

Architecture N-tiers et développement web

Joël Heinis

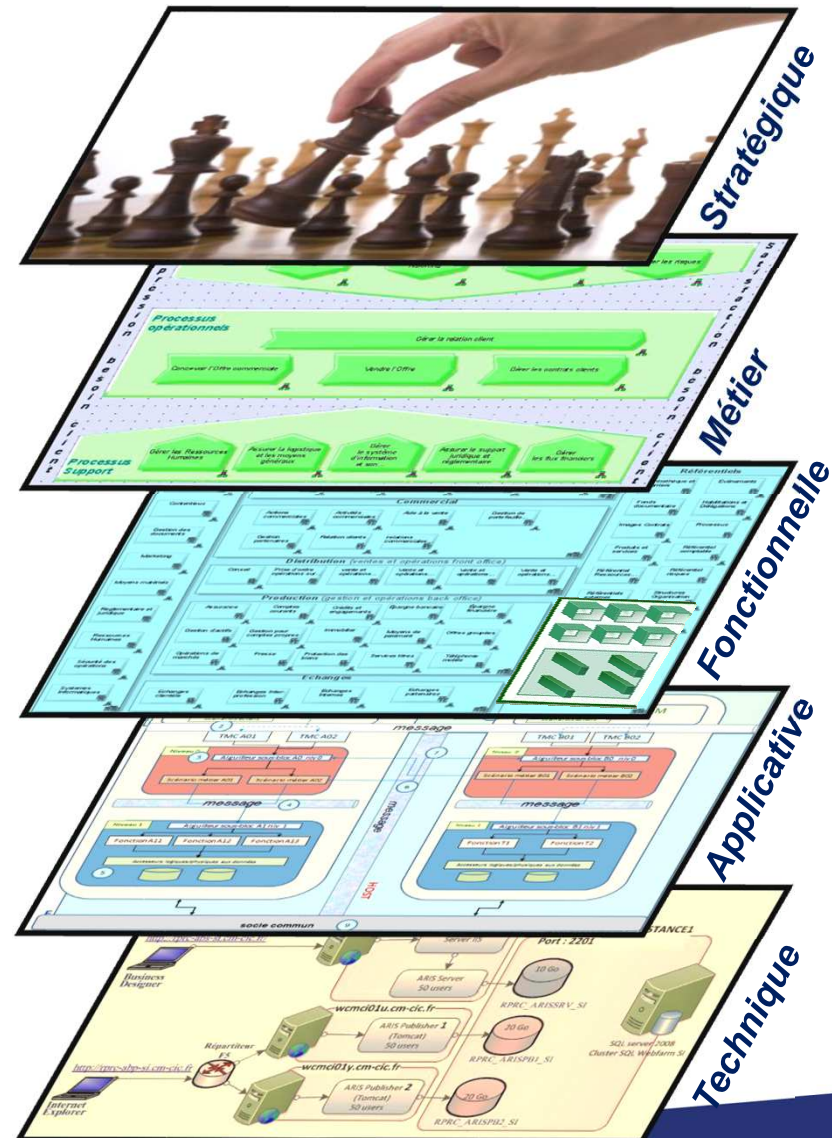
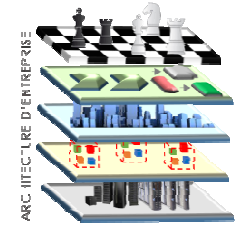
www.uha.fr



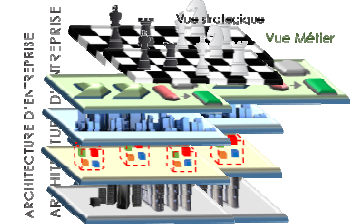
Chapitre

La structuration de l'Architecture d'Entreprise (AE)

Les cinq dimensions de l'AE



La dimension stratégique



Rôle et objectifs :

- Cette dimension représente la vision des stratégies métier et informatique et décrit :
 - Les orientations stratégiques,
 - les objectifs,
 - les ressources
 - l'organisation à mettre en place.

On y trouve les notions de ...

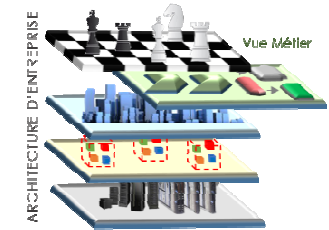
- Stratégie Métier (Augmenter ses parts de marchés, se développer à l'international, développer de nouveaux produits ...)
- Stratégie Informatique (Un système informatique unique, banque, assurances et services; Un réseau IP haut-débit données-voix prêt à supporter la visio-conférence ...)



Je suis concerné par ...

- Dans le cadre de mes activités quotidiennes, au travers des projets , évolutions, contrats de service, respect des indicateurs SMQ , je contribue à la réalisation des objectifs de l'entreprise

La dimension métier

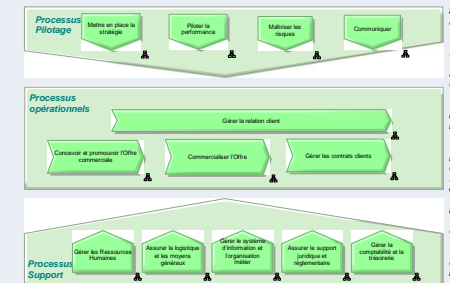
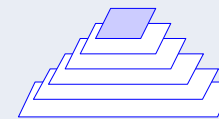


Rôle et objectifs :

- Décrire et comprendre le métier.
- Analyser, optimiser et mettre sous contrôle les processus.

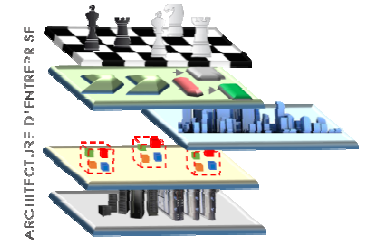
On y trouve les notions de ...

- Processus de l'ensemble d'une entreprise décrit avec une démarche d'analyse (vision descendante, encore appelée TOP – DOWN)
- Les processus sont décrits sur plusieurs niveaux de précisions
- Risques
- Contrôles ...



Je suis concerné par ...

- Les processus du métier que j'outille mais également les processus dans lesquels je suis moi-même acteur (processus de développement, de livraison, de mise sous contrôle, ...)



