

Systemes d'Information pour la Production

Introduction à l'Architecture Orientée Service
(Service Oriented Architecture – SOA)

E. Tranvouez (erwan.tranvouez@univ-amu.fr)

Quelques références

- ❑ Livre Blanc Softeam :
<http://www.softeam.fr/sites/default/files/files/Livre%20blanc-SOA%20Architecture%20Logique.pdf>
- ❑ Supports de cours I.A. Sadoune & O. Besnard (Supelec/Solucom) :
<http://idir.aitsadoune.free.fr/index.php/activites-denseignement>
- ❑ OpenGroup : <http://www.opengroup.org/standards/soa>
- ❑ Langage modélisation SOA de l'OMG :
www.omg.org/spec/SoaML

Propos Liminaires

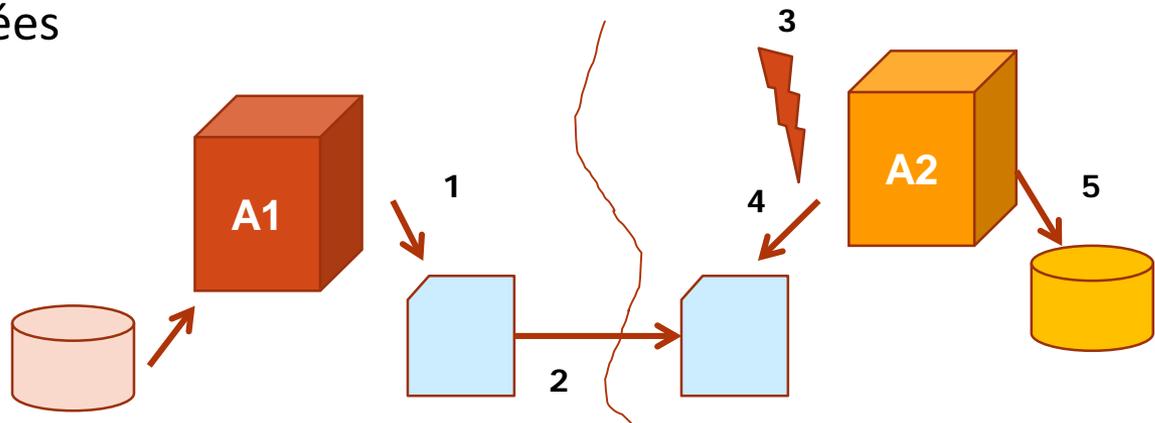
Retour sur l'intégration de SI

- ❑ Couplage de SI liés à leur spécialisation (CRM, ERP, SI métiers...)
- ❑ Nécessité d'échanger les informations => implique de définir :
 - **Mode de communication :**
 - Fichier
 - Via Connecteurs (ex. Connecteur JDBC MySQL)
 - Protocole/réseau
 - **Direction des flux :**
 - Flux monodirectionnel : modèle Client – Fournisseur
 - Flux bi-directionnel : échanges d'informations
 - **Configuration de l'intégration :**
 - Multi-points
 - Bus
 - Services

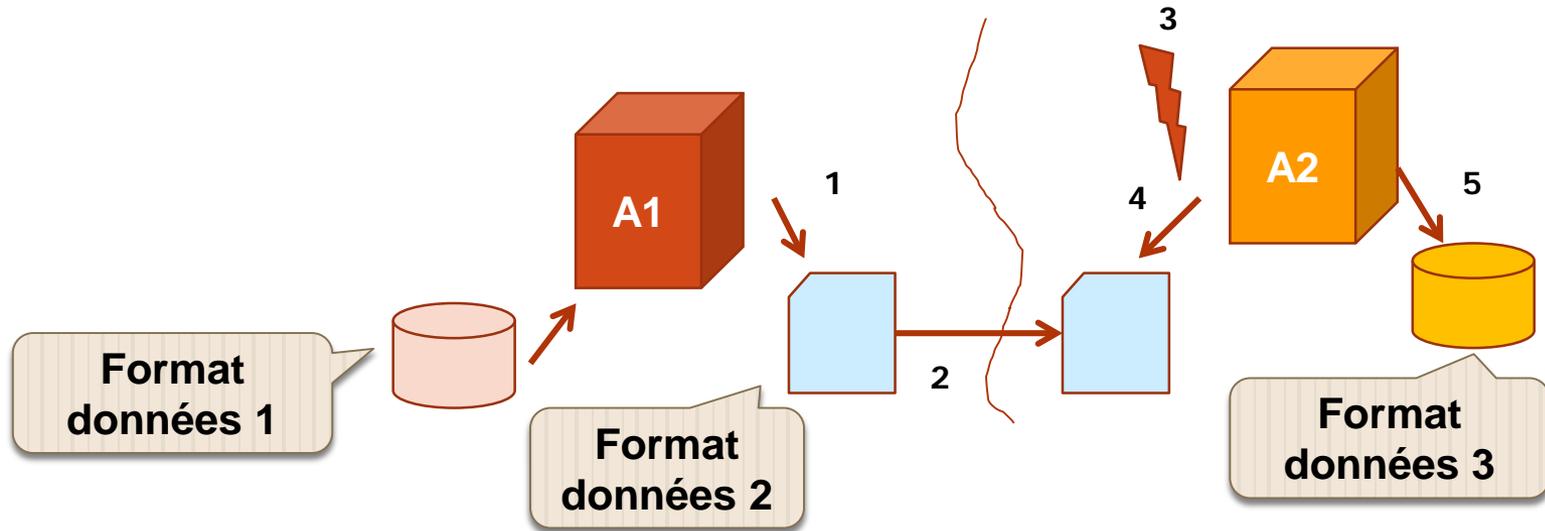
Retour sur l'intégration de SI : illustration échange par fichier(s)

❑ Illustration échanges données via fichier :

- Si fichier partagé : implique gestion droit d'accès/écriture (problème de l'accès concurrentiel à un fichier)
- Si « upload » (régulier) de fichier :
 1. Exportation données par application A1 (suppose format/contenu ok)
 2. Transport fichier (ftp par ex.)
 3. Evènement signalant la disponibilité du fichier (fréquence temporelle, événement applicatif)
 4. Chargement fichier par application 2
 5. Intégration données



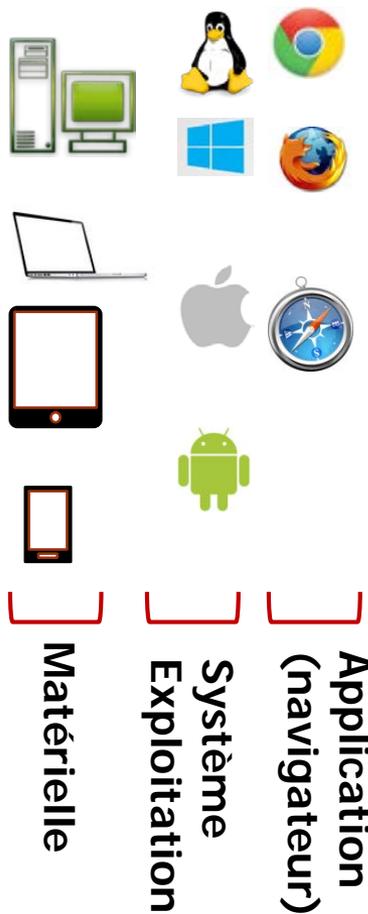
Retour sur l'intégration de SI : illustration échange par fichier(s)



- ❑ On retrouve ici l'ensemble des enjeux de l'intégration de SI :
 - ❑ Format d'échange de données
 - ❑ Protocole & Transport des données
 - ❑ Connectivité des SI
 - ❑ Mode de synchronisation (batch/temps réel)
 - ❑ Niveau de dépendance vis-à-vis des architecture interne des SI (A1 implémenté en un langage/dans un OS différents de A2).

Métaphore des Services Web : *Remember Internet ?*

Exemple d'architecture (simplifiée) d'un site d'e-commerce



TCP
HTTP
HTML



Server Web

TCP
SQL



Service Validat°
Paiement



SGBD

Variabilité

Remember Internet ?

□ HTTP :

- Protocole sans état (*pas de mémoire, traite les requêtes au fur et à mesure*)

- *Principe :*

1. *Connexion à un serveur (port 80 habituellement) via socket.*

2. *Envoi d'une requête HTTP. Ex :*

- *GET ressource localisation*

- *POST : envoie données au serveur web*

3. *Réception et traitement par le serveur*

4. *Renvoi du résultat (fichier HTML)*

```
GET /page.html HTTP/1.0
Host: example.com
```

- HTML+HTTP: **indépendants** ressources matérielles/logicielles (pour peu que l'application respecte les standards => W3C).

