

# **- Etude détaillée -**

## **Conception d'ensemble**

### **Élaboration d'une architecture de référence pour l'automatisation d'un SI**

**Mots-clé :** système d'information, USDP, UML, architecture applicative, SFG, SFD

Département Informatique  
INSA DE LYON

Youssef AMGHAR



# Plan du cours

---

1. Introduction
2. Etude de cas
3. Les différentes étapes pour la construction d'une architecture de référence
4. Interopérabilité des SI

# Objectifs de la conception d'ensemble

## Objectifs de production

- ✓ Établir les fondations d'une ARCHITECTURE DE REFERENCE ,
- ✓ Analyser le domaine à traiter en capturant la plupart des besoins fonctionnels exprimés sous forme de cas d'utilisation ,
- ✓ Définir les niveaux de qualité à atteindre, en termes de fiabilité, temps de réponse, ... grâce à des critères d'évaluation de l'architecture de référence :
  - satisfaction des besoins fonctionnels exprimés par les acteurs ,
  - robustesse pour résister à la phase de construction et à l'ajout de nouvelles fonctionnalités.
- ✓ Mesurer et identifier les risques les plus significatifs de nature à bouleverser le plan, les coûts, les délais.

## ■ Objectifs de gestion de projet

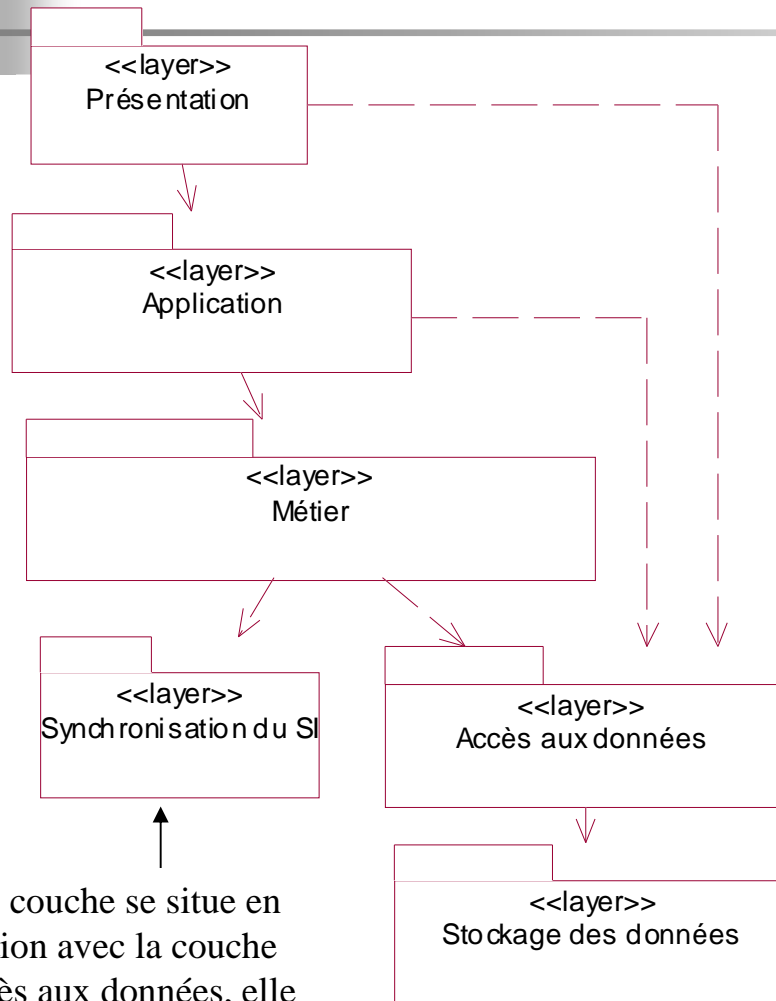
- ✓ Planifier le projet et éliminer ses plus haut risques.

# Définitions: Architecture de référence

**Construire une architecture de référence, c'est faire un choix à chacun des niveaux suivants :**

- ✓ **Architecture applicative ou logique:** obtenue en découpant le SI en blocs.
  - Données et traitements du bloc: services, méthodes, programmes, entités, tables, vues, ...(définition du bloc),
  - Composants logiciels assemblés au sein du bloc (composition du bloc),
  - Spécification formelle des échanges entre blocs (interface du bloc).
  
- ✓ **Architecture d'exécution :** regroupement de composants logiciels remplissant une fonction parmi:
  - Interface homme-machine: présentation + dialogue
  - Traitement: fonctions applicatives (métier, règles de gestion)
  - Données: gestion de données
  
- ✓ **Architecture technique et logicielle:** ensemble de composants techniques (machines, réseaux, logiciels de base) permettant de bâtir une solution informatique.
  
- ✓ **L'architecture opérationnelle ou organisationnelle**
  - localise les traitements du SI en déterminant les lieux de production et de communication de l'information ,
  - introduit une répartition des éléments de l'architecture applicative sur les postes de travail

# Architecture d'exécution : organisation en couches



Cette couche se situe en partition avec la couche d'accès aux données, elle permet au système à mettre en place de synchroniser ses flux de données avec le reste du système d'information.

INSA LYON

Cette couche a pour but de restituer les données à l'utilisateur, et de transformer ses actions en événements applicatifs.

Cette couche représente les objets de contrôle de l'application, elle a pour but d'implémenter les règles dynamiques de pilotage dans le cadre de l'application..

Cette couche représente les objets du métier et a pour but d'implémenter toutes les règles structurelles et dynamiques du domaine modélisé.

Cette couche interface les objets de la couche métier avec le stockage des données. Elle permet également un accès aux données depuis les couches application ou présentation.

Cette couche présente le schéma utilisé pour le stockage des données en SGBDR.

# Avantages de l'architecture applicative

- Découpage crédible d'un grand projet en sous-projets parallélisée
- Détermination d'une maille de sous-système testable individuellement
- Assemblage de blocs pour des tests d'intégration
- Répartition lors du déploiement du système (pourra évoluer en fonction de nouvelles contraintes ou politiques d'exploitation)
- Blocs indispensables, optionnels ou substituables
- Le contenu d'un bloc peut évoluer (évolution du modèle de données ou ajout de nouvelles fonctions)
- Dans une démarche de migration-évolution de système d'information le bloc sera l'unité constitutive des états stables composant la trajectoire d'évolution

**Portabilité - Interopérabilité - Intégration -  
Evolutivité**

**Modularité - Maintenabilité - Réutilisabilité -  
Stabilité**

- Conception d'ensemble -

# Les différentes vues d'un système selon USDP

Elle exprime la vue du système tel que la voit l'utilisateur final. Cette vue est donc orientée utilisateur. Elle s'exprime par l'intermédiaire des USE CASES.

Activité concernée : CAPTURE DES BESOINS

Elle exprime l'architecture de référence. Elle est le résultat de l'analyse du SI en termes de fonctionnalités, d'objets métiers, de découpage en packages et finalement d'applications.

Activité concernée : ANALYSE

**VUE METIER & FONCTIONNELLE**

**VUE LOGIQUE**  
Paquetages  
(classes principales)

**SYSTEME**

**VUE DE DEPLOIEMENT**  
Répartition des composants selon un patron d'architecture

**VUE COMPOSANT**  
Description des composants et des interfaces

Elle exprime comment les composants du système sont répartis sur le modèle d'infrastructure technique et les postes de travail.










Activités concernées : ANALYSE & CONCEPTION

Description des classes provenant du modèle d'analyse. Il s'agit des classes frontière, contrôle et entité, ainsi que la dynamique du comportement des objets de classes. Construction des sous-systèmes.

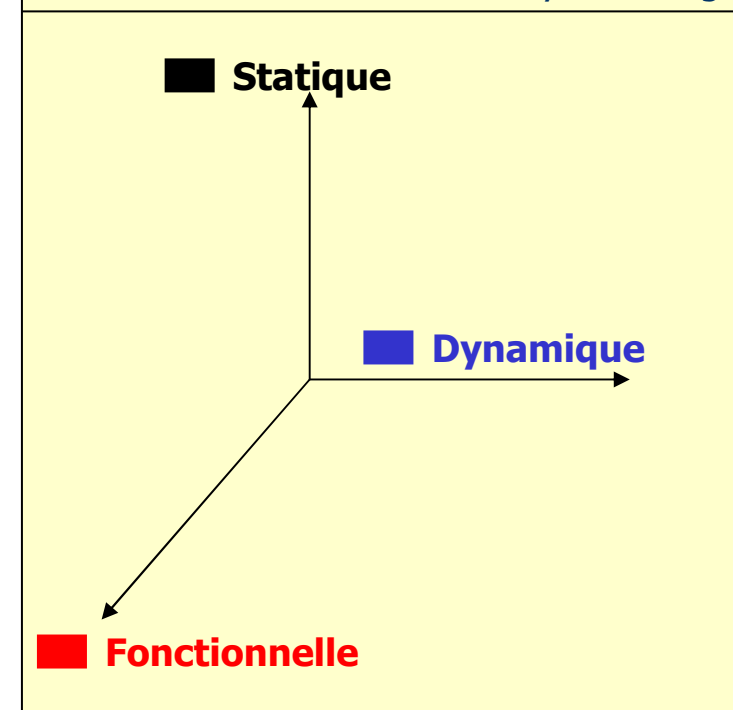
Activité concernée : CONCEPTION

# Les facettes d'un système

Dans chacune des vues d'un système, on utilise un ou plusieurs de ces diagrammes pour modéliser entièrement le système :

- les cas d'utilisations, 
- les diagrammes de classes,  
- Le modèle de paquetages,  
- les diagrammes de comportement :
  - diagrammes d'états-transitions, 
  - diagrammes d'activités, 
  - diagrammes de séquence, 
  - diagrammes de collaboration. 
- Les diagrammes d'implémentation :
  - diagramme de composants,
  - diagrammes de déploiement

Les différentes dimensions d'un système logiciel

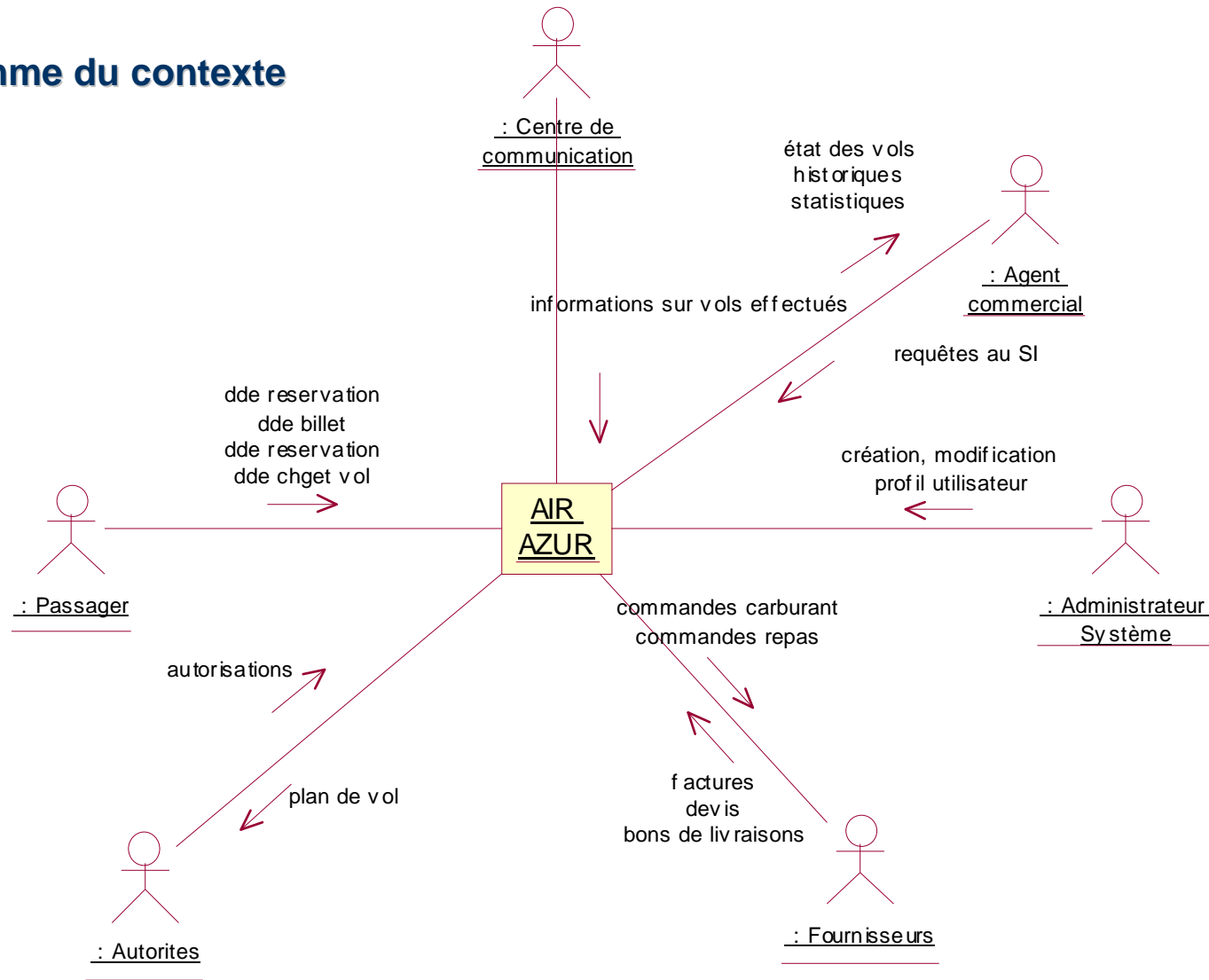




# Etude de cas : AIR AZUR

Extrait de l'ouvrage: OOM : la conception objet des SI  
M. Bouzeghoub et A. Rochfeld Editions Hermes, 2000

## Diagramme du contexte



## ■ Décision

- ✓ sur incitation du PDG de AIR AZUR :
  - étude en profondeur du SI de l'entreprise
  - recours aux nouvelles technologies informatiques
    - orientation objet,
    - informatique distribuée,
    - système d'exploitation non-propritaire
    - méthode USDP (UML)

## ■ Objectif de l'étude : élaborer une architecture de référence qui couvre :

- ✓ Architecture applicative
  - liste des acteurs, cas d'utilisation
  - modélisation des processus (diagrammes des activités)
  - modèle statique
  - paquetages (flux d'information)
  - réalisation des cas d'utilisation (diagramme de collaboration, de séquence)
- ✓ Architecture en couche et architecture C/S
- ✓ Architecture opérationnelle

# Les différentes étapes pour la construction d'une architecture de référence



# Description des étapes de la conception d'ensemble

## 1. ETAPE 1. SPECIFICATIONS FONCTIONNELLES GENERALES

- DESCRIPTION DES PROCEDURES
- IDENTIFICATION DES BLOCS OU PAQUETAGES

⇒ Livable SFG

## 2. ETAPE 2. SPECIFICATIONS FONCTIONNELLES DETAILLEES

- ANALYSE DES BLOCS
- (Modélisation des aspects statique et dynamique)

⇒ Livable SFD

## 3. ETAPE 3. VALIDATION

- CONSOLIDATION DES BLOCS

## 4. ETAPE 4. SPECIFICATIONS TECHNIQUES POUR UNE ARCHITECTURE OPERATIONNELLE

- A – ELABORATION DES ARCHITECTURES
  - ✓ LOGICIELLE
  - ✓ TECHNIQUE
  - ✓ ORGANISATIONNELLE
- B - DEPLOIEMENT DES BLOCS SUR L'ARCHITECTURE OPERATIONNELLE

⇒ Livable ST\*

\* ST : spécifications techniques



# Affinement des procédures : aspects organisationnels

## ETAPE 1

---

- Variantes de procédures
  - ✓ Étude des différents cas de comportement d'un processus en fonction :
    - des événements qui le déclenchent
    - des objets qu'il manipule
  - Un événement peut donner lieu à un traitement particulier
  - Exemple : cas d'une entreprise de BTP
    - processus affectation de matériel
    - événement : fax, téléphone, saisie informatique, papier
  - Les différents types d'objets ne mènent pas forcément au même traitement :
  - Exemple : cas d'une entreprise de BTP
    - *transporter une grue nécessite un convoi exceptionnel qu'il faut organiser*
    - *une bétonneuse nécessite un camion poids lourd avec une organisation légère.*



# Affinement des procédures : aspects organisationnels

## ETAPE 1

---

- Étude du mode dégradé et du mode inverse
- Mode dégradé
  - ✓ Étude d'un processus en face d'une panne
    - déterminer les comportements additionnels
    - identifier les pannes
    - certains processus ne nécessitent pas de fonctionnement en mode dégradé
  - ✓ Exemple : cas d'une entreprise de BTP
    - Processus « Affectation matériel sur les chantiers »
    - Si on suppose que le serveur de traitement peut tomber en panne, on prévoit des comportements additionnels :
      - impression des états d'affectation de matériel à une fréquence donnée
      - le listing permet de fonctionner
    - Processus « Facturation du matériel »
      - processus non sensible



# Affinement des procédures : aspects organisationnels

## ETAPE 1

- **Étude du mode dégradé et du mode inverse**
- **Mode inverse**
  - ✓ Étude d'un processus en présence d'un événement qui remet en cause le déroulement de tout ou partie d'un processus
  - ✓ Cas des annulations
  - ✓ Exemple : cas d'une entreprise de BTP
    - Processus « Affectation de matériels »
      - Erreur d'affectation
    - Processus « Facturation de matériels »
      - traitement de l'événement « Contestation »
- **Traitements des incidents**
  - ✓ Étude d'un processus en présence d'un événement qui remet en cause le déroulement de tout ou partie d'un processus
  - ✓ Exemple : cas cas d'une entreprise de BTP
    - Un matériel arrive au chantier et présente des dysfonctionnements
    - Un chantier se voir facturer un matériel qu'il n'a pas utilisé.



