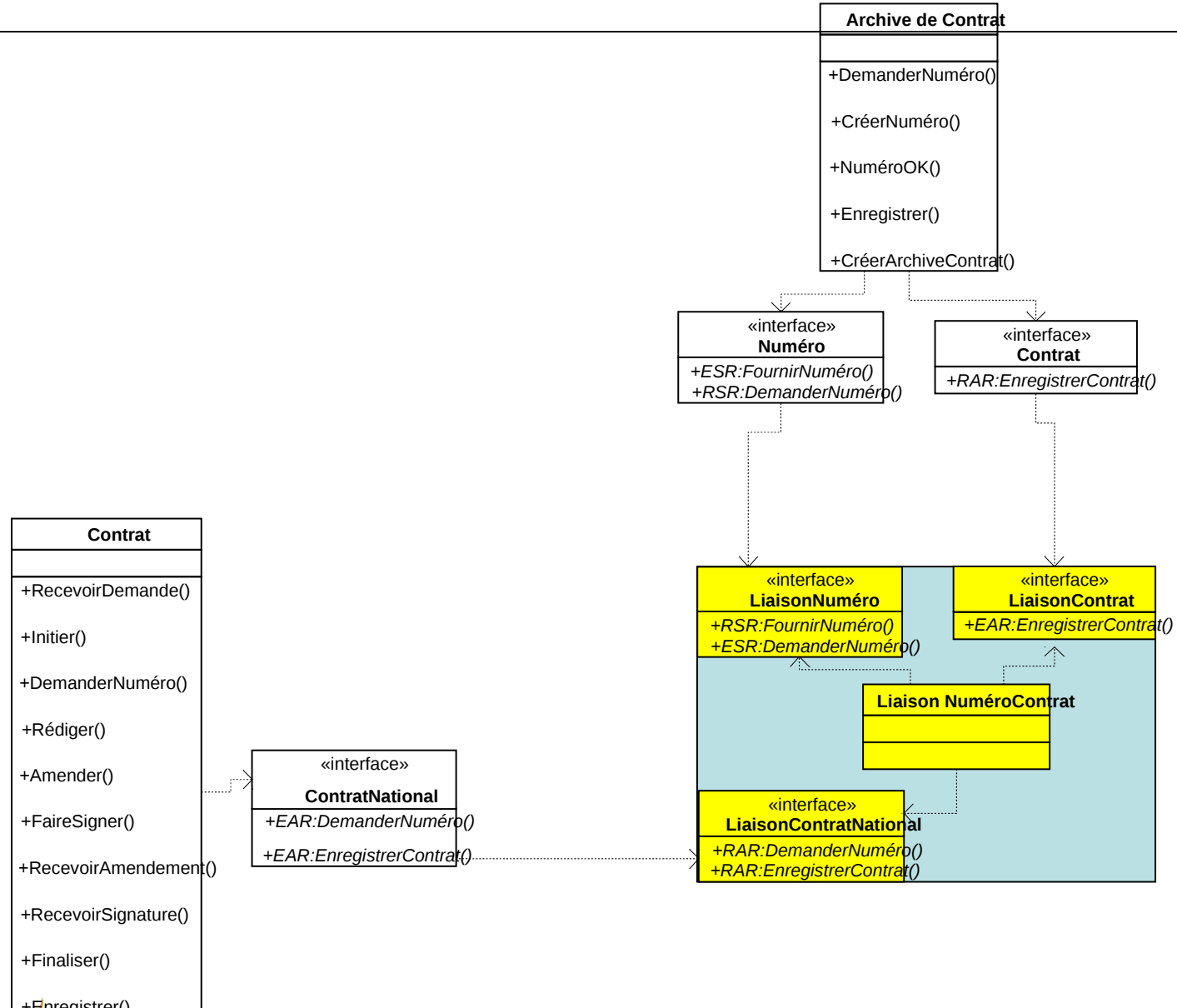
# Point de vue Information : Exemple



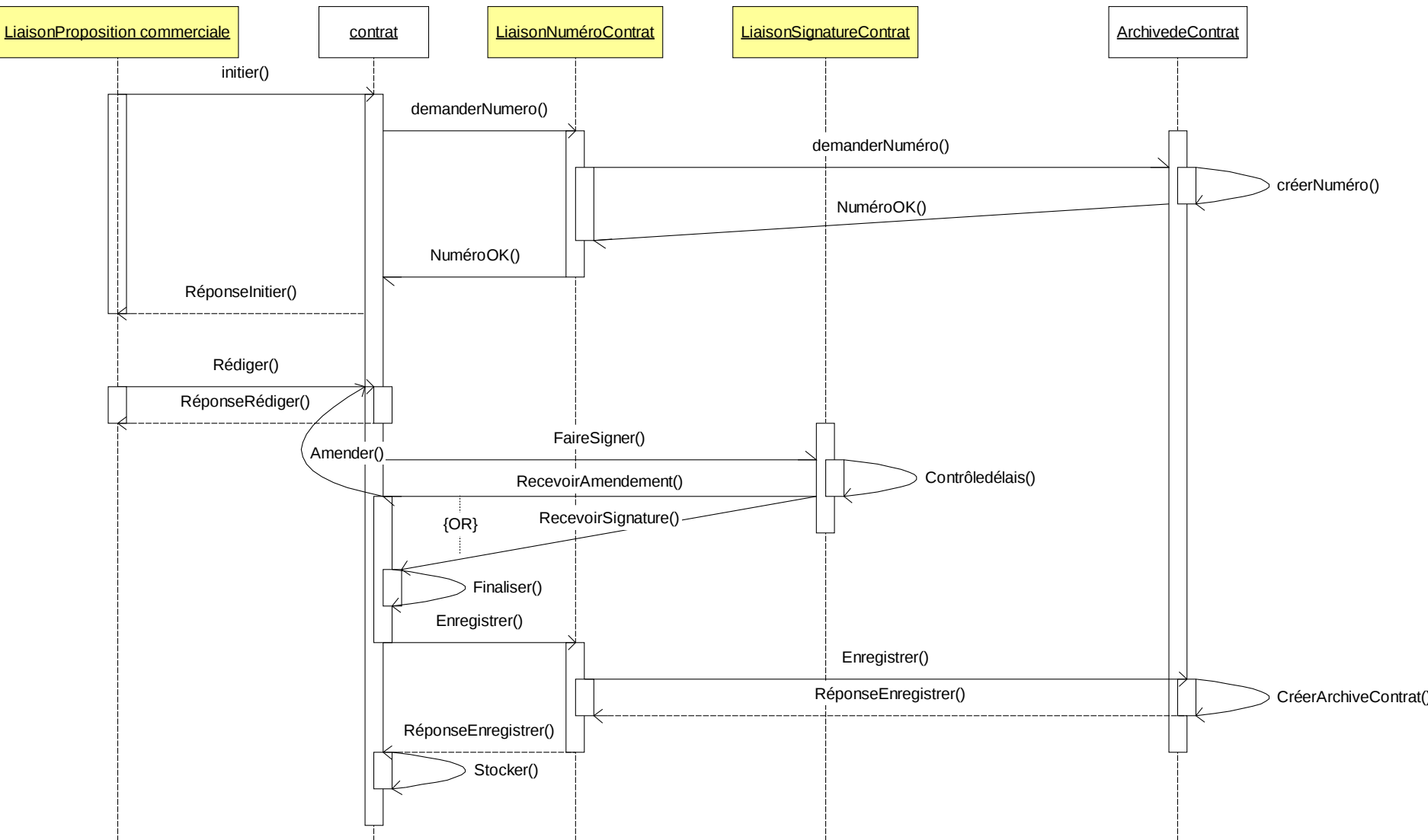
## ➤ Comment?



