

# Les Architectures Orientées Services

SOA

**Introduction Générale**

# Introduction aux Architectures Orientées Services

## Objectifs de la matière

L'architecture orientée service (SOA) est une façon de concevoir un système d'information (SI) d'entreprise, indépendamment des technologies mises en place pour la réaliser.

A l'issu du cours, l'apprenant sera capable de comprendre les SOA, d'identifier clairement les niveaux de granularité de services, de modéliser des services métier, de définir des contrats de service, de décrire des processus métier ,et les décliner en orchestration de services.

**Pré requis** recommandés :

**Unité d'enseignement** : UEM11

**Crédit** : 5

**Coefficient** : 3

**Mode d'évaluation** : Examen, contrôle continu TD et TP.

# Introduction aux Architectures Orientées Services

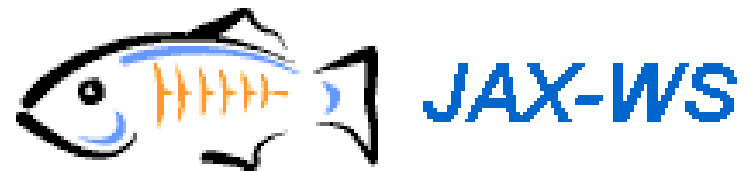
## Contenu de la matière :

- 1) Introduction générale
- 2) Introduction à SOA et Services
- 3) Contrat, message, et registre de service
- 4) Démarche SOA : constitution des services métier unitaires
- 5) Démarche SOA : constitution des services métier composés
- 6) Démarche SOA : processus métier
- 7) Pattern MVC pour SOA
- 8) La boîte à outils Service Web
- 9) Gérer la Sécurité dans les Services Web
- 10) Liens entre les concepts et les outils

# Introduction aux Architectures Orientées Services

## Contenu de la matière :

- TP et TD : Les Service Web SOAP
- L'API **Java** JAX-WS
- Logiciels : Eclipse JEE, Serveur Glassfish, soapUI, etc.



# Plan du cours

1. Evolution des architectures informatiques
2. L'accélération et ses conséquences sur les SI
3. Les limites des réponses proposées : Architecture fonctionnelle, urbanisation, EAI, etc.
4. Vers les architectures SOA
5. Après SOA, les Microservices

**Architectures** orientées **Services**



**?**

Une approche qui consiste à avoir et à exposer des ressources logicielles et matérielles disponibles (et détectables) sur le réseau en tant que services indépendants et ces services doivent atteindre un objectif métier.