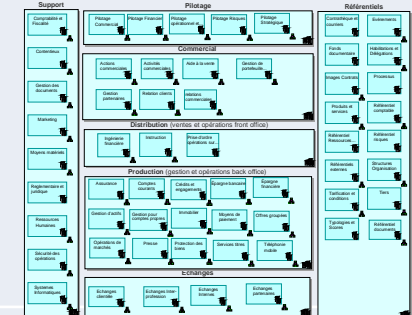
La dimension fonctionnelle

Rôle et objectifs :

- Trier et organiser les informations et les fonctions des systèmes d'information pour :
 - Avoir un catalogue des fonctions à présenter à d'éventuels partenaires
 - Faire le lien entre les processus et les applications en des termes métier
 - Nous aider à mutualiser et réutiliser les fonctions

On y trouve les notions de ...

- Zone fonctionnelle, quartier fonctionnel, îlot, bloc fonctionnel et fonction SI
- Modèle de Référence des Objets Métier
- Spécifications fonctionnelles



Je suis concerné par ...

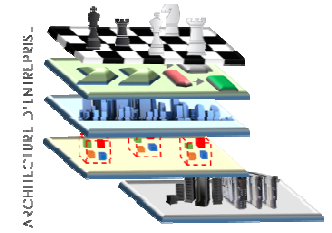
- Le référentiel des objets métiers lors de la conception des objets de gestion de mon application.
- Par la cartographie fonctionnelle lors de la définition des fonctions du SI, leur réutilisation et la cohérence de leur implémentation.

La dimension applicative



Rôle et objectifs :	<ul style="list-style-type: none">• Décomposer le système informatique en composants applicatifs qui outillent les éléments de l'architecture fonctionnelle.• Décrire et structurer les données : Comment sont-elles utilisées, organisées, stockées et échangées par les composants applicatifs.
On y trouve les notions de ...	<ul style="list-style-type: none">• 2 vues :<ul style="list-style-type: none">○ L'architecture applicative, qui rend compte de la décomposition du système informatique en ensembles, domaines, blocs, applications, composants et programmes<ul style="list-style-type: none">○ La façon de je vais développer, les patterns utilisés○ L'architecture des données, qui décrit et structure les données : modèles, bases, ...
Je suis concerné par ...	<ul style="list-style-type: none">• La création de mes applications, composants, services, bases de données et messages.

La dimension technique



Rôle et objectifs :	<ul style="list-style-type: none">• Décrire les composants logiques et physiques qui supportent les applications : machines, réseaux et composants réseau ...
On y trouve les notions de ...	<ul style="list-style-type: none">• Environnements, partitions, WebFarm, protocoles, SGBD, ...
Je suis concerné par ...	<ul style="list-style-type: none">• La répartition de mes données et de mes traitements sur les différents environnements.• Le choix des technologies adaptées à mon projet ...

Chapitre

Principes généraux de l'Architecture d'Entreprise

Il existe 6 grands principes en Architecture d'Entreprise

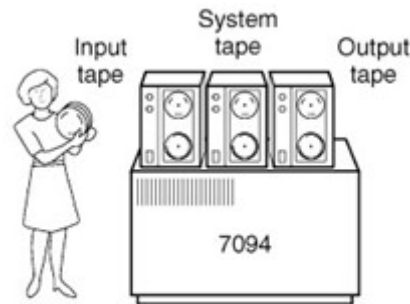
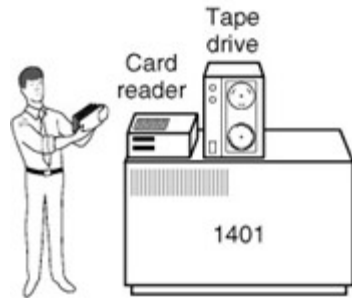
- La cohérence
- La modularité
- Le découplage
- L'interopérabilité
- La résilience
- La traçabilité et le contrôle

Chapitre

Les problématiques d'architecture

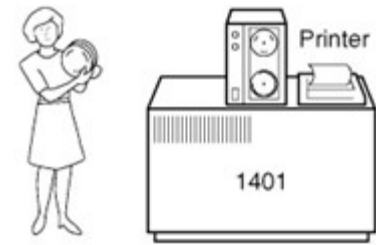
Les premières machines / architecture (1960 – 1970)

- Systèmes dits centraux



(c)

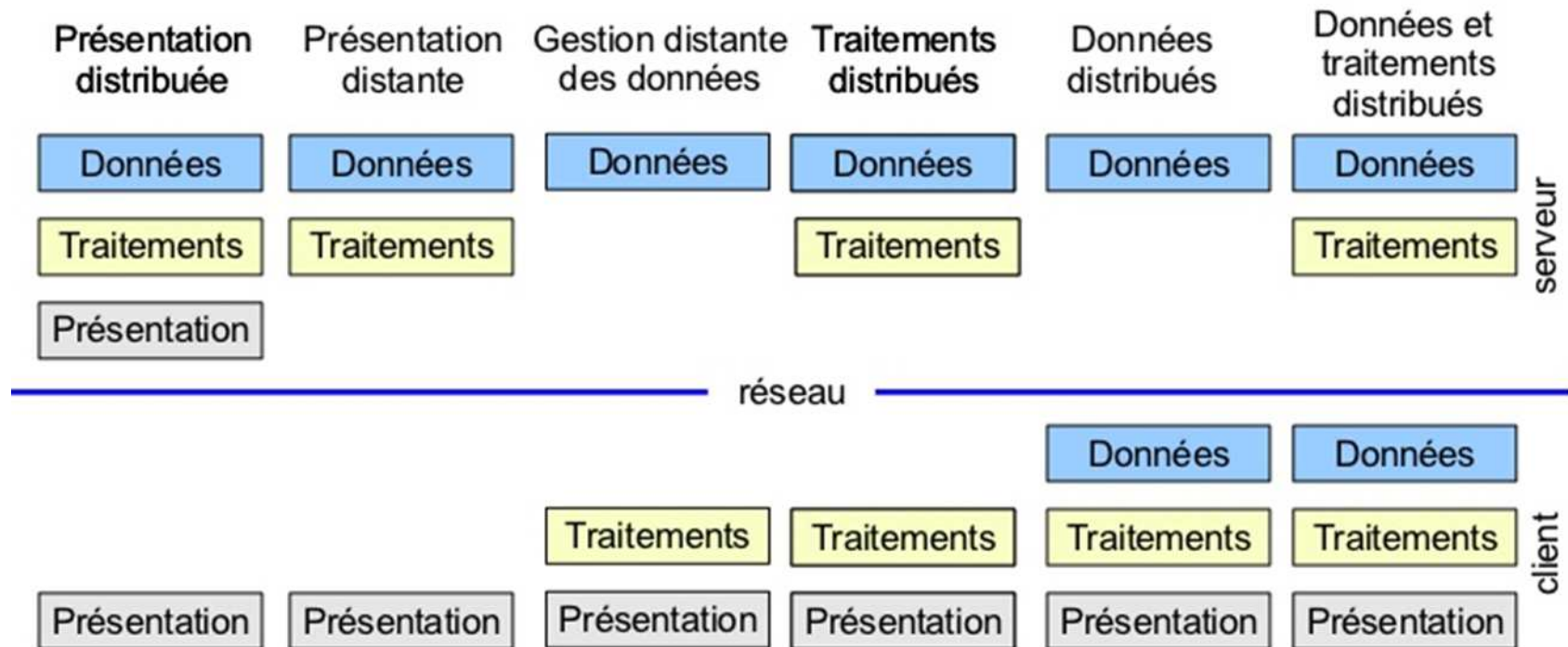
(d)