# Point de vue Traitement : Exemple



# Point de vue Traitement : Exemple



➤ **Avec quoi?**

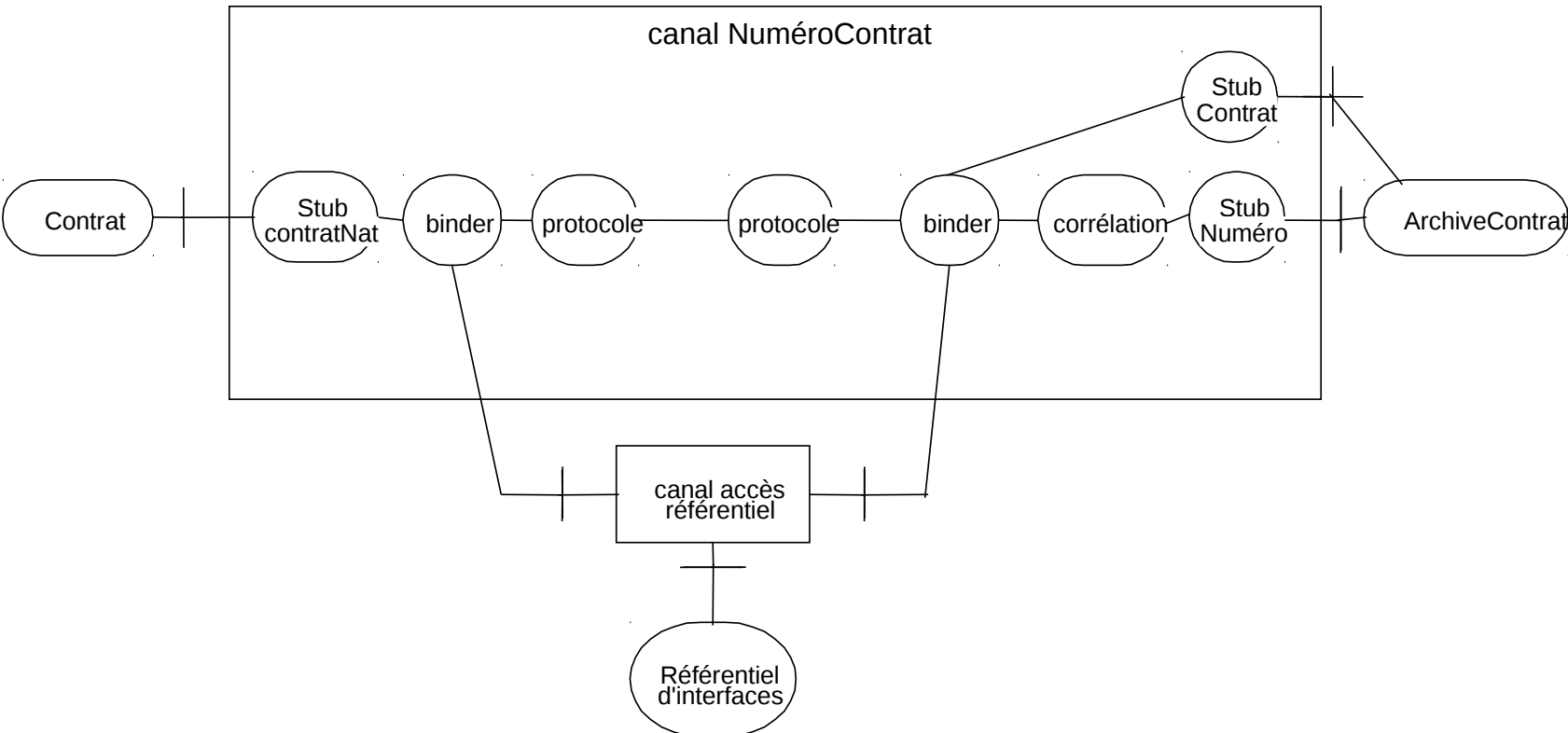
**Infrastructure de communication**

**Infrastructure de déploiement**

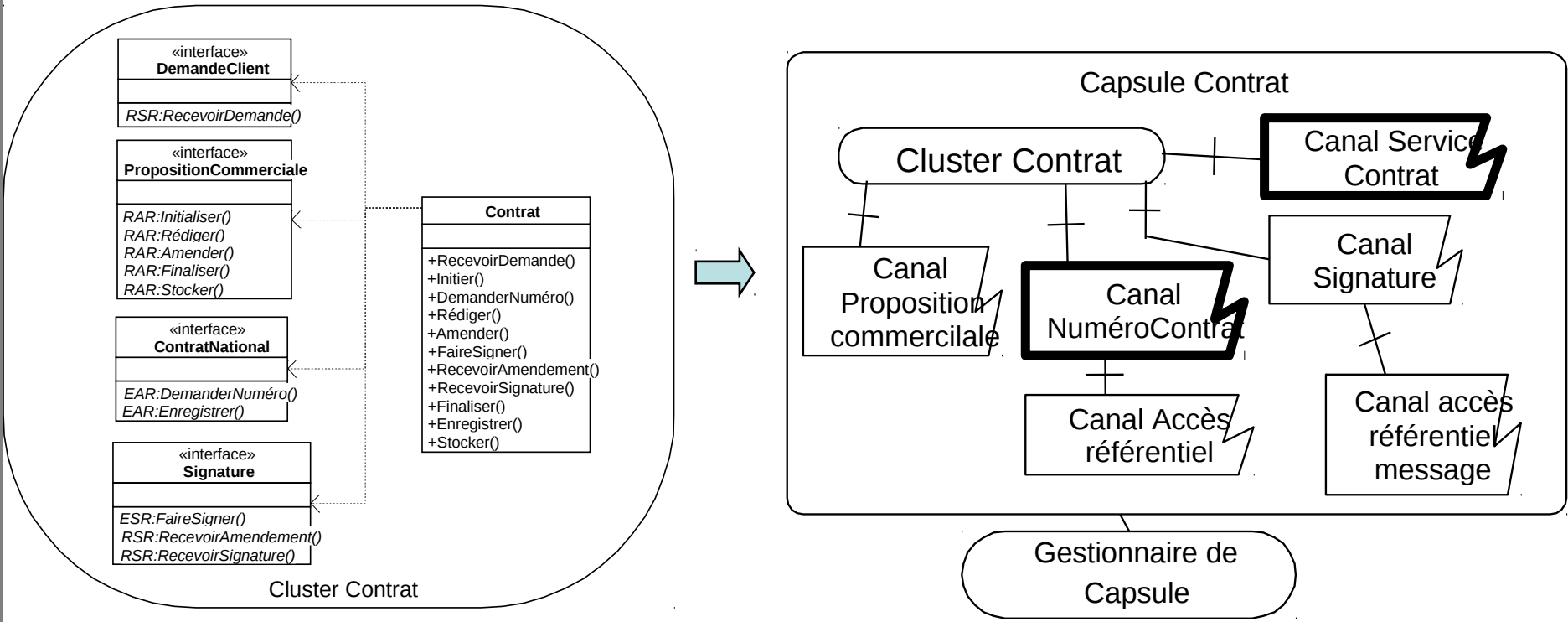


➤ Avec quoi?

