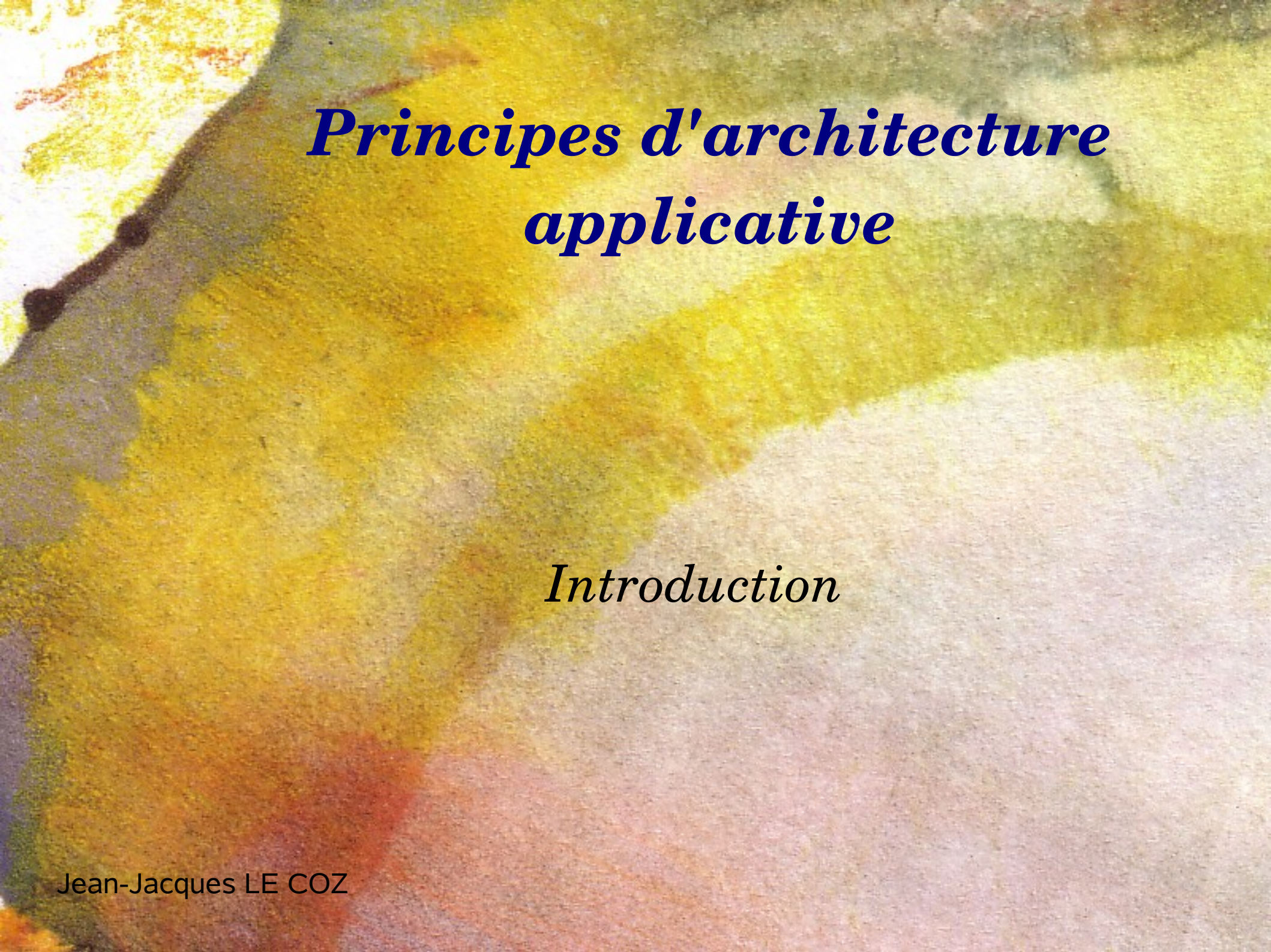


Principes d'architecture applicative

Introduction

Jean-Jacques LE COZ

Définition

- L'architecture logicielle décrit de manière symbolique et schématique les différents composants :

  D'un ou de plusieurs programmes informatiques

  Leurs interrelations

  Et leurs interactions

Styles architecturaux

- Architecture en appels et retours

  Approche par décomposition fonctionnelle

  Basée sur l'utilisation de fonctions

  - Extension modulaire
  - Extension au paradigme objet

- Forme dérivée

  Architecture distribuée

Styles architecturaux

- Architecture en couches

  Approche par bibliothèques

  - Une bibliothèque spécialisée utilise des bibliothèques moins spécialisées qui elles-même utilisent des bibliothèques génériques

  Conséquence du concept de réutilisation

Styles architecturaux

- Architecture centrée sur les données
  - Composant central responsable du CRUD
    - Base de données
    - Datawarehouse
  - Séparation des données, des traitements et de la présentation
  - Intégration par les données
    - Nécessite une grande stabilité
    - Peu extensible