## Architectures orientées Services

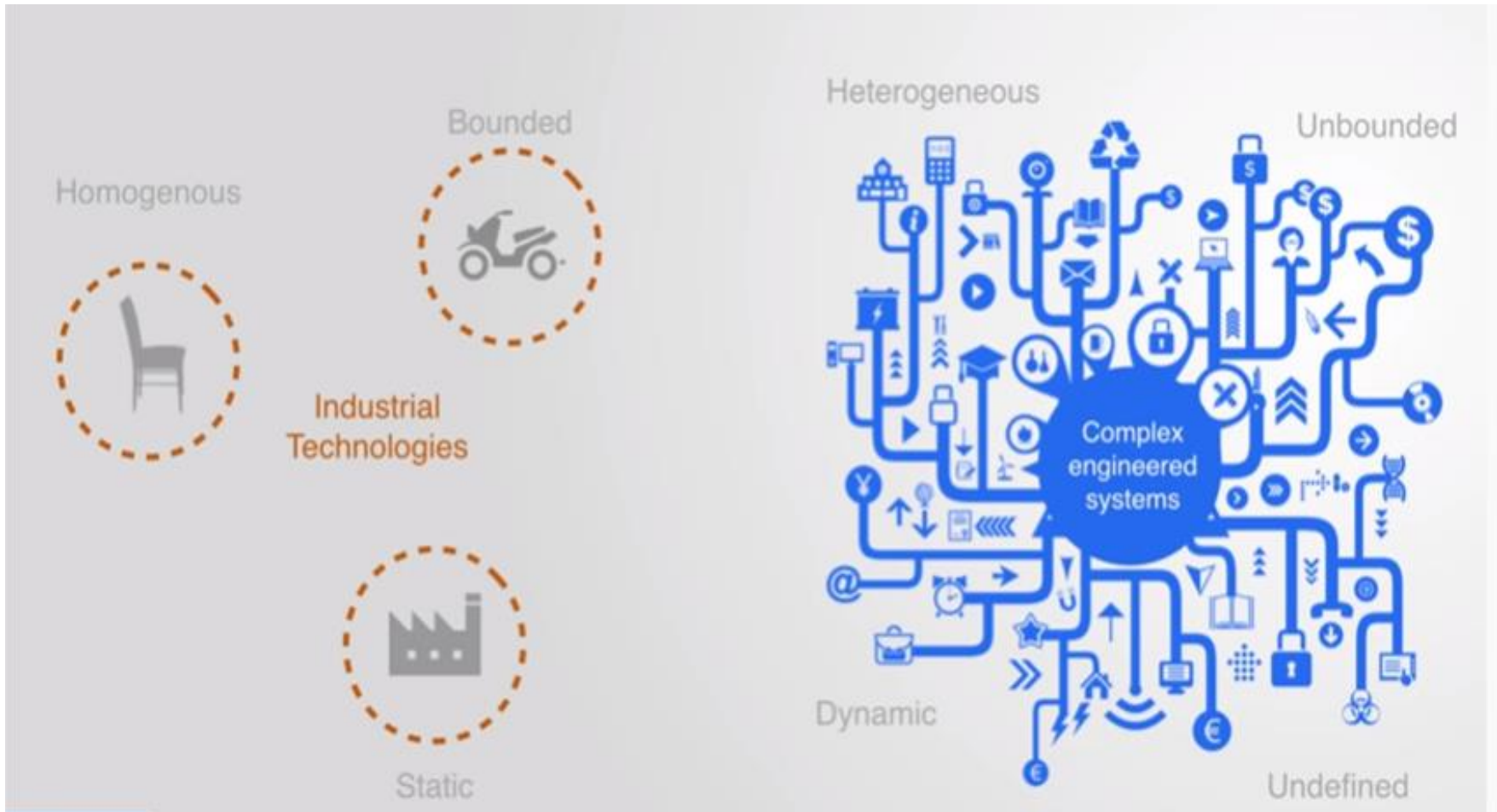


Concevoir/  
Organiser/  
Structurer

Logiciel/  
application/  
bloc  
applicatif/  
**SI**

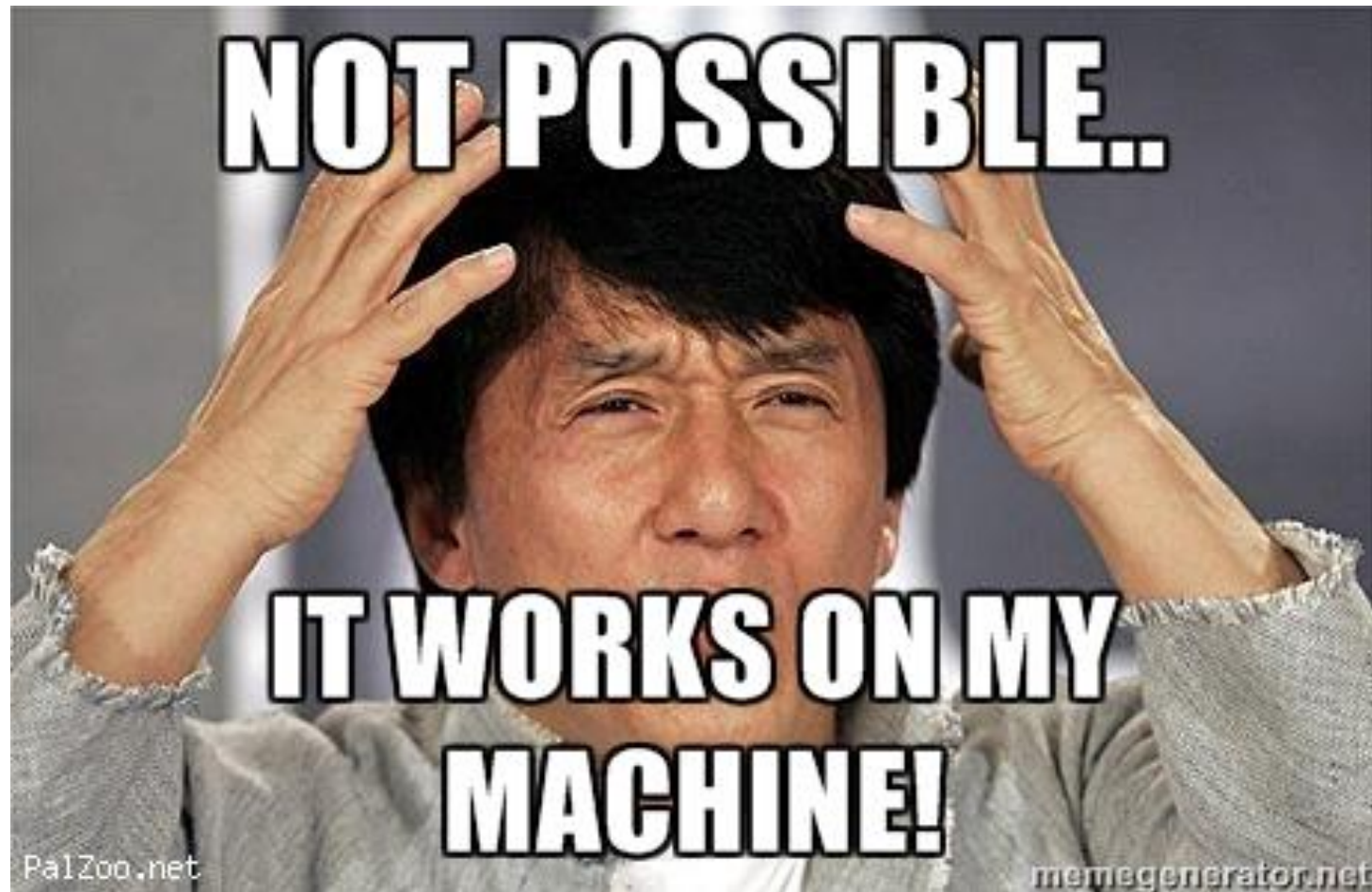
Comme étant  
**Services**

## Contexte : SOA

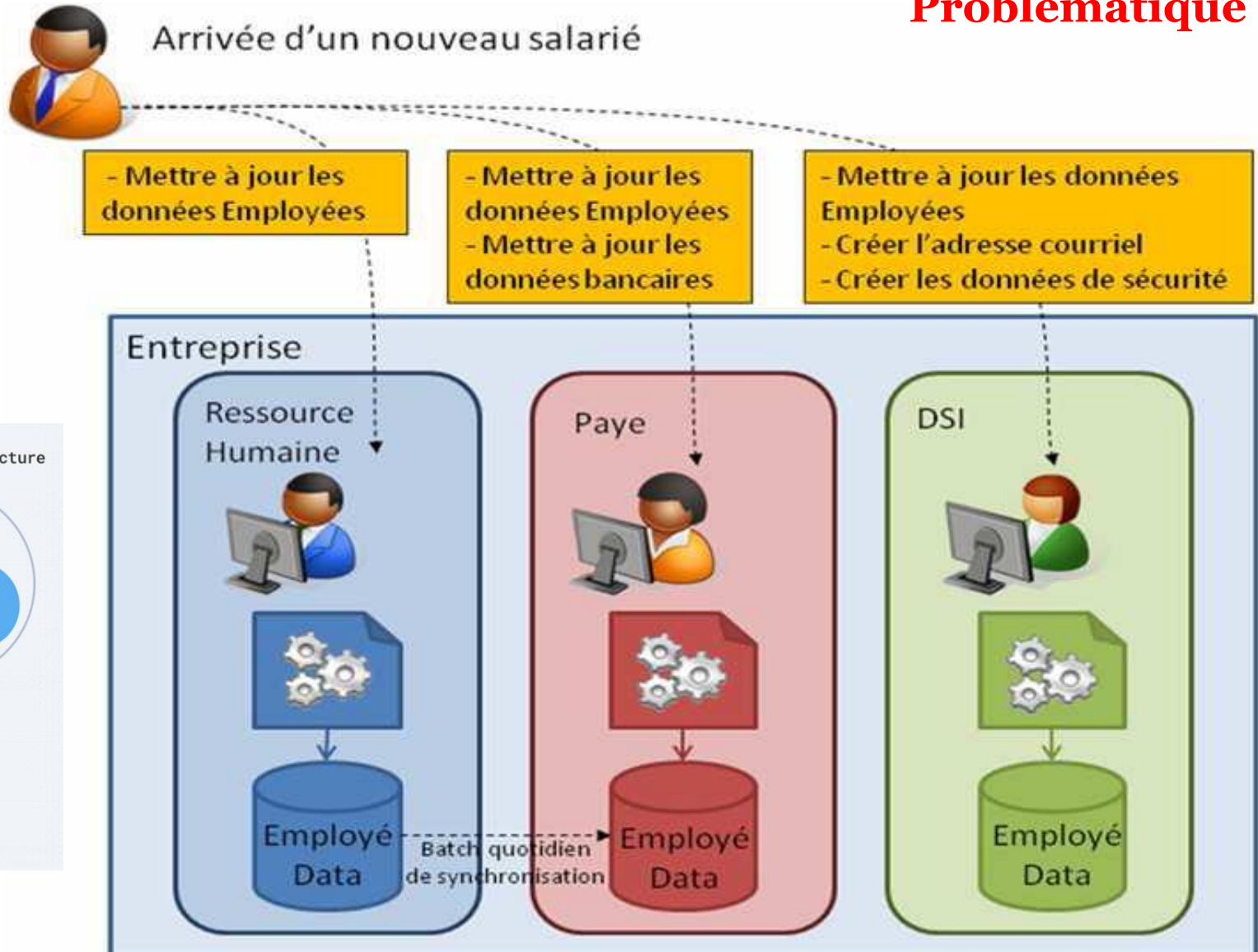




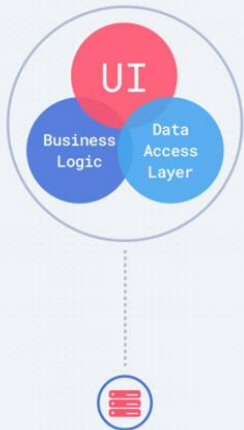
## Problématique



# Problématique



Monolithic Architecture



## Problématique



Monolith

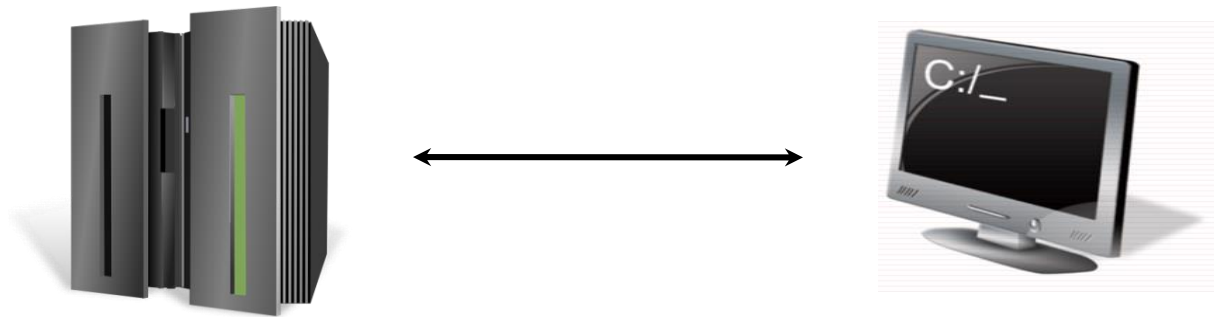


Service  
Oriented