## Infrastructure de communication

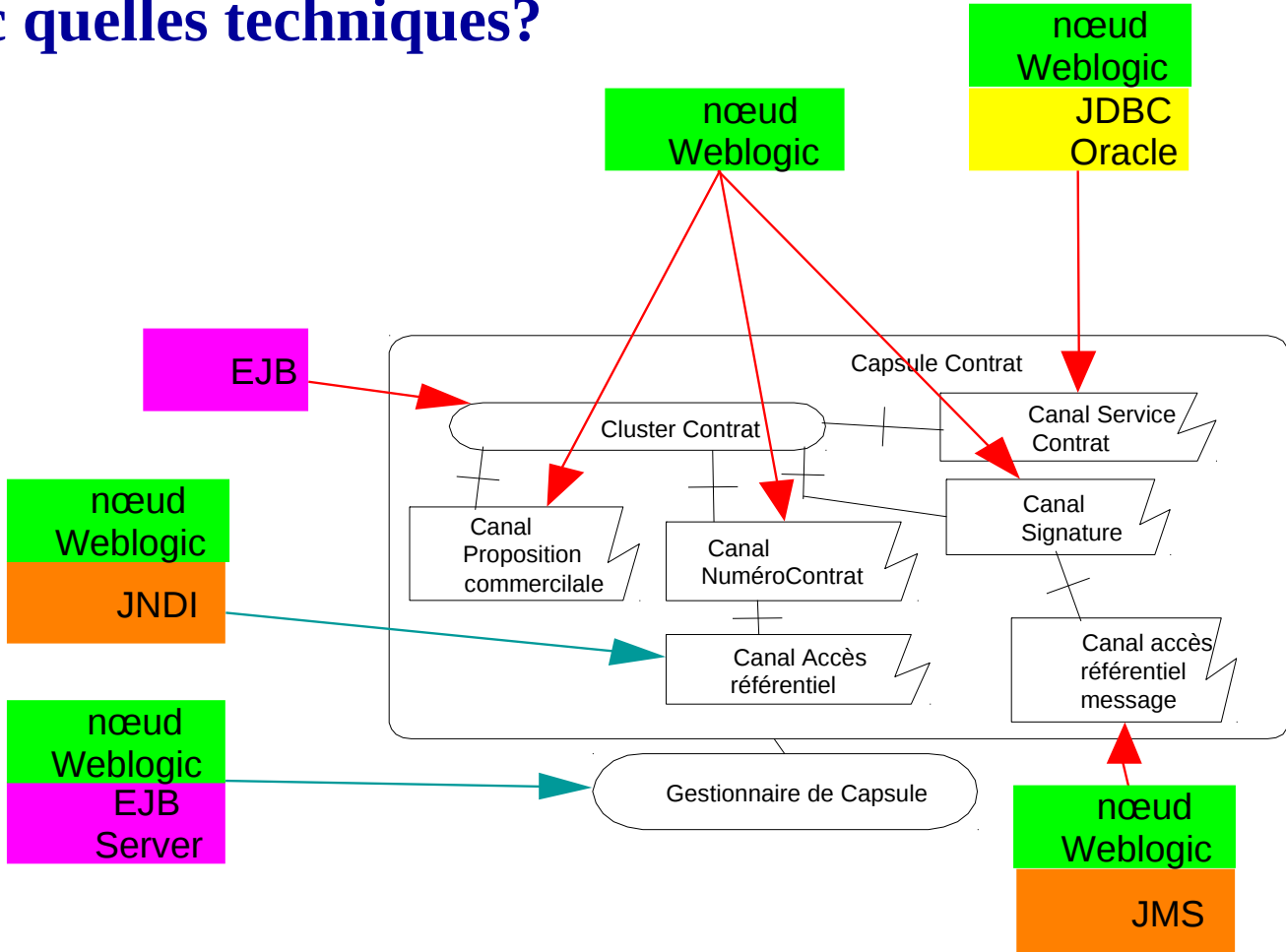


# Infrastructure de déploiement

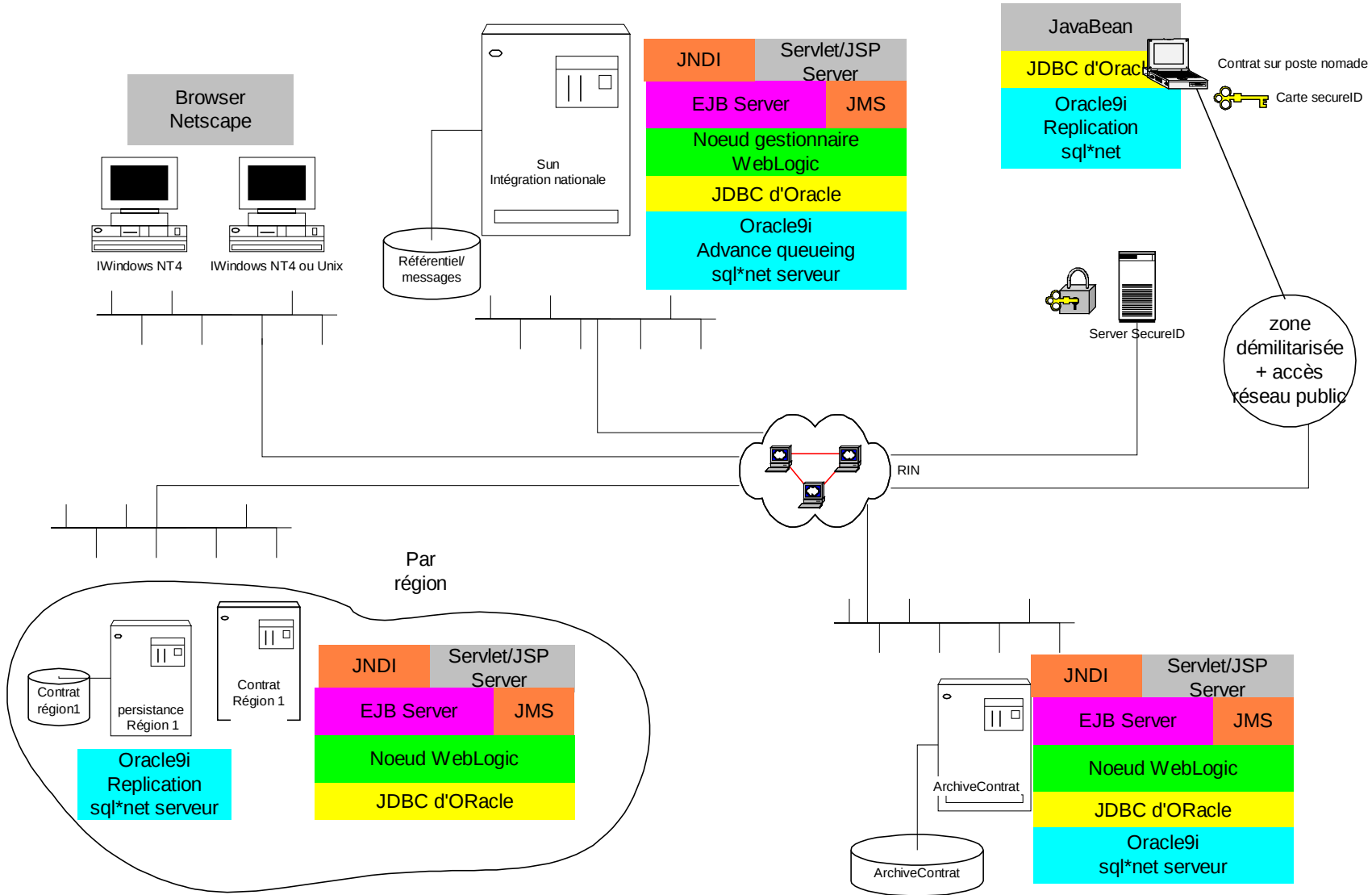


➡ Nœuds

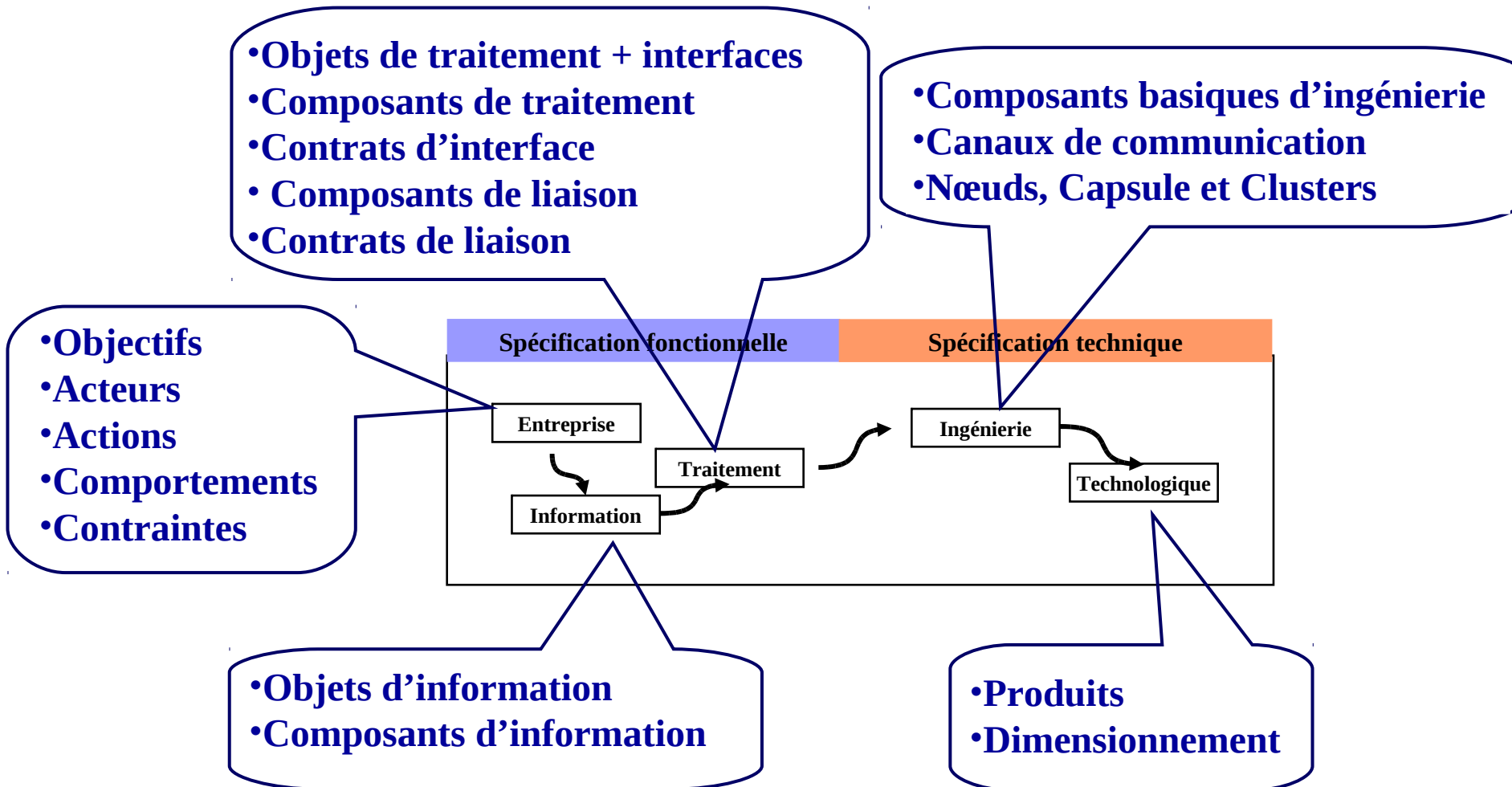
## Avec quelles techniques?