# Description des étapes

## ETAPE 1 : SPECIFICATIONS FONCTIONNELLES GENERALES (SF

### ACTIVITES USDP

- **CAPTURE DES BESOINS**
- **ANALYSE**

### RESULTATS

- La familles des procédures
- Cartographie des processus
- Urbanisation des applications
- Référentiels de données
- Description des applications

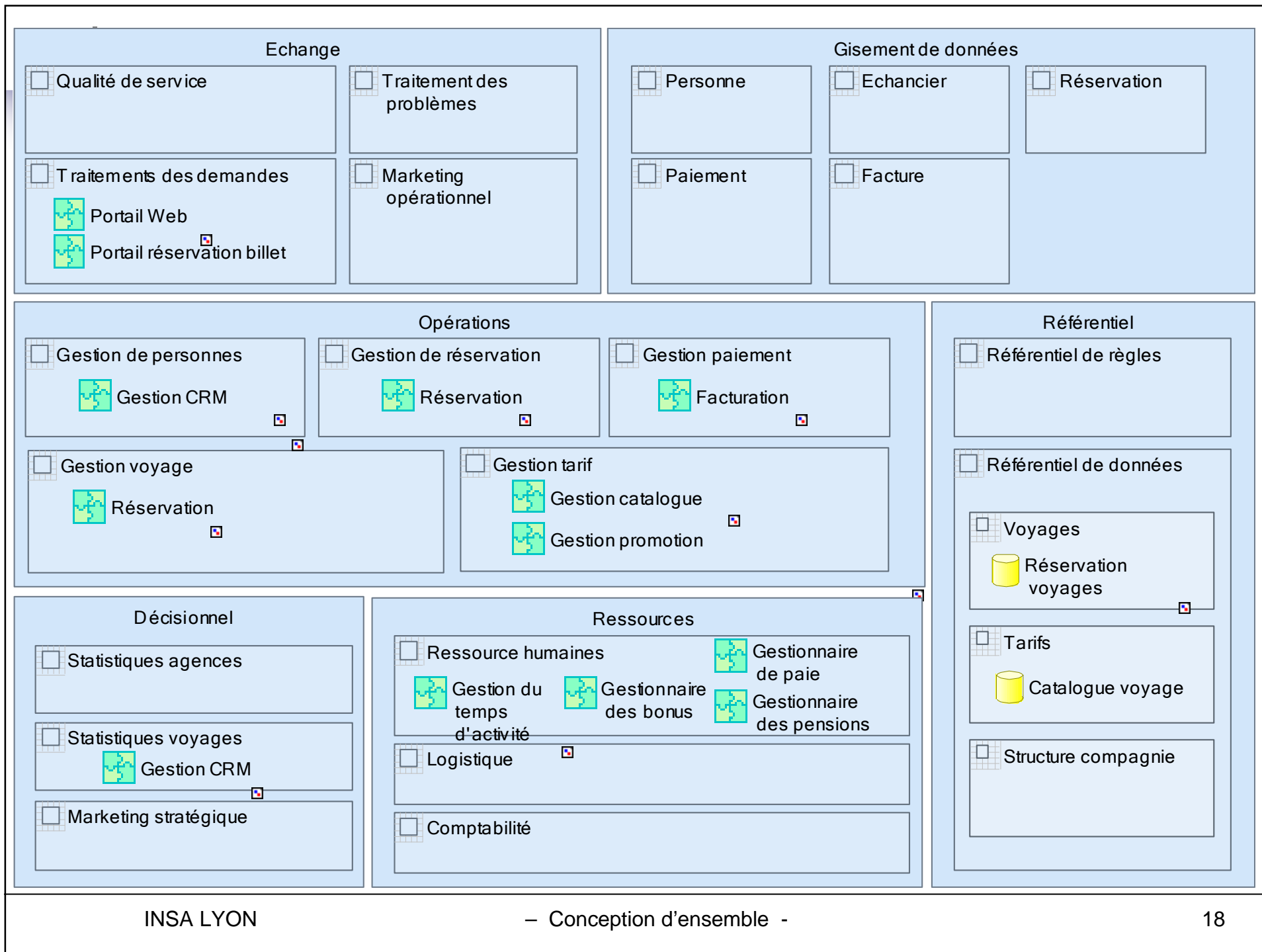
**En respectant les principes  
d'urbanisation**





# Urbanisme des systèmes d'information

- Concept développé pour offrir aux entreprises un cadre d'homogénéisation des composants de leur système d'information.
- Il souligne la différence entre:
  - ✓ **L'Urbanisme:** science, art et/ou technique de l'organisation spatiale des établissements humains
  - ✓ **L'Architecture:** art de construire, de disposer et d'orner les édifices
- Le rôle de l'urbanisme consiste à définir aux niveau fonctionnel et applicatif:
  - ✓ les différents composants du Système d'Information et leurs modalités d'assemblage; **c'est la technique de l'organisation des activités exercées.**
  - ✓ les points d'ancrage minimums, compatibles avec les standard en émergence sur le marché, permettant l'intégration d'une activité ou d'un produit supplémentaire.
- Au moins 6 zones
  - ✓ Zone d'échange, zone opérationnelle de traitement, zone opérationnelle de synthèse, zone référentielle, zone de pilotage, zone de gestion



# Identification des blocs

## ETAPE 1

Lors de ce processus, on déroule les macro-tâches suivantes :

1. Identifier et décrire les **cas d'utilisation (service)** et les **acteurs**
2. Élaborer le diagramme des objets métier
3. **Identifier des blocs (paquetages) extraits des cas d'utilisation et/ou du diagramme métier**
4. **Affectation des outils aux blocs (détermination des services)**

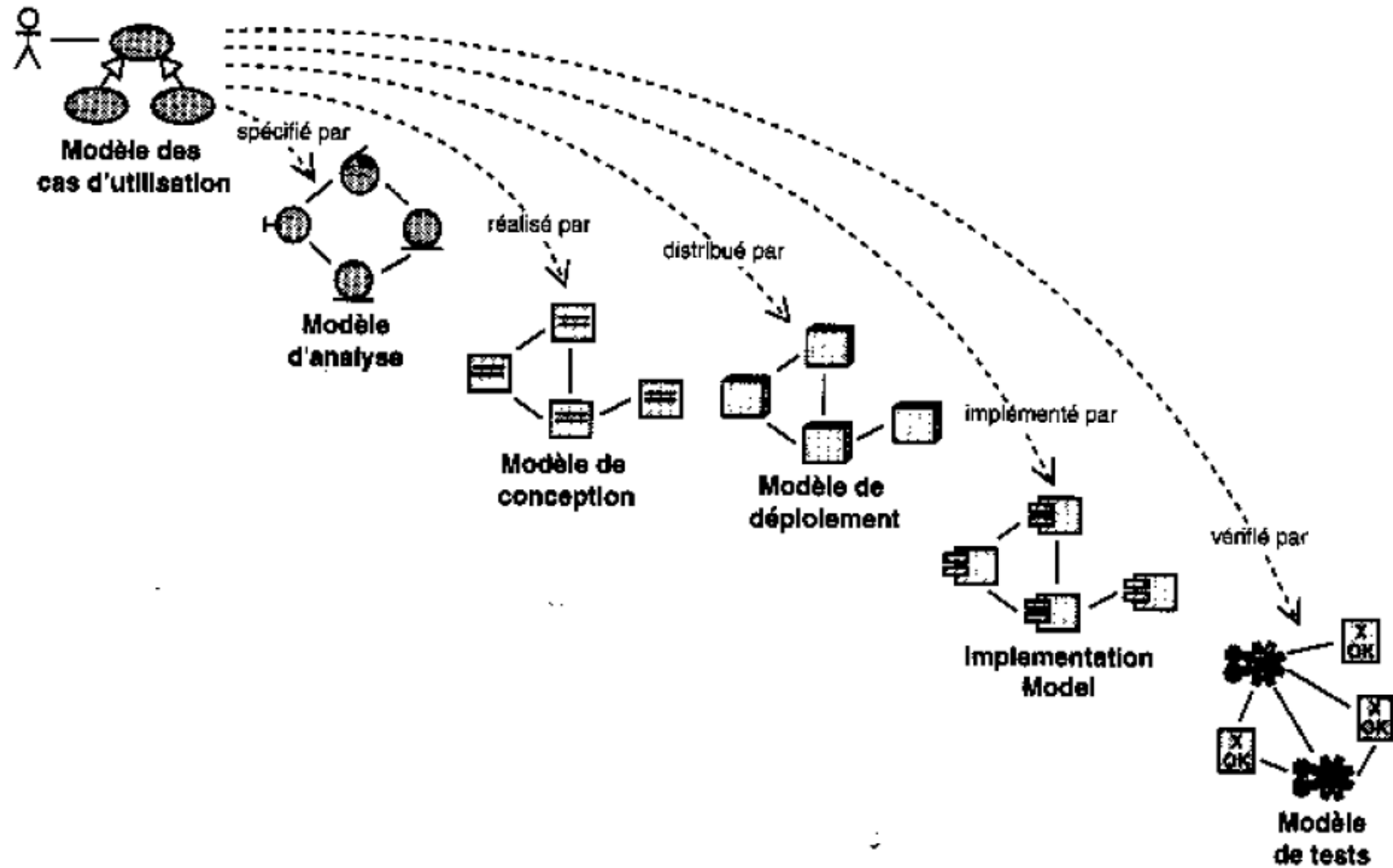
### Entrées du processus

- Modèles issus de l'étude préalable ou de la phase préliminaire de USDP
- Une liste des fonctionnalités,
- Les contraintes et les exigences

### Sorties du processus

- Fiches détaillées des cas d'utilisation
- Une liste d'acteurs avec leurs rôles
- Une liste d'applications
- Diagramme des objets métiers

# Stratégie des cas d'utilisation



# Fiche de cas d'utilisation

Expression des besoins

Titre :

**En-tête**

But :

Résumé :

Acteurs :

Date de création :

Date de mise à jour :

Version :

Responsable :

Post-conditions :

**Nominal**

Liste des services

Scénario nominal :

Scénario d'exception :

**Alternatives**

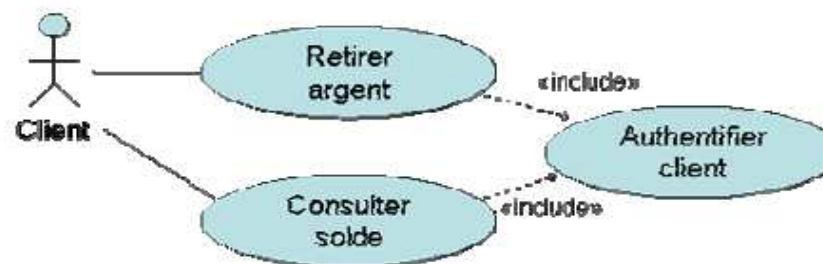
Scénario mode inverse :

Scénario mode dégradé :

# Structuration des cas

## ■ Include

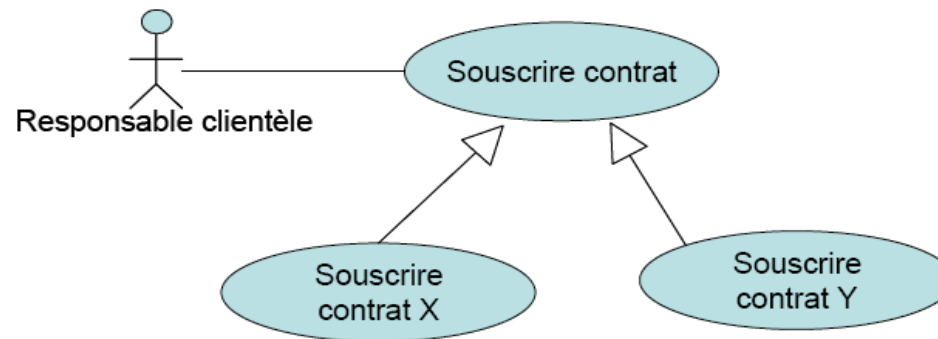
- ✓ La relation d'include n'a pour **seul** objectif que de **factoriser** une partie de la description d'un cas d'utilisation qui serait **commune** à d'autres cas d'utilisation.
- ✓ Le cas d'utilisation inclus dans **les** autres cas d'utilisation n'est pas à proprement parlé un vrai cas d'utilisation car il n'a pas d'acteur déclencheur ou receveur d'évènement.
- ✓ Une erreur classique est d'utiliser la relation « d'include » pour faire du découpage fonctionnel d'un cas d'utilisation en plusieurs « sous cas d'utilisation » qui s'enchaînent en fonction de certains critères.