Styles architecturaux

- Architecture en flot de données

  Composants reliés par flux de données

  - Sans valeur ajoutée

    Architecture par lot (batch)

  - En réseau et avec transformations

    Architecture de médiation

Styles architecturaux

- Architecture orientée objets
  - Les composants intègrent
    - Données
    - Comportements (traitements)
  - Communication et coordination
    - Mécanisme de passage de messages
    - Extension de l'architecture en appels et retours
  - Trois piliers
    - Encapsulation, héritage, polymorphisme

Styles architecturaux

- Architecture orientée agents
  - Les agents utilisent de manière intelligente les autres agents
  - Établissement de dialogues avec les autres agents
    - Négociation et échange

Historique

- Périodes
  - 1960 – 1970
    - Programmation structurée
    - Programmation modulaire
      - Diagramme de flux, langage Simula
  - 1970 – 1980
    - Séparation
      - Architecture statique
        - Dépendance fonctionnelle
      - Architecture dynamique
        - Activité, séquence, réseaux de Pétri

Historique

- Périodes
  - 1980 – 1990
    - Architecture orientée objet
    - Architecture client-serveur
      - Centrée sur les données
  - 1990 – 2000
    - UML, Patrons de conception
    - SGBD R/O
    - Réseaux ouverts (Internet)
    - Architecture à 3 couches web

Historique

- Périodes

  2000 – 2010

  - XML
  - Médiation de données
  - Développement orienté agent

    KQML (Knowledge Query Manipulation Language)

Modélisation de l'architecture

- Approche par les vues
- Liaison entre les vues
- Modèles

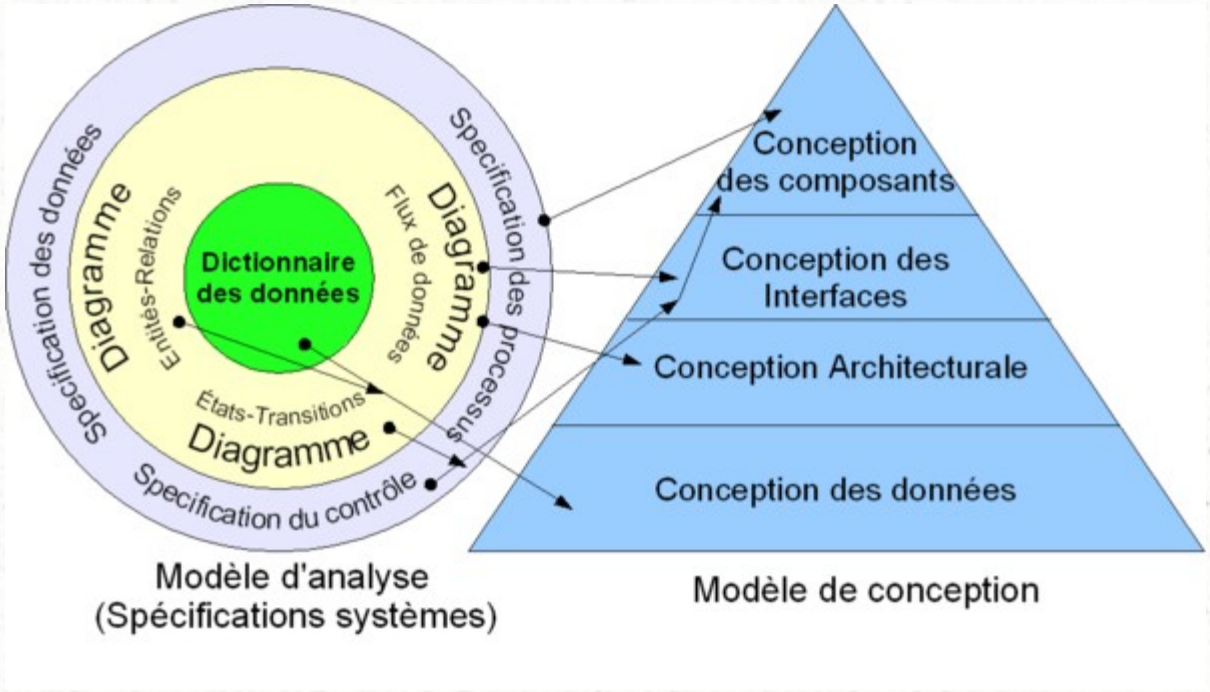
  Modèle d'analyse

  Modèle 4+1 vues (Krutchen)

  Cas d'utilisation (use case)

  Vues

  - Logique, processus, réalisation, déploiement


Spécification des données
Diagramme
Entités-Relations
Dictionnaire des données
Diagramme
Flux de données
Spécification des processus
États-Transitions
Diagramme
Spécification du contrôle
Modèle d'analyse
(Spécifications systèmes)
Conception des composants
Conception des Interfaces
Conception Architecturale
Conception des données
Modèle de conception