□ => Couplage faible :

- chaque application (le client, le serveur) garde sa logique fonctionnelle/applicative
- Pas de propagation des modifications applicatives (again, si respect standard) => Contre Ex. HTML5.

SI & Services ?

- ❑ On retrouve les enjeux suivants :
 - Comment faire communiquer 2 applications
 - Organiser cet échange (protocoles)
 - Caractériser l'information échangée
 - Gérer la complexité de l'intégration (propagation modifications/indépendance)
- ❑ Dans le cadre d'intégration d'applications (héritées, ouvertes)
- ❑ Parfois avec ouverture à l'extérieur (B2B, B2C)
- ❑ Dans un contexte financier contraint
 - Réduction des coûts de maintenance
 - Rationalisation de l'architecture SI de l'entreprise

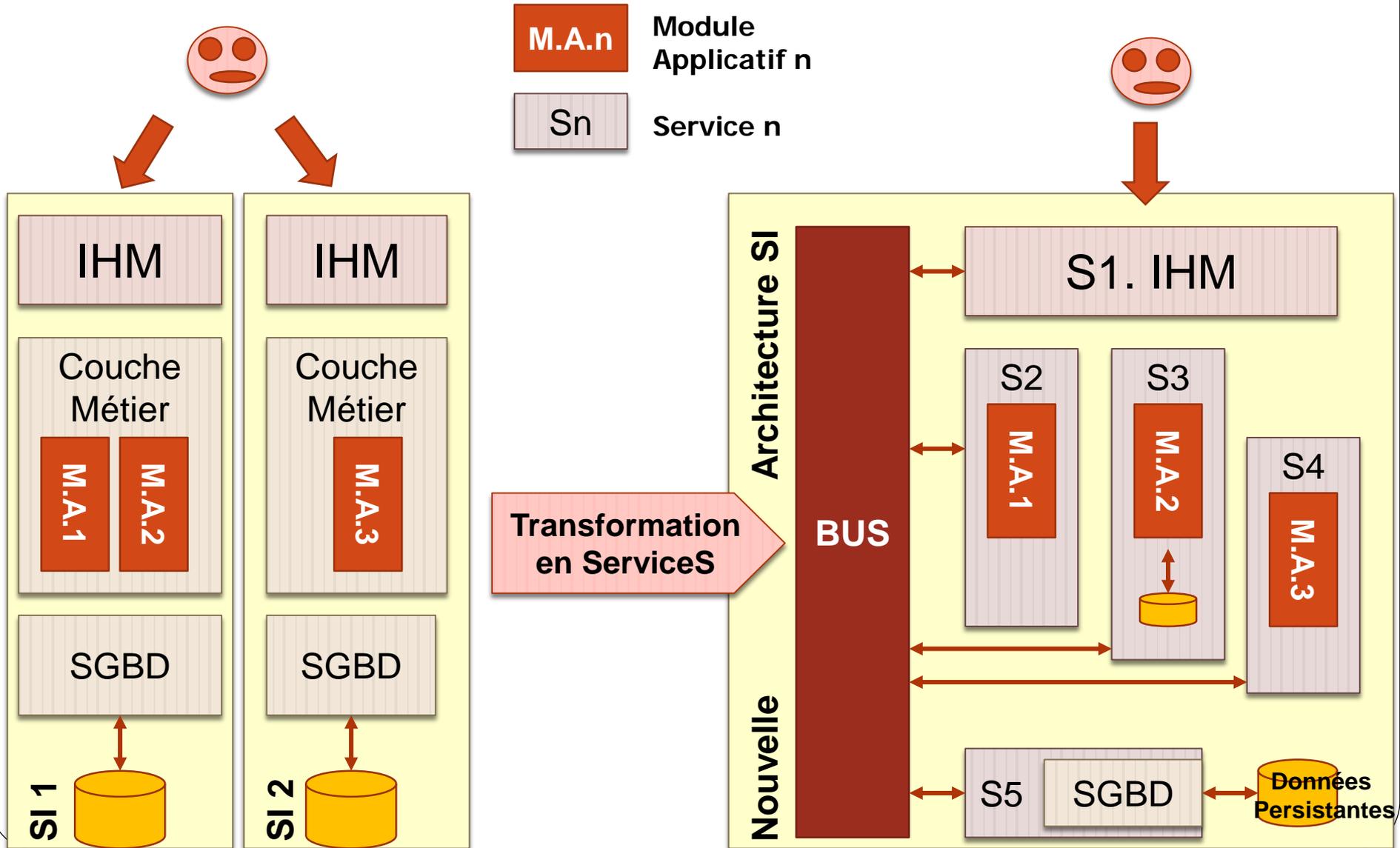
Présentation de l'Architecture Orientée Service (SOA)

Principe de l'approche SOA

- ❑ Reformulation de l'architecture SIs :
 - Les Applications SI deviennent des Services (voire sont décomposés en Services)
 - l'échange d'information entre applications devient des requêtes de **Services**
- ❑ Pas de standard officiel, mais tentatives de normalisation (OASIS, OpenGroup...) ou standards *de facto* (SOAP).
- ❑ Un service est alors :
 - Un module applicatif, un ensemble de traitements
 - Ouvert, répondant à des demandes d'information/traitements
 - Communiquant via une interface standard
 - Consommateur/producteur de requêtes (ou les deux)
- ❑ Le SI Global est alors construit comme un assemblage de Services en interaction (// avec POO).

Principe de l'approche SOA :

exemple de «*Servicification*» (terme anglais) de SI



Principe de l'approche SOA

- ❑ Approche pouvant intervenir pour :
 - L'Intégration de SI existants
 - Interactions entre SI supportées par une architecture orientée service
 - La conception de SI:
 - Identification de modules composant les SI réutilisables
 - Transformation de ces modules en Services
 - Indépendance de l'architecture SIA* des implémentations des modules
 - revient ~ à transformer des appels de fonction à des envois de requêtes
 - => amélioration modularité, réutilisation, maintenance des modules.
 - La Re conception de SI: càd restructuration de l'architecture SI existante.
 - Idem que conception mais appliqué à un existant
- ❑ Dans un contexte industriel :
 - de fusion d'entreprises (intégration de SI métiers existants)
 - d'optimisation des coûts (rationalisation, + ce qui suit)
 - d'ouverture de leur SI (B2B, B2C => production de Valeur)
 - Agilité des SI (gestion des évolutions)

Ex. Fusion Air France KLM

❑ Quelques chiffres

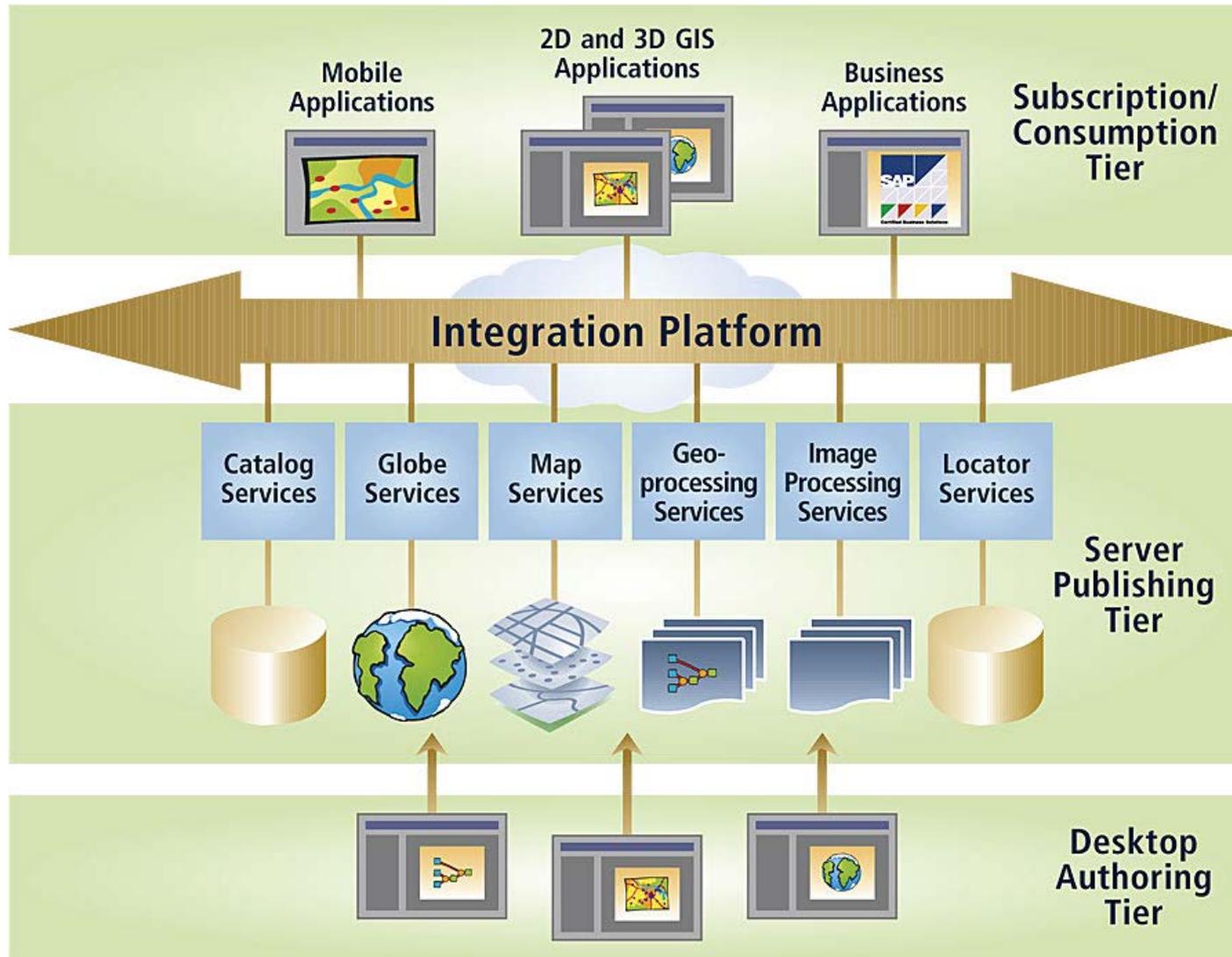
- Fusion en 2004
- => la 1^{ère} compagnie aérienne mondiale en CA, 3^{ème} en nb de passagers
- Suivie par un contexte économique défavorable :
 - Crise de 2008, concurrence compagnie low cost, développement des compagnies asiatiques ... => 2013 5^e compagnie en CA.