- Traitements par lots
 - saisie loin du lieu de production
 - ordinateur central : seul lieu de traitement et de stockage
- Hyper centralisation

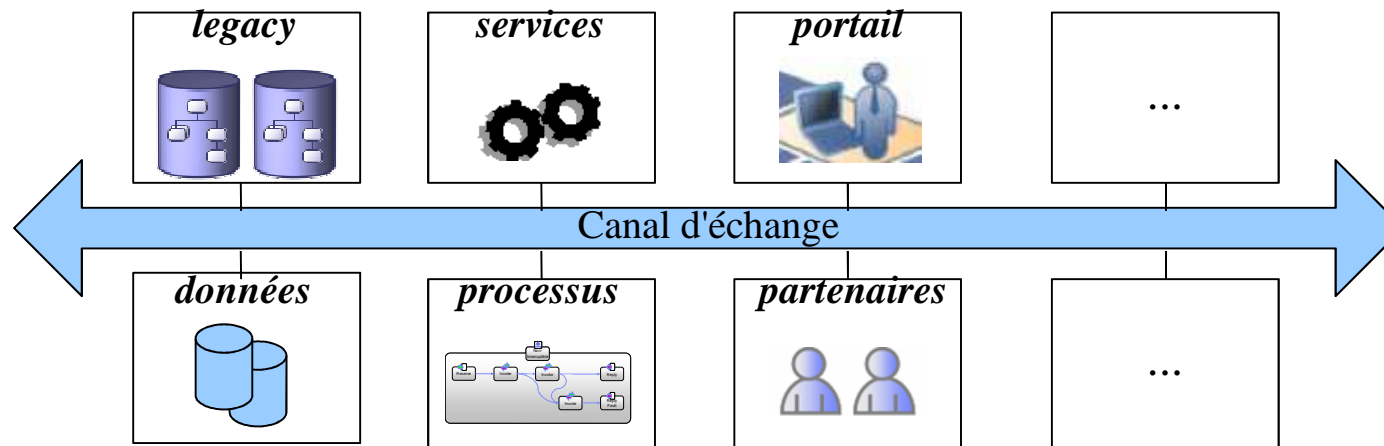
Client-Serveur

- Les évolutions matérielles permettent d'exploiter la puissance dans tous les composants du SI
 - GARTNER présente les différentes approches C/S



SOA - Une approche plus urbanisée

- L'urbanisation informatique définit l'organisation d'un SI
 - découper le SI en modules autonomes (zone, quartier, îlot, bloc)
 - localiser les zones d'échange d'informations (routes, ponts, tunels) qui permettent de découpler les différents modules



- Objectif : faire évoluer le SI au même rythme que la stratégie et l'organisation des métiers de l'entreprise

SOA - Principes fondamentaux

- **Il n'existe pas une recette pour garantir le succès de la mise en place d'une SOA mais des principes à respecter :**
 - Discussion entre métier et IT
 - Utilisation des cas d'utilisation métier
 - Utilisation de standards
 - Pas de remise en cause de l'existant lors d'évolutions technologiques
 - Découplage entre fournisseur et consommateur de services
 - Indépendance des ressources vis à vis de ceux qui les utilisent

Les Micro-services: Une réponses aux besoins de l'industrie (Time to market, Agilité)

- **Domain-Driven Design**
- **Intégration et Déploiement continue**
- **Virtualisation à la demande**
- **Automatisation des infrastructures**
- **Equipes de développement polyvalentes et autonomes**
- **Mise à l'échelle des systèmes (scaling)**
- **...**