Objectifs

- Structuration
- Couplage
- Souplesse
- Réduction de la complexité

Structuration

- Structuration en couches

  Séparation

  - Logique de **présentation**
  - Logique de **service**
  - Logique de manipulation des **données**

- *Permet le déploiement en environnement distribué*

Couplage

- Le couplage entre les couches est **faible**
- Le couplage des composants à l'intérieur d'une couche est **fort**
- *Facilite les tests de chaque couche de manière indépendante*

Souplesse

- Matérialisation des couches dans des **paquetages** *(package)* séparés
- Assemblage, groupement de composants
- *Souplesse de déploiement*

Réduction de la complexité

- Le comportement des composants **est connu à l'avance**
- Il y a une logique d'enchainement entre les composants
- Le **périmètre** des composants est **réduit** et banalisé
- *Le travail de développement est simplifié car fortement cadré*

Modèle en couches

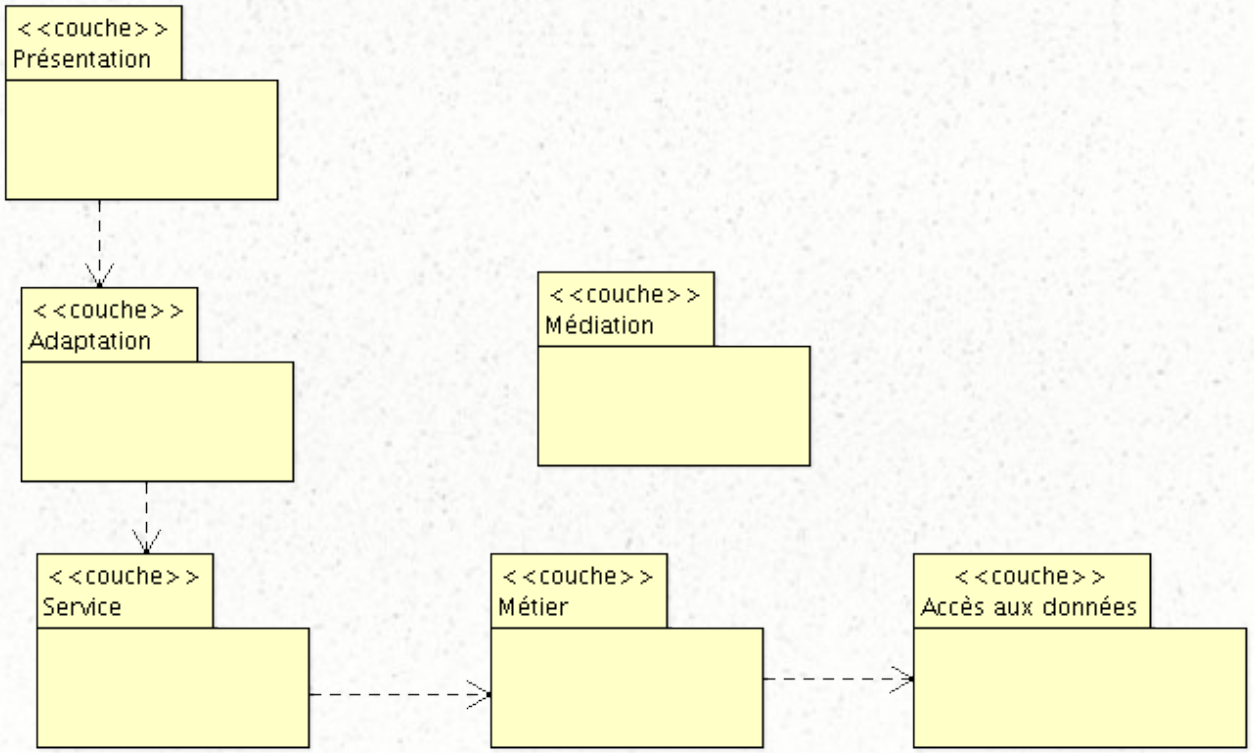
- Couches internes

  Présentation, Adaptation, Service, Métier, Accès aux données

- Couche externe

  Médiation

Schéma de définition des couches



Couche présentation

- Affichage à l'utilisateur

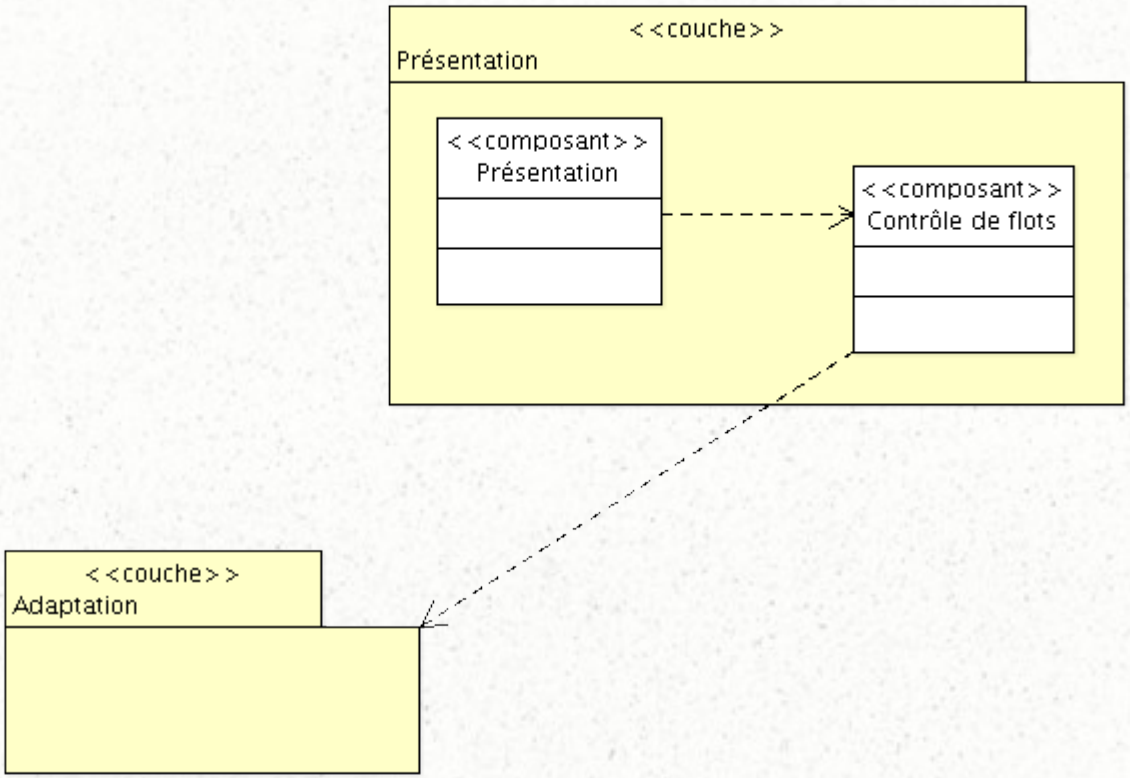
  IHM *(Interface Homme-Machine)*

- Plusieurs instances de présentation
- Deux types de composants

  Rendu à l'utilisateur (IHM)

  Logique d'enchaînement des IHM

  - MVC, MVC2, HMVC



Composants de la couche Présentation

Composants de la couche présentation

- Composants de la couche rendu

  Portlet – WebForm

  Servlet, JSP – ASP, ASP.NET

  JavaFX, RCP, XUL

  Swing – ActiveX, WinForm

- Composants de logique d'enchainement

  JSF, Struts, etc.