❑ Projet d'intégration des SI des 2 compagnies

- Projet lancé en 2006 (durée > 2 ans ?)
- Budget initial de 200 M€ (final ?)
- Objectifs :
 - Ciblage processus de réservation, gestion de capacités (avion)
 - 70% d'applications communes (sur 400 applications)
- Approche (processus) métier !
- => SOA

❑ Quels enjeux identifiez vous ?

Exemple d'intégration par service : Systèmes d'Information Géographiques



Concept SOA : *Service*

□ Définitions de Service

■ OpenGroup – (www.opengroup.org)

- *Is a **logical representation of a repeatable business activity** that has a **specified outcome** (e.g., check customer credit, provide weather data, consolidate drilling reports)*
- *Is self-contained*
- *May be composed of other services*
- *Is a "black box" to consumers of the service.*
- => Séparation du service « rendu » de son implémentation. Insiste sur l'aspect « composant » / Encapsulation

■ OASIS (www.oasis-open.org)

- *A mechanism to enable access to one or more **capabilities**, where the access is provided using a prescribed **interface** and is exercised consistent with **constraints** and **policies** as specified by the **service description**.*
- => insiste sur l'aspect description du service ~signature d'une fonction/méthode publique

Concept SOA : Contrat de service & Annuaires

- ❑ **Contrat de service / Interface** : description externe d'un service. Définit les termes régissant l'utilisation du service :
 - Description fonctionnelle du service (ce qu'il fait)
 - Règles et contraintes d'utilisation du service (pré conditions par ex.)
 - Qualité de service (QoS) : engagement en terme de sécurité, disponibilité, performance (e.g. tps de réponse moyen d'une requête, plages horaires), ... => WSDL (cf. SOAP)
- ❑ **Annuaire de Service** :
 - permet de trouver un service répondant à un besoin (données issues des contrats de service) => UDDI (Universal Description Discovery and Intégration – OASIS)
- ❑ **Enjeux ?**
 - D'un point de vue logique métier, permet d'estimer les besoins en terme de ressources (matérielles/logicielles)
 - D'un point de vue composition de services :
 - (Idéalement) composition de services dynamique (*à la volée*) impliquant éventuellement une optimisation multicritère (couverture du besoin, disponibilité etc..)

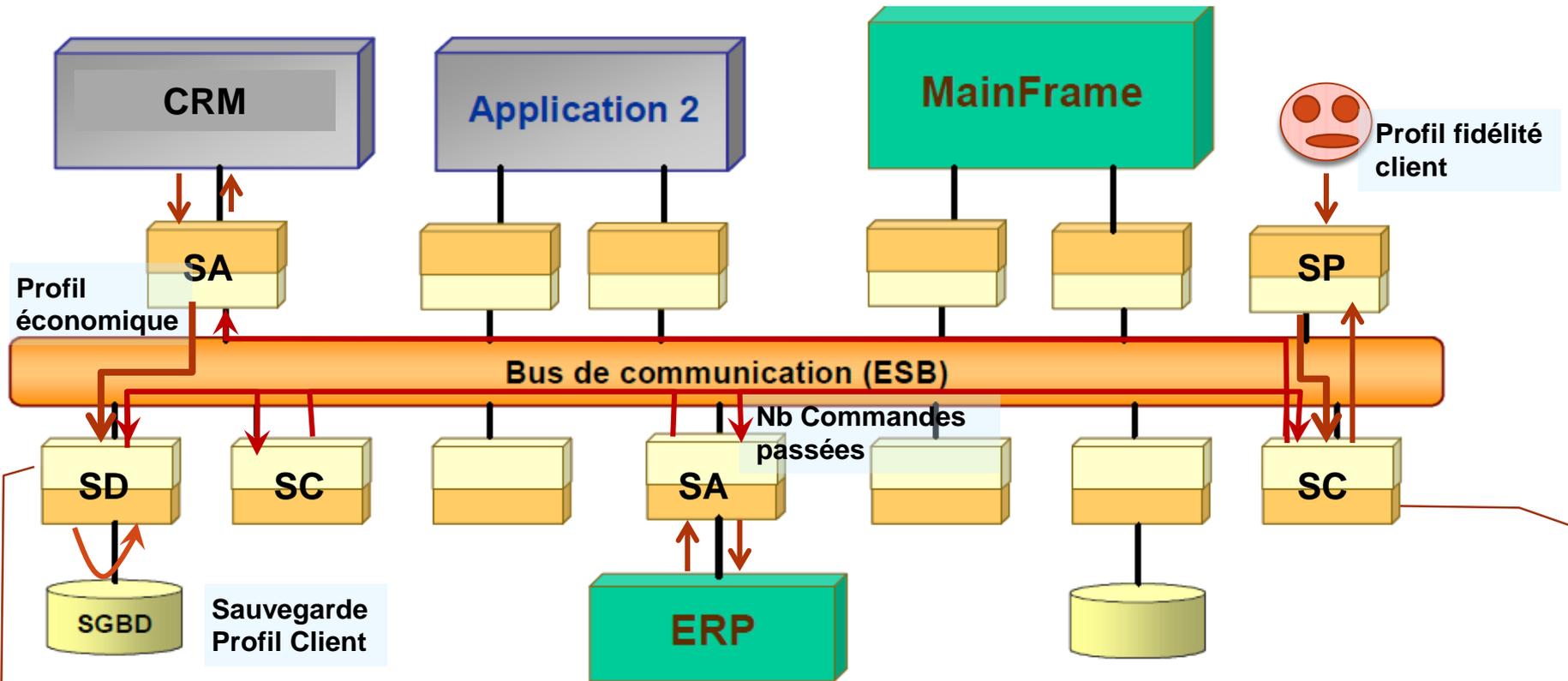
Concept SOA : Données

- ❑ **Un service ne tient pas compte de l'état des autres services** (// serveur web) ...
- ❑ **... mais l'état peut être passé en paramètre** afin d'avoir un traitement contextuel (vérification respect protocole)
- ❑ **Suppose format de données partagé entre les services**
- ❑ **Données transférées via messages** (format XML en général)
- ❑ Possibilité d'avoir des données partagées via SGBD commun (données persistantes)

Concept SOA : Typologies de Services

- ❑ L'invocation de service peut être réalisé via SOAP
- ❑ Typologies non standardisées mais on retrouve des préoccupations communes :
 - **Services de Présentation** : dialogue utilisateur comparable à l'approche 3 tiers dans les architecture S.I.
 - **Services de chorégraphie** : Service assurant l'enchaînement des requêtes auprès d'autres services (couche « métier »)
 - **Services applicatifs** : service assurant un traitement par un module applicatif / SI
 - **Service de Données** : Service assurant la récupération d'information (accès aux données persistantes)
- ❑ Le transport des messages peut s'appuyer sur HTTP (on parle alors de Web Services) avec une technologie
 - REST (Representational State Transfer) sur laquelle est basée internet (paramètres dans la requête HTTP)
 - ou SOAP (Simple Object Access Protocol) permettant l'envoi de message structurés (XML)