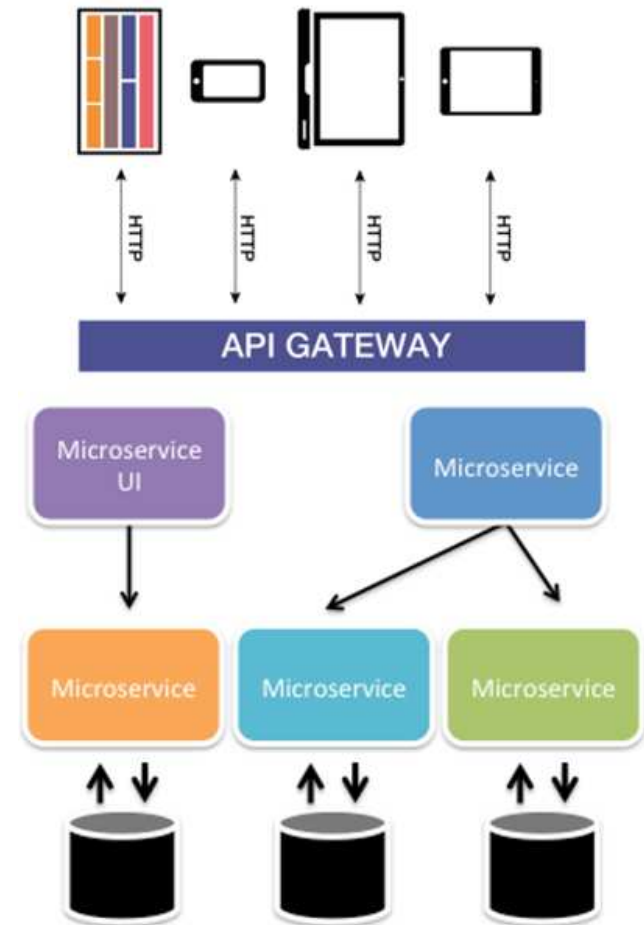
.... Et les microservices

www.uha.fr



Micro-Service: Définition

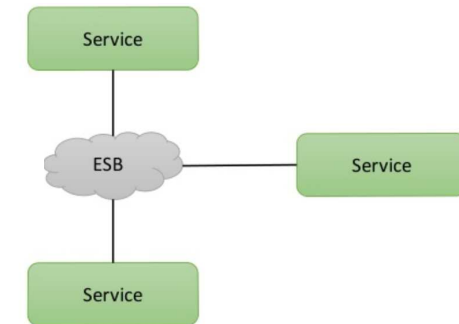
- Un microservice est un service concentré sur une seule fonctionnalité (Single Responsibility Principle).
- La notion de microservice est apparue en 2011
- L'interface exposée doit être la plus simple possible.
- Même s'il peut exister des interdépendances entre les différents microservices, chacun doit être autonome et déployé séparément.
- L'architecture microservice (MSA) est une évolution de la SOA et reste complètement compatible avec celle-ci



MSA/SOA - Ce qui évolue par rapport à une démarche SOA classique

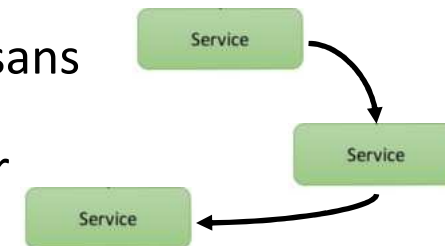
- **SOA**

- Basée sur un Bus de service ESB, organe central de l'architecture
- MOM intégré pour l'asynchronisme
- Les applications sont "enrobées" de services web possédant un contrat fort
- L'organisation des équipes restent inchangés (principe SOA non appliqué)
- Le routage intelligent (L'ESB va déduire le destinataire)



- **MSA**

- Basée sur des canaux de communications pauvres sans médiation
- Associé ou non à un bus d'événement (broker) pour l'asynchronisme
- Les applications sont découpés en service fonctionnel
- L'organisation des équipes suit le découpage fonctionnel



Caractéristique d'un microservice (vue conception)

Domain
Driven
Design

Explicitly
Published
Interface

Single
Responsibility
Principle

Lightweight
Communication

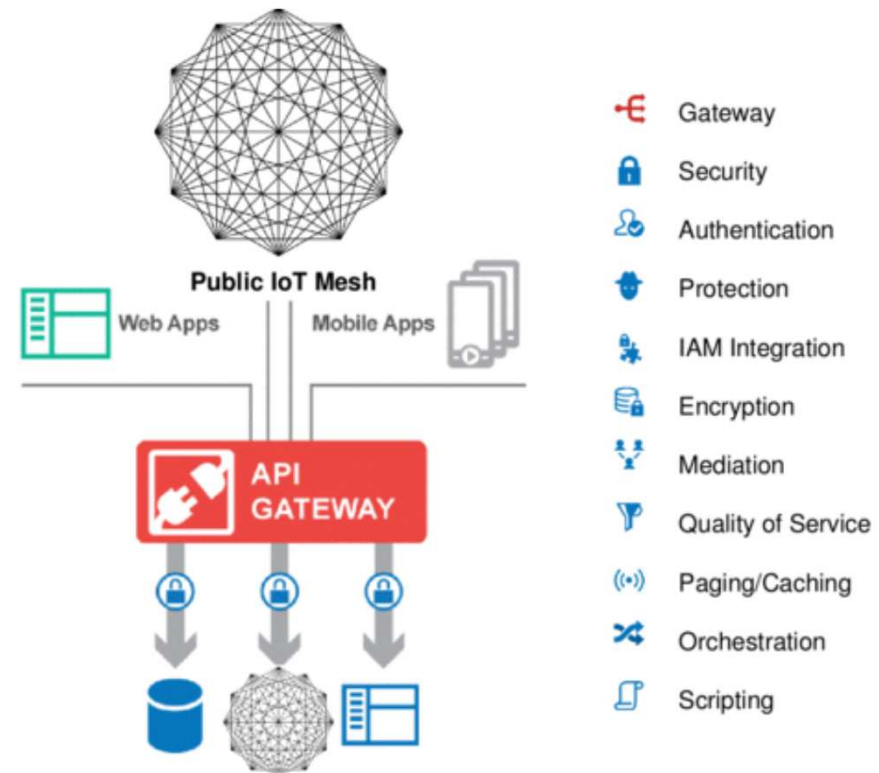
Independent
DURS

Caractéristique d'un microservice (vue production)