  Listener

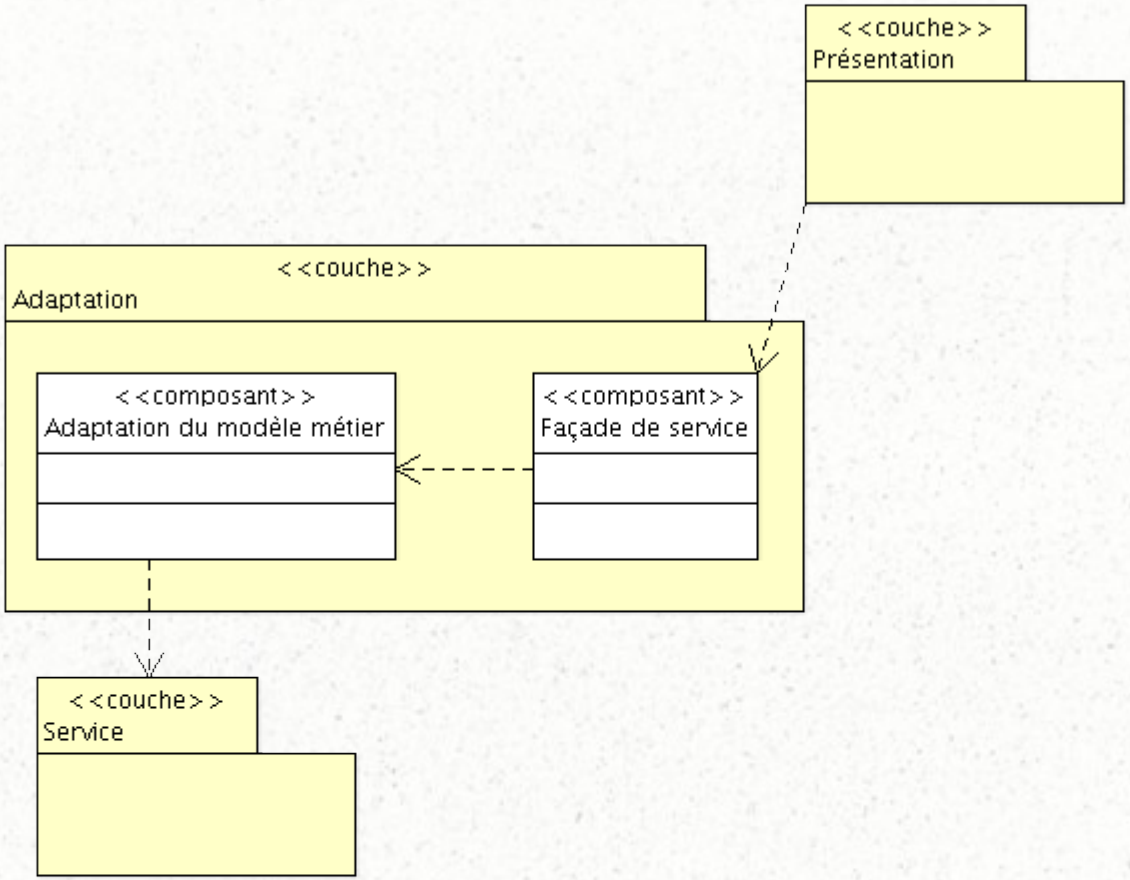
Couche adaptation

- Ensemble de **façades** orienté *(use case)* cas d'utilisation

  Pour masquer la complexité du métier et des services

  Indépendant des technologies de l' IHM

- Ensemble de composants effectuant l'**adaptation** entre le modèle métier et le modèle de présentation



Composants de la couche Adaptation

Couche service

- Expose l'implémentation de l'ensemble des **cas d'utilisation** *(use case)*

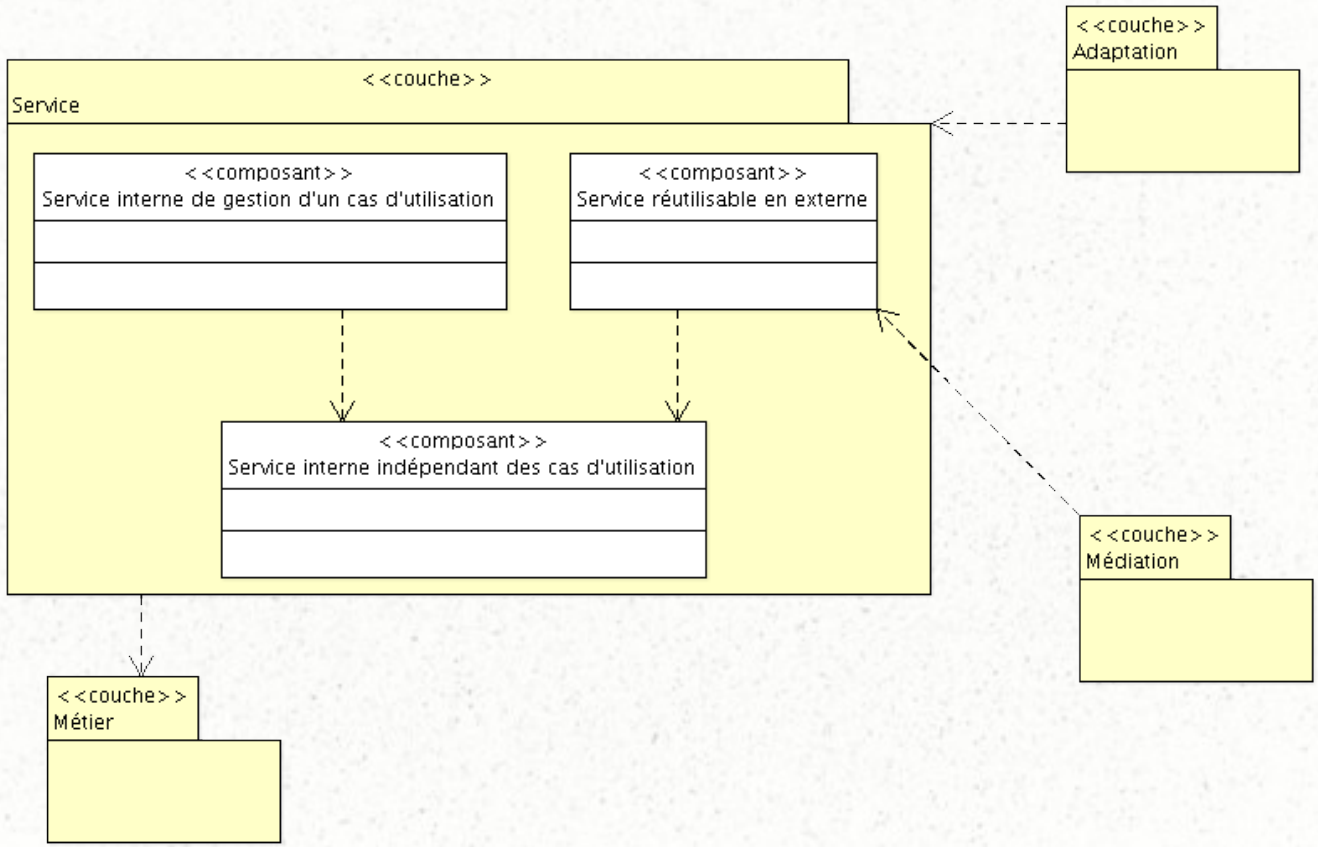
  De manière interne (services internes)