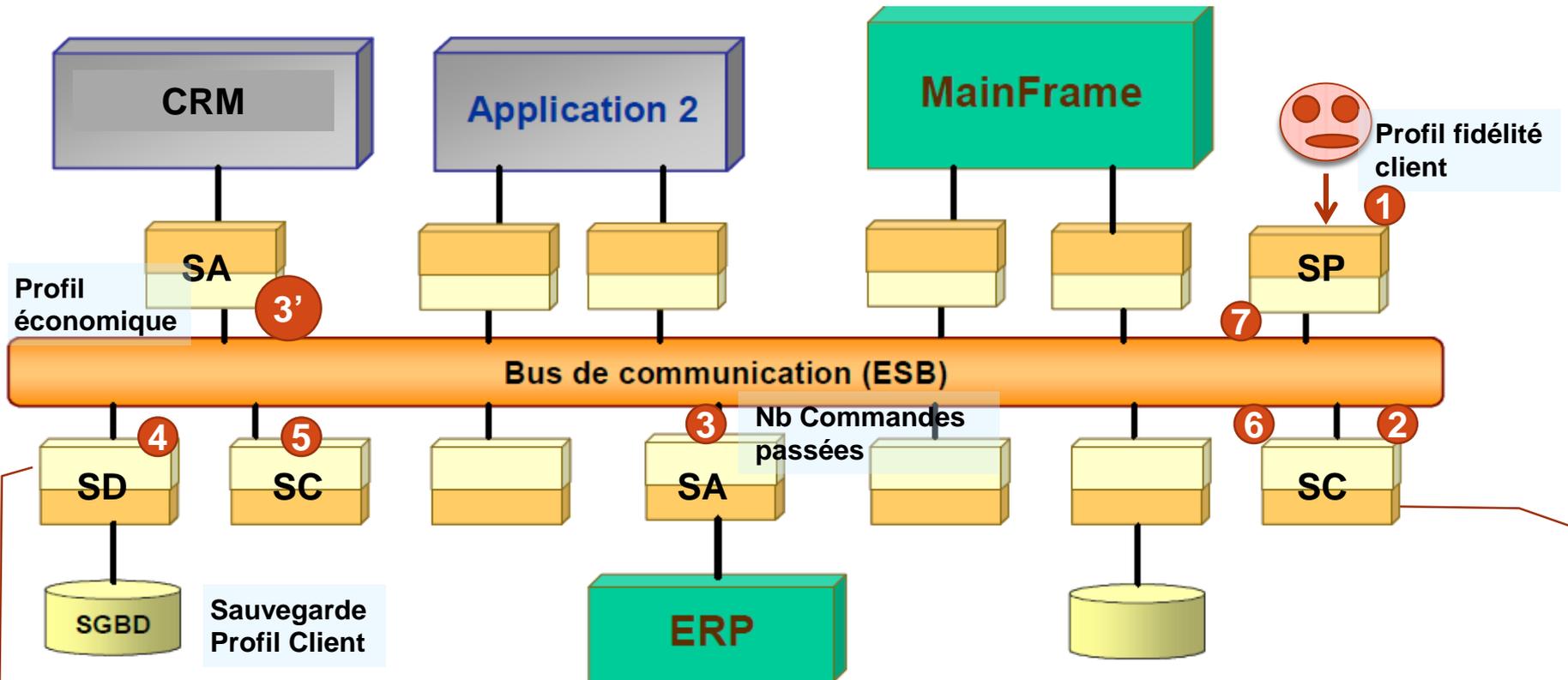
Illustration d'intégration d'applications par SOA



Vue EXterne : spécification externe du service (contrat de service)

Vue INterne : non visible des services consommateurs (boîte noire).
Décrit les ressources logicielles du service (applications), lien de dépendance entre services, protocoles ...
Inclus lien physique et traitements données (rôle de connecteur)

Illustration d'intégration d'applications par SOA (idem sans animation)



Vue EXterne : spécification externe du service (contrat de service)

Vue INterne : non visible des services consommateurs (boîte noire).
 Décrit les ressources logicielles du service (applications), lien de dépendance entre services, protocoles ...
Inclus lien physique et traitements données (rôle de connecteur)

Concept SOA : Explain the joke ...

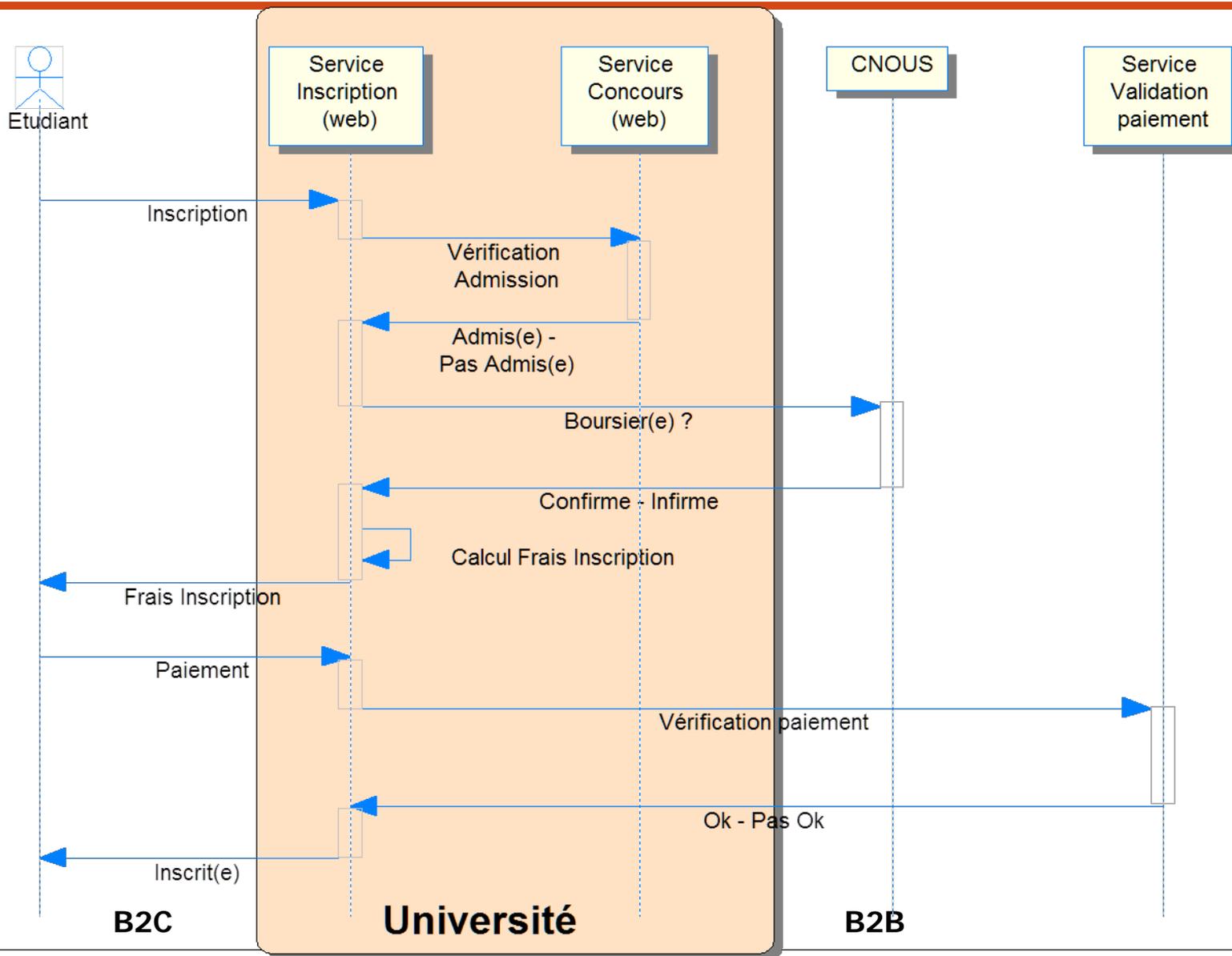


Concept SOA - Illustration sur processus inscription à l'université

□ Illustration :

- Modélisation du processus d'inscription en ligne impliquant :
 - validation de la légitimité de l'inscription (résultat concours)
 - Validation du statut de boursier (=> CNOUS)
 - Validation paiement (service extérieur)
 - Délivrance justificatifs inscription.
- ... avec un diagramme de séquence UML.
Les lignes de vies représentant un service...
- Remarques :
 - Auparavant, inscription piloté par Personnel de l'université (étudiant présente documents, personnel saisie informations)...
 - ... « Maintenant » : saisie Information par étudiant => B2C, économie temps saisie => changement organisationnel.
 - Service concours simplifié : on peut avoir un ou plusieurs applications gérant les concours (APB, SCEI, Concours sur titre...), certaines internes à l'université (périmètre SI de l'université => intégration), d'autres externes (B2B).