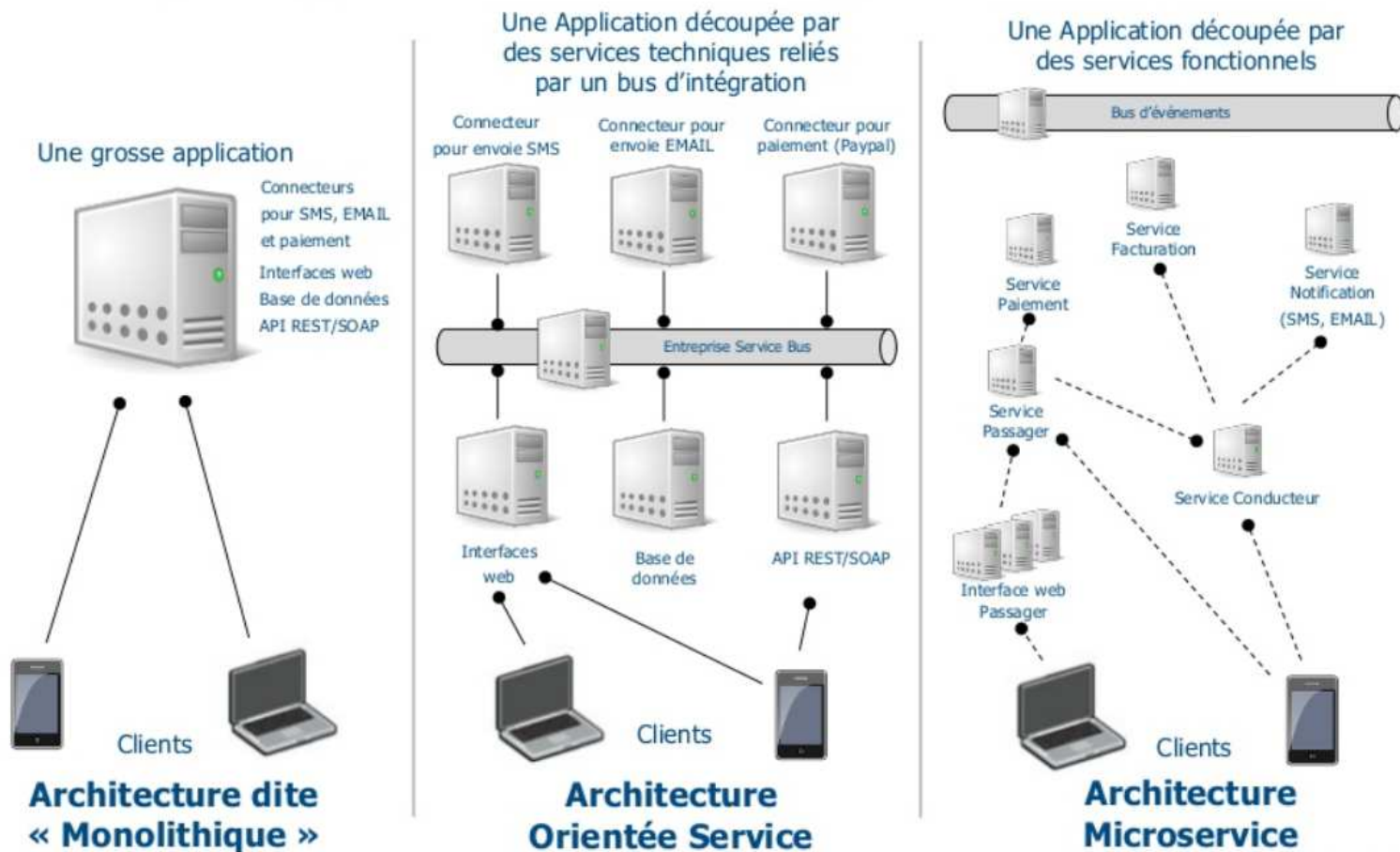


Services MESH: Se concentrer sur le métier

- **Service MESH (ou side-car)**
est un proxy qui va
permettre de gérer un
ensemble de contraintes
techniques liées au micro
service
 - Sécurité
 - Déploiement
 - Cache
 - ...



Synthèse de l'évolution des architectures techniques



Sensibilisation architecture - 5/12/2019

Chapitre

Les langages, normes, outils,
applications, ... du web

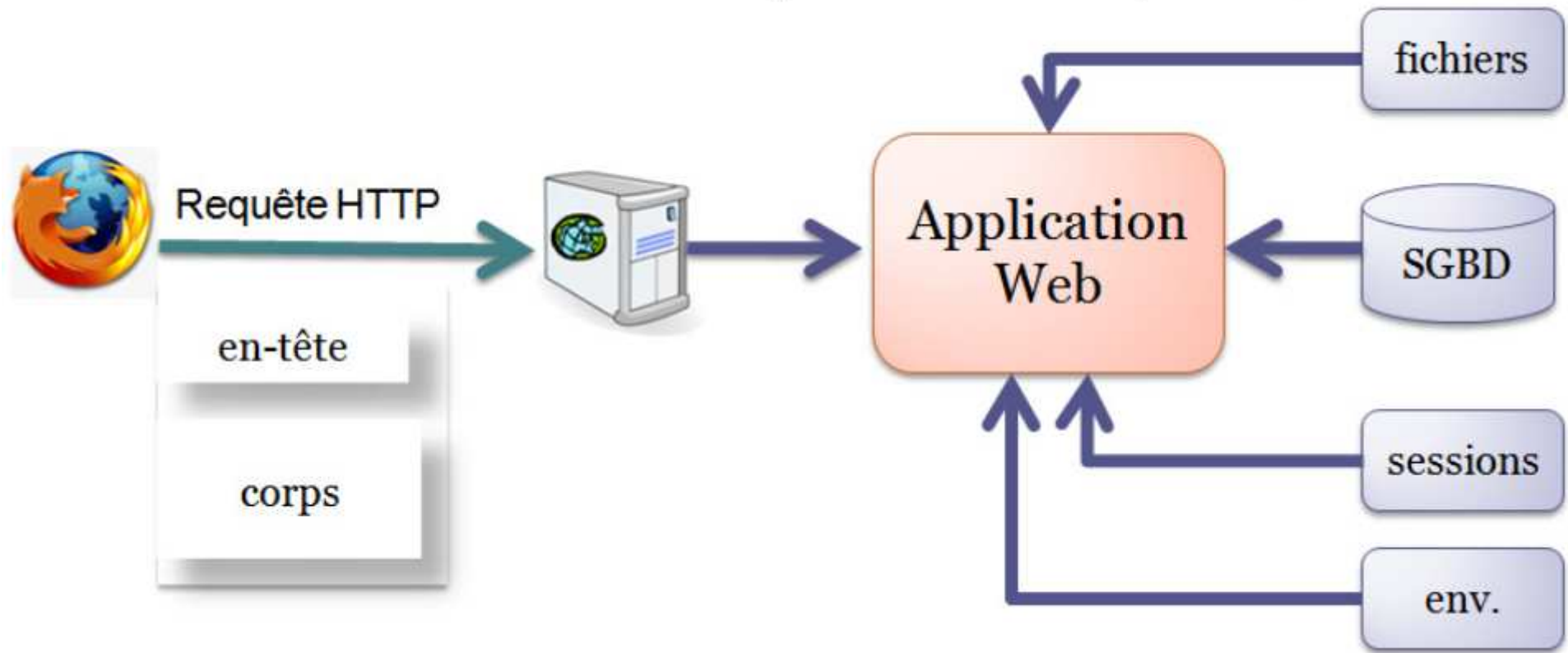
De nombreux acronymes et termes barbares