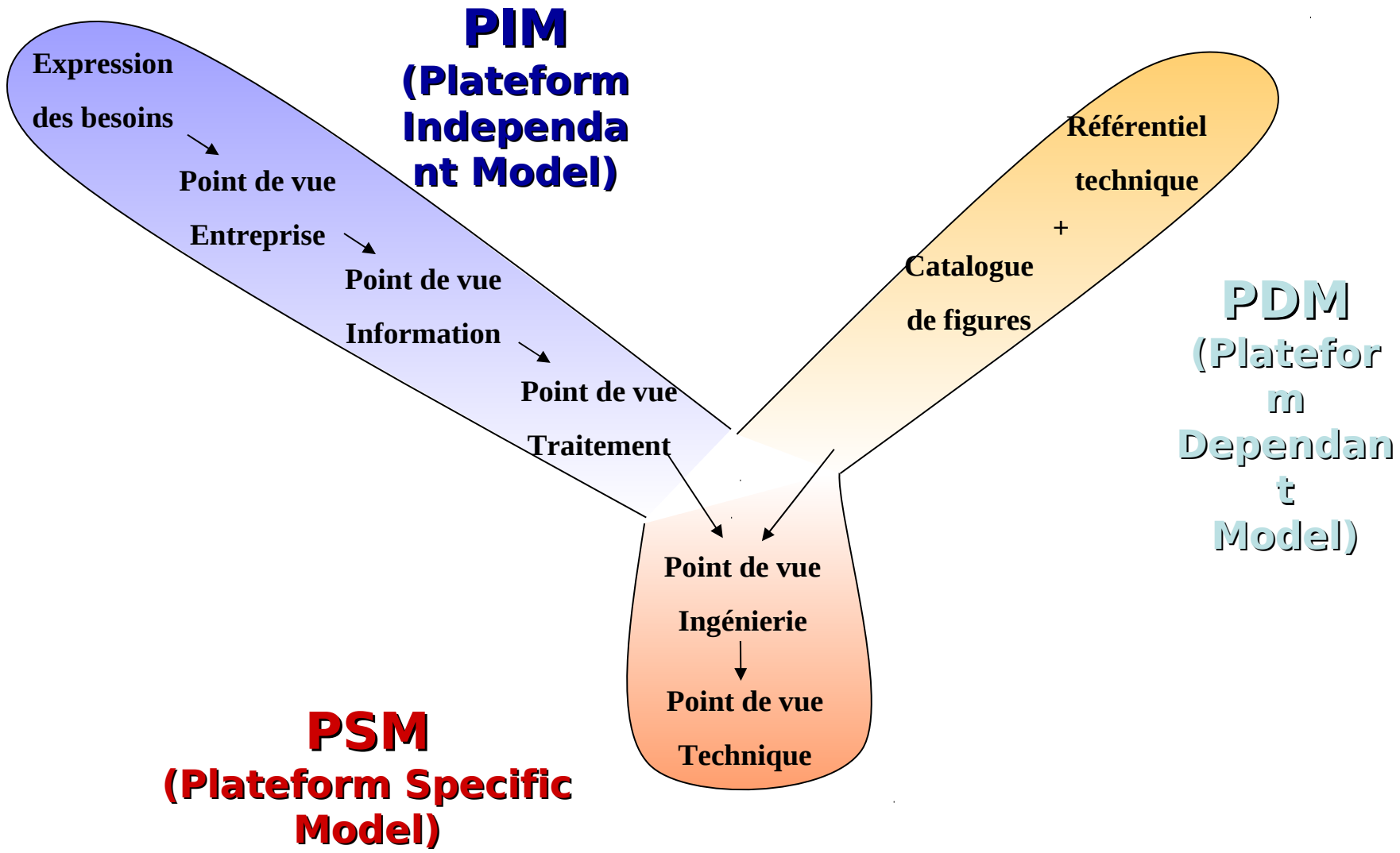
# Point de vue Technologique : Exemple



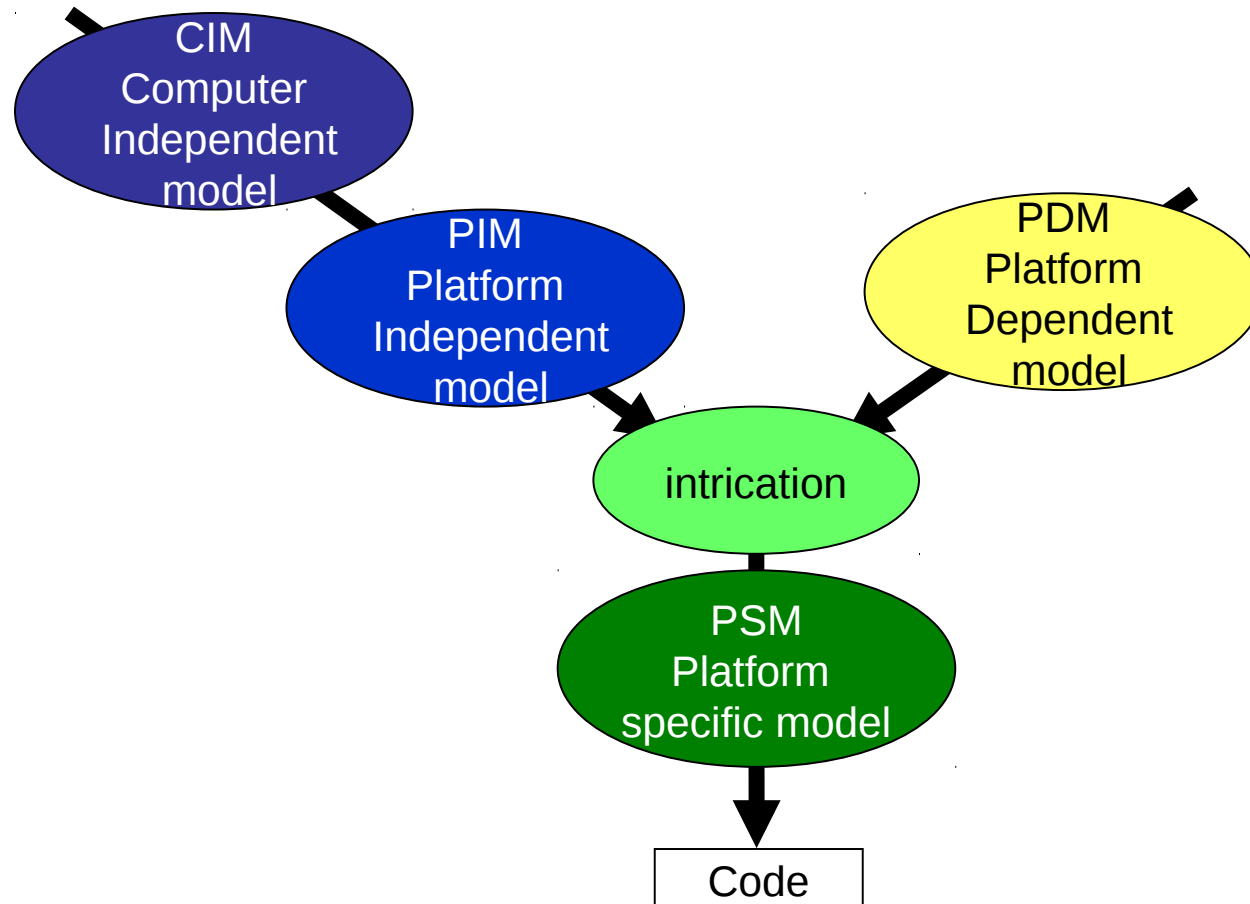
# Exemple : Synthèse



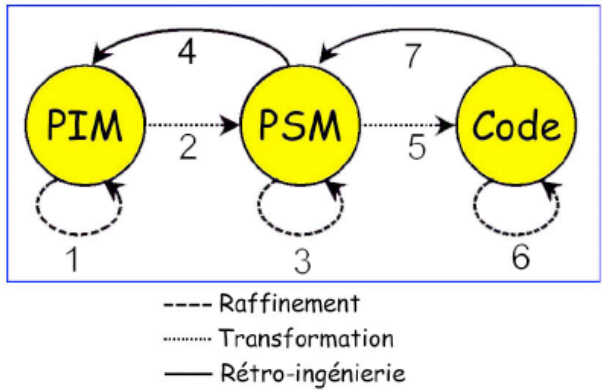
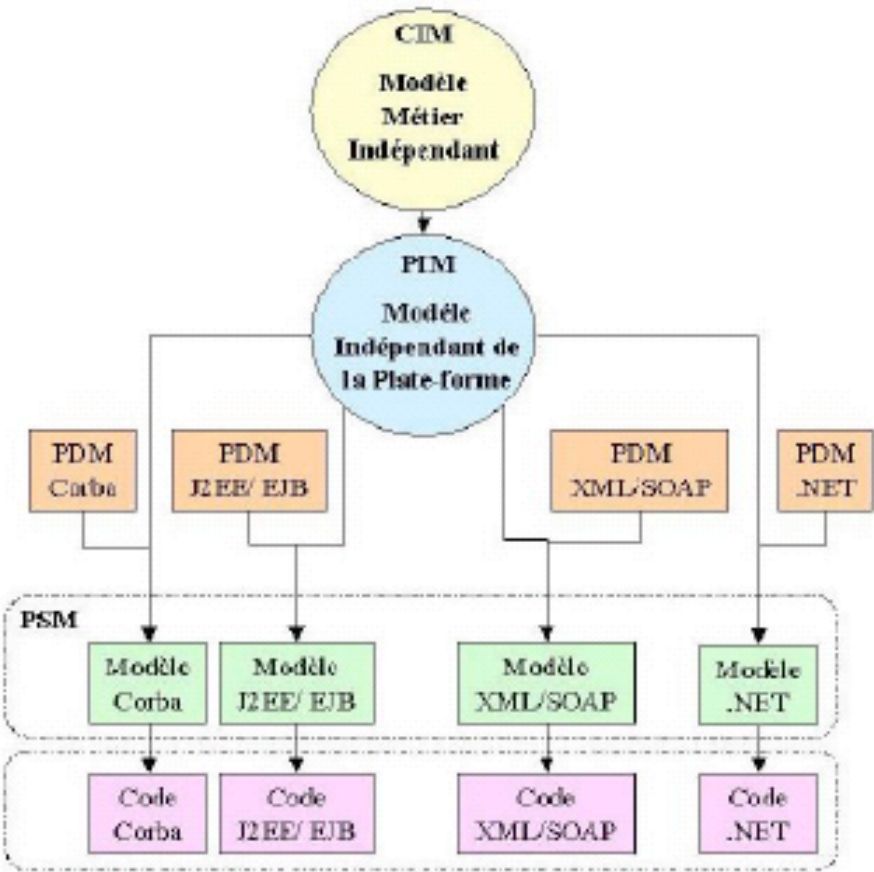
- Lacune de Dasibao
  - Ne formalise pas (encore) un modèle de processus
    - Autrement que dans la vision de modèles de séquences



- L'initiative MDA est motivée par le souci de réduire les tâches de reconception des applications nécessitées par l'évolution versatile constante des modes et technologies informatiques



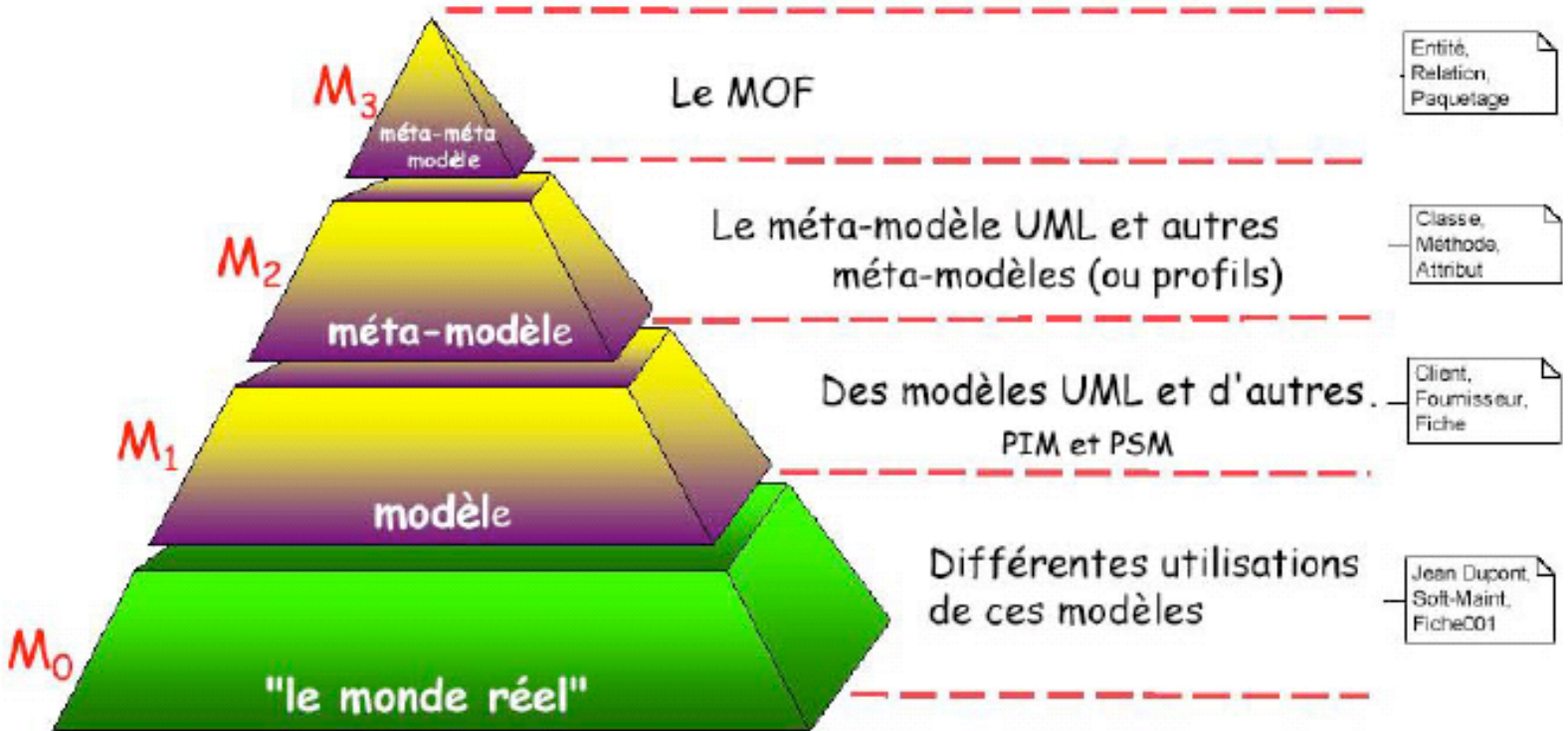
# Model Driven Architecture : principes de séparation des modèles





# Nécessité de méta modélisation pour une ingénierie des applications

- UML étend son statut de représentations graphique de concepts applicatifs pour traiter des objets formels traitables par des algorithmes
- Pour cela UML a nécessité d'être formellement modélisé par lui-même



# Méta - modélisation

Layer	Contents	Example
Meta-meta-model	Language to describe meta-models	<pre> classDiagram     class MetaEntity     class MetaRelationship     MetaEntity .. &gt; MetaRelationship : IsA     MetaEntity --&gt; MetaRelationship : HasSource (1:1 to 0:n)     MetaEntity --&gt; MetaRelationship : HasDestination (1:1 to 0:n)         </pre>
Meta-model	Language to describe models	<pre> classDiagram     class Class     class Attribute     Class .. &gt; Attribute : IsA     Class --&gt; Attribute : contains (1:1 to 0:n)         </pre>
Model	Language to describe data	<pre> classDiagram     class Customer {         Name         Address     }     .. .. Customer : IsA         </pre>
Data	Data	<pre> classDiagram     class PierreDupont {         Pierre Dupont         8, rue La Fontaine         Paris     }     PierreDupont .. .. Customer : IsA         </pre>

# Convergence ?

- Deux communautés qui doivent coopérer
  - Les données, domaine privilégié du W3C avec les standards XML
  - Les traitements, domaine privilégié de l'OMG, avec les standards UML
  - XML n'est rien sans traitements applicatifs
  - UML n'est rien sans données à traiter
- Les deux communautés ont développé des capacités de modélisation, selon des modèles formels de grammaire eBNF.
- Les deux communautés peinent à faire converger leurs visions
  - L'OMG considère les standards XML comme des standards d'implémentation.
  - Les tenants d'UML cherchent à créer des modèles XML xsd et owl à partir de diagrammes UML, ce qui nécessite des « profils » avec des conventions à l'expérience non formalisables.
- Il serait nécessaire de reconnaître la nature ontologique du MOF
  - Et de le transcrire en OWL
  - Toute transformation de modèle peut se décrire par une transformation XML

- Les Nations Unies ont créé l'UN/CEFACT en vue d'améliorer la coordination à l'échelle mondiale de la facilitation des procédures commerciales.
- L'UN/CEFACT fait la promotion
  - de l'initiative de normalisation de « Core Component Types »
  - des standards ebXML , conjointement avec OASIS :
    - **ebXML Messaging Services**
      - Maintaining and advancing ebXML Message Service Specification, which provides a secure method for exchanging electronic business transactions using the Internet.
    - **ebXML Registry**
      - Developing specifications for interoperable XML registries and repositories.
    - **ebXML CPPA Collaboration Protocol Profile and Agreement**
      - Developing Collaboration Protocol Profiles (CPPs) and Collaboration Protocol Agreements (CPAs).
    - **ebXML Implementation, Interoperability, and Conformance**
      - Facilitating the creation of interoperable ebXML infrastructures and applications.

# ebXML et ses objectifs (source site ebXML)

- L'initiative ebXML a débuté en Novembre 1999 à San José, California. Pour répondre aux objectifs fixés, ebXML a développé un ensemble de principes généraux sur les plans fonctionnels et techniques qui devraient être respectés dans la suite des travaux. Ces principes sont les suivants :
  - Permettre, en utilisant XML, de conduire des opérations de commerce électronique de manière simple et universelle .
  - Utiliser les spécifications techniques XML du W3C ayant acquis le statut de recommandation de la manière la plus étendue possible.
  - Fournir un standard trans-sectoriel ouvert et interopérable pour le commerce "B to B" et "B to C".
  - Fusionner les composants de structure et de contenu d'initiatives XML divergentes en un unique standard XML applicable à l'ensemble des relations d'affaires.
  - Fournir une dynamique initiale suffisante pour que les ressources communes engagées actuellement dans la mise au point de solutions verticales à court terme puissent être mobilisées dans un effort commun de développement de solutions horizontales portant sur le long terme.
  - S'appuyer sur des regroupements horizontaux et verticaux des participants de l'industrie et du commerce.
  - Eviter les solutions propriétaires qui influencent la liberté des choix financiers et logiciels des utilisateurs d'ebXML et contraignent ces derniers à acheter, installer ou supporter des produits logiciels ebXML uniques pour mener à bien leurs échanges d'informations d'affaires.
  - S'efforcer de minimiser le coût du commerce électronique.
  - Fournir un support multilingue.
  - Se conformer aux règles et spécifications nationales et internationales du commerce.
  - Proposer un chemin de migration permettant à partir de l'EDI standardisé de développer des standards XML EDI.