Concept SOA - Illustration sur processus inscription à l'université



Modélisation Architecture Service

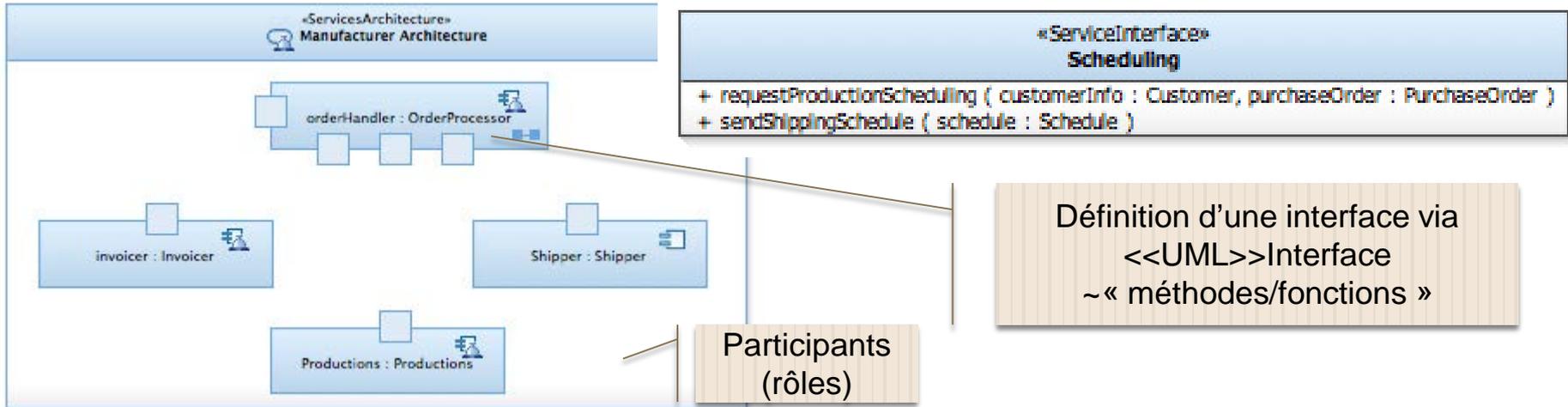
❑ Nombreux standards :

- ToGaf de l'Open Group
- SOAML (SOA Modeling Language) de l'OMG

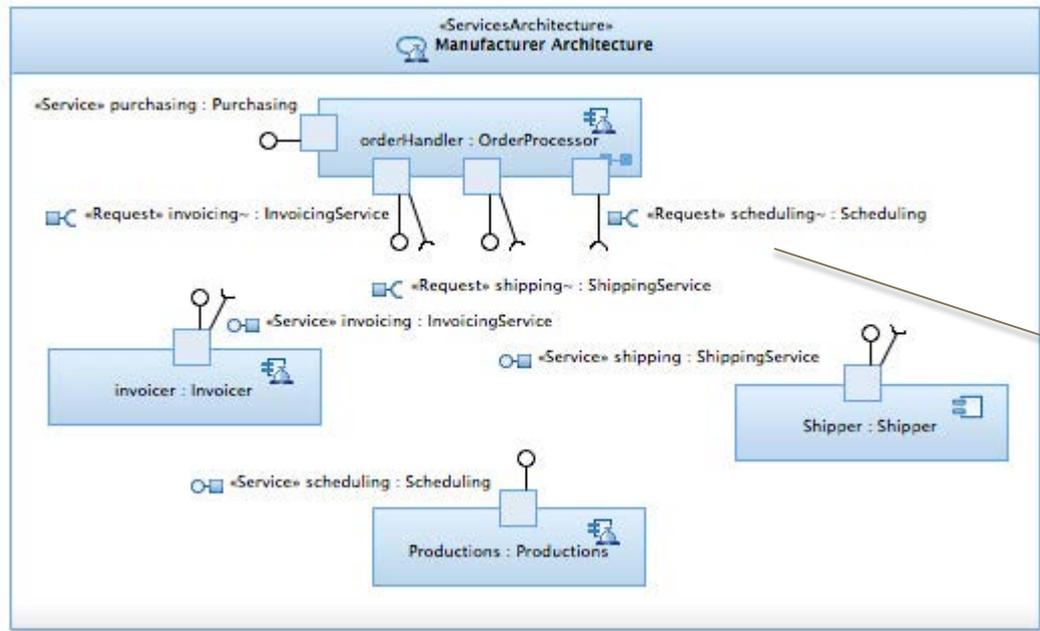
❑ SOAML :

- Extension d'UML appliqué à SOA (diagramme de classe, de séquence & activité...)
- Plus complexe (ou riche selon le point de vue) que BPMN, puisqu'on ajoute des contraintes, contexte d'utilisation plus détaillées (BPMN = processus métiers ~MOT de Merise, SOAML = spécifications architecturale ~MLT de Merise)
- Les services sont décrits :
 - Par leurs entrées/sorties i.e. Services proposés (requêtes acceptées) et services requis (requêtes qu'ils enverront) reliés à des ports (// serveur web => port 80).
 - Par leurs interfaces, leurs contrats

Modélisation Architecture Service : illustration SOAML (IBM)



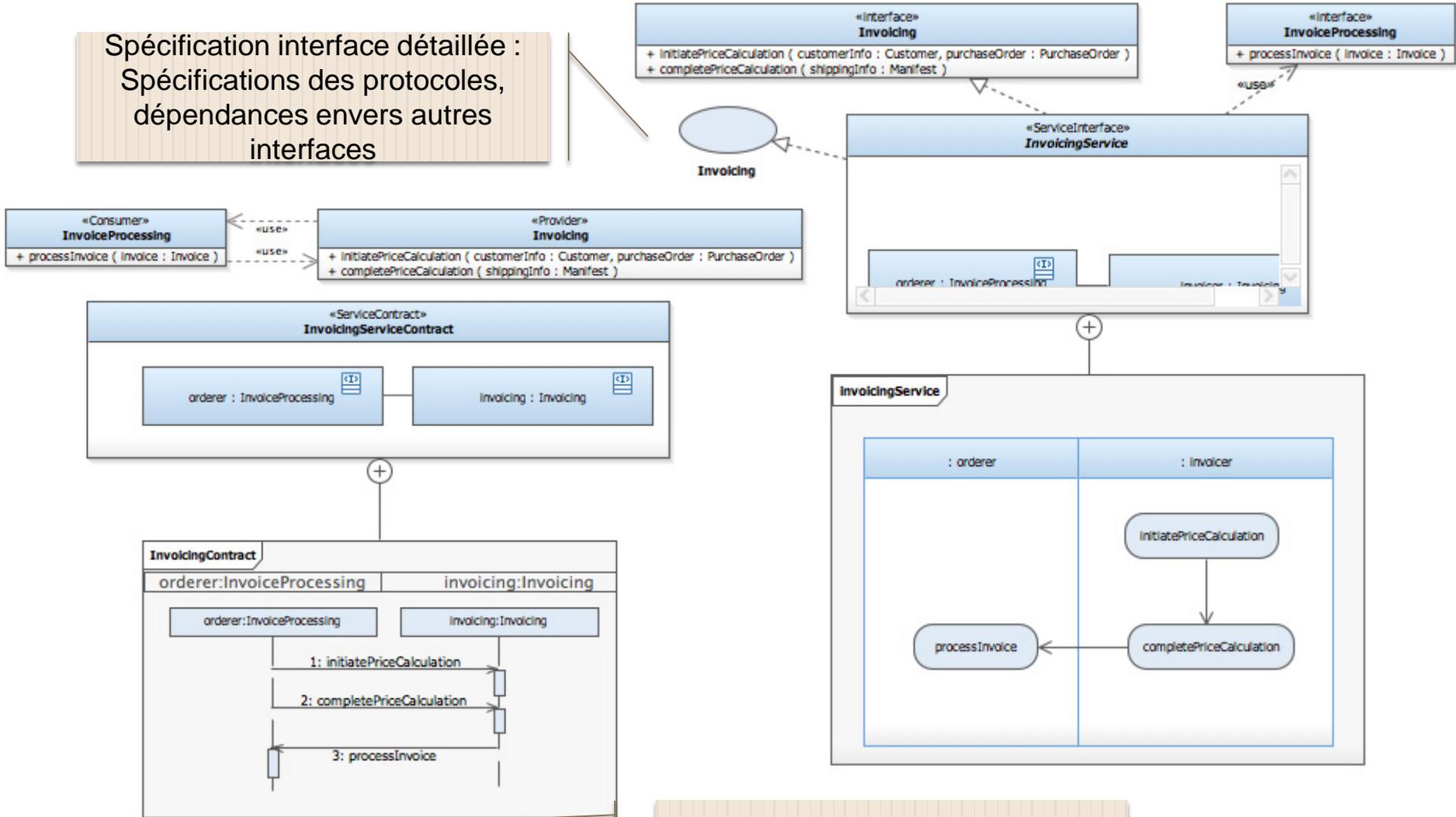
Définition d'une interface via <<UML>>Interface ~« méthodes/fonctions »