  De manière externe (services externes)

- Service orienté ***use case***

  Pour chaque cas expose une méthode

  - Implémentation de l'interaction entre l'acteur et le système



Composants de la couche service

Services internes

- Services qui implémentent des cas d'utilisation qui n'ont pas besoin d'être connus par des systèmes externes :

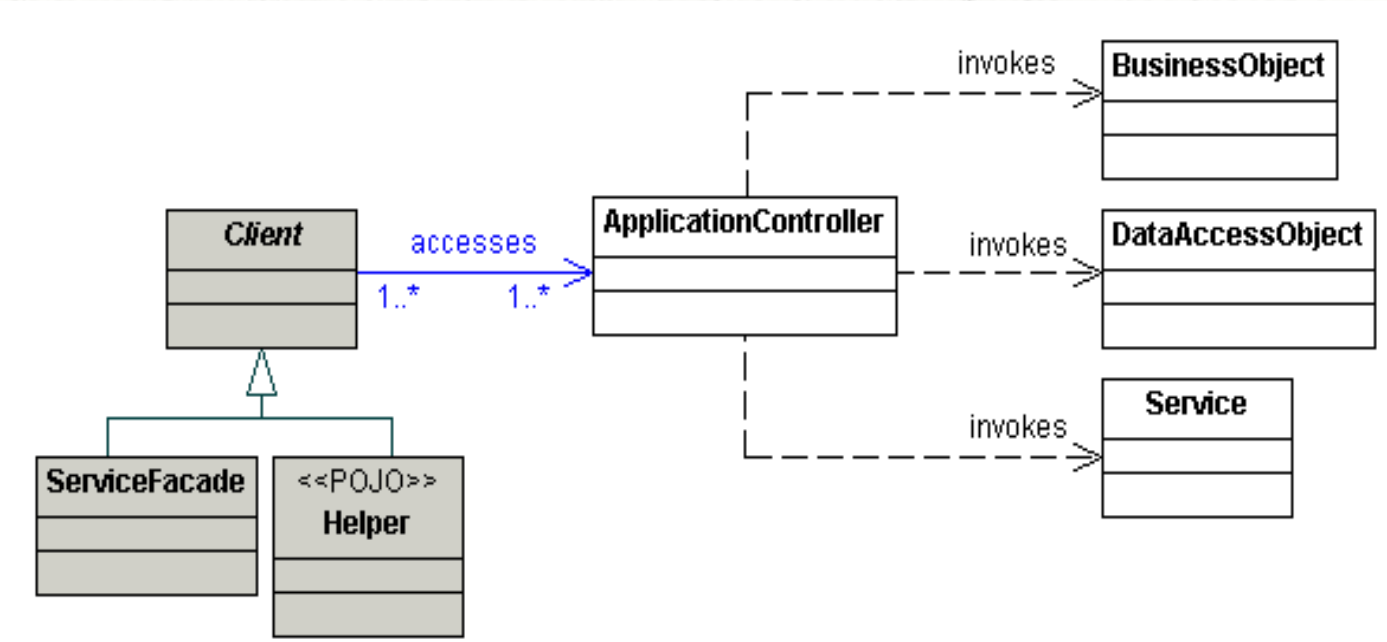
  Service orienté cas d'utilisation

  - Type de service transactionnel

    Chaque activité = 1 transaction dans le SI

  Service transversal

  - Service pour plusieurs cas d'utilisation
  - Patron *ApplicationService*, *finder*



Patron de conception *Application Service*

Services externes

- Services mis à disposition pour le reste du système d'information :
  - Service sans état
  - Service distribué
- Exemple : Web service

Couche métier

- Vue sous forme d'objets métier sur le modèle de stockage des données
- Trois types de composants :