- **ebXML Messaging Services**

- The ebXML Message Service (ebMS) defines the message enveloping and header document schema used to transfer ebXML messages over a communications protocol such as HTTP or SMTP and the behavior of software sending and receiving ebXML messages.
- The ebMS is defined as a set of layered extensions to the base Simple Object Access Protocol [SOAP] and SOAP Messages with Attachments [SOAPAttach] specifications. This document provides security and reliability features necessary to support international electronic business. These security and reliability features are not provided in the SOAP or SOAP with Attachments specifications.

# ebXML Messaging Services

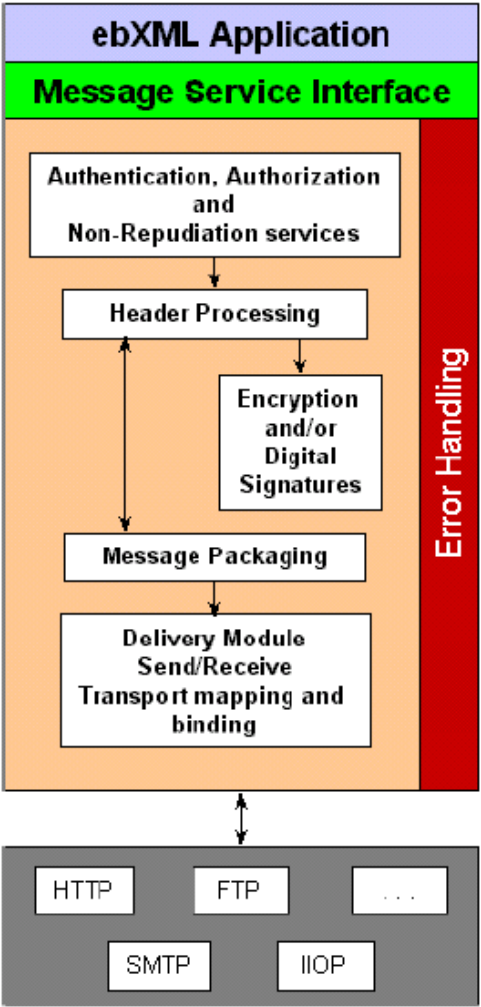


Figure 1.1 Typical Relationship between ebXML Message Service Handler Components

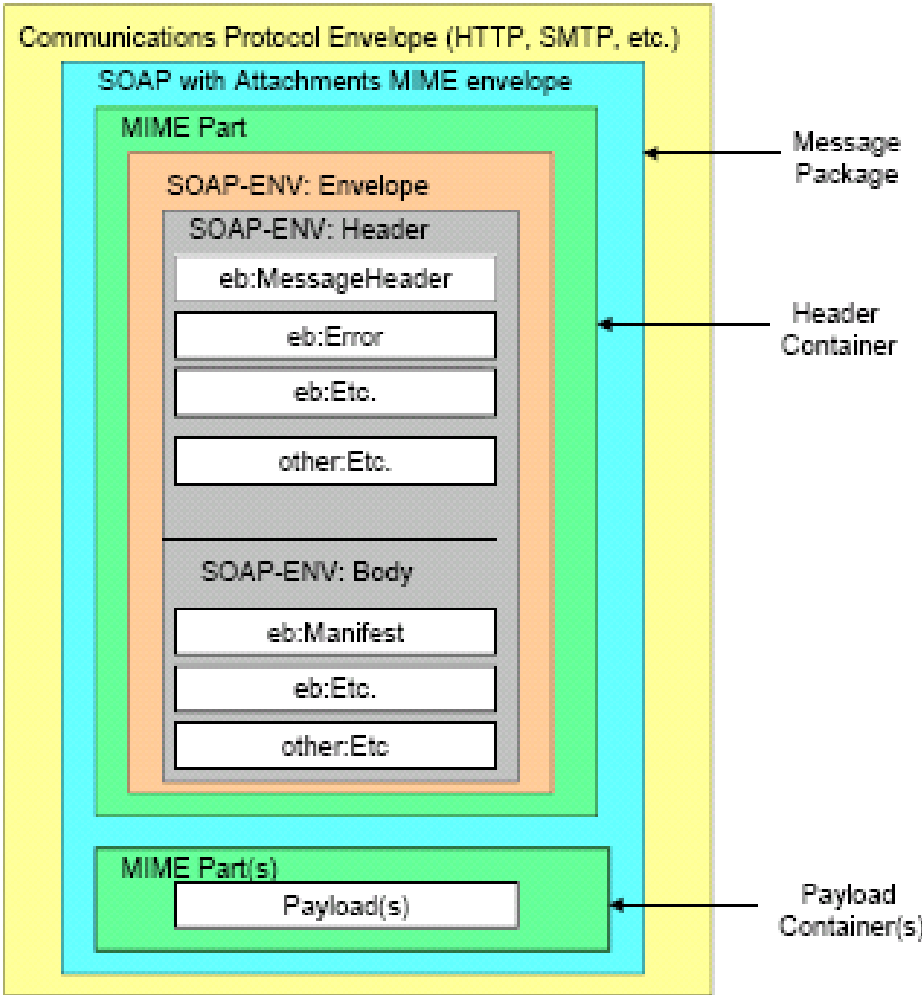
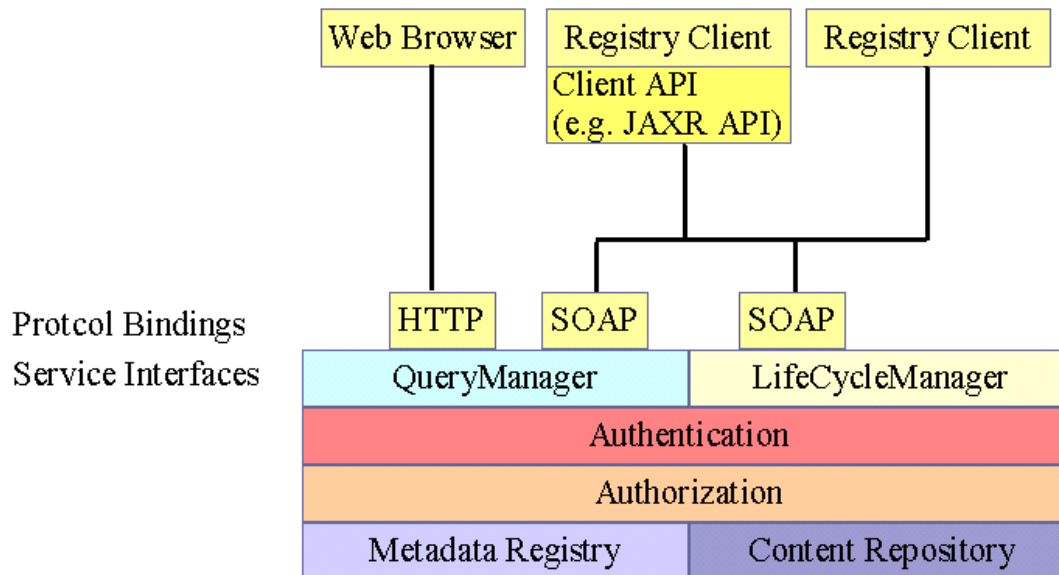


Figure 2.1 ebXML Message Structure

- Un répertoire ebXML est destiné au stockage de toutes sortes de contenu électronique tel que documents XML, textes, images, sons et vidéos. Les Instances de tels contenus sont définis comme entités de référence. Ces entités sont rangés dans un référentiel défini dans les répertoires ebXML





# ebXML CPPA Collaboration Protocol Profile and Agreement

Figure 1: Structure of CPP & Business Process Specification in an ebXML Registry

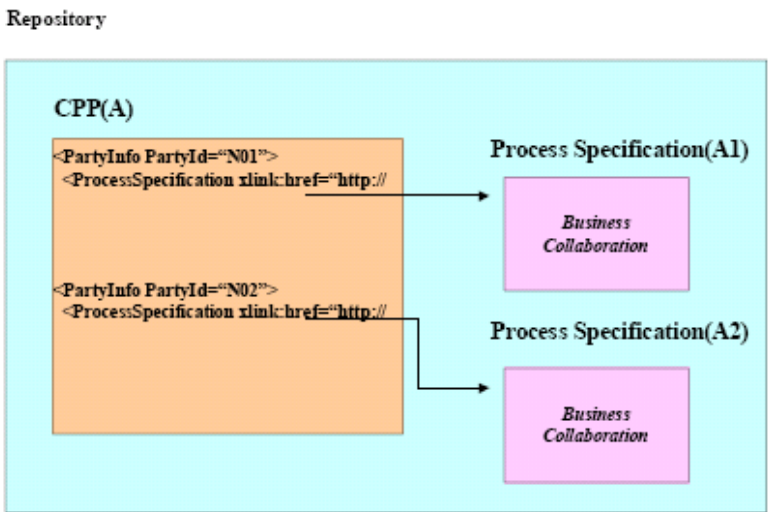


Figure 2: Overview of Collaboration-Protocol Profiles (CPP)