# Evolution des architectures informatiques

## 60's/70's

Mainframe



Data – Logic - Presentation

- ✓ Facilité de mise en œuvre et de gestion
- ✓ Fiabilité – Disponibilité et intégrité des data
  - Scalabilité (Utilisateurs, fonctions, etc.)
  - Coûts d'acquisition et d'exploitation élevés

# Evolution des architectures informatiques

## 80's

Client/serveur



Data



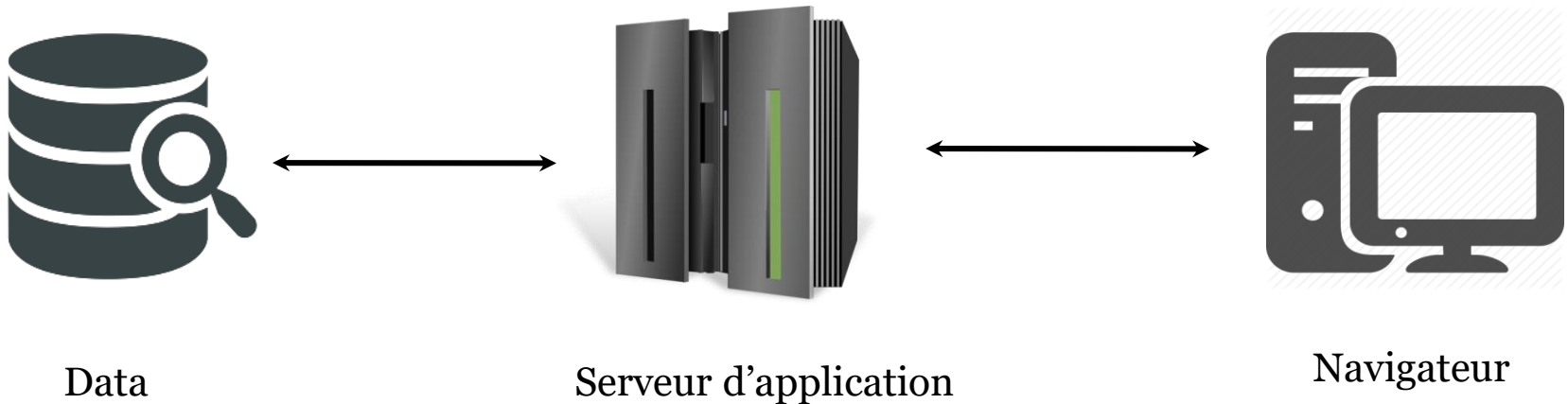
Logic/Validation - Presentation

- ✓ Scalabilité
- ✓ Faibles coûts
  - Maintenance + hétérogénéité
  - Duplication des informations