Participants(rôles) avec les services qu'ils consomment et ceux qu'ils proposent

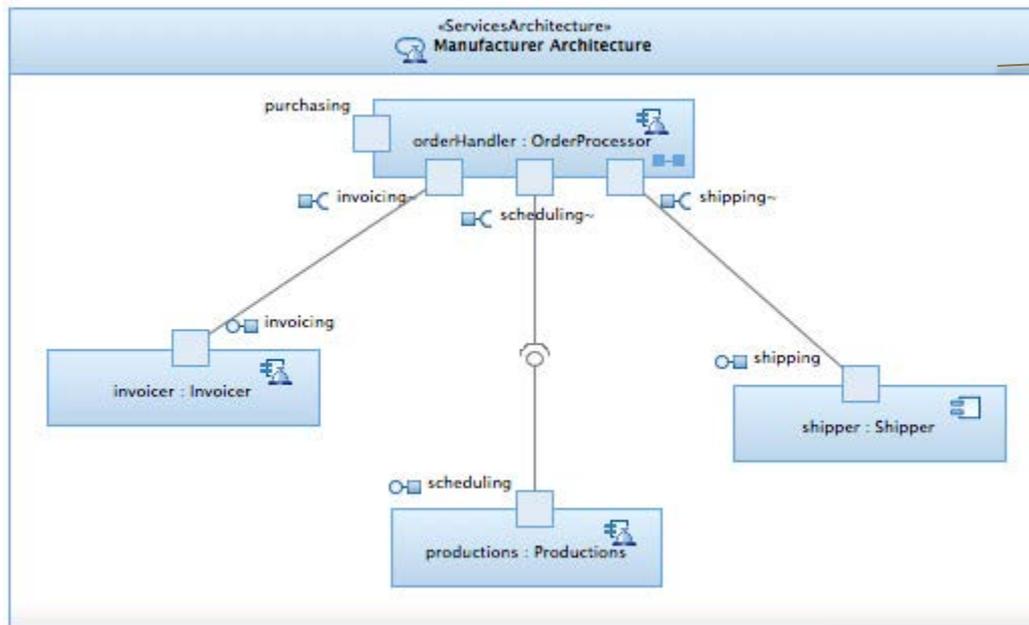
Modélisation Architecture Service : illustration SOAML (source IBM)

Spécification interface détaillée :
Spécifications des protocoles,
dépendances envers autres
interfaces



Spécification contrats de service

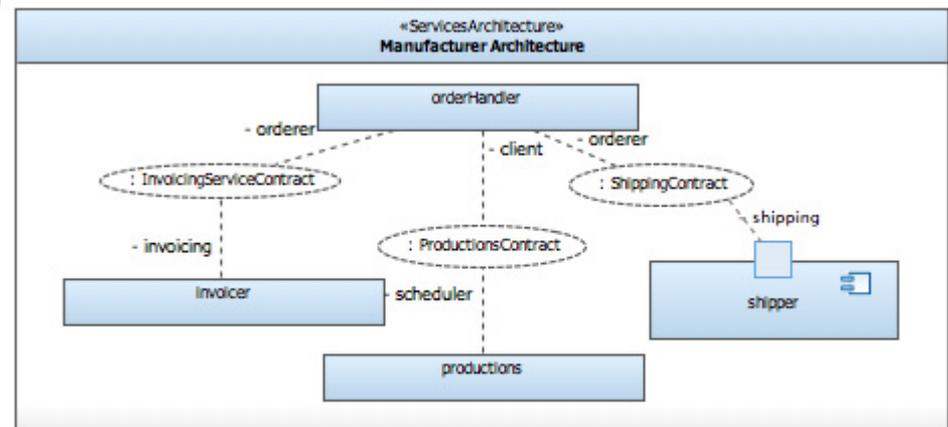
Modélisation Architecture Service : illustration SOAML (source IBM)



Spécification contrats de service

Etc...

Cf. outils de modélisation
www.modelio.org



Mise en œuvre de projets SOA

- ❑ A connu une mauvaise presse (après un buzz) suite à nombreux échecs dont les causes fréquemment nommées sont :
 - Approche technologique de l'intégration/conception sans prise en compte des enjeux (processus) métiers
 - => modélisation des processus métiers (**BPMN !**) et alignement avec les SI existants pour identifier les couplages et/ou décompositions des fonctions SI ... et **non l'inverse !**
 - Sous-estimation de la complexité du projet
 - Sous-estimation de l'impact organisationnel
 - Implication direction
 - Estimation de la maturité de l'entreprise pour une démarche SOA => cf. ci-après.

Sentiment
de déjà vu ?

OpenGroup Service Integration Maturity Model (OSIMM)

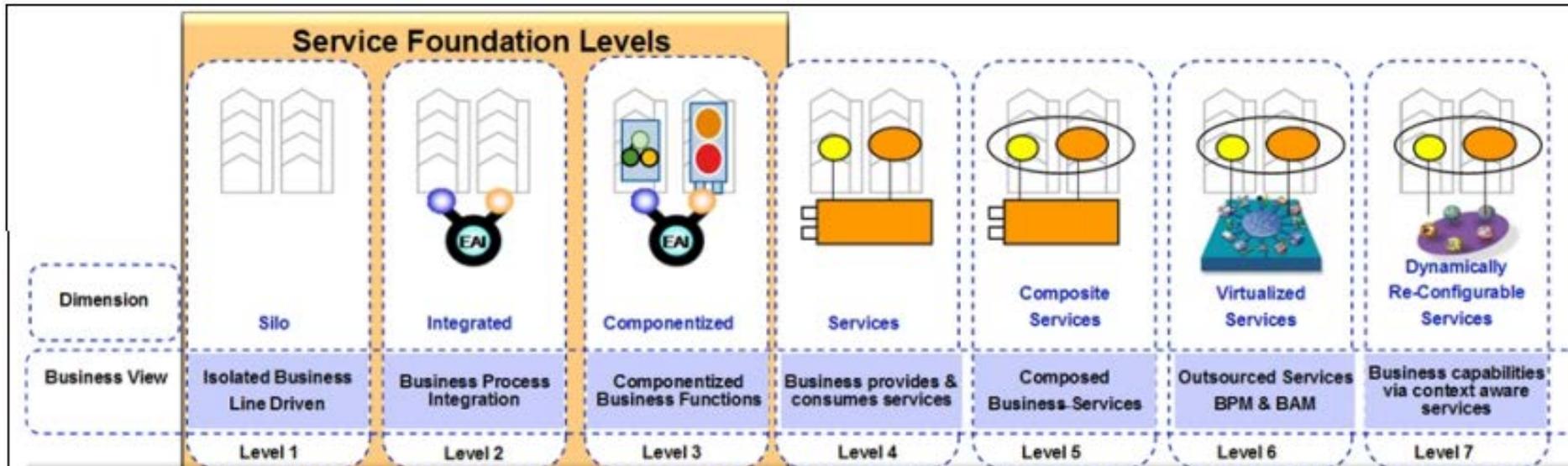
30
44

- ❑ Approche désormais classique d'audit d'une organisation vis-à-vis d'un enjeu GL/SI :
 - savoir où on est pour savoir (jusqu')où on va (chemin à parcourir, jusqu'où aller en fonction des moyens/contraintes...)...
 - Approche basée par des questions permettant de mesurer quantitativement (indicateurs) le niveau de maturité...
 - Défini selon 7 dimensions (point de vue) :
 - **Métier** : caractérise les pratiques et processus en cours, moyens alloués aux SI...
 - **Gouvernance & Organisation** : inclus compétences « indoor »
 - **Méthodes** : maturité GL, gestion de projet...
 - **Applications** : degré de réutilisation des applications
 - **Architecture** : topologie SI actuelle, techniques d'intégration, connaissances SOA...
 - **Information** : vision données ...
 - **Infrastructure & Management** : capacité de l'infrastructure actuelle...