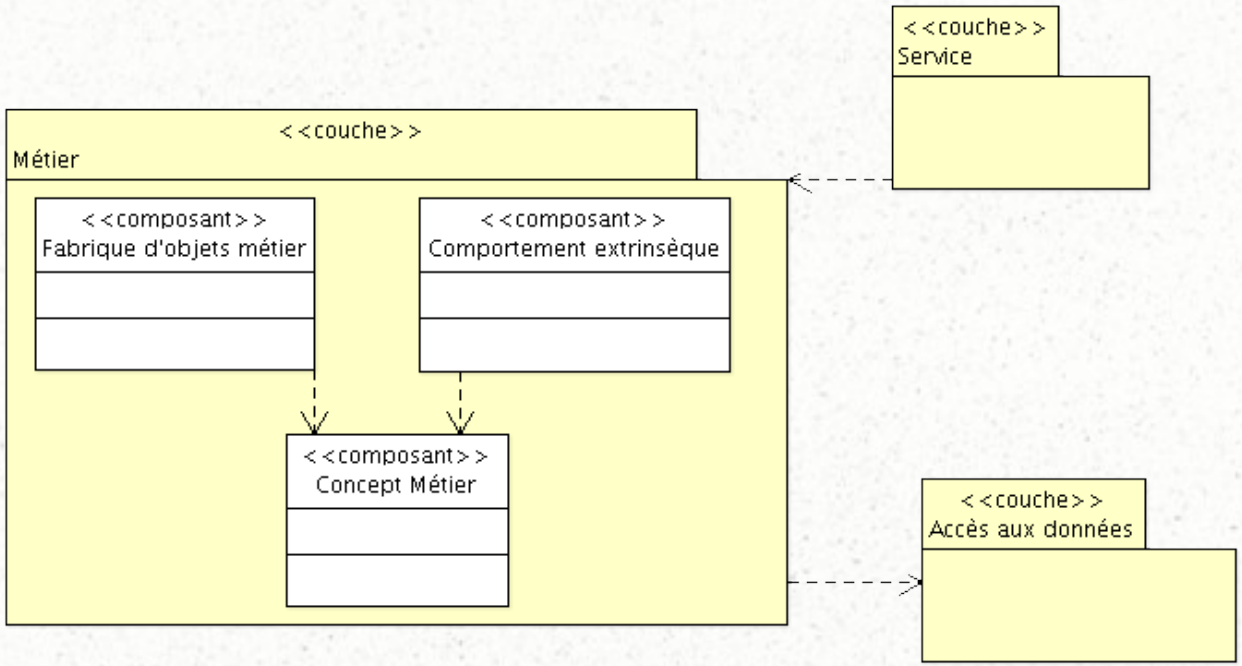
  Objet métier

  Fabrique d'objet métier (CRUD)

  Objet technique

  - Implémentation de règles extrinsèques aux objets métier

    manipulation, intégrité, etc.