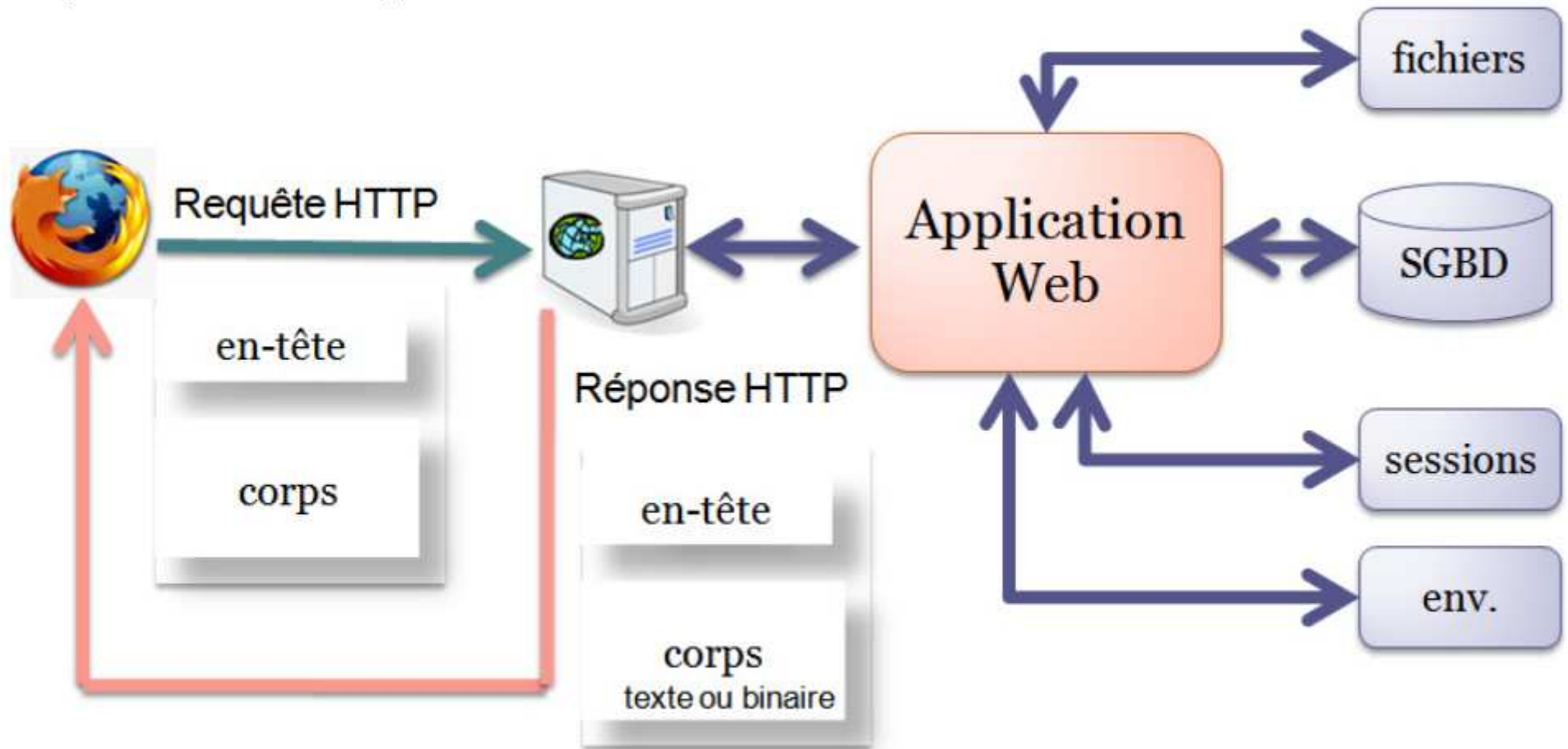
Chapitre

Web?