# Evolution des architectures informatiques

**90's/00's**

3 tiers/Web

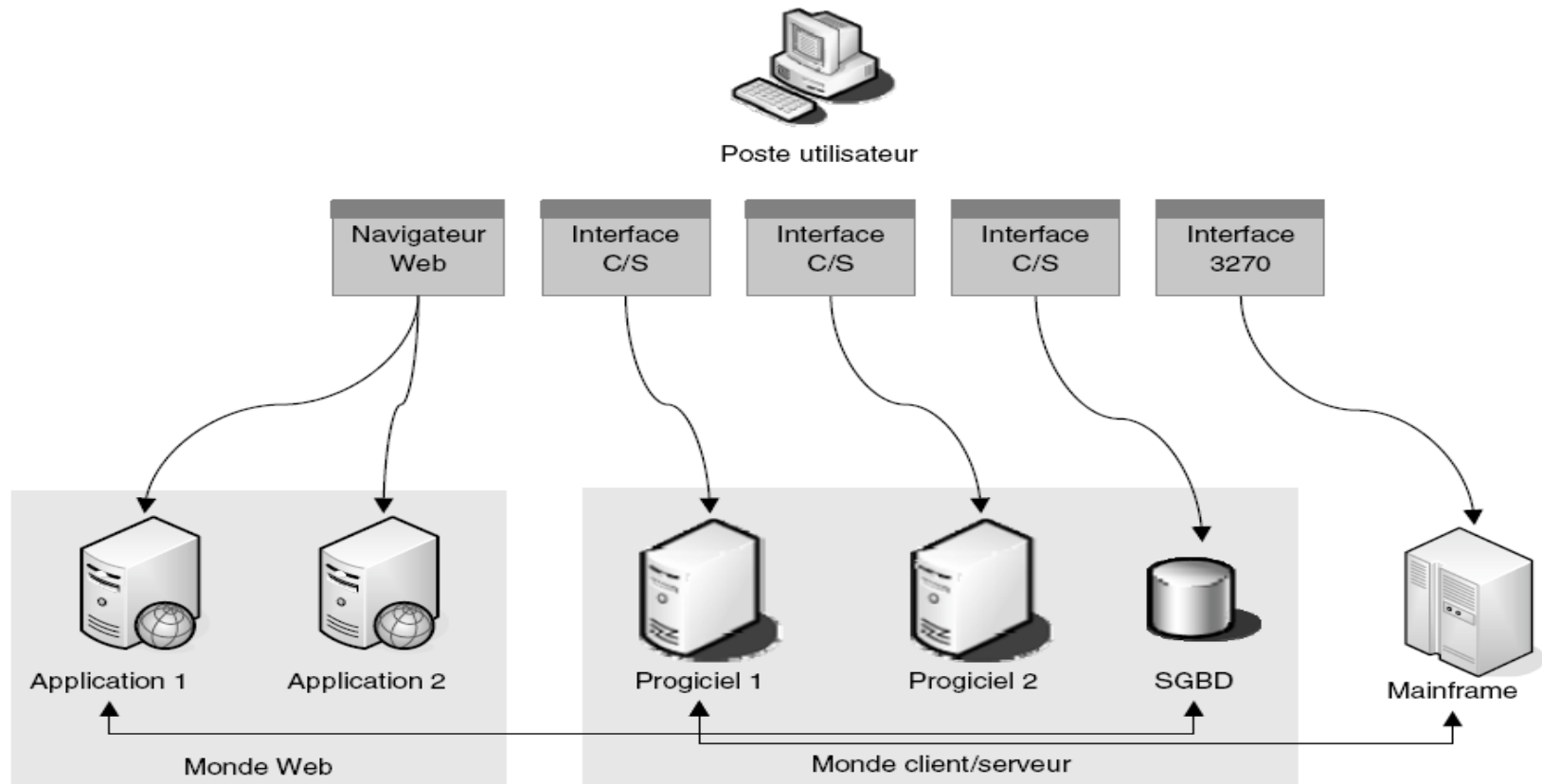


- ✓ Scalabilité + Accès à distance
- ✓ Faibles coûts de gestion
  - Complexité
  - Morcellement du SI

## Problème 1 : Gestion de l'Hétérogénéité et Intégration

**Résultat** : Des SI hétérogènes, redondants, morcelés, et qui ne partagent pas leurs informations.

Cohabitation et intégration de données ou d'applications



# Problème 2:Communication

- Comment déclencher, en réponse à un événement dans un sous-système donné, un traitement dans un autre sous-système qui lui est étranger ?
- Comment assurer la consistance et la propagation des données entre plusieurs sous-systèmes ?



# Problème

## 3: Redondance

Le SI doit pérenniser ses briques existantes et favoriser la **réutilisation**.

# Accélération des marchés et ses conséquences sur le SI

## 1. **Le besoin d'agilité** (= capacité du SI à supporter des évolutions rapides)

- Compétitivité + Mondialisation : être capables de livrer de nouveaux produits de plus en plus vite.
- Fusion + Acquisition : être capable d'absorber rapidement d'autres SI, souvent hétérogènes sur le plan technologique.
- Le SI doit être capable de s'adapter rapidement à une fusion ou à une réorganisation des équipes.
- Développement d'une offre sur la base d'une nouvelle technologie.

# Accélération des marchés et ses conséquences sur le SI

## **2. L'intégration des acteurs externes (et personnalisation)**

- Offrir de plus en plus de portes d'entrées aux consommateurs et sous-traitants.
- Supporter des processus de plus en plus automatisés et faisant intervenir une grande variété d'acteurs.
- Offrir une disponibilité à toute épreuve.

## Bilan de ces évolutions et accélérations

- Les SI doivent **s'aligner** sur les besoins techniques et métiers :

### **1. Technique :**

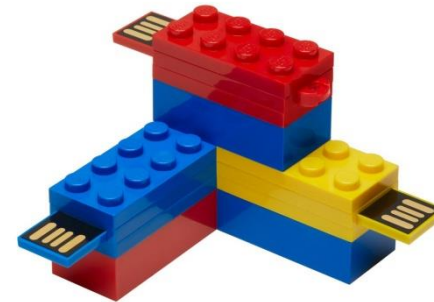
- Maîtrise de l'hétérogénéité et de l'intégration.
- Communication.
- Redondance.

### **1. Métier :**

- Agilité.
- Intégration d'acteurs externes.

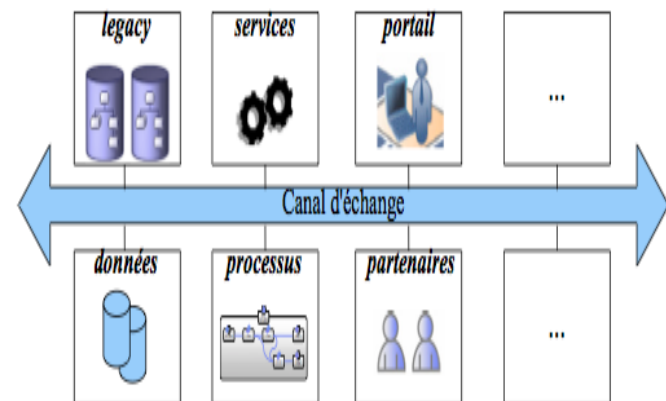
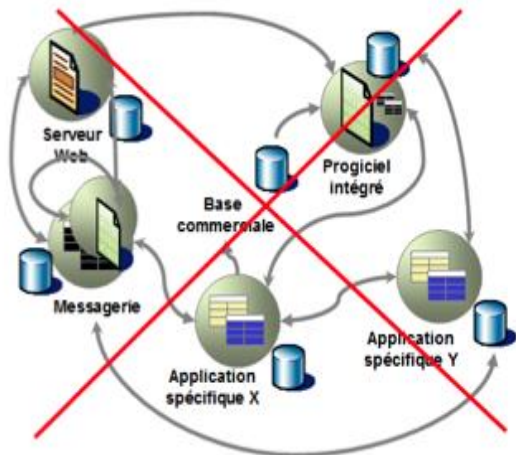
=> **Rationaliser** le SI.