Composants de la couche métier

Couche d'accès aux données

- Encapsule l'accès au support de stockage des objets métier

  Transparence de la localisation

- Utilisée par les fabriques d'objets métier
- Solutions

  Patron **DAO** *(Data Access Object)*

  Cadriciel *(Framework de persistance)*

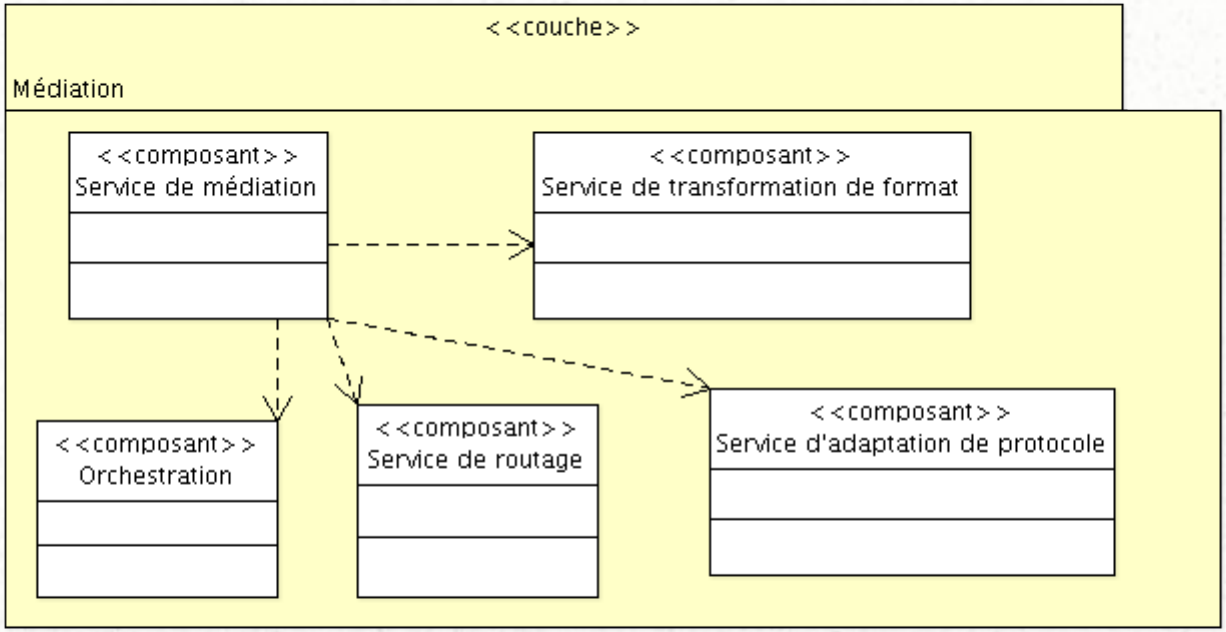
Couche de médiation

- Assure le découplage entre les différentes applications du SI

  Au niveau couche service

  - Routage, transformation, adaptation de protocoles et d'orchestration

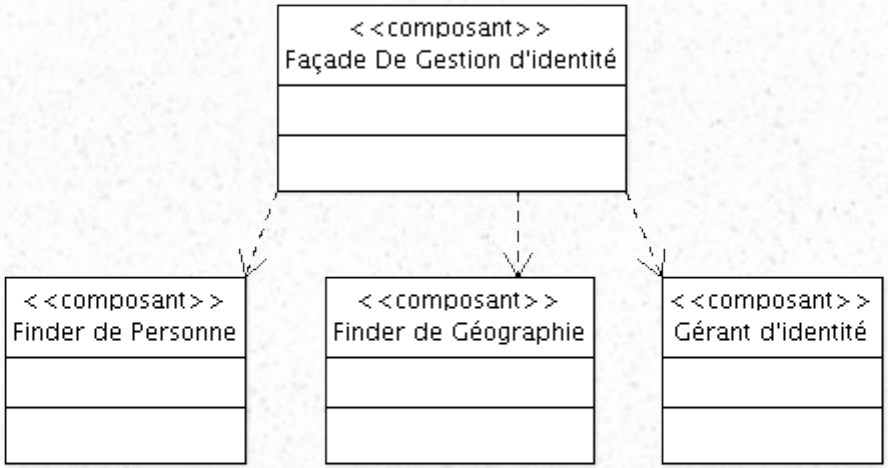
- Couche externe aux applications
- Unique point de passage depuis ou vers l'extérieur d'une application



Composants de la couche de médiation

Notion de façade

- Chaque cas d'utilisation est implémenté par une seule façade.
- Elle masque le nombre et la nature des services utilisés dans la couche service pour implémenter le cas d'utilisation.



Exemple de façade sur trois composants de la couche service

Notion d'agrégat

- La complexité du modèle métier impose :

  De créer des objets dédiés à l'affichage

- Afin de masquer cette complexité aux développeurs d'IHM
- Exemple :

  Les modèles de certains composants Swing (TableModel, TreeModel, …)

  DTO (Data Transfer Object)

Paquetage

- A chaque implémentation de la couche présentation est associée la même couche adaptation
  - Quelques limitations à la réutilisation
  - Mais maximisation de la réutilisation

Couche service

- Basée sur le patron **façade**
- Composants
    - Exposent des interfaces
    - Orientés cas d'utilisation
    - Indépendants des cas d'utilisation
    - Réutilisables
    - Événementiels
    - Transactionnels
    - Sécurisés

Couche métier

- Factorisation des concepts métier
- Abstraction vis à vis du stockage
- Cycle de vie des objets métier
  - Pris en charge par des fabriques
- Cadriciels de persistance
  - EJB3, JPA (JAVA EE, JAVA SE)
  - ObjectSpaces (.Net)

Couche d'accès aux données

- Couche optionnelle