# Structuration des cas

- **Généralisation / spécialisation**

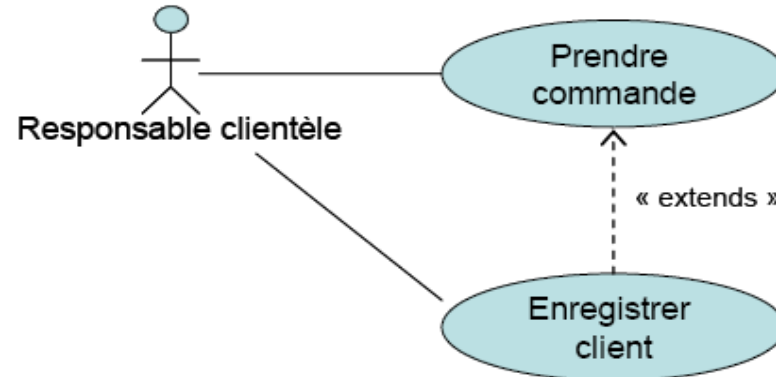
- ✓ Cette relation est consistante à dire que l'on a un cas d'utilisation dit « de base », générique, qui décrit des séquences d'événements et d'autres cas d'utilisation qui héritent de ce comportement de base et le spécialise suivant différents critères.
- ✓ Exemple :



# Structuration des cas

## ▪ Extends

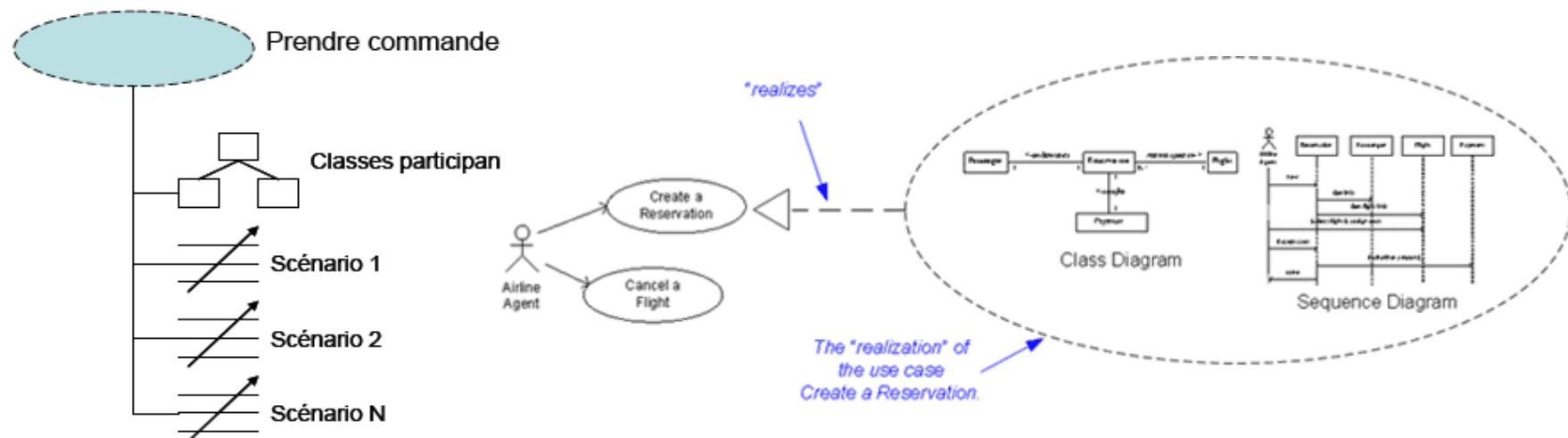
- ✓ La relation « d'extend » est probablement la plus utile car elle a une sémantique métier au contraire des 2 autres qui sont plus des artifices d'informaticiens.
- ✓ On dit qu'un cas d'utilisation X étend un cas d'utilisation Y lorsque le cas d'utilisation X peut être appelé au cours de l'exécution du cas d'utilisation Y.  
Exemple :





# Les « Use case Realization »

- Les « use case realization » sont un moyen de regrouper un diagramme de classes et des diagrammes de séquences ou de collaboration.
  - ✓ On retrouvera dans le diagramme de classes les classes qui mettent en oeuvre le cas d'utilisation associé au « use case realization » (structure des classes et relations entre classes).
  - ✓ On retrouvera dans les différents diagrammes de séquences (ou de collaboration) une documentation des différents événements échangés entre les objets afin de réaliser les différents scénarii décrit dans le cas d'utilisation.



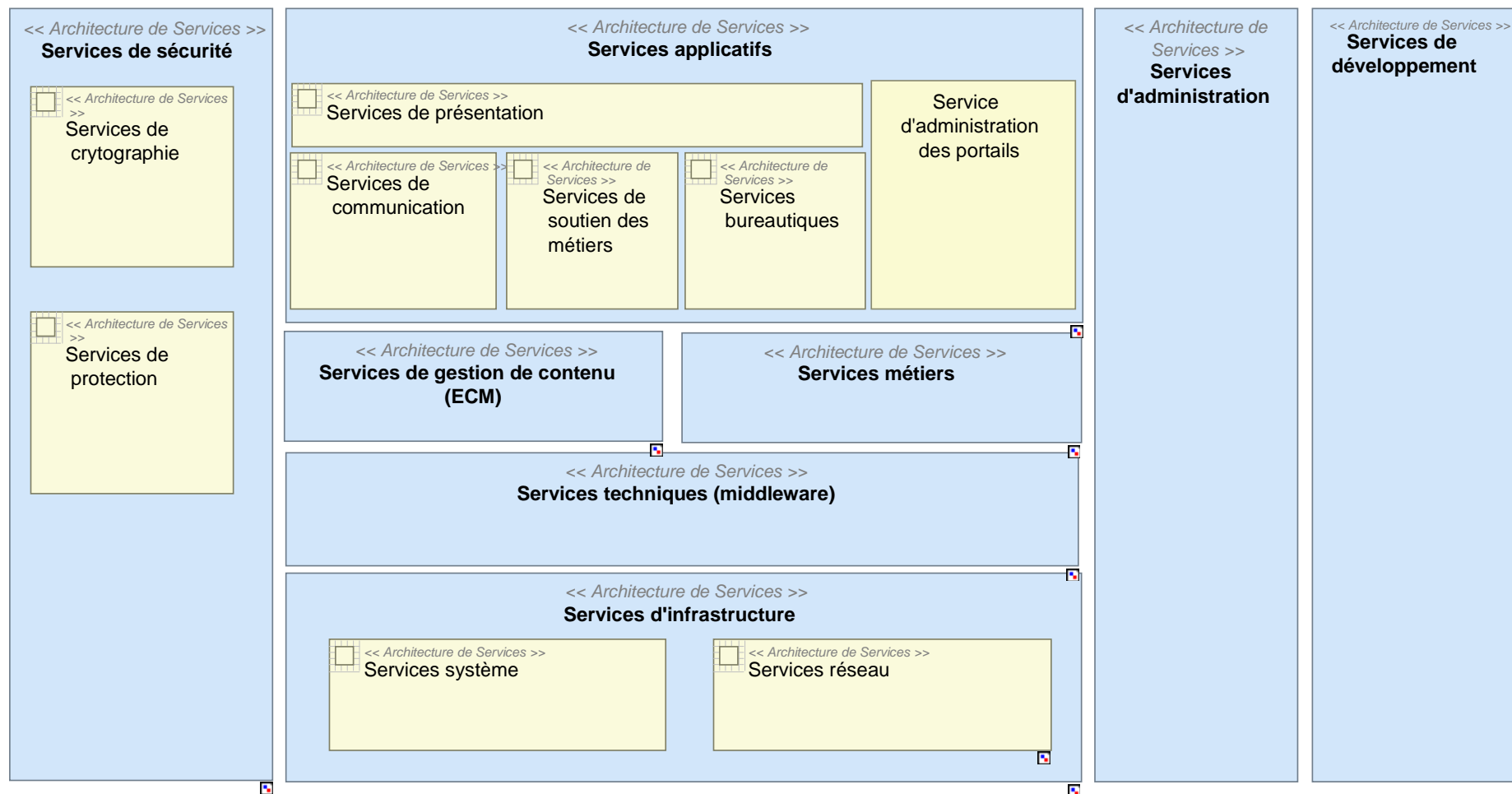
<http://www.ibm.com/developerworks/rational/library/>

# Définition des applications

- Les applications se déterminent par regroupement de fonctions de l'utilisateur tout en respectant la définition des postes de travail.
- Un découpage idéal en cas d'utilisation, permet d'affecter un nombre entier de cas à une application (ce n'est pas toujours vrai).
- La distribution des composants facilite leur réutilisation puisque les mêmes services sont accessibles par plusieurs applications.
- On peut prendre comme critère de transformer les paquetages d'analyse en composants.
- Une application se décrit en terme de :
  - ✓ rôle,
  - ✓ procédure
  - ✓ données manipulées
  - ✓ poste de travail
  - ✓ acteurs responsables

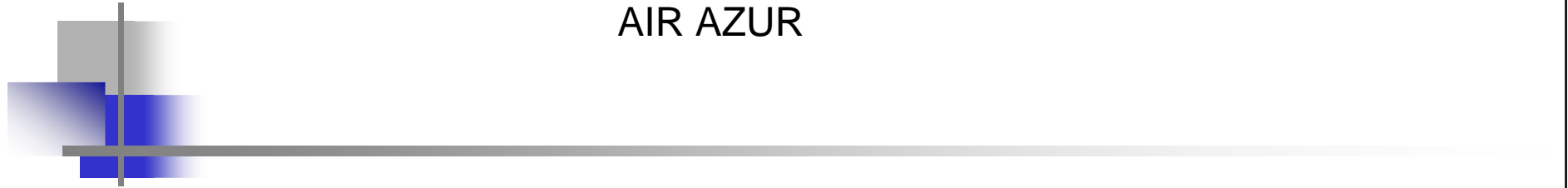
Application	A1
Rôle (services fournis)	cette application a pour but de permettre à une catégorie d'utilisateurs de suivre l'état des stocks des magasins et lancer des ordres d'approvisionnements fournisseurs.
Postes de travail	Département D1, secrétariat. Exploitation (windows, authentification)
Procédure	permet d'automatiser les opérations conceptuelles O1 et O3 de la procédure P1.
Données	cette application s'appuie sur le modèle externe ME1. Elle utilise les entités E1 et E2 en lecture; l'entités E4 et l'association A2 en écriture; l'association A1 en
Acteurs	lect/ecr Cet application est destinée au Chef de département et à la secrétaire

# Schéma de principe d'une architecture de services



# Illustration

AIR AZUR



## Zone opérationnelle

### Gestion des approvisionnements

- Gestion des commandes,
- Gestion des livraisons,
- Gestion des marchés
- Gestion des fournisseurs

### Gestion opérationnelle

- Vente de billets, réservation, annulation
- Préparation et suivi des vols
- Gestion des passagers et des clients,

### Gestion des entités organisationnelles

- Gestion des entités organisationnelles (régions, escales, agences).

## Zone ressource

### Gestion du référentiel

- Fournisseurs
- Clients
- Appareils

## Zone ressource

### Gestion financière et comptable

- Gestion de la trésorerie.