## Bilan de ces évolutions et accélérations



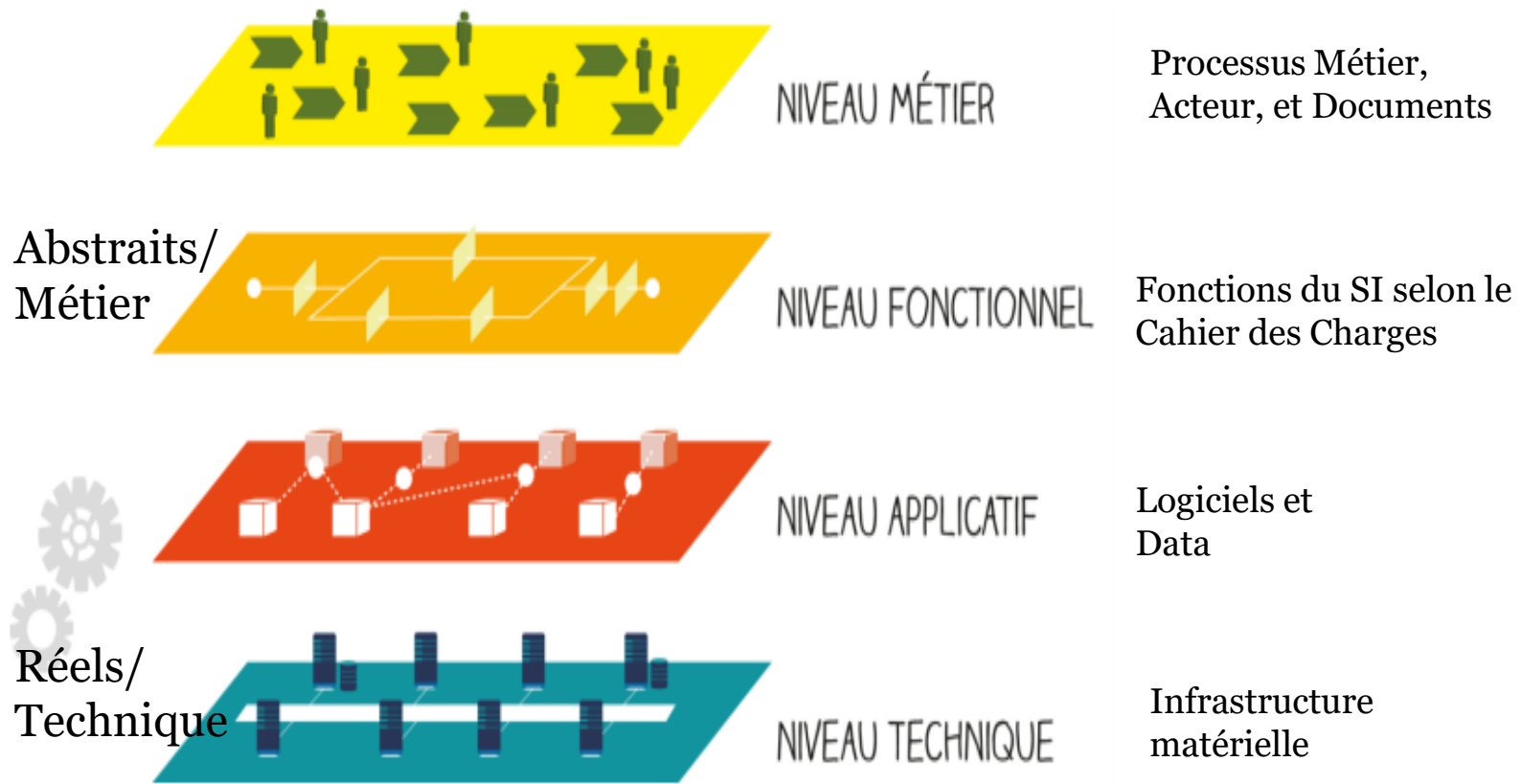
Architecture en Spaghetti

Architecture Urbanisée



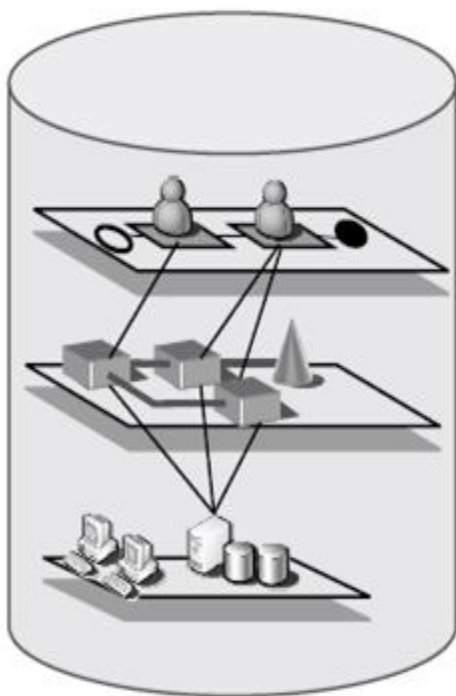
## Les solutions proposées et leurs limites