- Objets DAO (Data Access Object)

  Un DAO = un objet métier

  Délégué de la fabrique

  Communication

  - Par DTA (Data Transfer Object)
  - Par Value Object

Couche médiation

- Support de l'urbanisation
  - Quartiers, blocs
- Assure l'indépendance entre applications
  - Producteurs, consommateurs de services
- Fonctions des composants
  - Adaptation des formats entre applications
  - Exposition d'un service sous un protocole différent de son implémentation
  - Orchestration, bus, connectivité

Urbanisation : définition

- L'urbanisation informatique définit

  l'organisation d'un SI à l'image d'une ville

- Permet à un SI d'accompagner la stratégie de l'entreprise

  meilleur rapport coûts/qualité/délais

- Elle permet d'améliorer la réactivité et de n'investir que dans les produits et services générateurs de valeur ajoutée

  Retour sur investissement

Urbanisation : découpage

- 3 niveaux

  Zone, quartier, ilot

- Zone

  Différents systèmes du SI

- Quartier

  Sous-système du SI

- Ilot

  Plus petite entité du SI

Urbanisation : vue stratégique

- Vue constituée
  - Des objectifs de l'entreprise
  - Des objectifs du SI à urbaniser
  - De la mise en correspondance de ces deux objectfs
    - Alignement stratégique sur le métier
- Par l'identification des axes d'analyse
  - Cartographie des connaissances
  - Cartographie du SI

Urbanisation : approche

- Approche par les processus

  Identification des processus

  - Flux d'information
  - Périmètre des processus (bornage)

    Conditions de démarrage et d'arrêt
  - Interactions potentielles
  - Outils

Urbanisation : méthode

- Analyse

  Modélisation des processus

  - BPM

    Business Process Management

    Business Process Modeling

  - Langage BPMN (OMG)

    Business Process Modeling Notation

Urbanisation : méthode

- Développement

  Génération des programmes

  - BPEL

    Business Process Execution Language

    Format XML

    Issu du langage WSLF

    - Web Service Flow Language