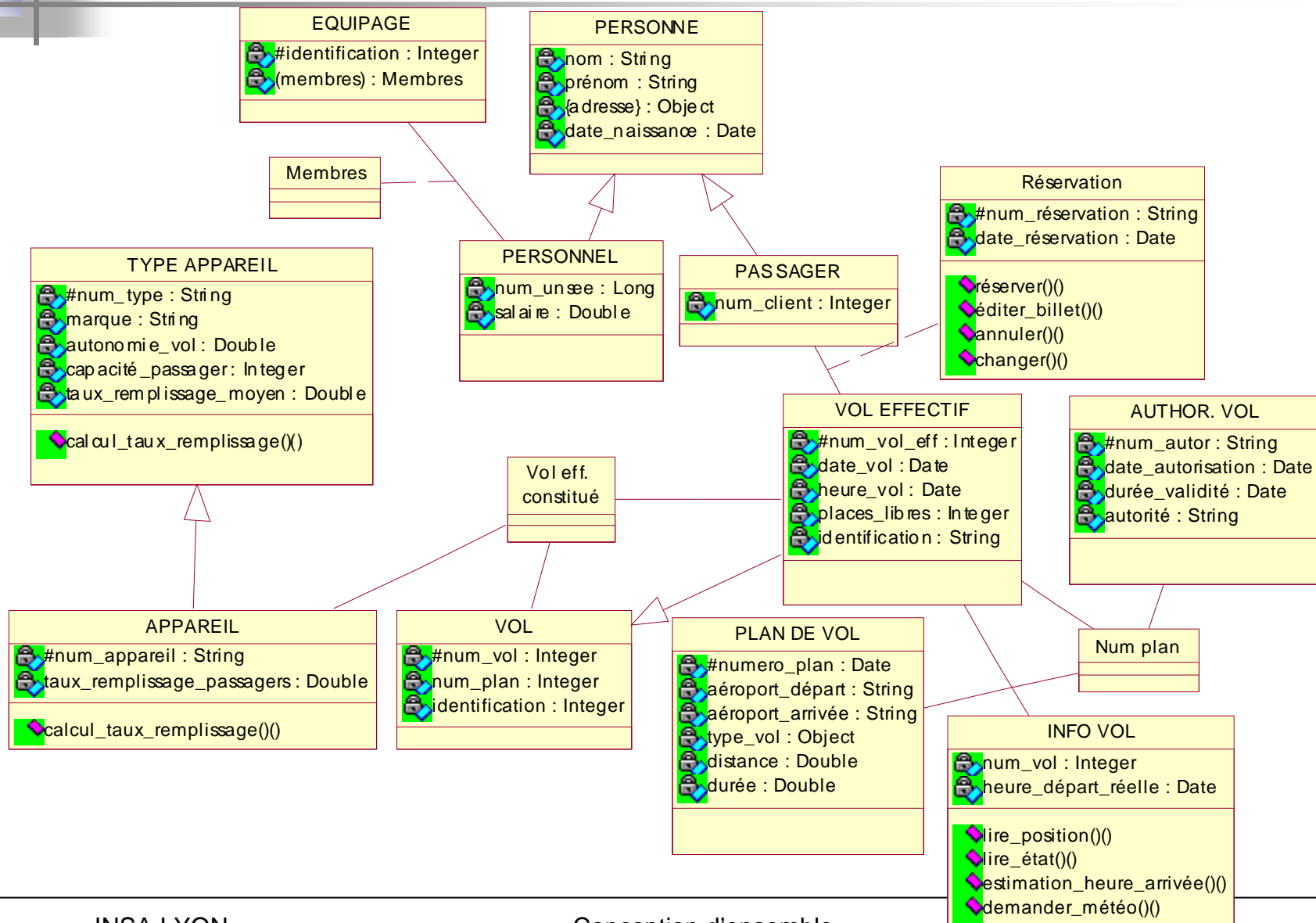
### Gestion des Ressources Humaines



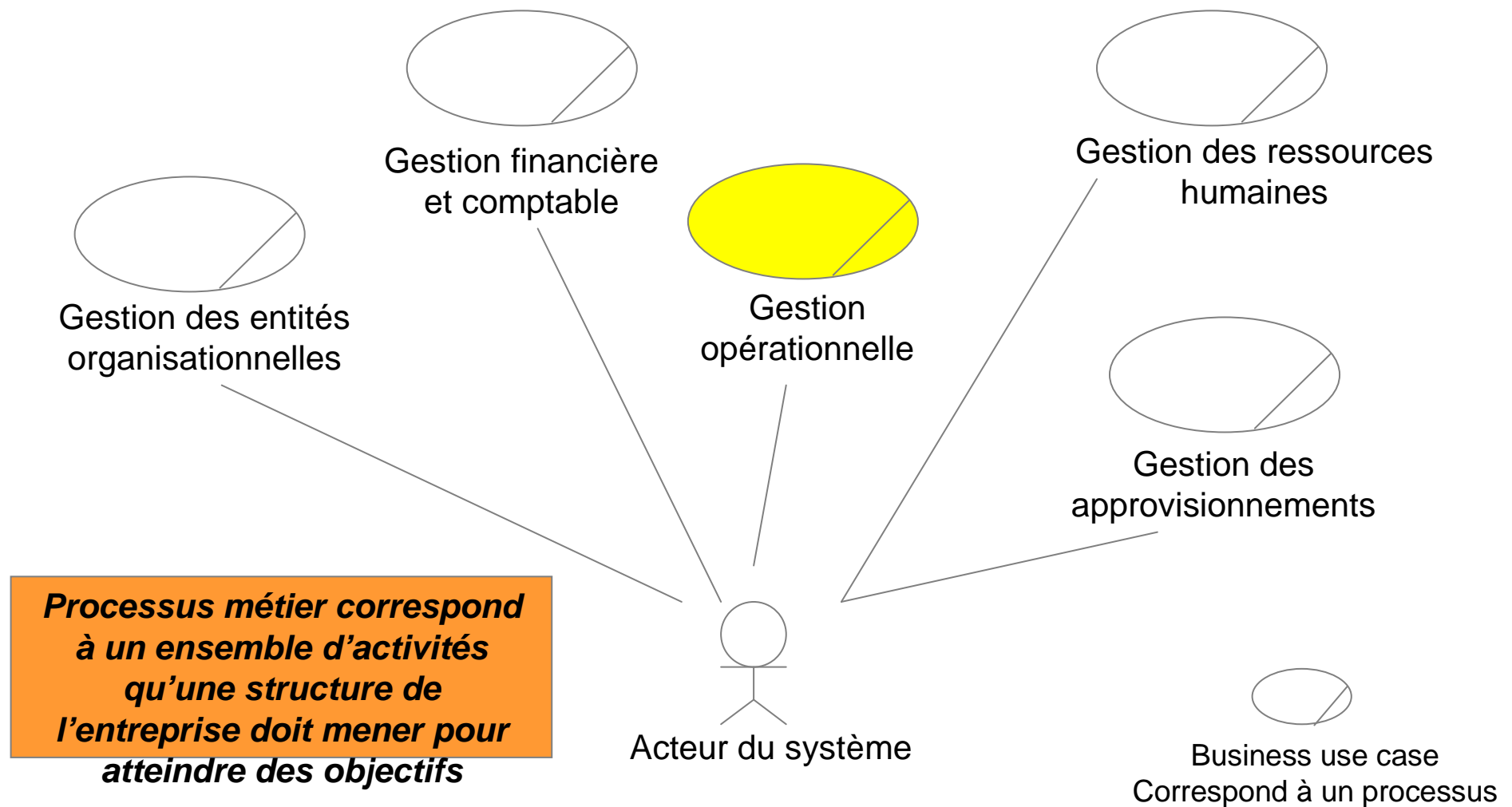
Zone référentielle

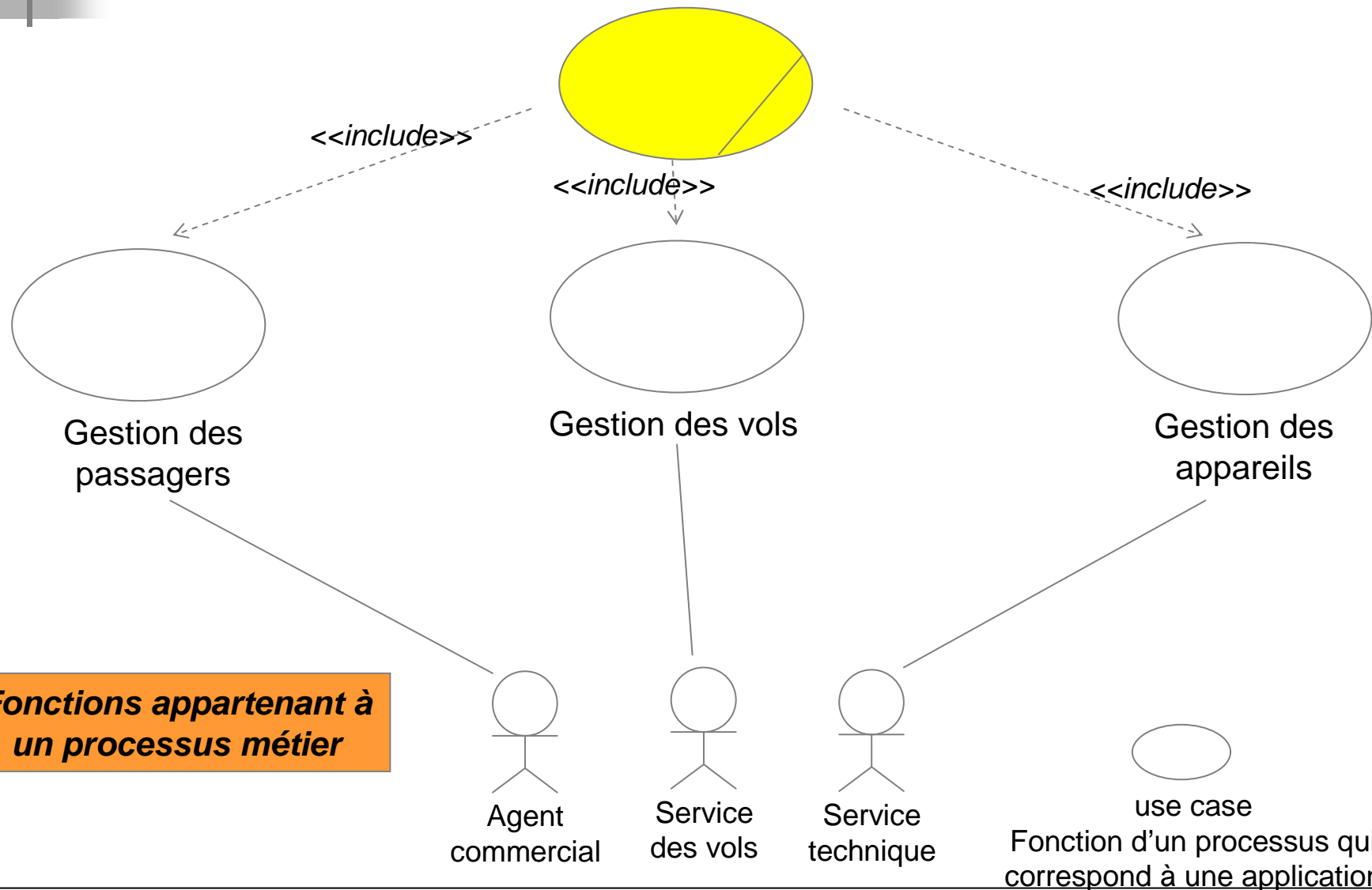
# Diagramme statique des objets métier

Analyse  
**AIR AZUR**  
**ETAPE 1**



## Modélisation de l'entreprise en termes de cas d'utilisation







# Description du cas : gestion des passagers

**AIR AZUR**  
**ETAPE 1**

Fiche de cas d'utilisation

Expression des besoins

**Titre** : Gestion des passagers

**But** : assure les réservations, les ventes et les annulations sur les vols.

**Résumé** : créer des réservation sur des vols. Édition de billets. Modification et annulation de réservations.

**Acteur** : Agent commercial

**Post-conditions** : réservations enregistrées dans la base, billets édités.

Date de création : 01/08/2006

Date de mise à jour : 15/10/2006

Version : 1.1

Responsable : Azur

**Liste des services** : réserver, acheter, rembourser, ...

**Scénario nominal** : Un agent commercial sollicite le système pour effectuer une réservation et éditer un billet.

Post-condition : une réservation est enregistrée dans la base et un billet est imprimé.

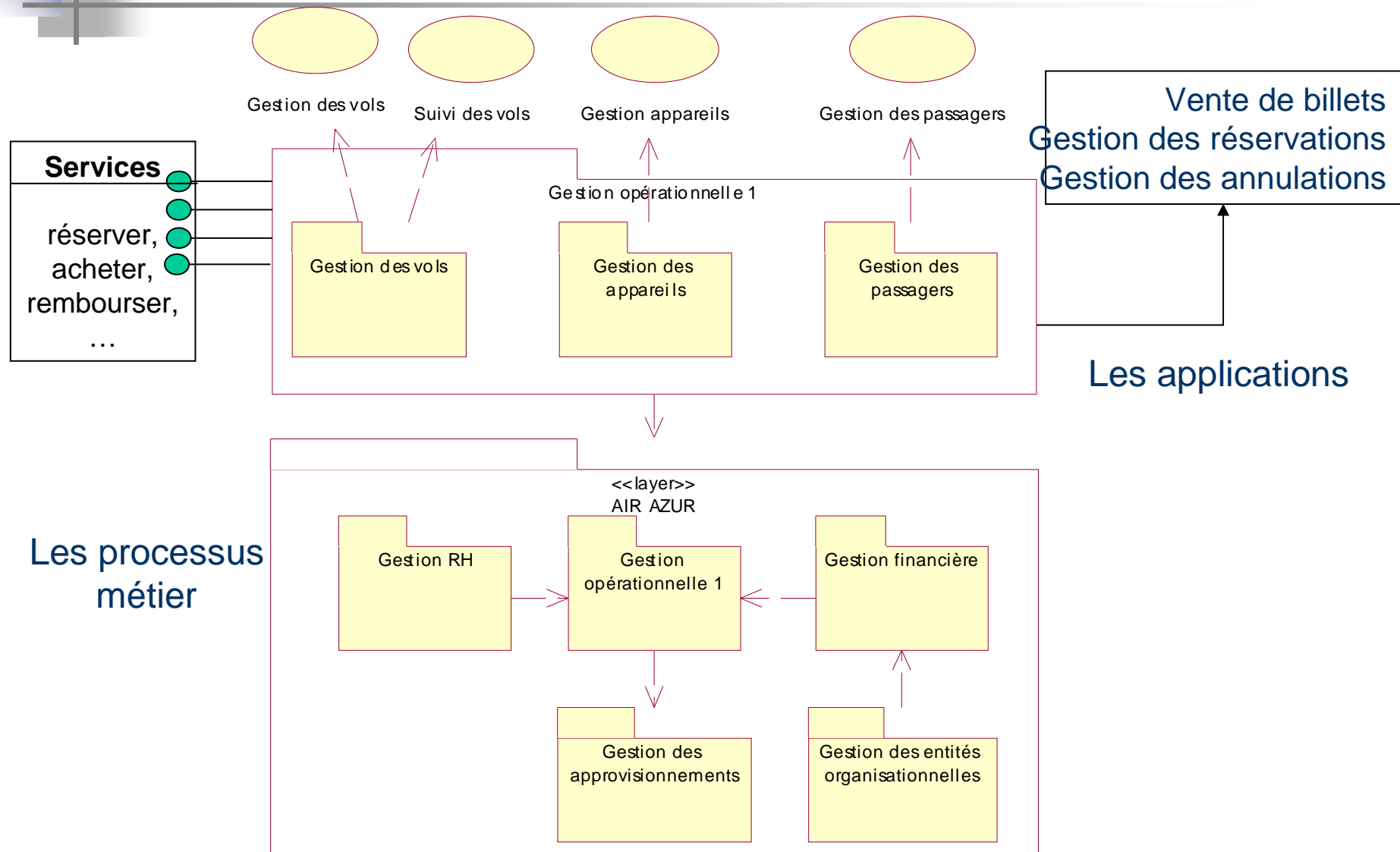
**Scénario d'exception** : Changement de vol et Remboursement

**Scénario mode inverse** : Annulation

**Scénario mode dégradé** : Panne du poste de l'agent

# La liste des processus métiers et des applications

**AIR AZUR**  
**ETAPE 1**





# Description des postes de travail (1/2)

**AIR AZUR**  
**ETAPE 1**

- Liste des postes de travail :
  - ✓ Poste directeur région,
    - Statistiques (Consultation chiffres d'affaires, bénéfices, exploitation, ...)
    - Analyse de données
  - ✓ Poste agent commercial
    - Gestion des passagers (vente de billets, réservation, annulation)
  - ✓ Poste de gestion des commandes et des livraisons
    - Gestion des carburants et des plats repas
  - ✓ Poste chef d'agence
  - ✓ ...



## Description des postes de travail (2/2)

**AIR AZUR**  
**ETAPE 1**

- Pour chaque poste, il faut décrire
  - ✓ La configuration : OS, ressources physiques
  - ✓ Le « bureau » : applications accessibles, environnement de travail, ressources partagées
  - ✓ Les habilitations : droits, mot de passe, protection des applications
  - ✓ Les acteurs



# Description des étapes

## ETAPE 2 : SPECIFICATIONS FONCTIONNELLES DETAILLEES (SFD)

- ✓ ANALYSE DES BLOCS
- ✓ (Modélisation des aspects structurel, dynamique et fonctionnel)

### ACTIVITES USDP

- ✓ **ANALYSE**
- ✓ **CONCEPTION**



**Architecture  
applicative**

### MODE OPERATOIRE (tâches)

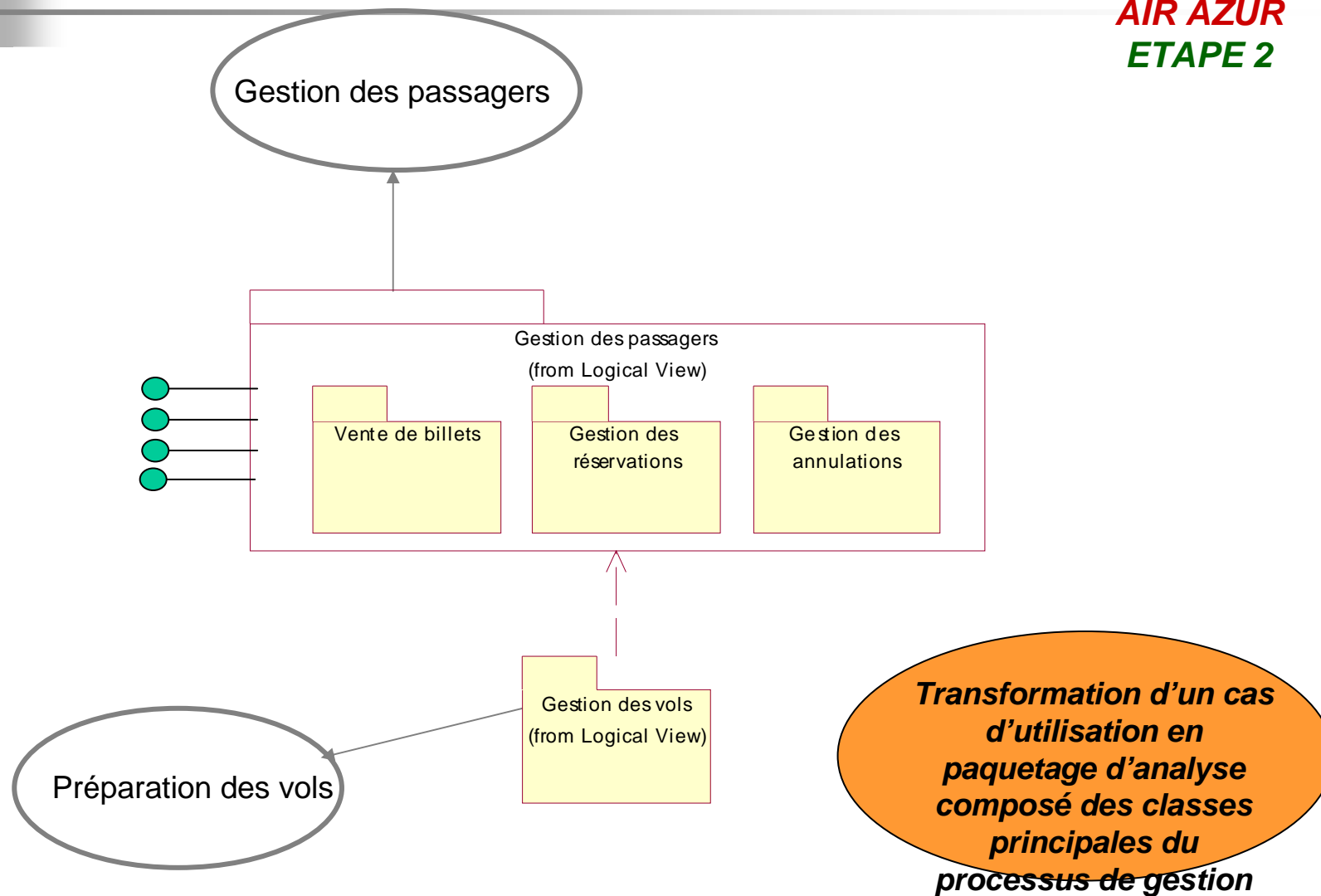
- ✓ Décrire chaque bloc obtenu à l'étape 1.
- ✓ Détailler les paquetages en décrivant les classes principales
- ✓ Analyser la dynamique des classes
- ✓ Décrire les interactions entre classes