**Démarche d'urbanisation** = comparer le SI à une ville



Cartographie : Modèle de référence

## Les solutions proposées et leurs limites

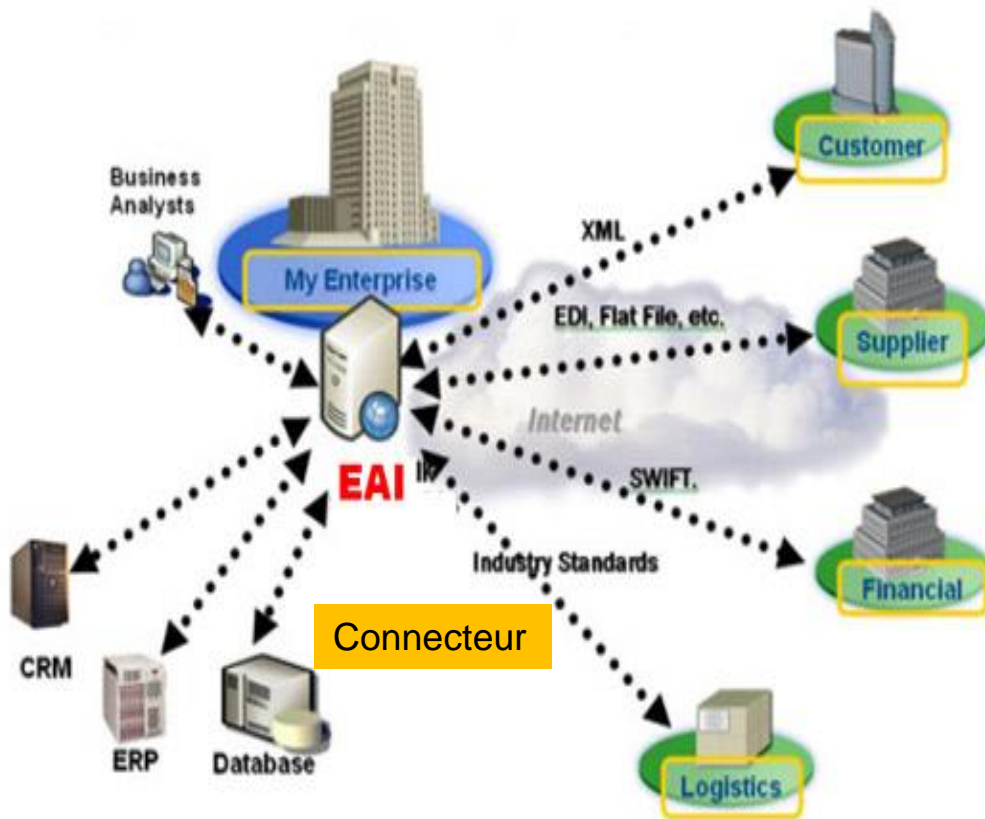


- Pas de communication
- Duplication
- Pas de réutilisation
- pas d'évolutivité

Fonctionnement en Silo – Applications Monolithiques

# Les solutions proposées et leurs limites

## Les EAI



✓ Communication Asynchrone

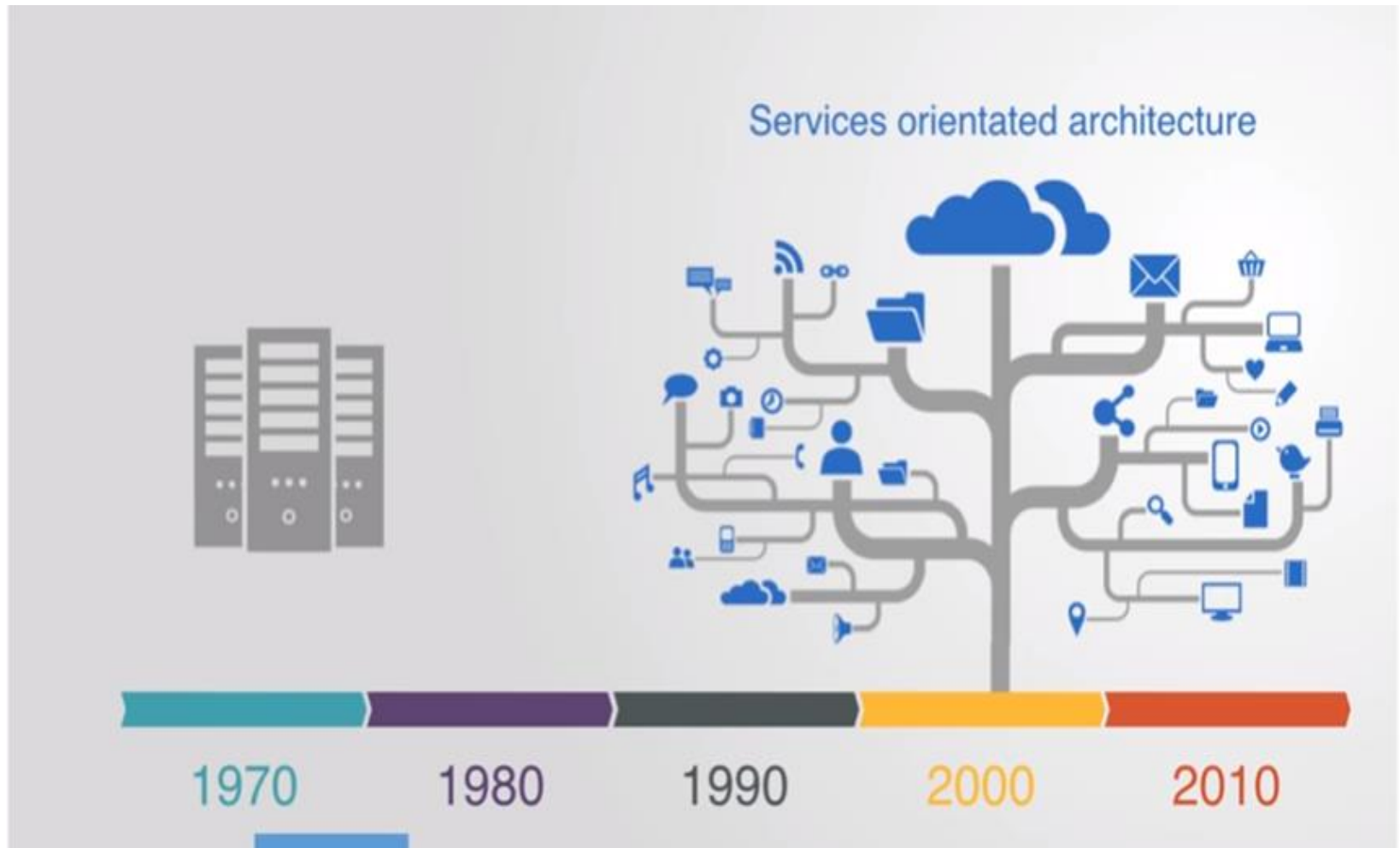
✓ Configuration non répétitive

- Paramétrage long et fastidieux
- Solutions propriétaires et coûteuses
- Non disponibilité de certains connecteurs
- Single Point of Failure

Broker EAI (Routage, transformation, fonctionnalités inter-applications, monitoring)



## Vers des AOS



Isoler et découper la complexité de l'application/bloc applicatif/SI en modules autonomes

## Vers des AOS



Isoler et découper la complexité de l'application/bloc applicatif/SI en modules **autonomes**