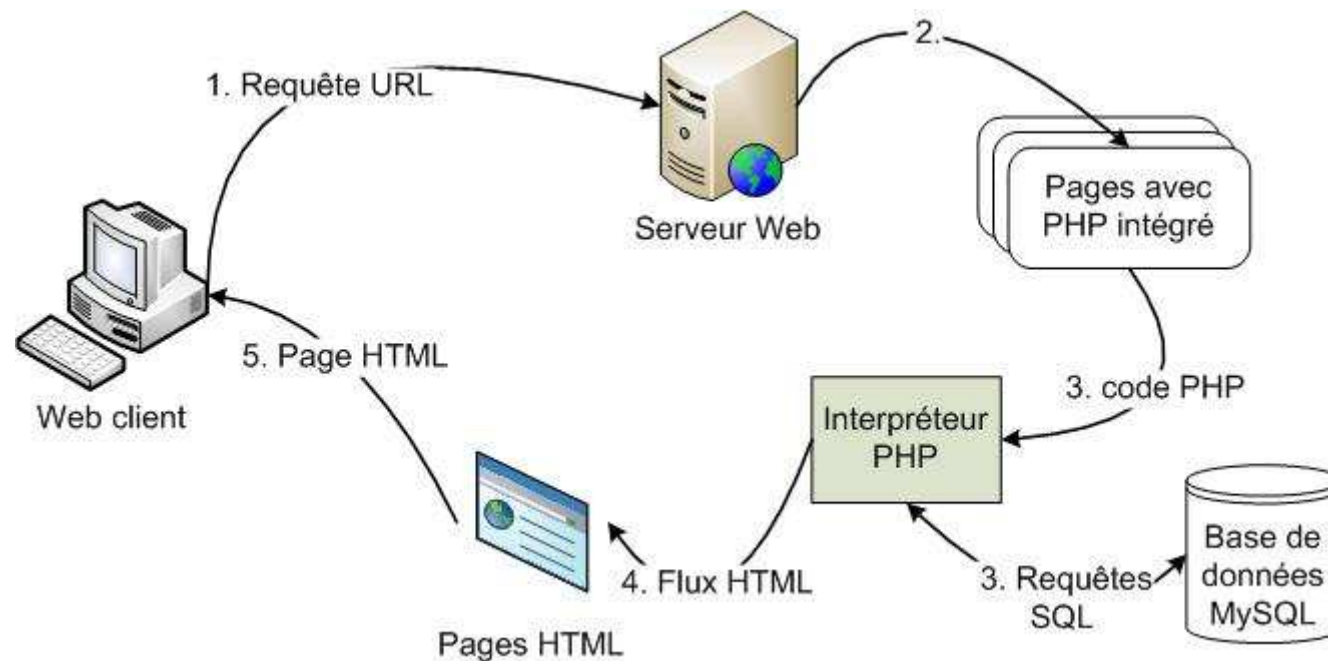
Appel du serveur par le client



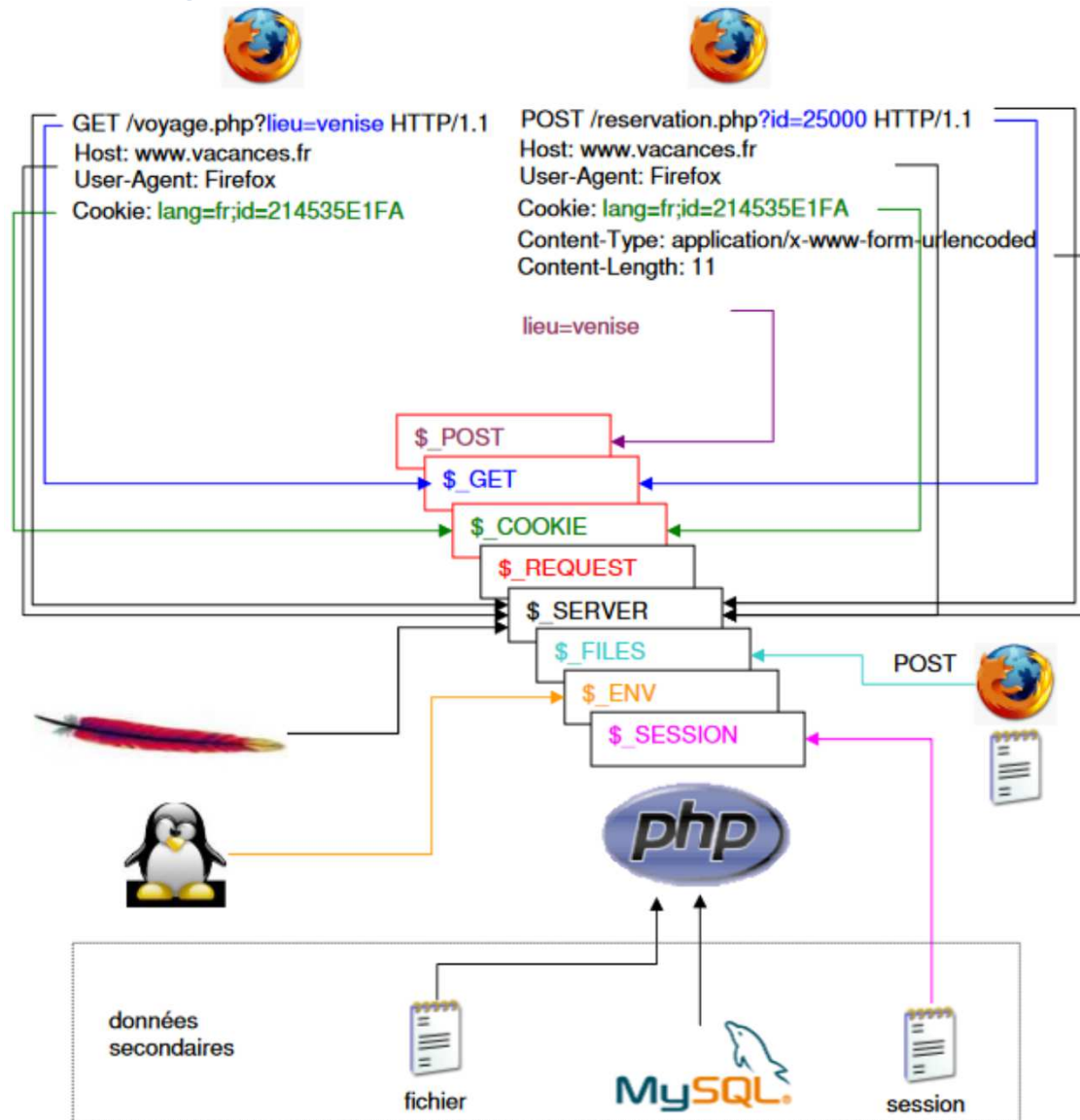
Réponse du serveur au client



Exemple architecture Apache / MySQL / PHP



Une requête coté PHP cela donne ...