OpenGroup Service Integration Maturity Model (OSIMM)

31
44

- ❑ En fonction de l'audit doit permettre de classer l'organisation selon 7 niveaux de maturité : Silo, Intégré, « Composantifié », Service, Services composés, Services Virtualisés, Services dynamiquement reconfigurables.
- ❑ Exemple : niveau de maturité du point de vue Métier



SOAP & Services Web

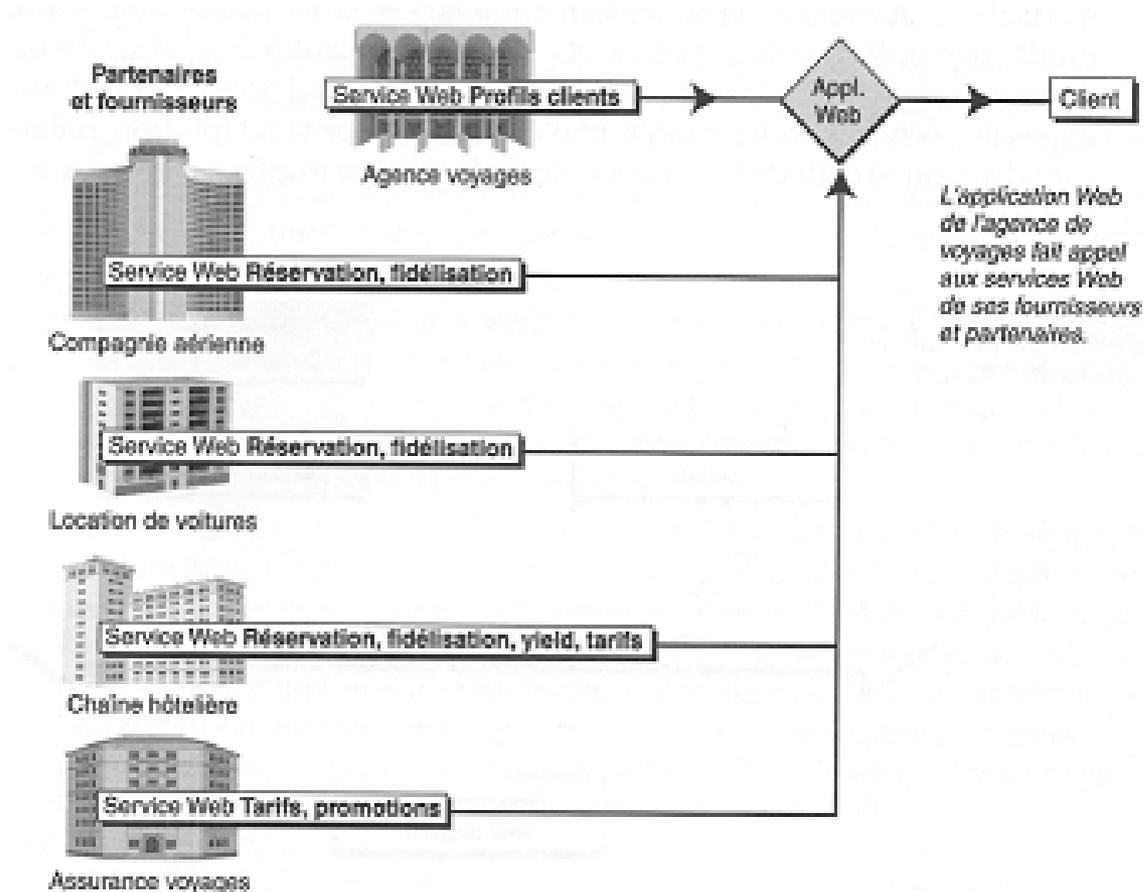
Définition d'un Service Web (W3C)

- ❑ Un service web est un système logiciel identifié par une URI dont les interfaces publiques et les liens sont définis et décrits en utilisant XML.
- ❑ Sa définition peut-être découverte par d'autres logiciels.
- ❑ Ces logiciels peuvent alors interagir avec le service web d'une façon pré décrite par sa définition en utilisant des messages XML convoyés par des protocoles internet

Avantage et usages

- ❑ Gestion de la complexité d'application lourde en décomposant par exemple en Services Web spécialisés (donc plus simple)
- ❑ La composition sera accomplie par des séquences de requêtes sur les serveurs
- ❑ Support de la communication: le web (ie http, html)
- ❑ Exemple d'applications :
 - Applications B2C (Business to Customer – e commerce)
 - Applications B2B (Business to Business) : ex, intégration des fournisseurs
 - Place de marché
 - Fournisseur d'applications

Exemple : Site internet « voyageur »



Application Web comme agrégation de services Web (extrait de: « Services Web » éd. Eyrolles)

En d'autres termes

□ Les services web :

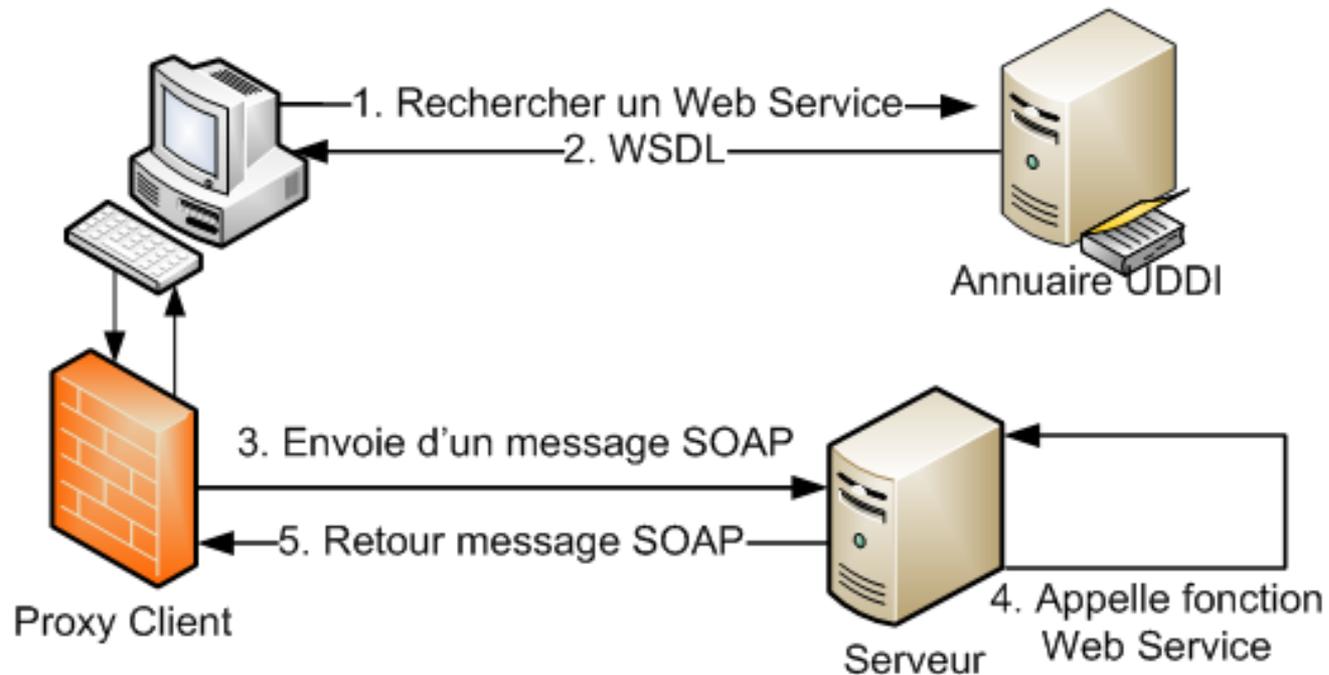
- Accès via le web (HTTP)
- Sont décrits en XML (fonctions, capacités)
- Enregistrées / Découverts dans des registres (fonction annuaire qui assure une souplesse car pas de « cablage en dur »)
 - Implique un langage de requête permettant par exemple de savoir les fonctions remplies par un service web (avec par exemple des informations de type qualité de service – par ex. temps de réponse ...)
- Echanges des messages XML (contenant les requêtes/résultats)
- Fonctionnent indépendamment dans le sens où ils assurent leur fonctions en autonomie
- Interagissent pour répondre à une question plus générale non détenue par un seul service
- Masquent la complexité de l'architecture distribuée aux « utilisateurs » (clients), ainsi que l'architecture interne des services web (requêtes XML)