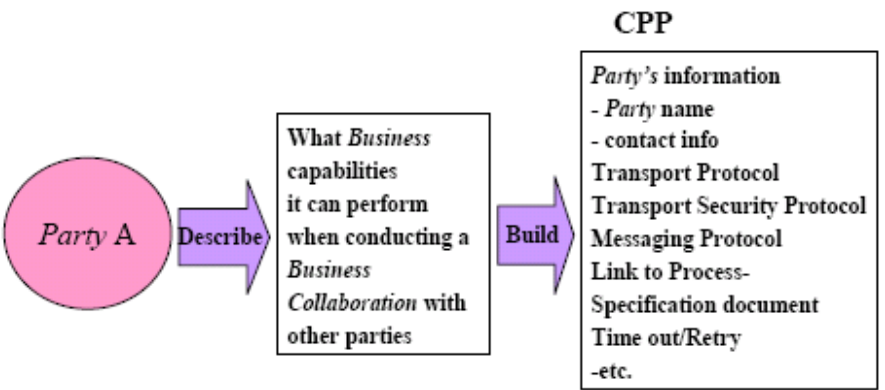
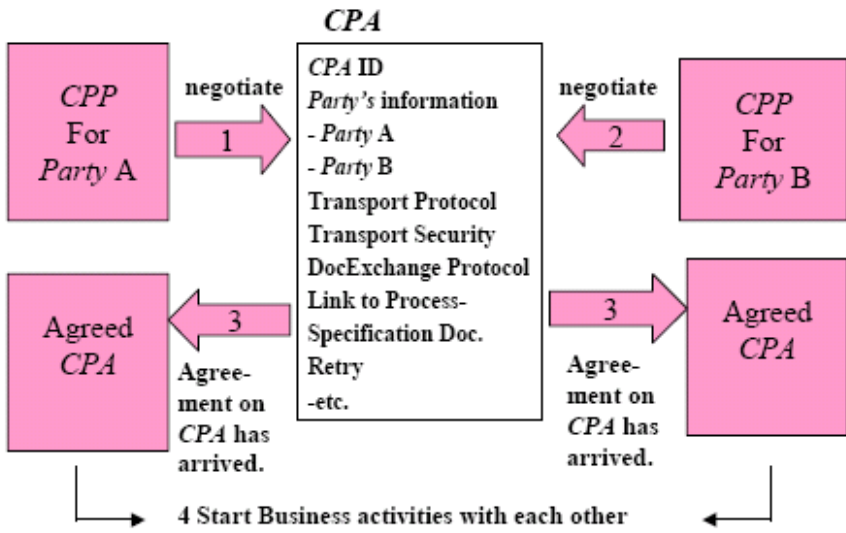
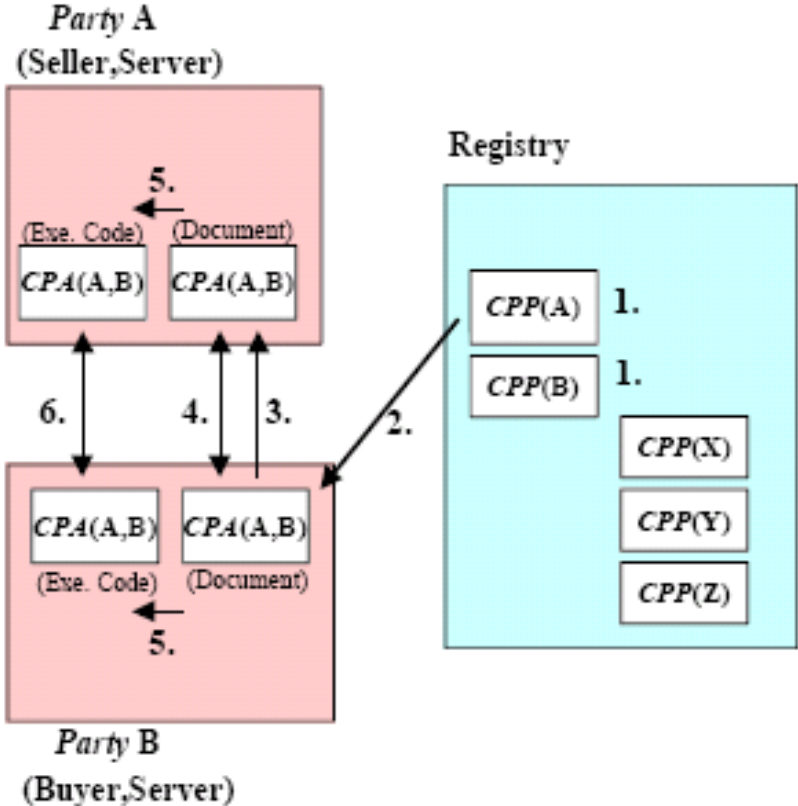


Figure 3: Overview of Collaboration-Protocol Agreements (CPA)



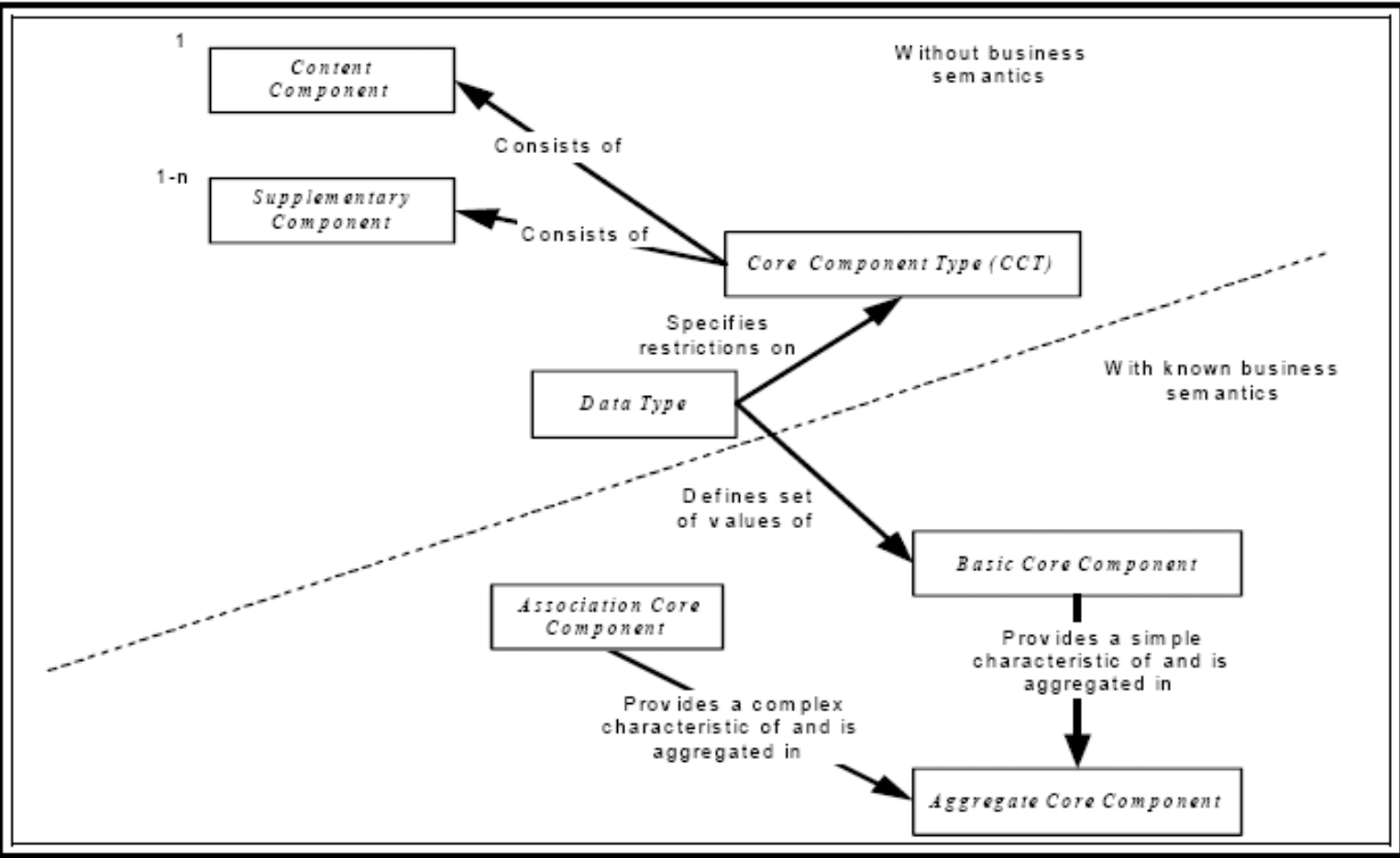
**Figure 4: Overview of Working Architecture of CPP/CPA with ebXML Registry**

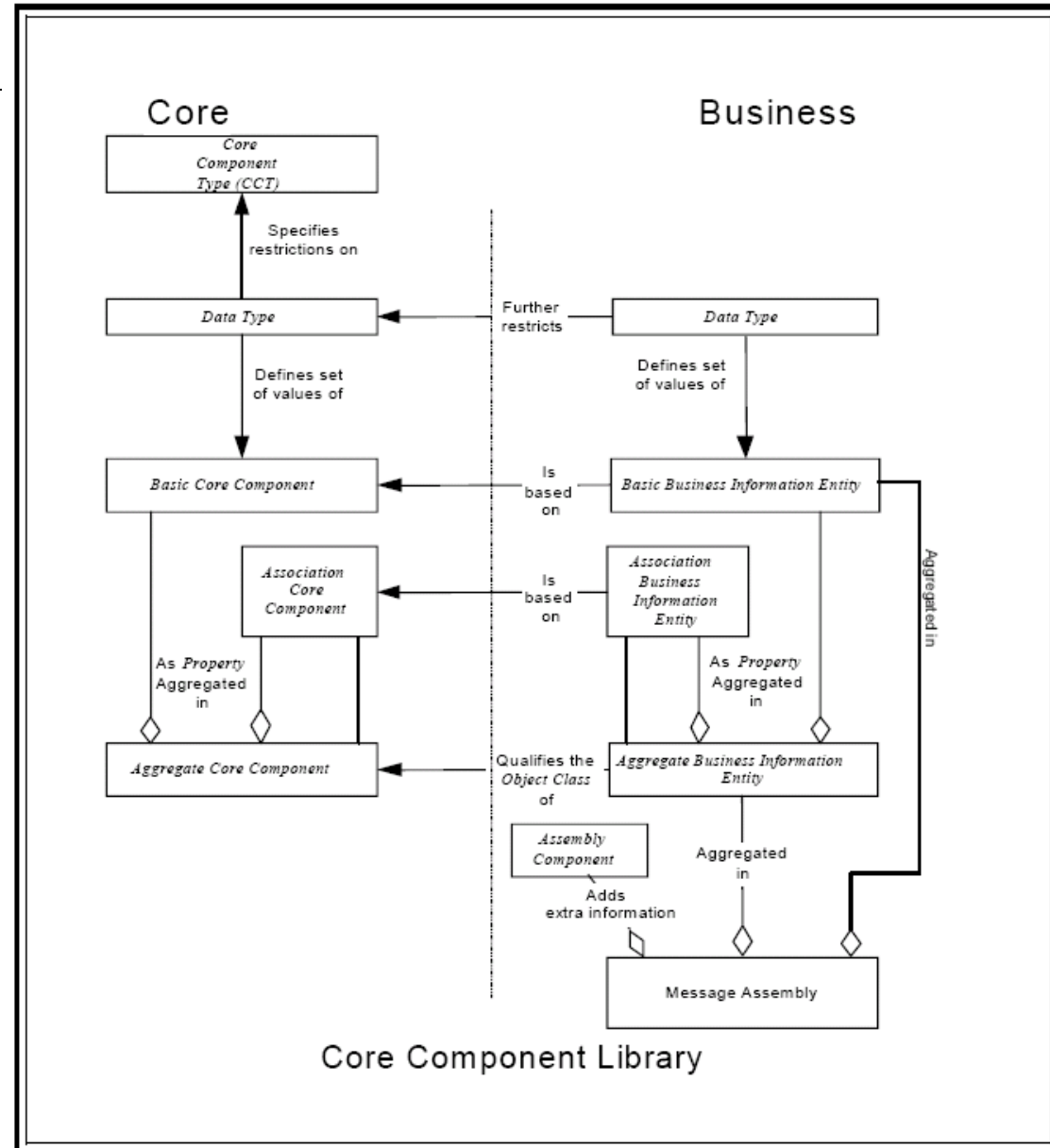
1. Any *Party* may register its CPPs to an ebXML Registry.
2. *Party B* discovers trading partner *A* (Seller) by searching in the Registry and downloads *CPP(A)* to *Party B*'s server.
3. *Party B* creates *CPA(A,B)* and sends *CPA(A,B)* to *Party A*.
4. *Parties A* and *B* negotiate and store identical copies of the completed *CPA* as a document in both servers. This process is done manually or automatically.
5. *Parties A* and *B* configure their run-time systems with the information in the *CPA*.
6. *Parties A* and *B* do business under the new *CPA*.