Chapitre

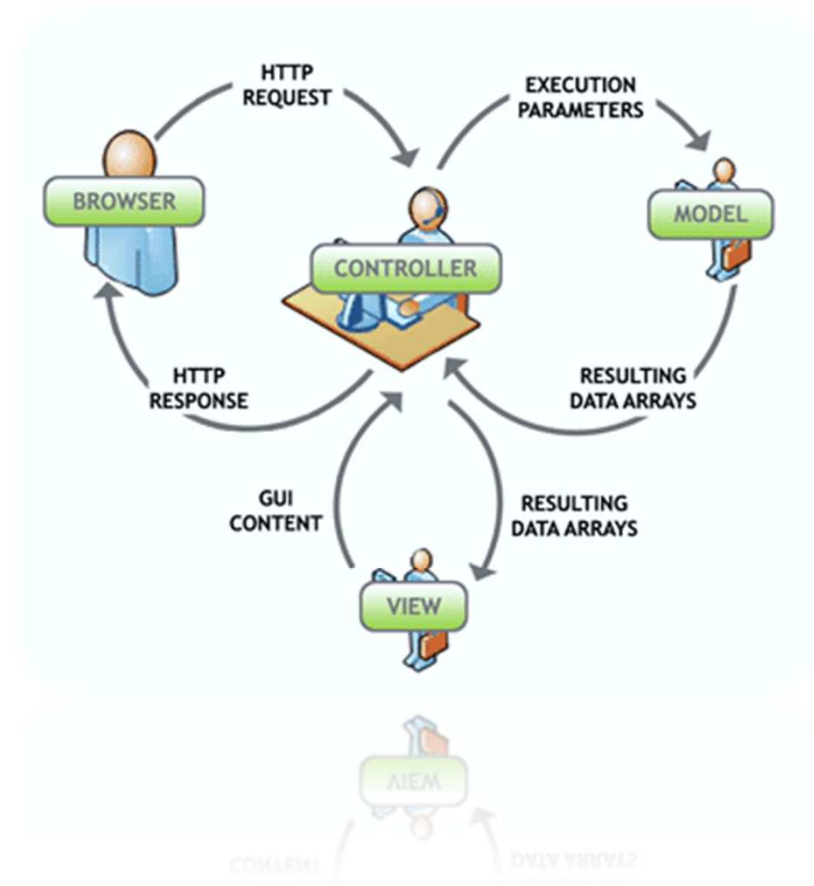
Les patron de conception

Principaux design pattern

- Singleton
- Factory
- Model View Controller
- Injection de dépendances
- Fluent
- Facade
- Les interfaces
- Les traits
- Adapter
- Decorator
- Observateur
- Middleware
- Event Manager

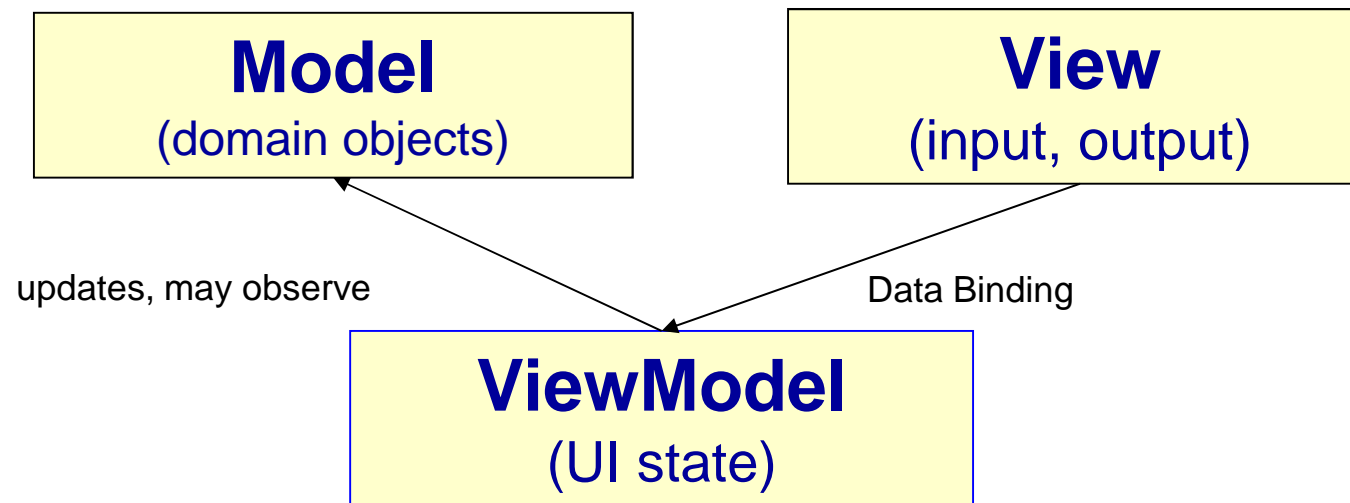
MVC – Model View Controller

- **Model** = interaction avec la base de données
- **Vue** = rendu au client
 - HTML, JSON
- **Controller** = gestionnaire de l'application
- ... et Dispatcher, Router



MVVM - Model-View-ViewModel

- Version Microsoft du MVC



`View.DataContext = ViewModel;`

Chapitre

Les Web Services

Qu'est ce qu'un Web Service ?

- **Définition [W3C]**

- Un Web service est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML.
- Sa définition peut être découverte par d'autres systèmes logiciels.
- Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.

- **Exposition**

- Langage WSDL utilisé pour décrire le service
- Similaire à IDL mais basé sur XML

- **Activation**

- Protocole Web au-dessus de HTTP (RPC XML, SOAP)
- Autres protocoles possibles ...

Les évolutions historiques

- **XML-RPC (1998)**
- **SOAP (2001)**
 - Un nouveau Protocole : SOAP
 - Paradigme orienté service : WSDL
 - Découverte automatique des services (dynamacité) : UDDI
- **REST (2000/2010)**
 - Simplifier les choses

SOAP

- **Simple Object Access Protocol**
- **Protocole d'échange de messages (client / serveur)**
- **Basé entièrement sur XML**
- **Standard W3C (Initiative IBM et Microsoft)**
 - Actuellement SOAP 1.2
- **Concepts**
 - Message = Enveloppe (Header + Body)
- **Extensibilité**
 - Porté sur HTTP, SMTP, ...

WSDL

- **Web Services Description Language**
- **Langage de définition de Web Services**
- **Basé entièrement sur XML**
- **Standard W3C (Initiative IBM et Microsoft)**
 - Actuellement WSDL 2.0
- **Définition de l'interface, de l'URL et du port du Web Service.**
- **Utilise le système de typage de XML Schéma**
- **Associé à SOAP**

REST

- **Acronyme de REpresentational State Transfert défini dans la thèse de Roy Fielding en 2000.**
- **REST n'est pas un protocole ou un format, contrairement à SOAP, HTTP ou RCP, mais un style d'architecture inspiré de l'architecture du web fortement basé sur le protocole HTTP**
- **Il n'est pas dépendant uniquement du web et peut utiliser d'autre protocoles que HTTP**

Méthodes (Verbes)

- Une ressource peut subir quatre opérations de bases CRUD correspondant aux quatre principaux types de requêtes HTTP (GET, PUT, POST, DELETE)
- REST s'appuie sur le protocole HTTP pour effectuer ces opérations sur les objets
 - Création, Ajout → POST
 - Lecture → GET
 - Mise à jour → PUT, PATCH (? POST)
 - Suppression → DELETE