### RESULTATS

- ✓ Description des classes des applications et des données

# Illustration: Paquetages (classes principales)

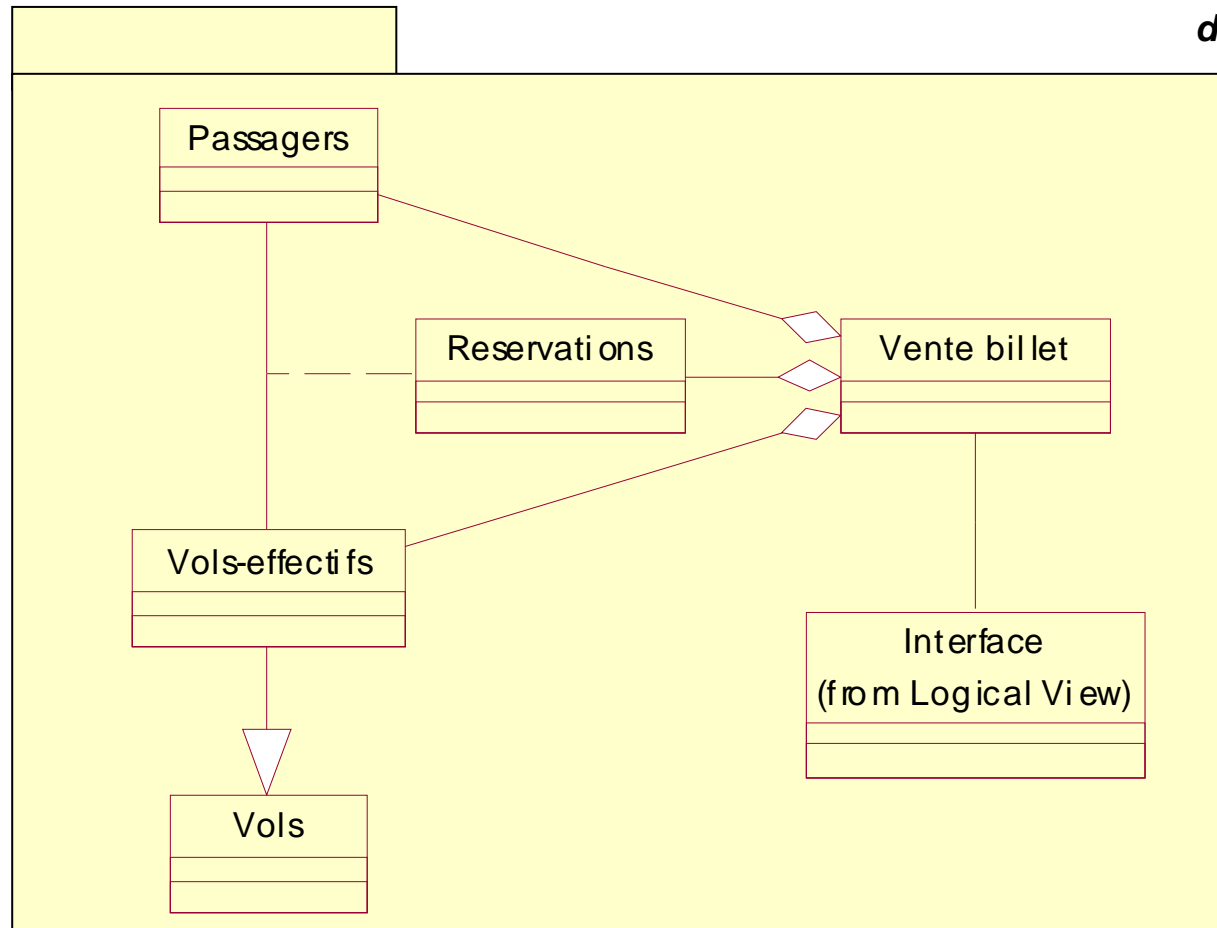
**AIR AZUR**  
**ETAPE 2**



## Illustration: Paquetages (classes principales)

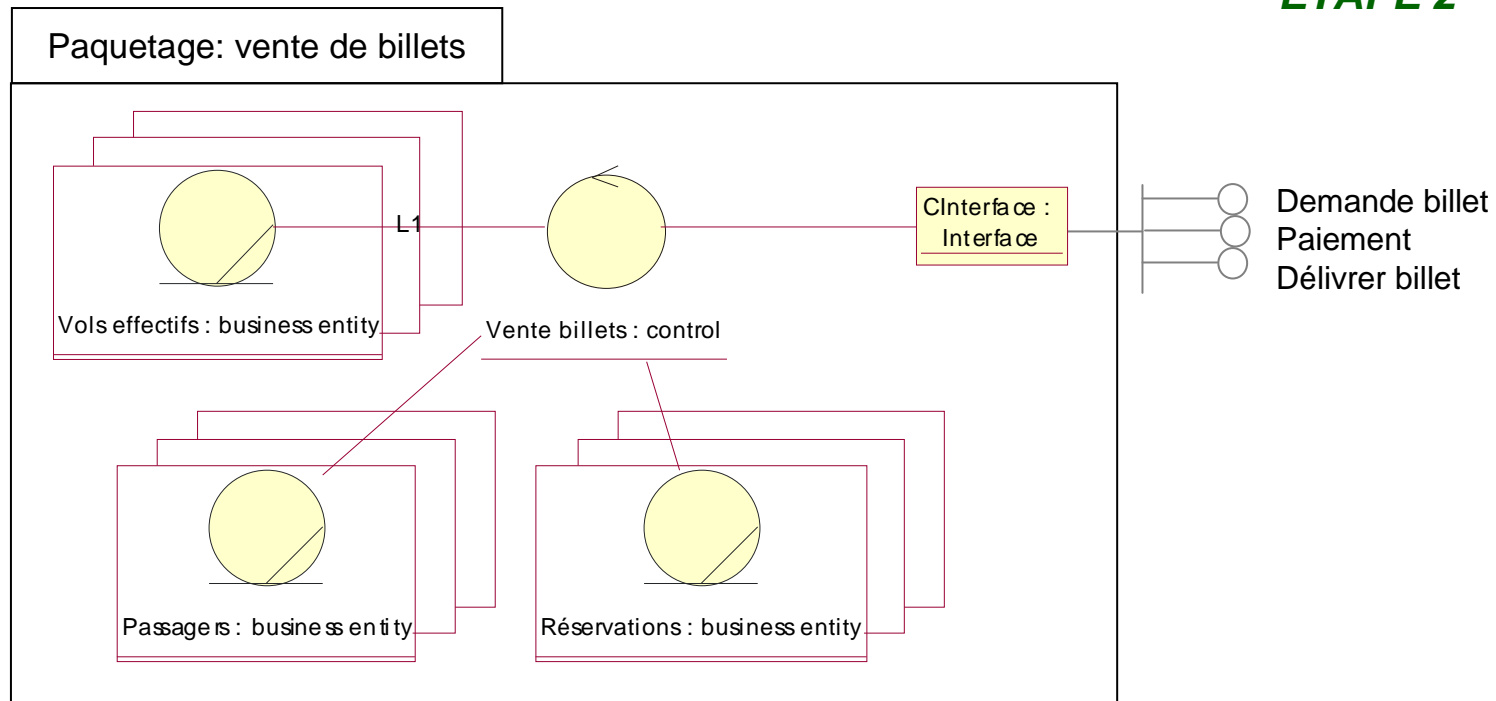
- Description du paquetage : vente billets

*Chaque paquetage de service est défini en terme de classes principales*

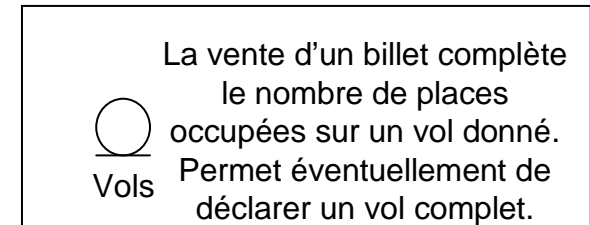
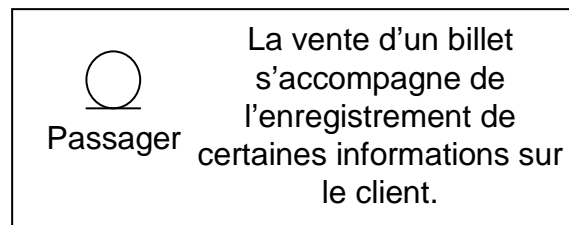
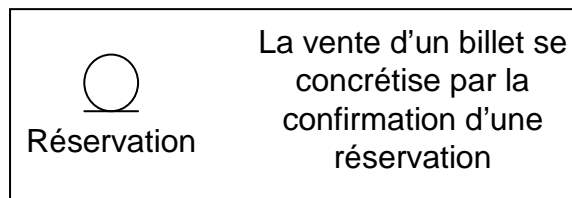


# Illustration: Paquetages (stéréotypes)

**AIR AZUR**  
**ETAPE 2**

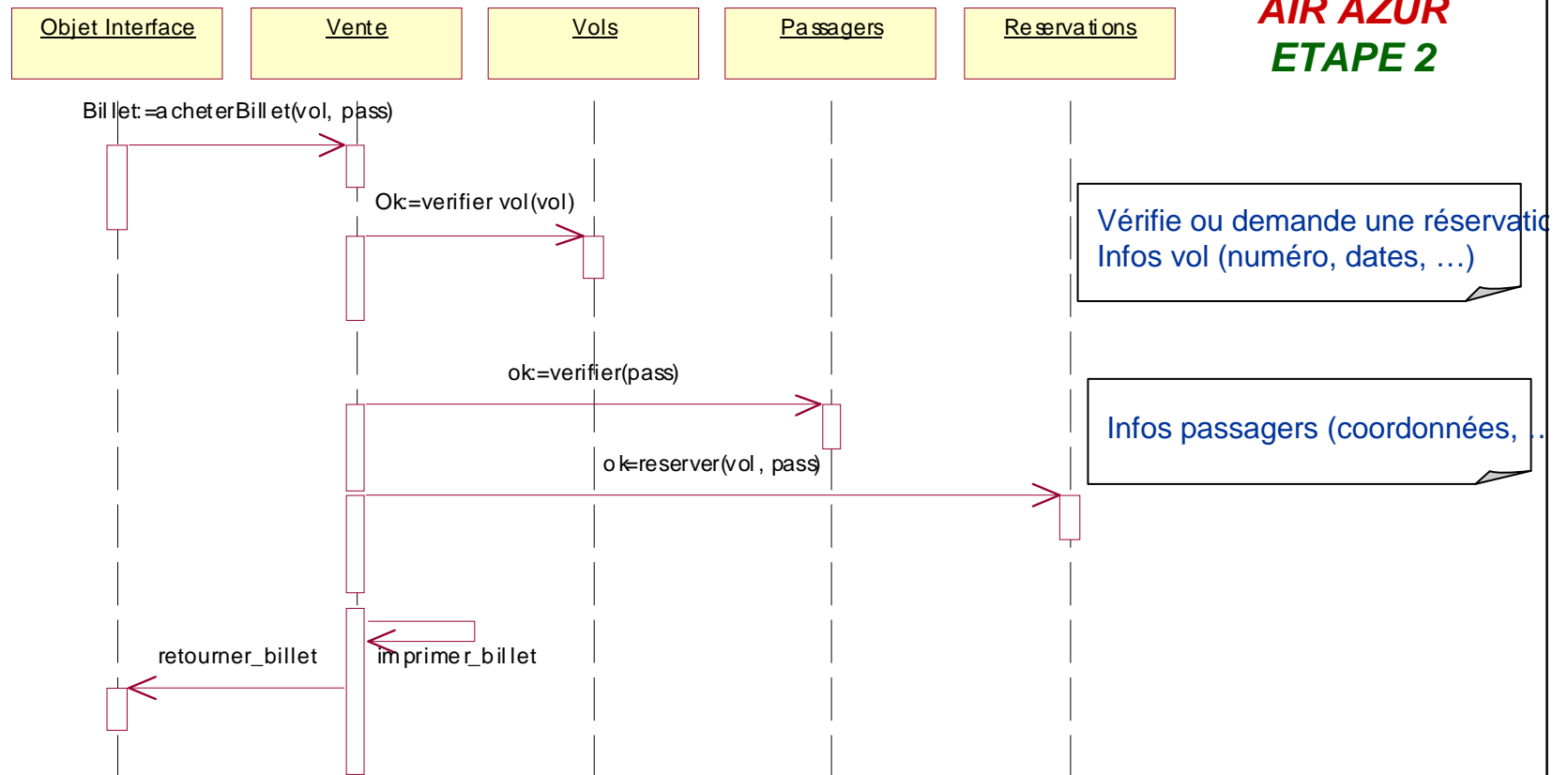


## Rôle des classes entité





# Illustration: réalisation d'un scénario



*Description de la dynamique d'un cas. Cette description peut se faire à l'aide d'un modèle quelconque de la dynamique. On choisit ici de mettre en évidence les interactions entre objets.*

- Classes frontière
  - ✓ **Interface client**
    - Identifie l'agent
    - Récupère les données sur l'achat ou la réservation
    - Transmet ces données aux classes de contrôle : vente de billet & réservation
    - Édite le billet
  
- Classes de contrôle
  - ✓ **Vente de billet**
    - crée un billet
    - modifie un billet
    - émet le billet pour le passager
  - ✓ **Gestion des réservations**
    - s'assure que le vol aux horaires indiqués existe bien
    - créer et modifie une réservation
    - met à jour le vol effectif
  - ✓ **Gestion des annulations**
    - supprime une réservation

***On définit (sous forme textuelle) les responsabilités des classes identifiées en terme de rôles et services que remplit chacune des classes.***