## Vers des AOS

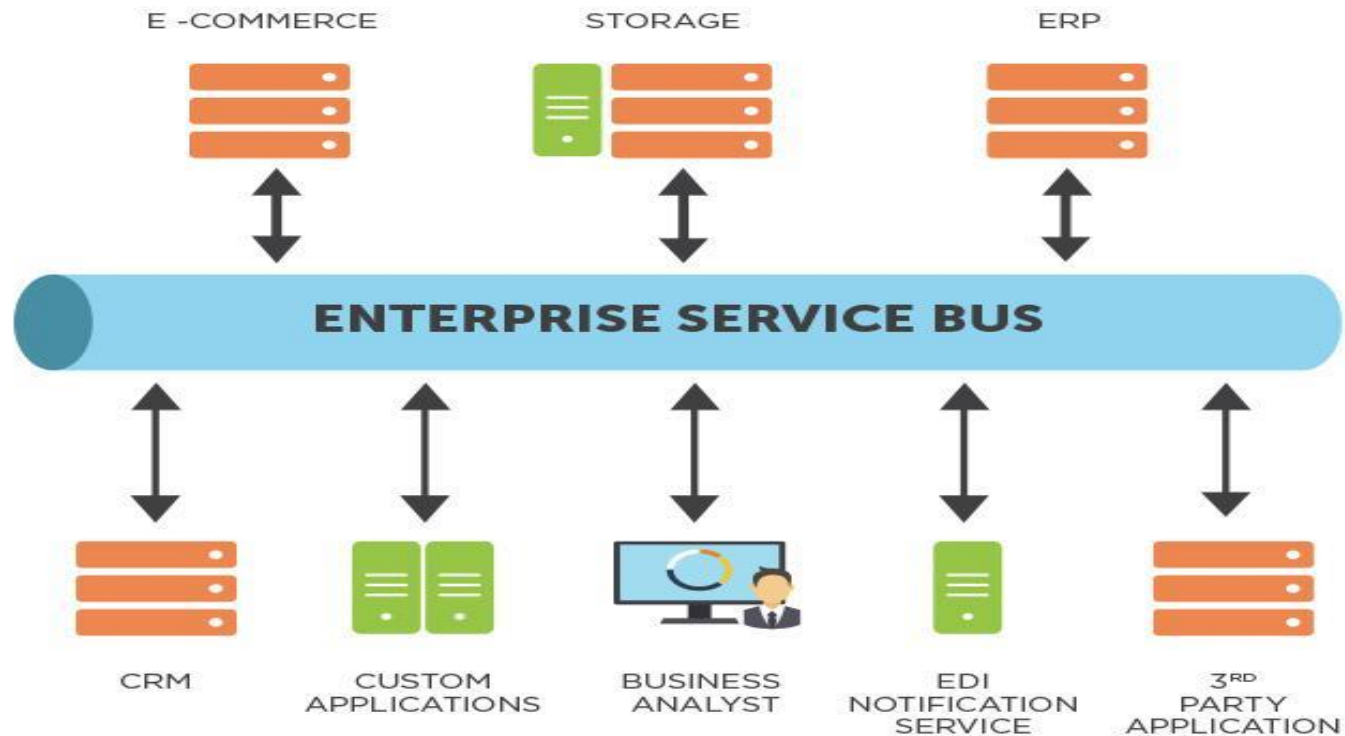


Créer un unique moyen de **communication** avec différents services

Vers des AOS

ESB est utilisé comme plate-forme sur laquelle est réalisée une architecture SOA  
Un intermédiaire de médiation

## Enterprise Service Bus (ESB)



## Vers des AOS

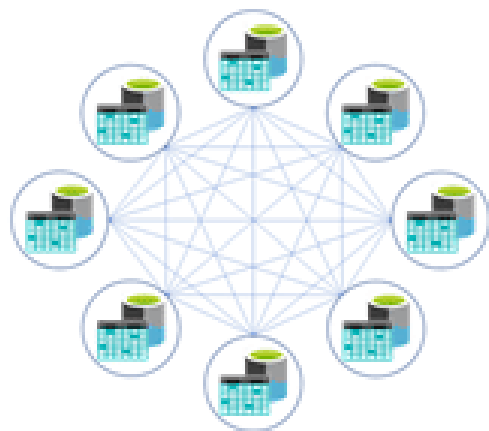
### **EAI** Vs. **ESB** (Entreprise Service Bus)

- EAI est un concept large qui décrit tous les types de modèles d'intégration, mais ESB n'est qu'un exemple de technologie qui permet les EAI, implémentation des EAI.
- ESB agit comme un **intermédiaire** entre les services, tandis que EAI est le modèle d'intégration en **étoile**.
- Les **ESB** sont des centres **d'échanges** et se focalisent sur la mise à disposition des données. Les **EAI** sont davantage des centres **d'intégration**.
- L'EAI intervient au niveau même des applications à partir de **connecteurs** spécifiques. Les données sont transportées vers un point central appelé **Middleware**.
- L'ESB n'intervient pas directement dans les applications du système mais via des **modules de services** au sein de chacune d'entre elles.
- Un ESB est construit sur des **standards** ouverts comme le **XML** pour la description des **messages** et l'échange de données.

# Vers des AOS

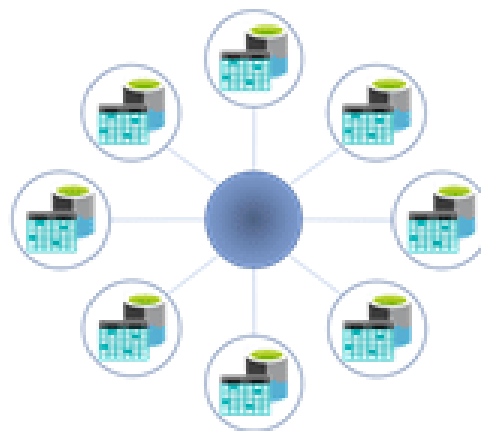
## ESB Vs EAI

Le fameux « effet spaghetti »



Couplage inter-applications

Solution EAI



Centre d'intégration

Solution ESB



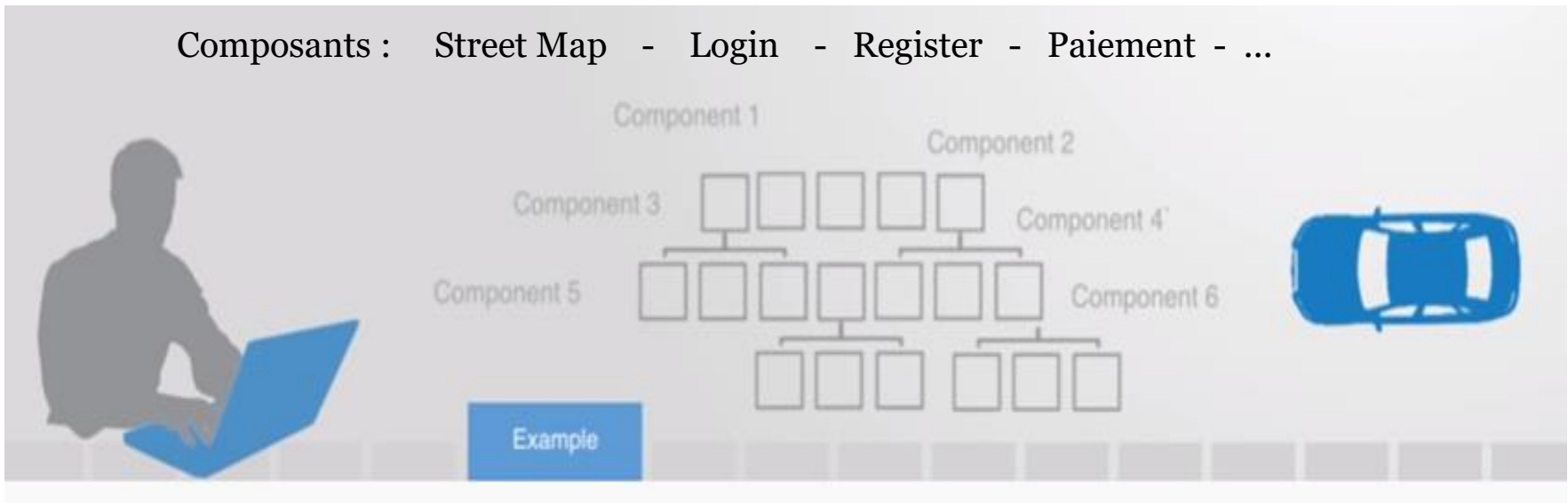
Centre d'échanges

## Vers des AOS

**Exemple** : application web pour payer en ligne sa place de parking

Sans SOA

Composants : Street Map - Login - Register - Paiement - ...