# Core Component Types de l'UN/CEFACT

- Méthodes de structuration pour établir des typologies communes de données
- En c





- A l'expérience UDDI s'avère insuffisant pour une exposition efficace de web services
  - Une conception informaticienne de librairies de services dont les conditions d'usage et la pertinence n'est pas définie
  - UDDI est une forme arborescente d'annuaire plate, inadaptée de recueil des connaissances sur les services disponibles.
  - UDDI ne fournit pas de logiques de choix d'accès aux services exposés
- Un service sur le web n'est qu'une ressource web comme une autre
  - Au même titre qu'une page HTML, il est éligible pour recevoir des métadonnées.
  - Pour qu'elles soient accessibles avec qualité (sans bruit et sans silence), les métadonnées de ressources web se structurent en connaissances
  - les connaissances s'organisent en graphes, et les graphes s'expriment en XML en XTM ou en RDF/RDFS/OWL...
- Les technologies du Web Sémantique peuvent elles répondre aux besoins des architectes d'applications et des urbanistes de systèmes d'information ?
- Les technologies du Web sémantique peuvent elles automatiser la construction dynamiques d'applications cohérentes ?

# Exemple de requête et de contenu UDDI

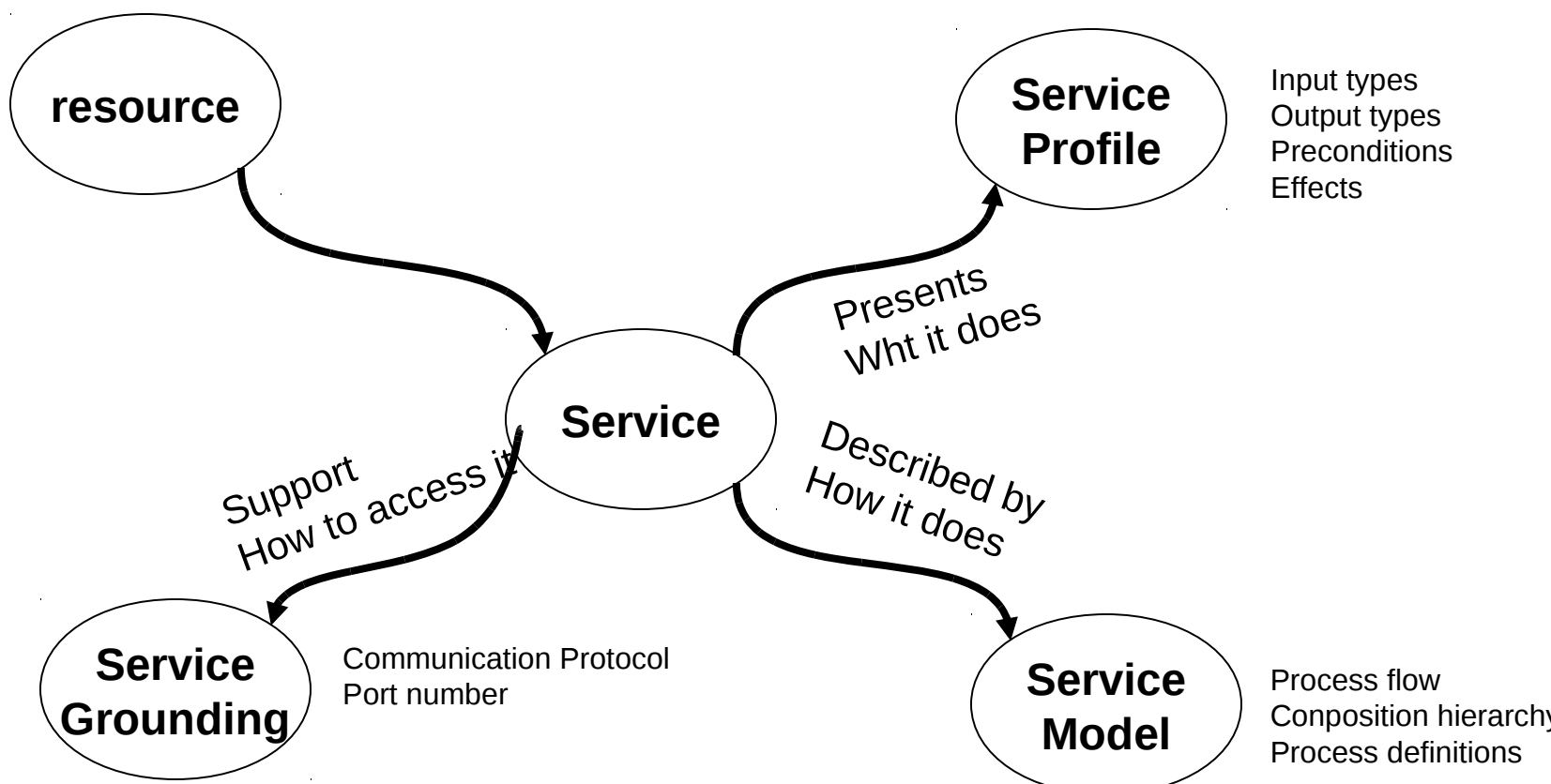
```
<?xml version="1.0" encoding="UTF-8"?>
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <find_service businessKey="*" generic="1.0" xmlns="urn:uddi-org:api" maxRows="5">
      <name>MaPetiteEntreprise</name>
    </find_service>
  </body>
</Envelope>

<businessList generic="1.0" operator="W3Québec" truncated="false" xmlns="urn:uddi-org:api">
  <businessInfos>
    <businessInfo businessKey="3894572309850239485723049857">
      <name> MaPetiteEntreprise </name>
      <description xml:lang="fr"> MaPetiteEntreprise fournit ... </description>
    <serviceInfos>
      <serviceInfo businessKey="1234567890" serviceKey="0987654321">
        <name>Exemple de service "Mon Service" </name>
      </serviceInfo>
    </serviceInfos>
  </businessInfo>
</businessInfos>
</businessList>
```

# Initiatives et Convergences 3 : OWL-S

- OWL-S est un acronyme de Web Ontology Language for Services
- OWL-S forme un jeu d'ontologies destinées à la description de la découverte de services
  - Service.owl
  - Grounding.owl
  - Profile.owl
  - Time-entry.owl
  - Resource.owl
  - Process.owl
  - ObjectList.owl
  - Expression.owl
  - ActorDefault.owl

# OWL-S : Concepts principaux





# OWL-S : Service Profiles, Process Model, Service Grounding

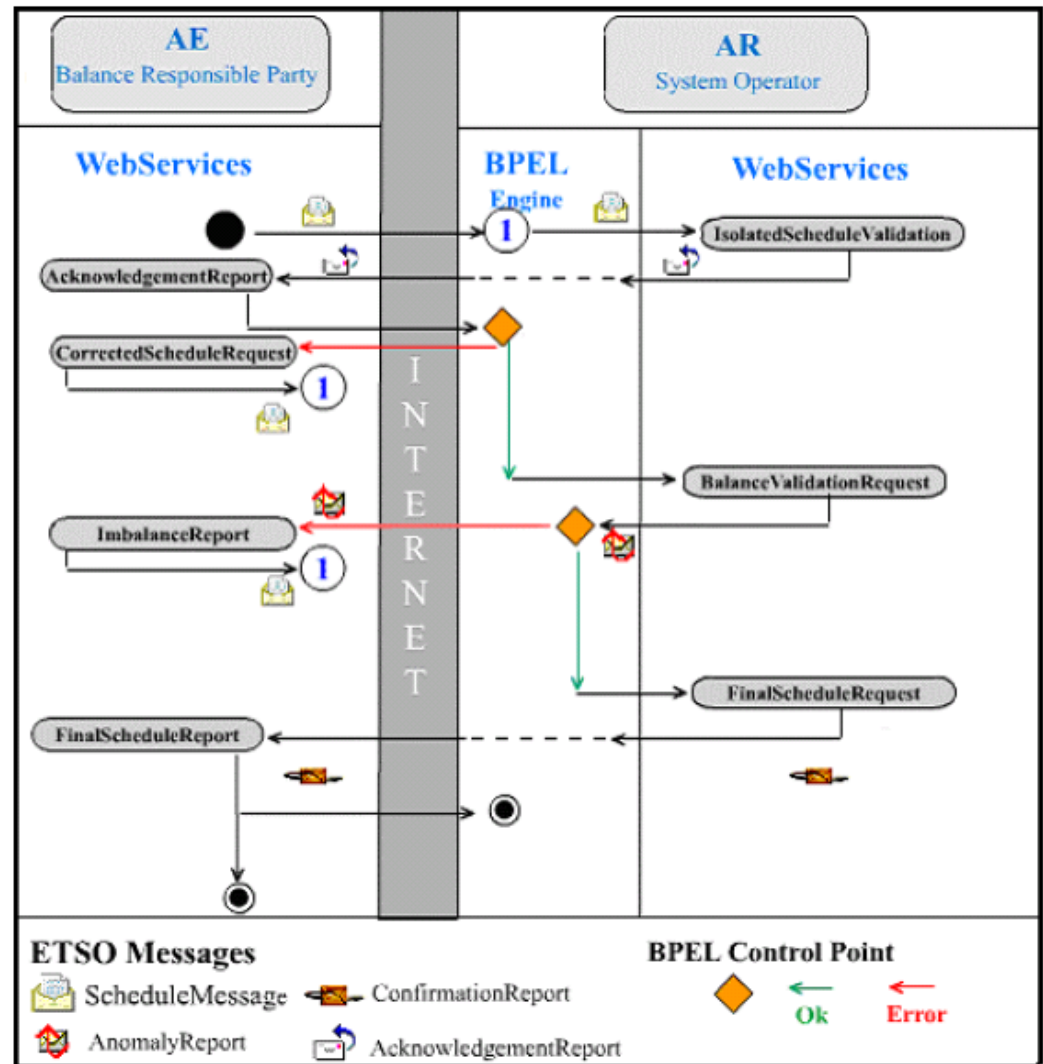
- *OWL-S Service Profiles*
  - Définit ce que produit un service
    - Entrée, sortie, préconditions, effets
  - Apporte des informations non opératoires: nom, catégorie, QoS,...
  - OWL-S permet théoriquement d'effectuer des calculs pour les préconditions et les effets
  - OWL-S est extensible : les profils peuvent s'organiser en hiérarchie de spécialisation.
- *OWL-S Process Model*
  - Définit la composition de services :
    - Séquence, concurrence: split, split+join,...
    - choix
    - Conditions, boucles
  - Gestion des flux d'informations
- *OWL-S Service Grounding*
  - Protocoles de communication
  - Accès effectif au service

- Initiatives de recherche sur la mise oeuvre dynamique (à la volée) de services
  - La découverte dynamique
  - L'organisation automatique des appels Pourquoi ?
    - choisir le meilleur service en fonction du contexte
    - remplacer un service défaillant par un autre
  - Composition avec
    - modèle "Romain"
    - modèle Mealy
    - modèle Golog
- Initiatives de recherche sur la validation d'architectures SOA
  - Identifications de deadlocks, garantie de service
  - Fiabilité, fonctionnement en mode dégradé
  - Monitoring/Model Checking