- Classes entités

- ✓ **Passager**

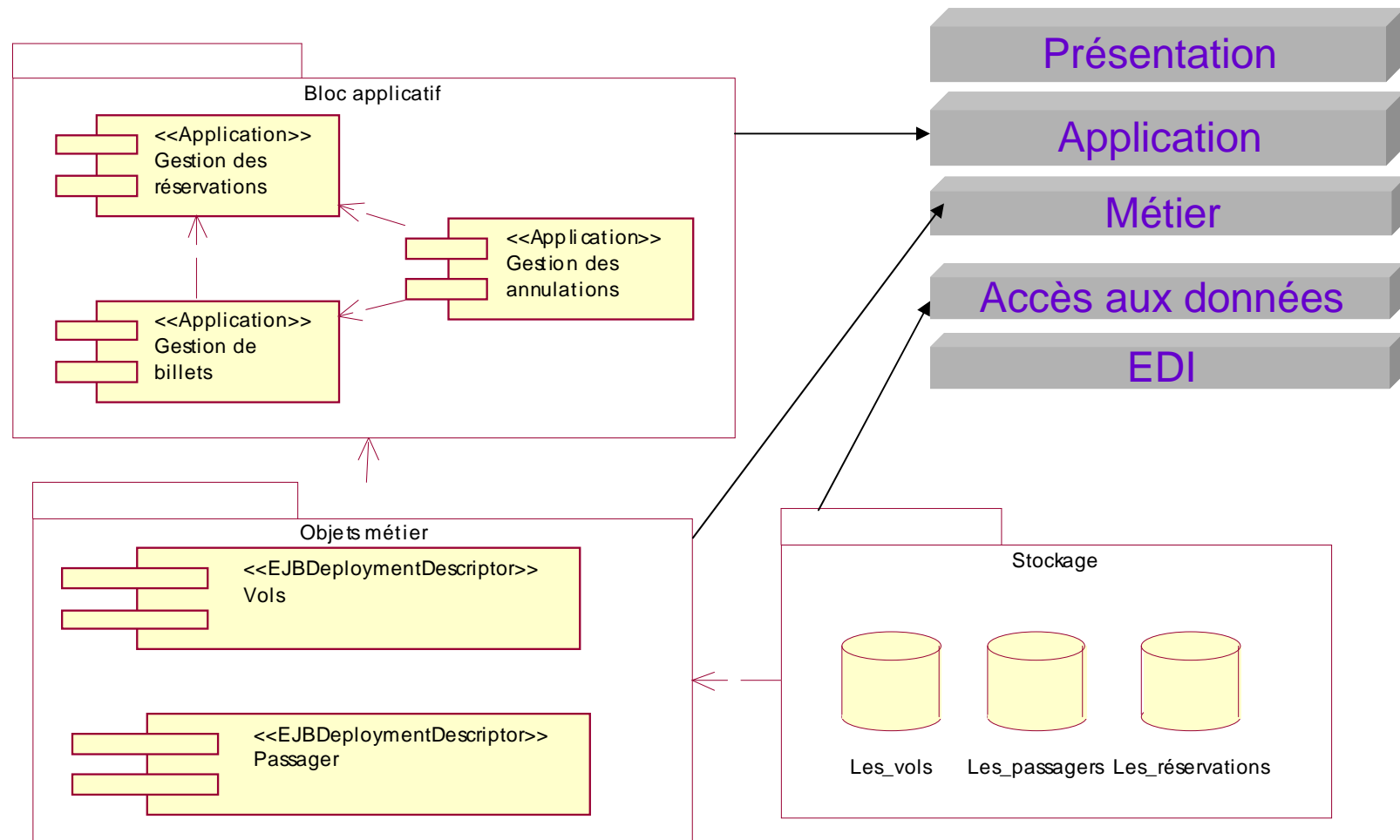
- numéro de passager
- nom et adresse
- date de naissance
- calculer âge
- créer un passager
- modifier, supprimer

- ✓ **Vol effectif**

- numéro vol
- date du vol
- heures du vol
- places libres
- identification
- enregistre un vol
- supprime un vol
- annule un vol

- Etc ...

*Les classes « entités » qui sont souvent maîtrisées (classiques, existant dans de nombreuses applications standard) sont directement décrites en terme d'attributs et de signatures de méthodes.*



# Description des étapes

## ETAPE 3 : VALIDATION

CONSOLIDATION DES SPECIFICATIONS (validation des modèles)

### ACTIVITES USDP

- ✓ ANALYSE
- ✓ CONCEPTION

**Architecture  
applicative**

### MODE OPERATOIRE (tâches)

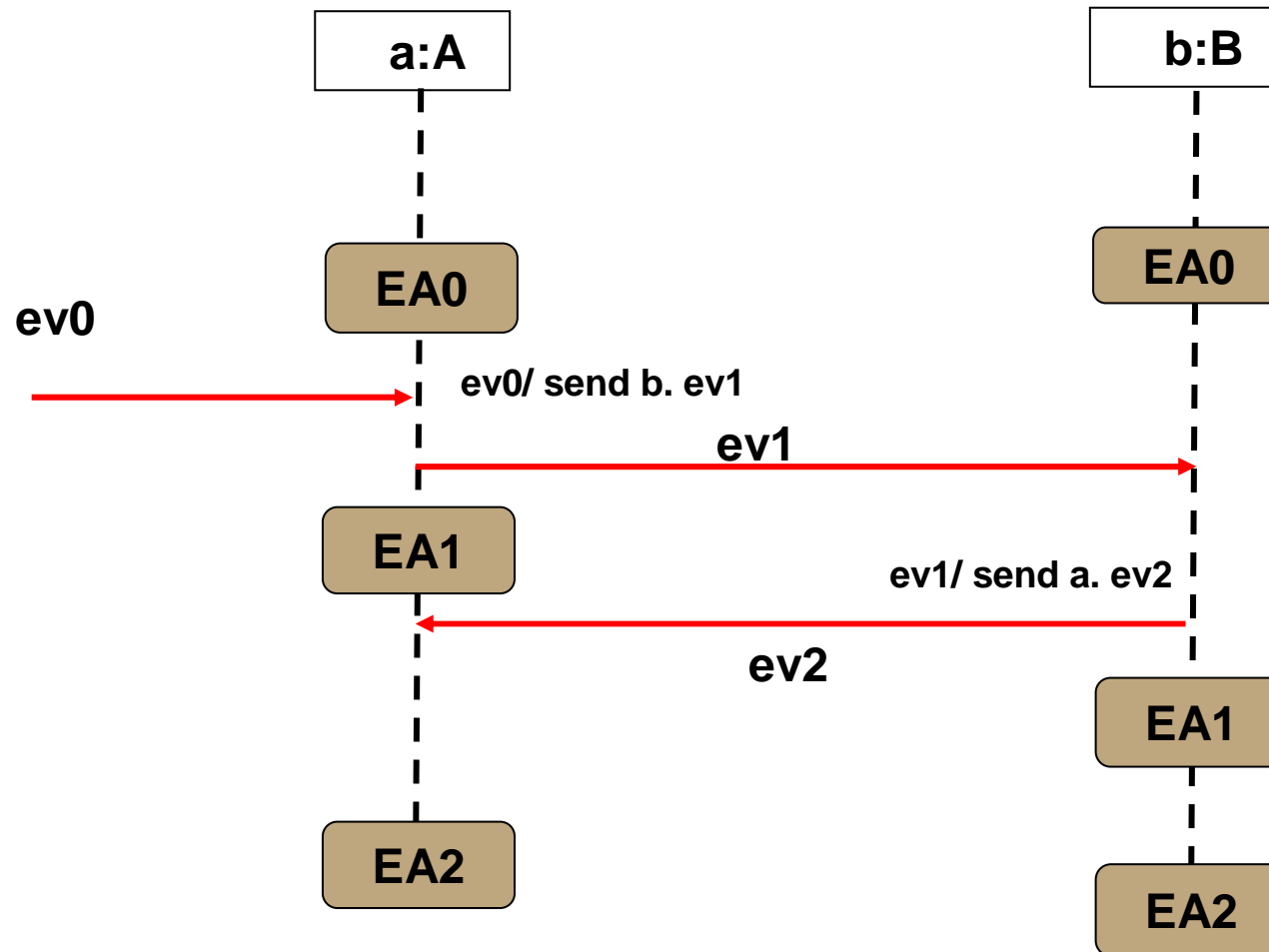
- ✓ Décrire les flux d'échange entre les blocs
- ✓ Compléter le diagramme métier
- ✓ Valider les diagrammes (dynamique et fonctionnel) obtenus à l'étape précédente

### RESULTATS

- ✓ Description fine des applications

- Lien très fort entre diagramme d'interaction (DI) ou diagramme de collaboration (DC) et diagramme d'états (DE).
- Les DE sont précis,
  - ✓ ils permettent de valider les DI
  - ✓ Ils peuvent inciter à compléter des DI existants ou créer de nouveaux
- Exemple
  - ✓ Relation entre 2 types de diagrammes.
  - ✓  $DS(a(A), b(B))$
  - ✓ L'interaction entre ces 2 objets consiste en l'enchaînement de 2 évènements:
    - - *Suite à la réception de l'évènement  $ev0$ ,  $a$  envoie  $ev1$  à  $b$ ,*
    - - *En retour,  $b$  envoie  $ev2$  à  $a$*

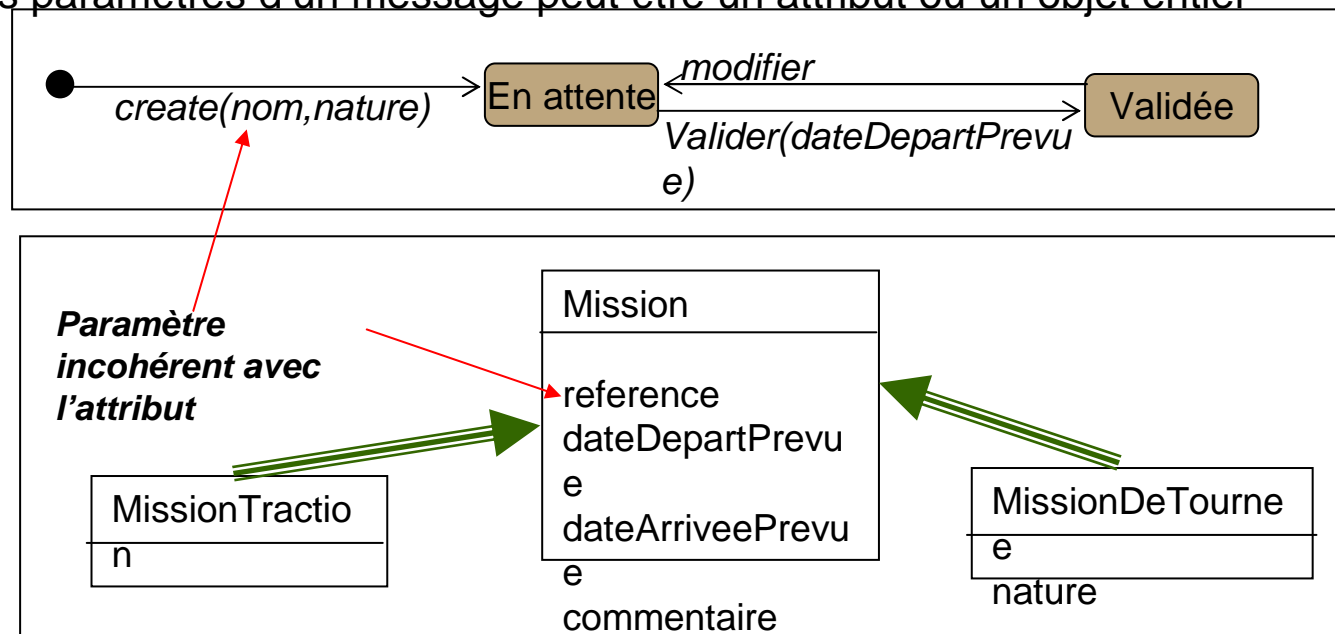
## Validation DI/DE (exemple)



## Validation DS/DD

### Confronter les modèles statiques et dynamiques

- ✓ Un message d'un DD peut être un appel d'opération sur un objet par un autre objet
- ✓ Un événement ou une action sur une transition peut correspondre à l'appel d'une opération
- ✓ Un DI met en jeu des objets
- ✓ Une opération peut être décrite par un DE
- ✓ Une condition de garde peut consulter des attributs
- ✓ Une action sur une transition peut manipuler des attributs
- ✓ Les paramètres d'un message peuvent être un attribut ou un objet entier







## Description des étapes

### ETAPE 4 : SPECIFICATIONS TECHNIQUES POUR UNE ARCHITECTURE OPERATIONNELLE

- ✓ ELABORATION D'UNE ARCHITECTURE OPERATIONNELLE
- ✓ DEPLOIEMENT DES BLOCS

#### ACTIVITE USDP

- ✓ ANALYSE
- ✓ CONCEPTION



Architecture  
opérationnelle

#### MODE OPERATOIRE (tâches)

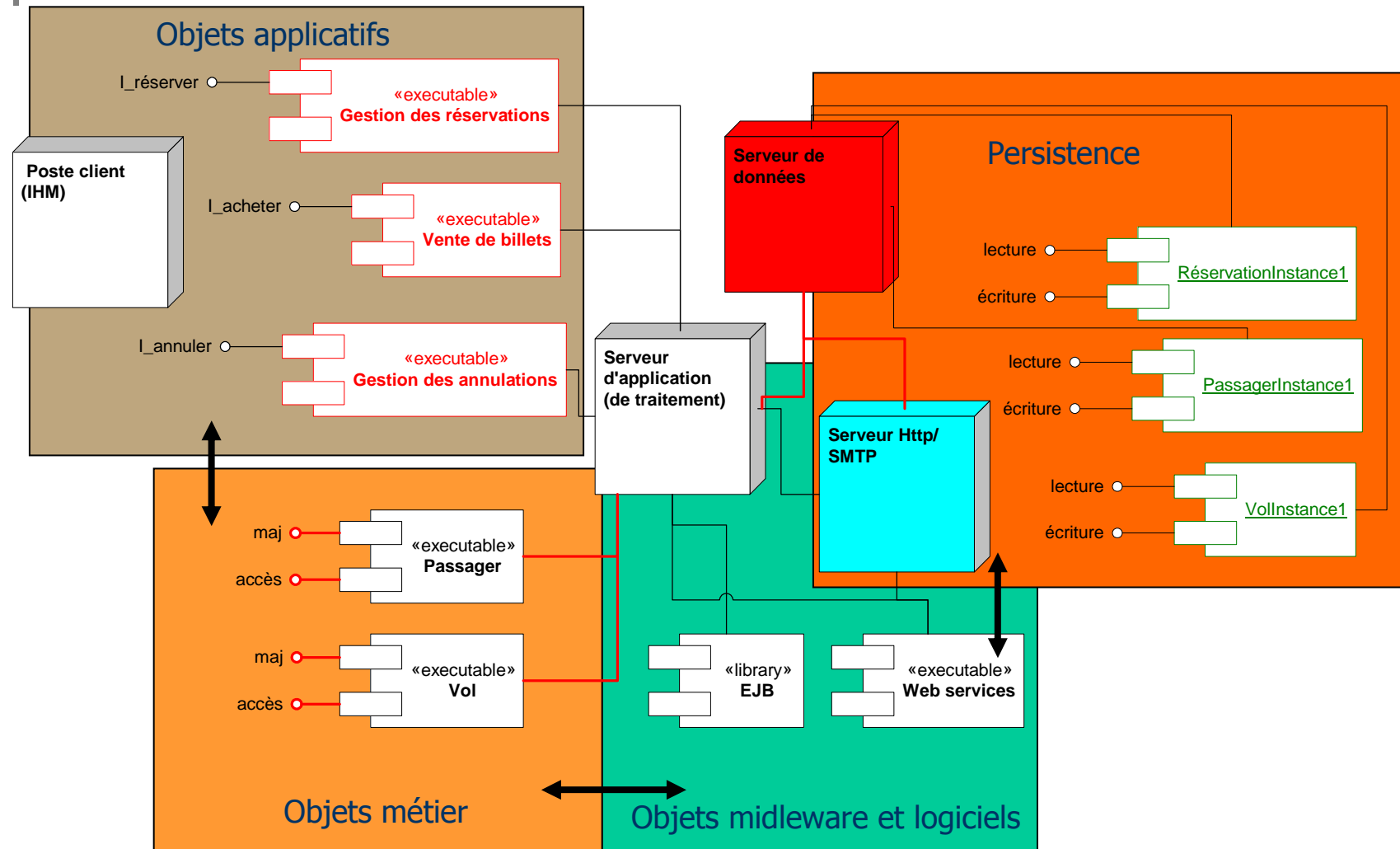
- ✓ Choisir des patterns d'architecture
- ✓ Définir l'architecture logicielle
- ✓ Spécifier l'architecture technique
- ✓ Répartir les classes des blocs de l'étape 2 sur les nœuds de l'architecture