# Vers des AOS

Avec SOA



Google Maps



Paypal



Facebook





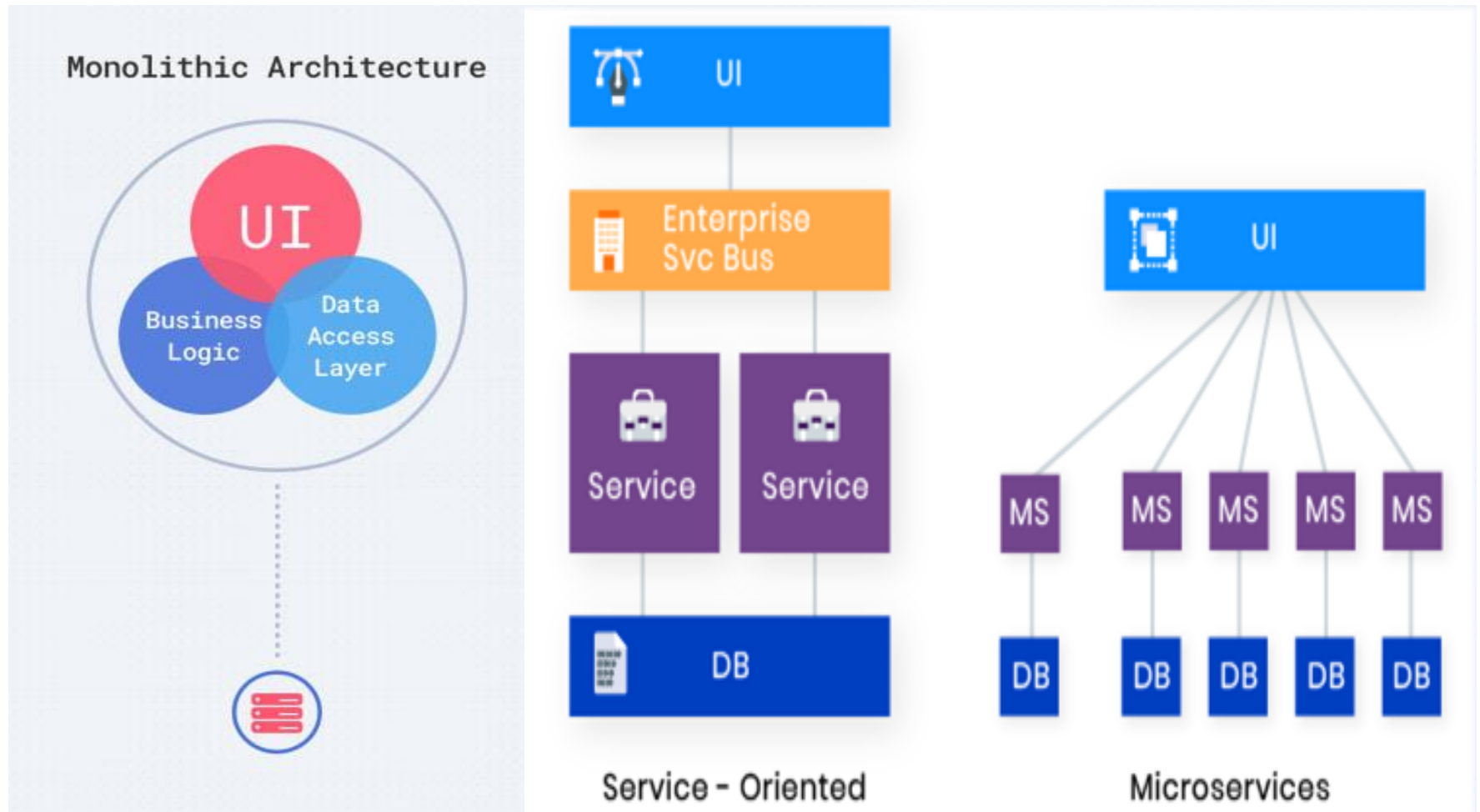
**SOA n'est pas une technologie, ni une recette,  
encore moins un produit.**

SOA est d'abord un ensemble de concepts constituant un modèle cohérent d'architecture pour faciliter la conception d'un SI flexible via l'émergence de **services** intégrant/réutilisant des applicatifs existants.

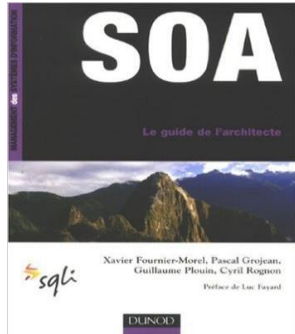
**SOA n'est pas une technologie, ni une recette,  
encore moins un produit.**

Comme son nom le suggère, l'élément clé de SOA est le **Service**. Un **Service** est un composant logiciel distribué, exposant les fonctionnalités à forte valeur ajoutée d'un domaine métier.

## Après SOA, Les Microservices



# Ressources



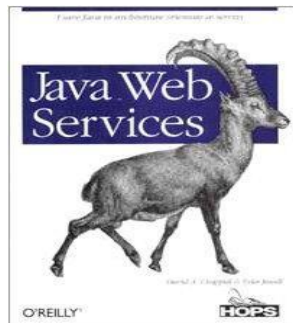
## **Le guide de l'architecte du SI**

- ✓ Auteur : Xavier Fournier-Morel, Pascal Grosjean, ...
- ✓ Éditeur : Dunod
- ✓ Edition : Octobre 2006 - 302 pages - ISBN : 2100499726



## **SOA Principles of Service Design**

- ✓ Auteur : Thomas Erl
- ✓ Éditeur : Prentice Hall Ptr
- ✓ Edition : Juillet 2007 - 608 pages - ISBN : 0132344823



## **Java Web Services**

- ✓ Auteur : David Chappell & Tyler Jewell
- ✓ Éditeur : O'Reilly
- ✓ Edition : Mars 2002 - 276 pages - ISBN : 0-596-00269-6

# Ressources

## **Engineering Long-Lasting Software: An Agile Approach Using SaaS and Cloud Computing**

- ✓ Auteur : Armando Fox and David Patterson
- ✓ Éditeur : Strawberry Canyon LLC
- ✓ Edition : Aout 2012 - 412 pages - ISBN : 0984881212

Cours – Mickael Baron – SOA et Microservices

- ✓ [http://mbaron.developpez.com/#page\\_soa](http://mbaron.developpez.com/#page_soa)

Cours – Koushik Kothagal - Developing SOAP Web Services with JAX-WS -  
[https://javabrainz.io/courses/javaee\\_jaxws/lessons/Introduction-to-Web-Services](https://javabrainz.io/courses/javaee_jaxws/lessons/Introduction-to-Web-Services)

Service-Oriented Architecture

- [https://www.youtube.com/watch?v=\\_\\_dFJOSR-aFs](https://www.youtube.com/watch?v=__dFJOSR-aFs)