□ => forme d'implantation d'une architecture SOA

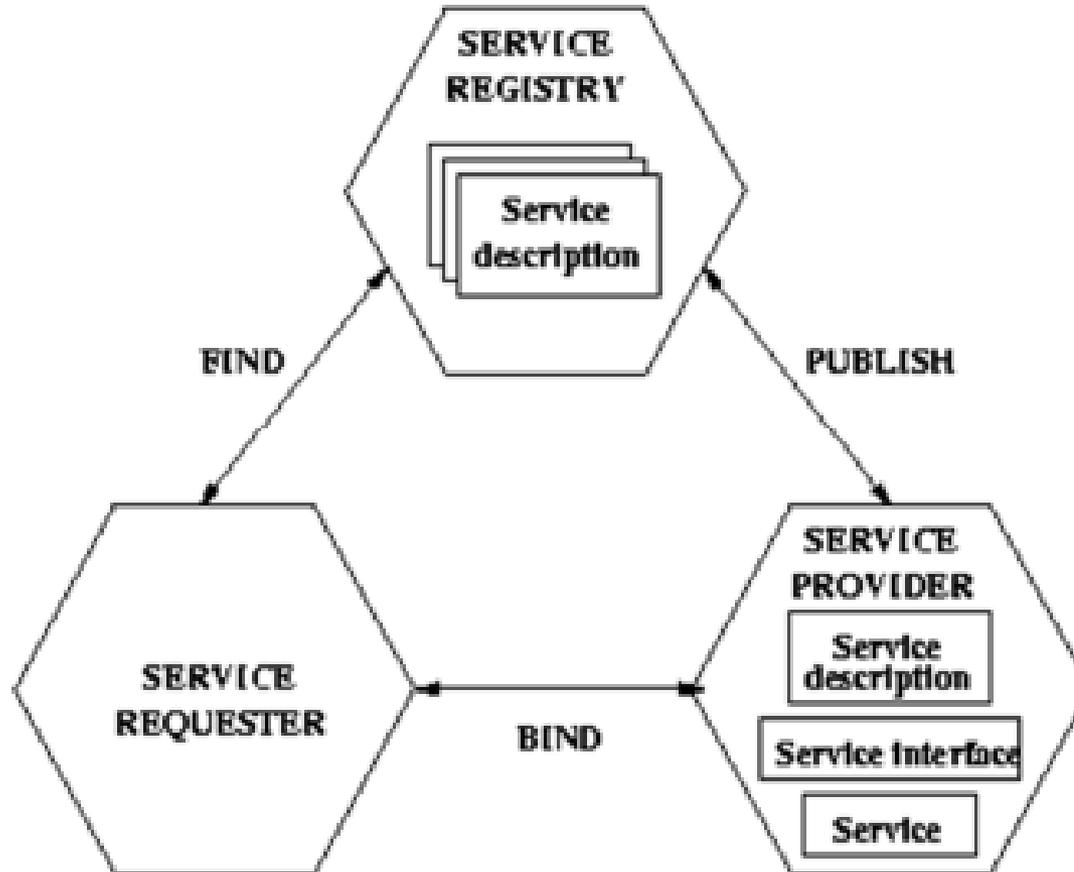
Déroulement ...

- ❑ Soit un client désirant utiliser une fonction f d'un service web
- ❑ Scénario
 1. Définition (langage WSDL) et description (XML) du service Web
 2. Publication du service ie se déclare auprès d'un annuaire de Service UDDI
 3. Le client effectue une requête auprès d'un annuaire (UDDI) => le client s'enregistre auprès du fournisseur de service (expression du besoin)
 4. Invocation du service (traitement de la requête) => SOAP
 5. Composition de service éventuelle (lorsque le besoin nécessite plusieurs fonctions détenue par plusieurs services)

Zoom SOAP : exemple



Déroulement ...



Source: G. Alonso

Langages supports du scénario précédent

- ❑ UDDI (*Universal Description, Discovery and Integration*) : annuaire des services web
- ❑ WSDL (*Web Service Description Language*) : Description de l'interface (fonctions assurées) du Service web sous un format XML
- ❑ SOAP (*Simple Object Access Protocol*) : protocole de communication entre web services, basé sur XML, précise la fonction invoquée et ses paramètres / le résultat de l'exécution de la requête..
- ❑ HTTP: protocole de transport par lequel passent les autres protocoles

Zoom WSDL

- ❑ Un document WSDL est un document XML structuré en 4 parties :
 - **Types** : définition des types de données utilisées par le service web
 - **Messages** : définition des données (types) nécessaires pour une opération (sorte de paramètre)
 - **Fonctions du SW** (portTypes) : opérations réalisables par le SW (interface) en tant qu'échanges de messages (ie requête)
 - Simple (requête de ou vers le SW ne nécessitant pas de réponse/résultat)
 - Question / Réponse (ex. simple ci après), à destination ou de la part du SW.
 - **Liens** (Bindings) : Décrit le contexte d'exécution des opérations (mode d'exécution : SOAP par exemple....)

- On pourrait faire un parallèle avec la POO : cela pourrait ressembler à ce qu'on obtiendrait en modélisant en XML un diagramme de classe UML...

Zoom WSDL : exemple

□ Structure d'un document WSDL (XML) :

■ **Définition** : Élément racine contenant les éléments suivants :

- `<definitions name="HelloService" targetNamespace="http://www.examples.com/wsdl/HelloService.wsdl" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://www.examples.com/wsdl/HelloService.wsdl" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> </definitions>`

- **Types** : définition des types de données utilisées par le SW (entier, réel, chaîne de caractères...)
- **Messages** : définition des données (types) nécessaire pour une opération (sorte de paramètre) échangées entre le client et SW

- `<message name="SayHelloRequest">
 <part name="firstName" type="xsd:string"/> </message>
<message name="SayHelloResponse">
 <part name="greeting" type="xsd:string"/> </message>`

Zoom WSDL : exemple

❑ Fonctions du SW (portTypes) : opérations réalisables par le SW (interface)

```
■ <portType name="Hello_PortType">  
  <operation name="sayHello"> ←  
    <input message="tns:SayHelloRequest"/>  
    <output message="tns:SayHelloResponse"/>  
  </operation> </portType>
```

❑ Liens (Bindings) : précise le contexte d'exécution de l'opération

```
■ <binding name="Hello_Binding" type="tns:Hello_PortType">  
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>  
  <operation name="sayHello">  
    <soap:operation soapAction="sayHello"/>  
    <input >  
      <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"  
        namespace="urn:examples:helloservice" use="encoded"/> </input>  
    <output> <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"  
      namespace="urn:examples:helloservice" use="encoded"/>  
    </output> </operation> </binding>
```

Zoom SOAP : exemple d'outils de validation

The screenshot displays the WSDL SOAP Analyzer interface. The top section is titled "WSDL" and contains three dropdown menus: "Services" set to "XigniteCurrencies", "Ports" set to "XigniteCurrenciesSoap", and "Operations" set to "GetRealTimeCrossRateAsString". Below these, the "Actions" section includes a "URL:" field with "http://www.xignite.com/xCurrencies.asmx" and a "SOAP Action:" field with "http://www.xignite.com/services/GetRealTimeCrossRateAsString". The "Version:" field has radio buttons for "1.1" and "1.2", with "1.2" selected.

The main area is split into two tabs: "Request" (selected) and "Attachments". The "Request" tab shows the following XML code:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns0="http://www.xignite.com/services/">
<SOAP-ENV:Header>
<ns0:Header>
<ns0:Username>STRING</ns0:Username>
<ns0:Password>STRING</ns0:Password>
<ns0:Tracer>STRING</ns0:Tracer>
</ns0:Header>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<ns0:GetRealTimeCrossRateAsString>
<ns0:From>STRING</ns0:From>
<ns0:To>STRING</ns0:To>
</ns0:GetRealTimeCrossRateAsString>
</SOAP-ENV:Body>
```

Below the XML editor are buttons for "Send", "Request settings:", "Open", "Save", and "Regenerate". There is also a checkbox for "Open response in editor".

The "Response" tab shows the following XML code:

```
<!-- Auto generated server sample response. -->
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns0="http://www.xignite.com/services/">
<SOAP-ENV:Header/>
<SOAP-ENV:Body>
<ns0:GetRealTimeCrossRateAsStringResponse>
<ns0:GetRealTimeCrossRateAsStringResult>
<ns0:Outcome>OUTCOMETYPES</ns0:Outcome>
<ns0:Message>STRING</ns0:Message>
<ns0:Identity>STRING</ns0:Identity>
```