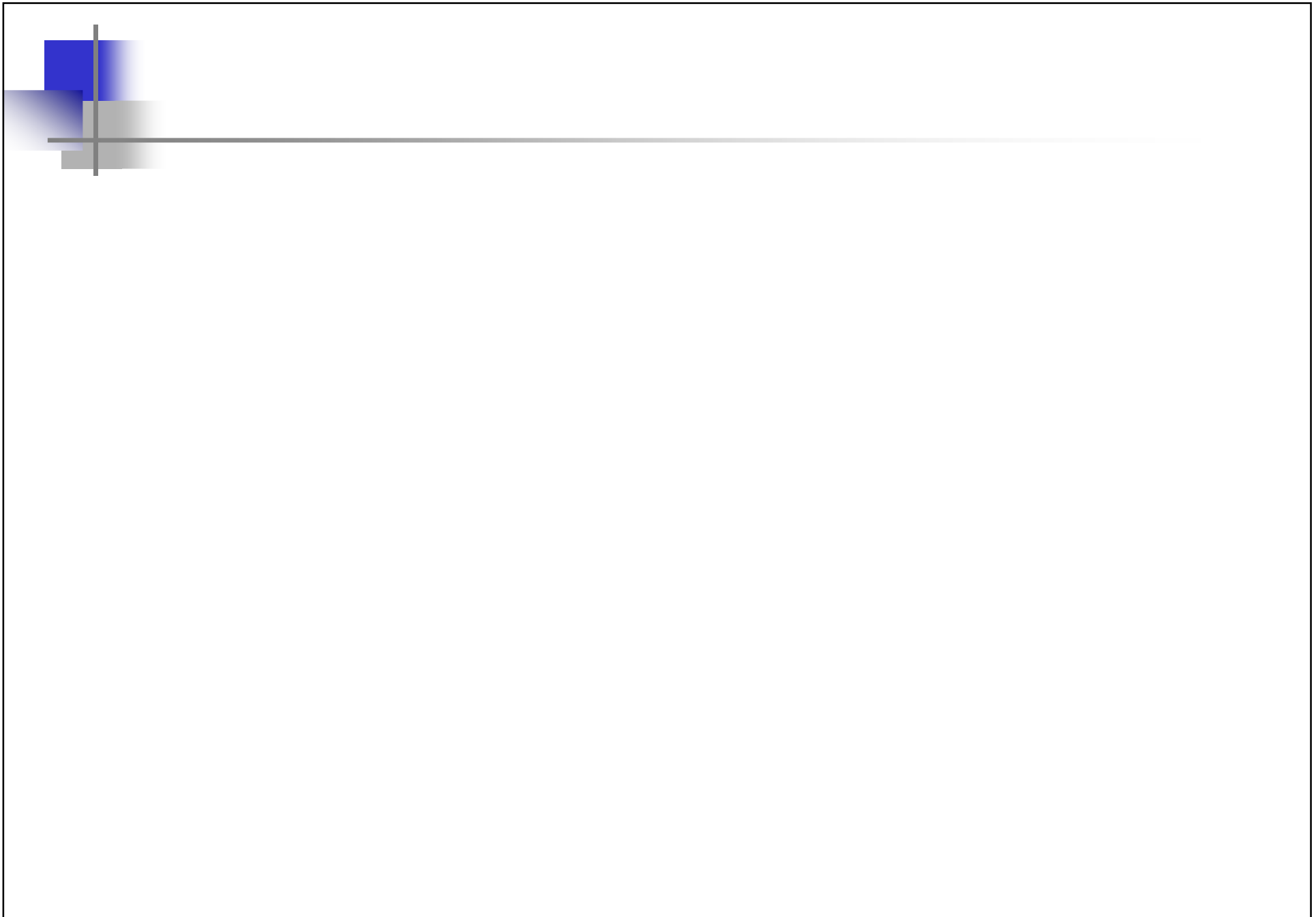
#### RESULTATS

- ✓ Une architecture opérationnelle
- ✓ Une architecture d'exécution
- ✓ Une répartition des applications et des composants

- Applications au siège :
  - ✓ GRH
  - ✓ Gestion des passagers et des clients,
  - ✓ suivi des vols,
  - ✓ Gestion des approvisionnements,
  - ✓ gestion financière et comptable,
  - ✓ Gestion des entités organisationnelles (régions, escales, agences).
- Applications dans les agences :
  - ✓ Gestion des passagers et des clients,
  - ✓ suivi des vols,
  - ✓ Gestion des commandes,
  - ✓ gestion de la trésorerie.
- Applications dans les régions :
  - ✓ Gestion des passagers et des clients,
  - ✓ préparation des vols,
  - ✓ Gestion des marchés,
  - ✓ gestion de la trésorerie,
  - ✓ Gestion des agences.
- Applications dans les escales :
  - ✓ Gestion des passagers et des clients,
  - ✓ Gestion des appareils,
  - ✓ Gestion des livraisons,
  - ✓ gestion de la trésorerie (vols transitant par l'escale).

# Vue d'exploitation



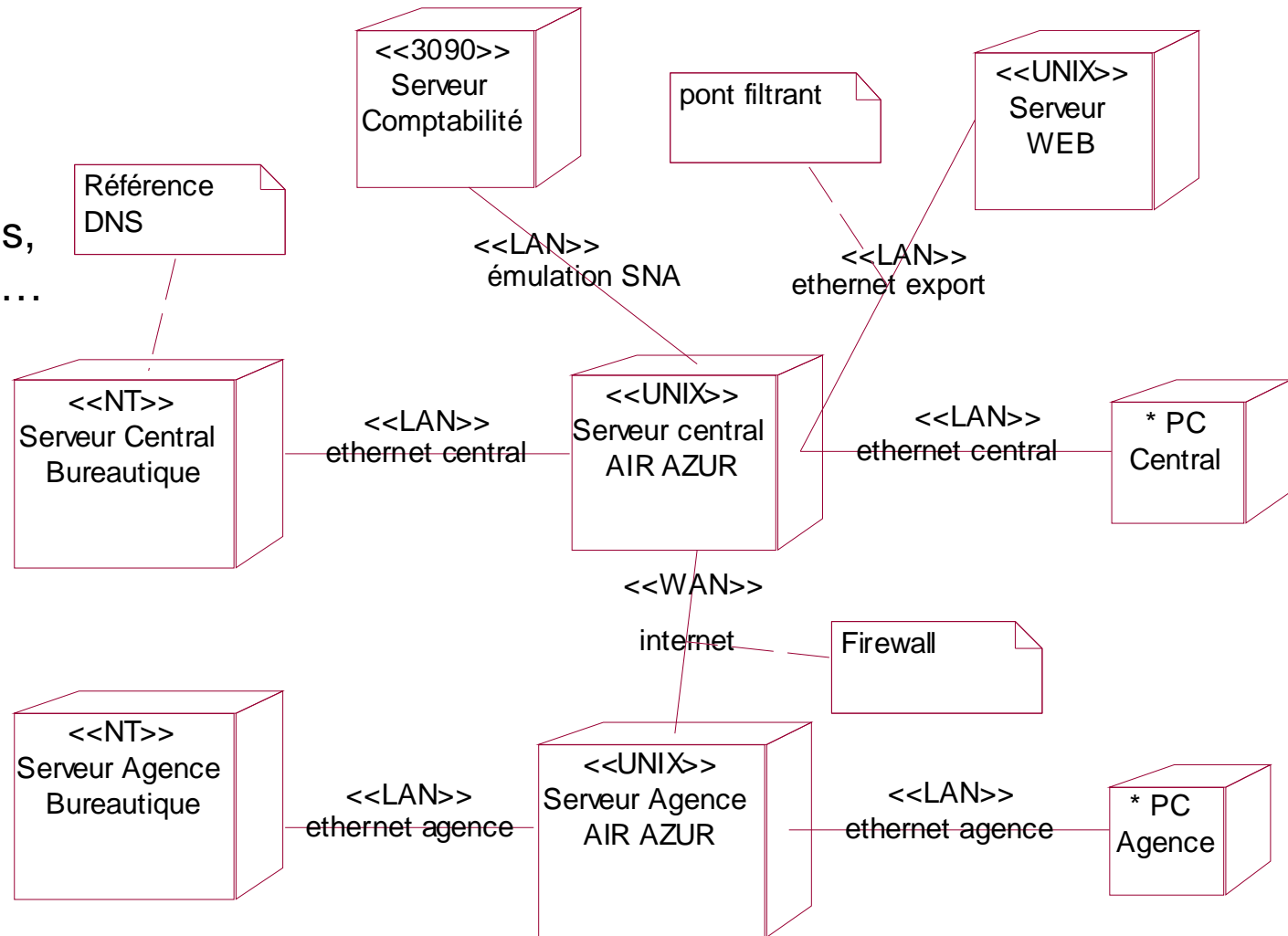


# Vue logique de déploiement

## Serveurs

:données,  
traitement, mails  
Web, impressions,  
dns, sauvegarde...

## Postes : lourd, léger



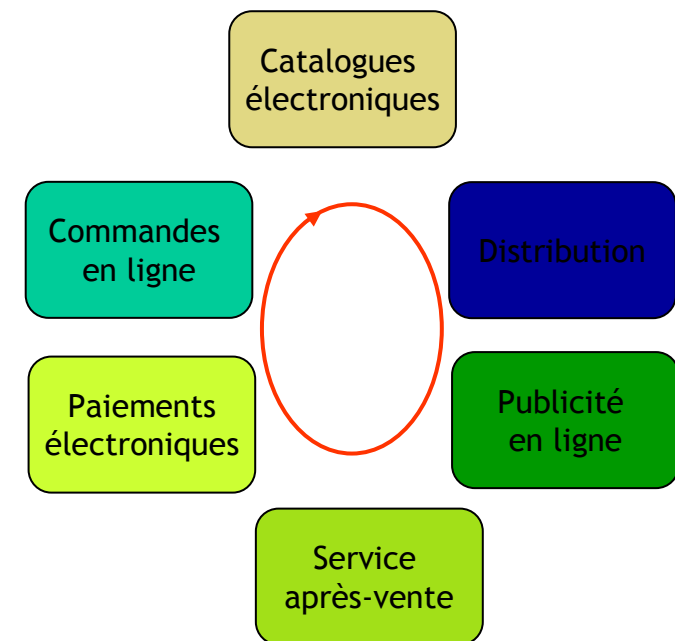
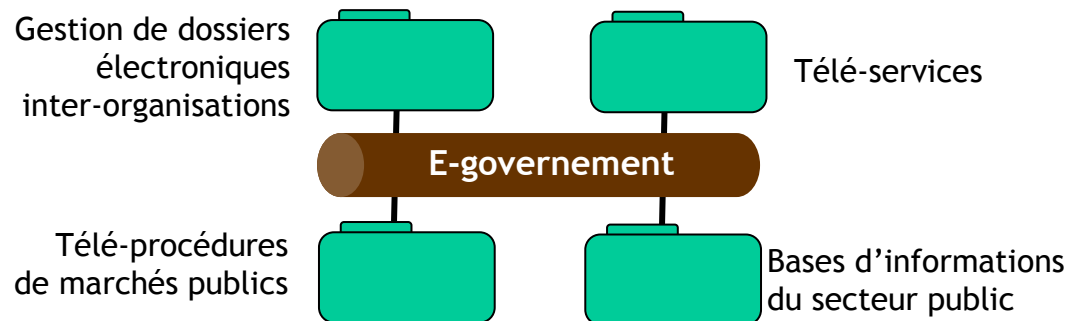


# Interopérabilité de SI

---

# Interopérabilité des systèmes d'information

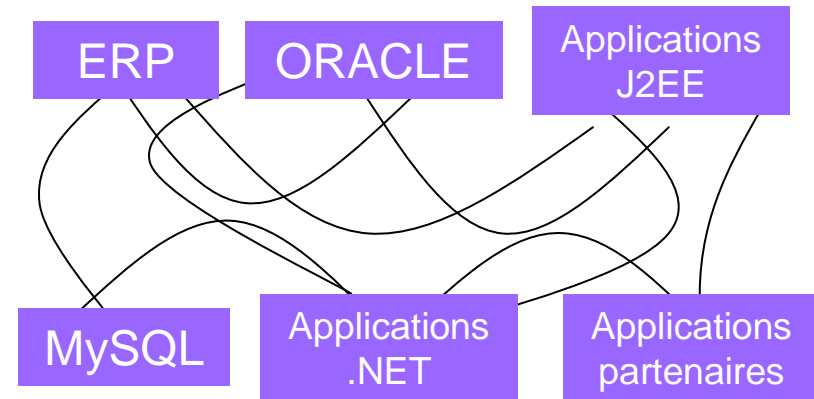
- Business to Consumer (B2C)
- Business to Business (B2B)
- Application to Application (A2A)
- Business to Application (B2A)
- Business to Administration (B2A): e-gouvernement



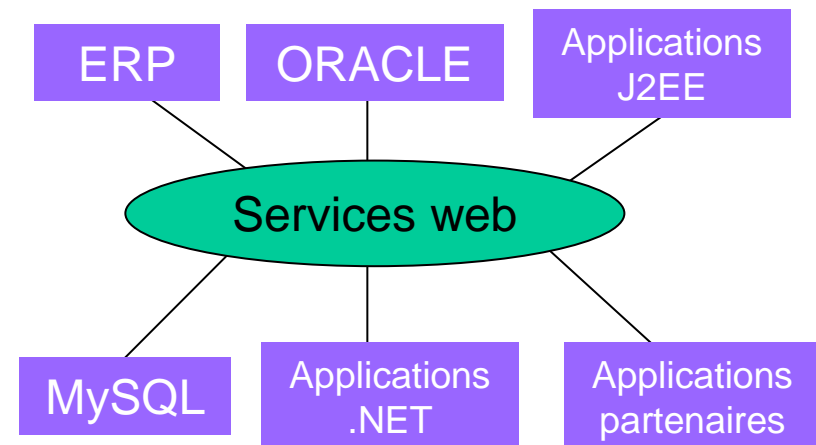
# Services web

- Problématique de l'intégration des données et des applications

- ✓ Liaisons encodées en dur
- ✓ Plateformes hétérogènes
- ✓ Évolutions difficiles