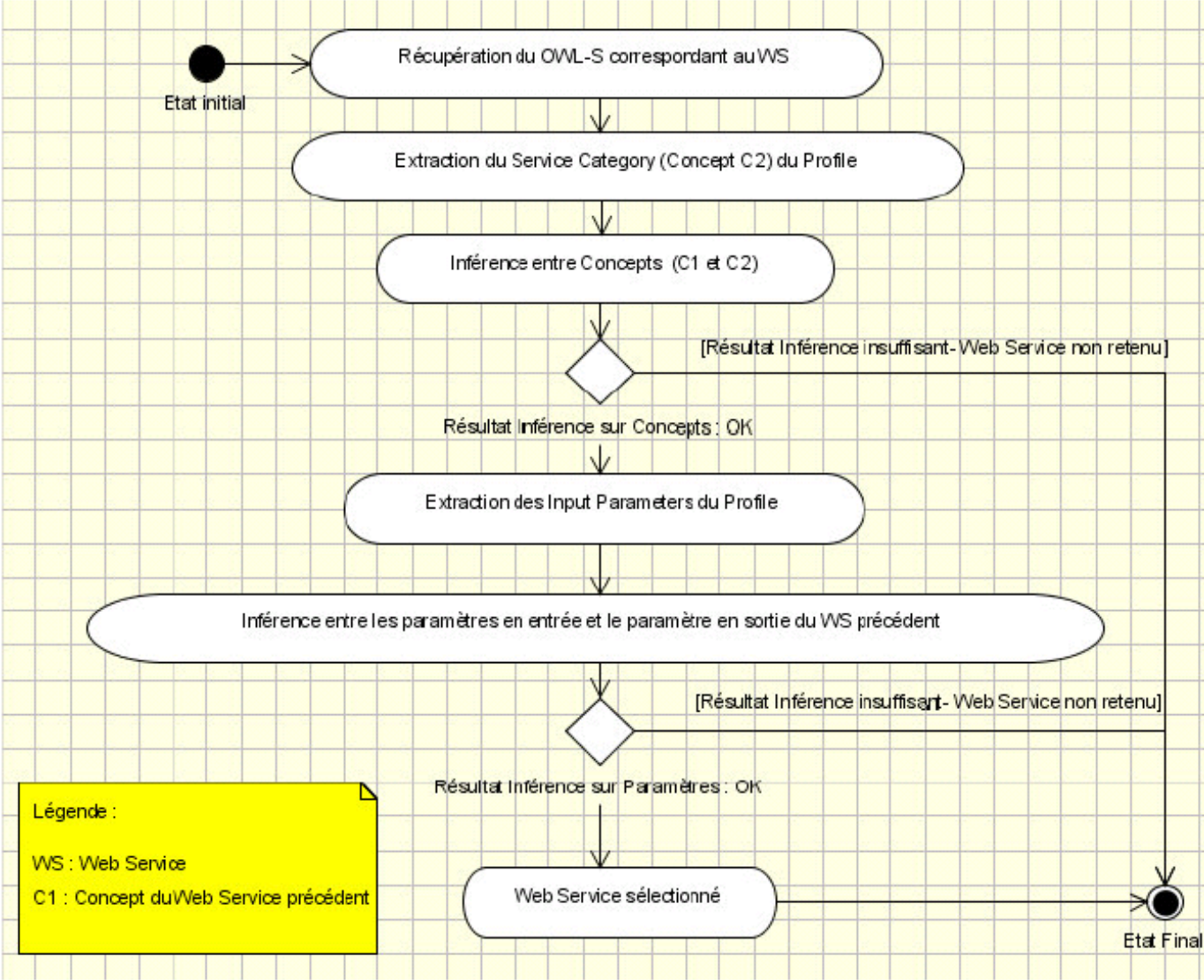
# Un exemple prototype d'utilisation de OWL-S

Figure 2 - Liste des Web Services intervenant dans la validation ESS.

- Orchestration de messages échangés pour la conduite de réseaux électriques
- Exemple issu du rapport de stage DESS effectué à EDF R&D par Hervé Pérez



# Diagramme d'états/transitions du prototype



# Inférences effectuées, architecture du prototype

Inférences entre les concepts paramètres sortie et paramètres entrée

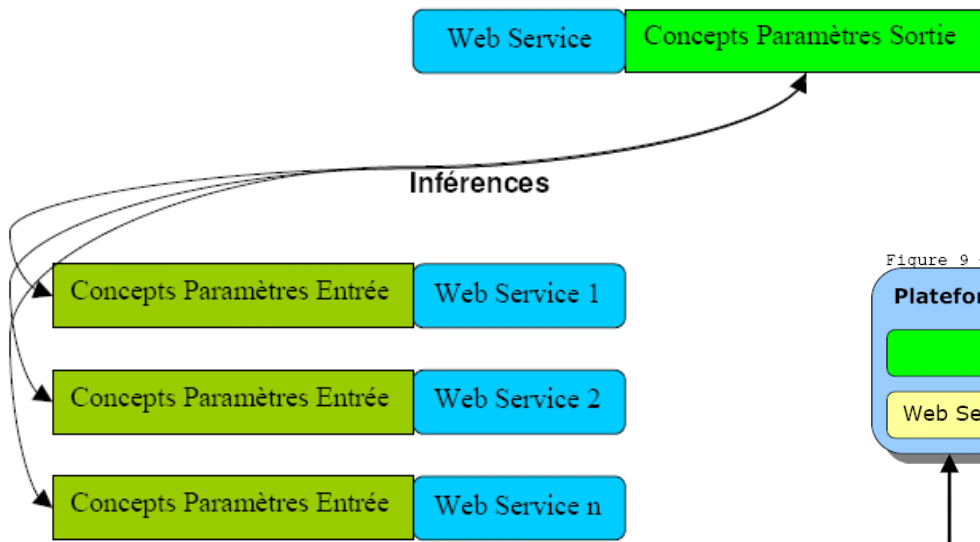
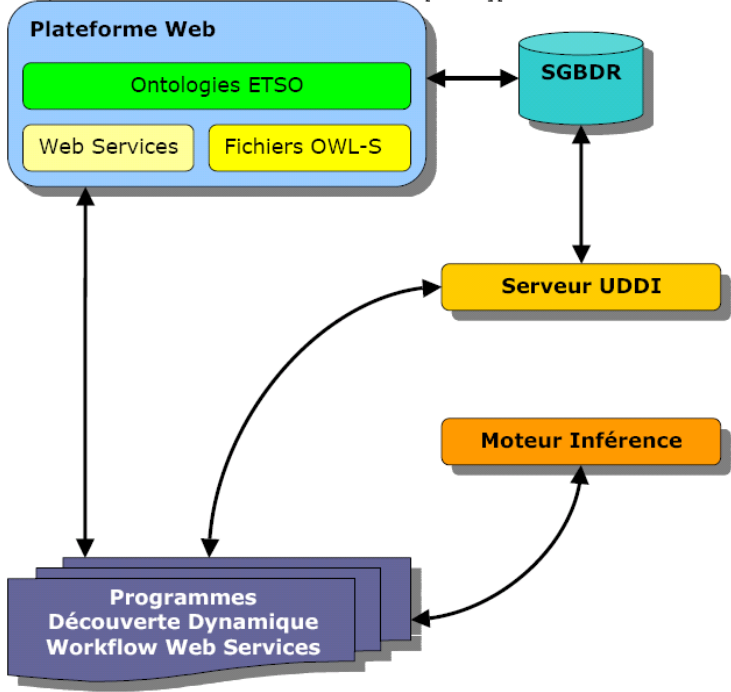
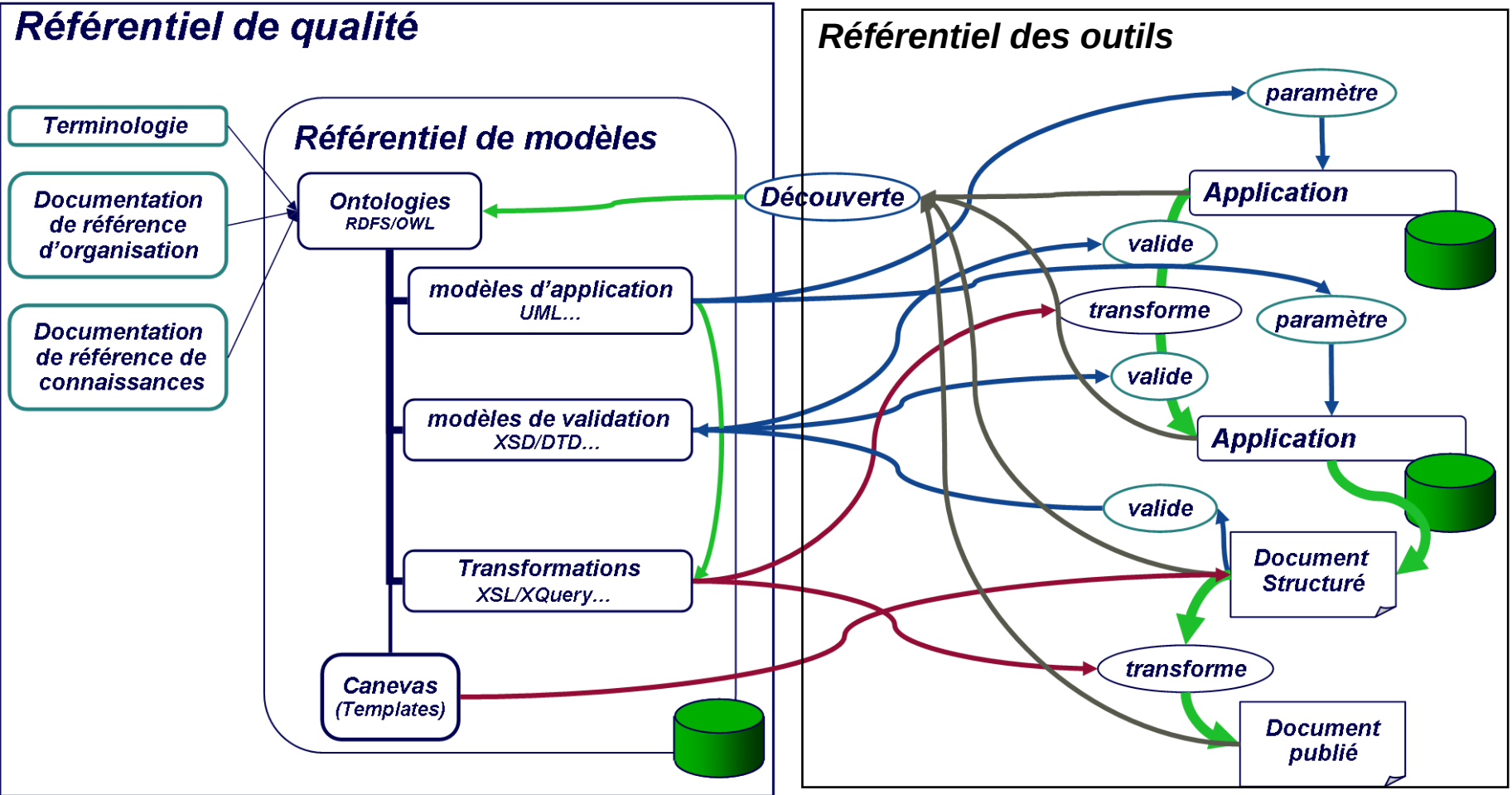


Figure 9 - Schéma de l'architecture du prototype



- Les principes des architectures orientées services impactent l'urbanisation des systèmes d'information
- Les technologies de couplage faible donnent une continuité entre les méthodes et outils de l'architecte et de l'urbaniste
  - Les architectures SOA ne disent rien sur la taille d'un service, ni sur sa largeur (étendue des actions possibles), ni sur sa hauteur (quantité de calculs effectués)
  - Une application peut être vue en soi comme un service, et exposée comme telle.
- Les démarches de maîtrise sémantique sont utiles et applicables à chaque niveau, de par l'identité de nature entre par exemple :
  - La découverte de service et la cartographie d'un système d'information
  - L'ordonnancement d'appels de services et les processus d'entreprise
  - ...

# services d'ontologies pour l'urbanisme de système d'information



Fin du module