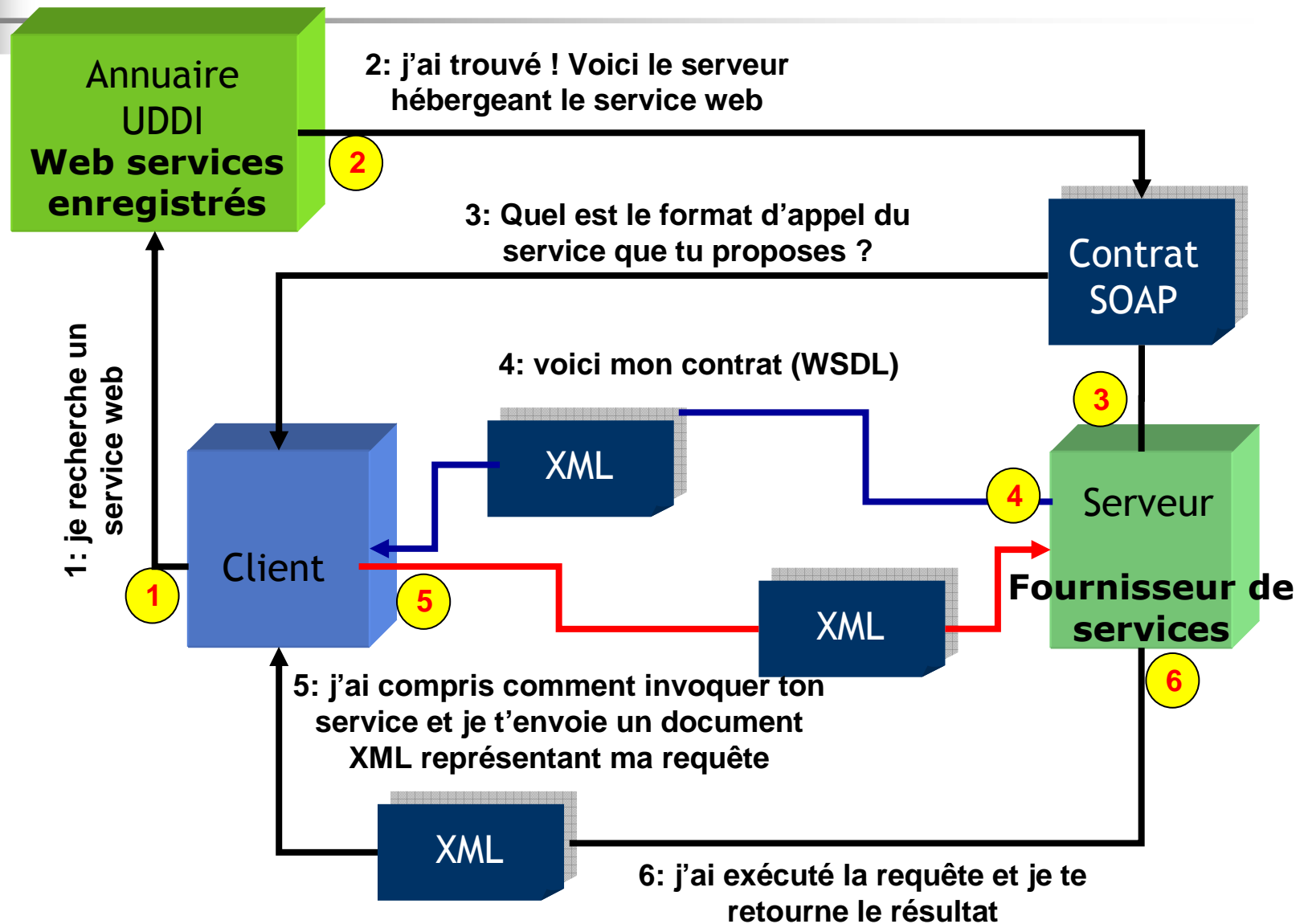


- liaisons réalisées à l'exécution
- standards
- Indépendants des plates-formes
- Configuration flexible

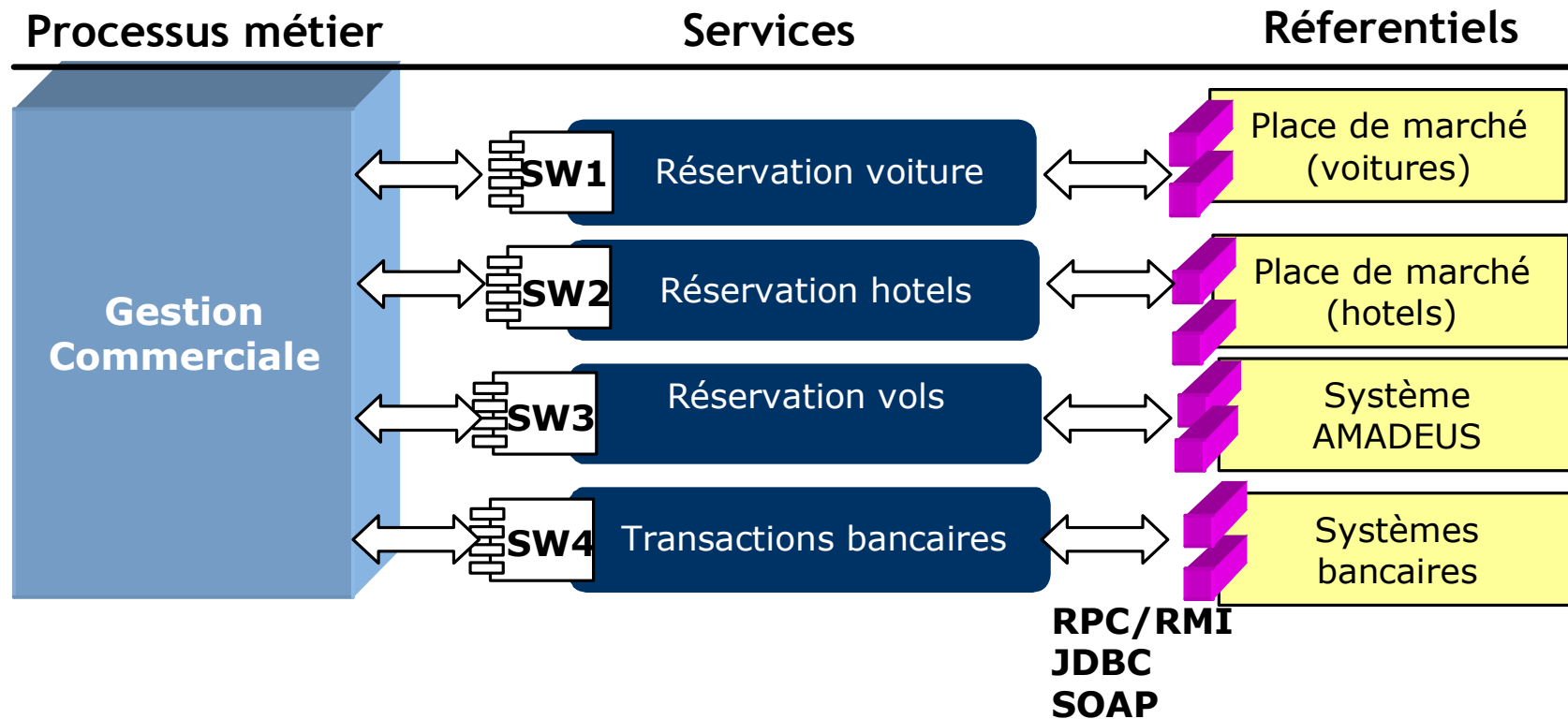




# Les services web

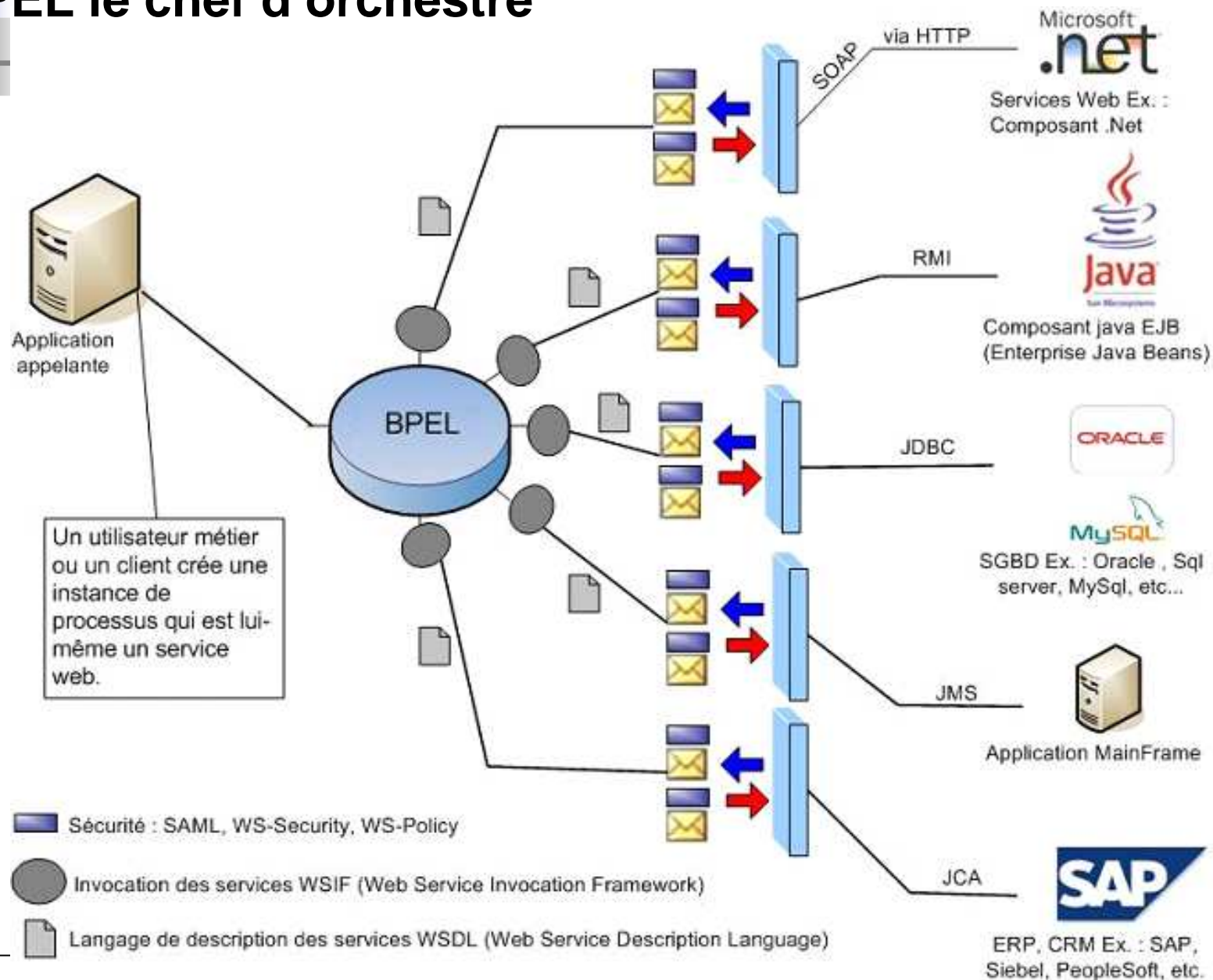


## Les services web: synoptique d'utilisation



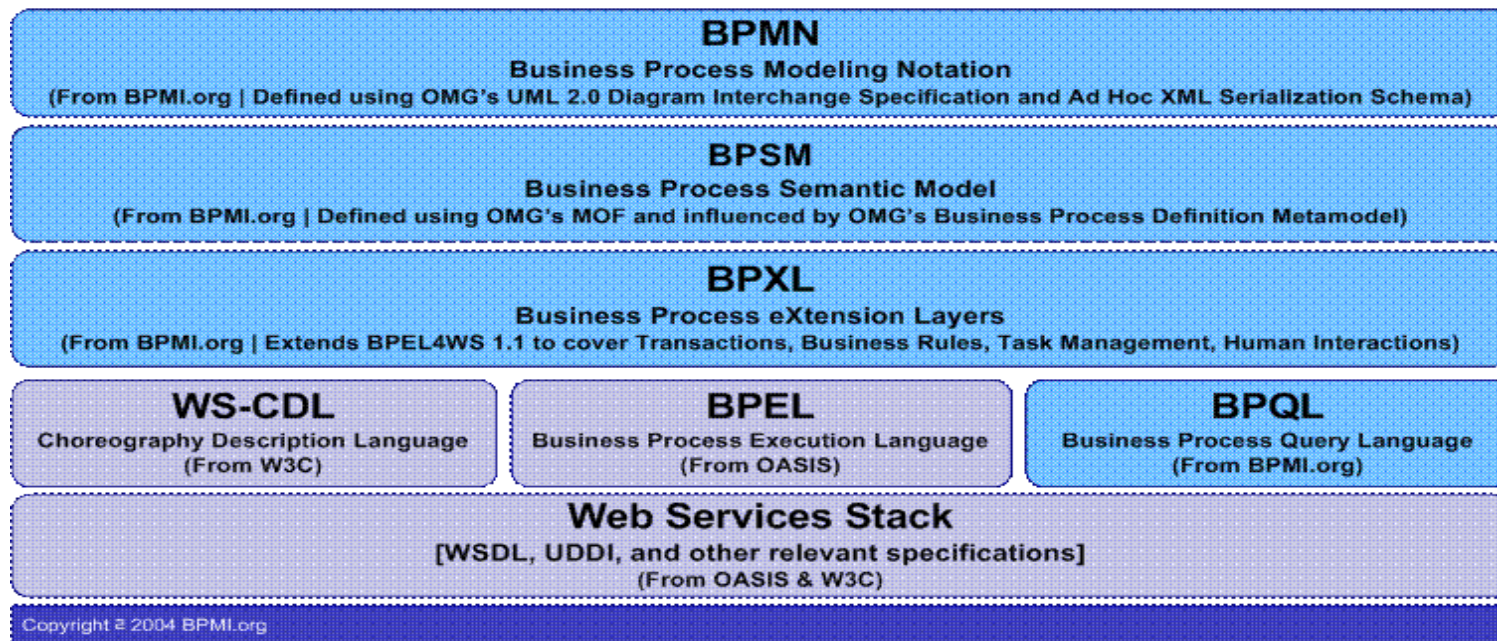
[http://www.active-endpoints.com/PlayTour/AWFPro\\_FlashTour\\_Intro+BuildProcess.htm](http://www.active-endpoints.com/PlayTour/AWFPro_FlashTour_Intro+BuildProcess.htm)

# BPEL le chef d'orchestre



# Les standards de l'OASIS [www.bpmi.org](http://www.bpmi.org)

- OASIS: Organization for the Advancement of Structured Information Standards
  - ✓ Fondée en 1993, plus de 5000 participants représentant plus de 600 organisations de 100 pays.
  - ✓ Microsoft, IBM, BEA, SAP, et Siebel en Mai 2003
- **Business Process Modeling Language (BPML)**
  - ✓ Grammaire XML en compétition (remplacée par) avec BPEL
- **BPMI travaille sur d'autres niveaux de représentation des processus et notamment BPMN**



# Références bibliographiques

- USDP (Unified Software Development Process, J. Rumbaugh, Ed. Eyrolles
- UML en action, P. Roques, F. Vallée, Ed Eyrolles
- UML à SQL. Conception de bases de données , Soutou, Christian - 2002, Éditeur : EYROLLES
- UML pour l'analyse d'un système d'information. Le cahier des charges du maître d'ouvrage, Hugues, Jean/Leblanc, Bernard/Morley, Chantal - paru le 30/05/2002, Éditeur : DUNOD -
- UML et les Design Patterns, Larman, Craig - paru le 27/02/2002, Éditeur : CAMPUS PRESS / MACMILLA
- De Merise à UML , Collectif, - paru le 01/10/2001, Editeur : EYROLLES
- UML par la pratique. Etudes de cas et exercices corrigés , Roques, Pascal - paru le 24/04/2001 , Editeur : EYROLLES
- Merise et UML pour la modélisation des systèmes d'information. Un guide complet, Gabay, Joseph - paru le 10/03/2001, Editeur : DUNOD
- UML la notation unifiée de modélisation objet. De Java aux EJB, avec CD-Rom, 2èm, Lai, Michel - paru le 11/04/2000, Editeur : DUNOD
- Modélisation objet avec UML. Avec CD-Rom, 2ème édition, Muller, Pierre-Alain/Gaertner, Nathalie - paru le 22/03/2000, Editeur : EYROLLES
- Bouzeghoub, Rochefeld, OOM : la conception objet des SI , Editions Hermès, 2000
- Urbanisation des systèmes d'information, J. Sassoon, 1998