Architectures Modernes des Systèmes d'information

Agnès CREPET

Mars 2018

Mines Saint-Etienne

@agnes_crepet

Depuis 15 ans... Développeuse - Architecte logiciel - Entrepreneuse - Enseignante

Mon expérience

… depuis 2001

Architecte sur plusieurs projets de refonte de SI

Expertise dans l’écosystème java

Coach Méthodes Agiles

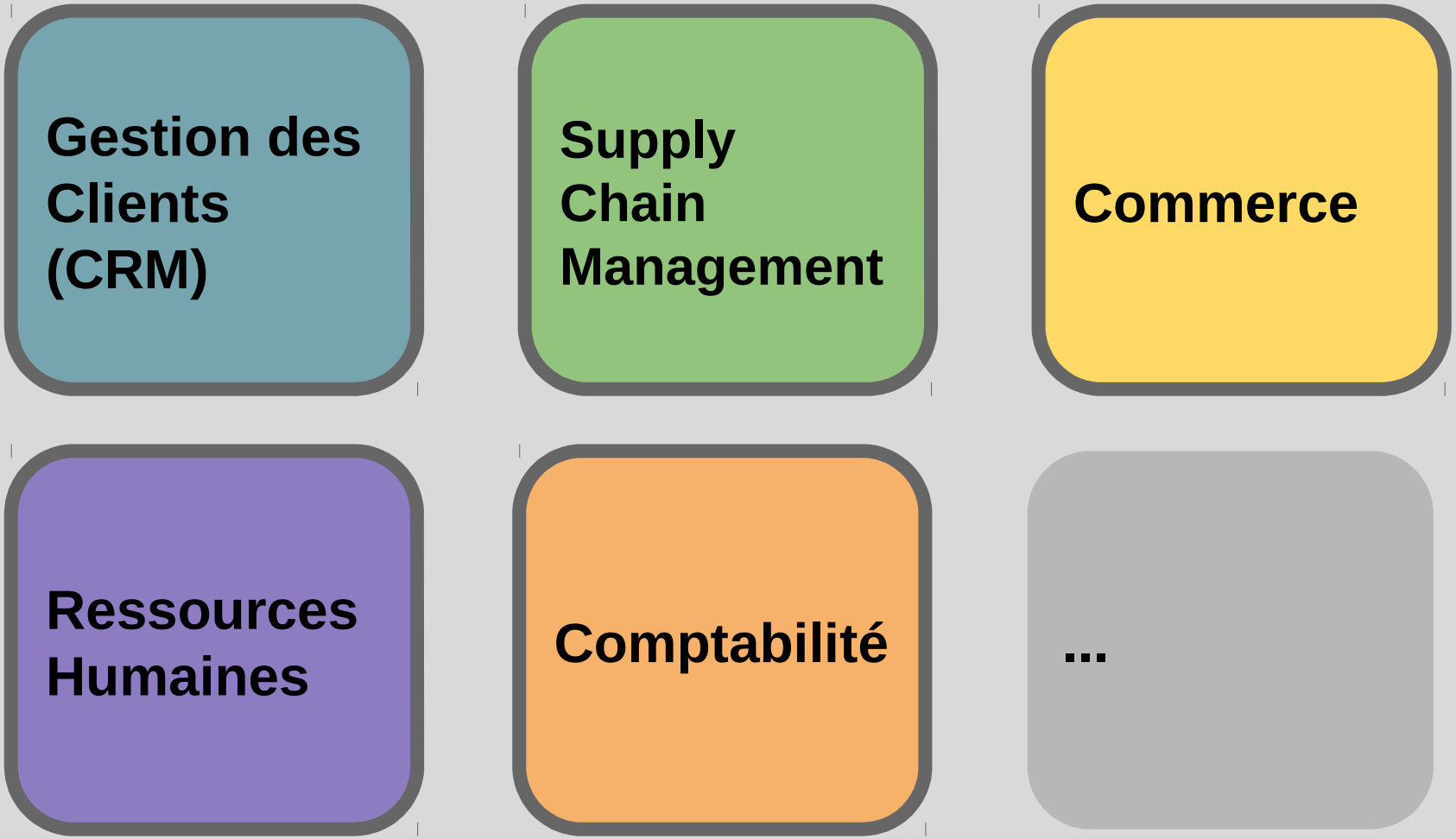
© Ton Hendriks

Le sujet du jour

Comment faire **communiquer…**
… les applications d’un système d’information?

Applications en silos

Applications en silos...
...qui doivent communiquer

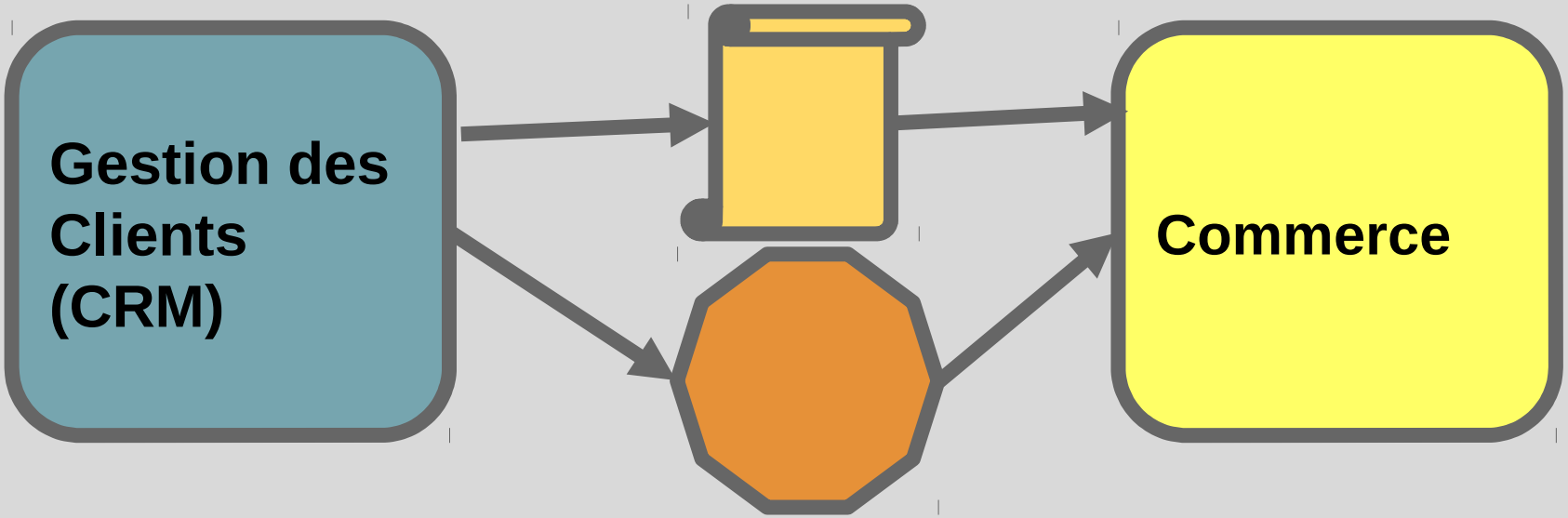


Comment les faire communiquer

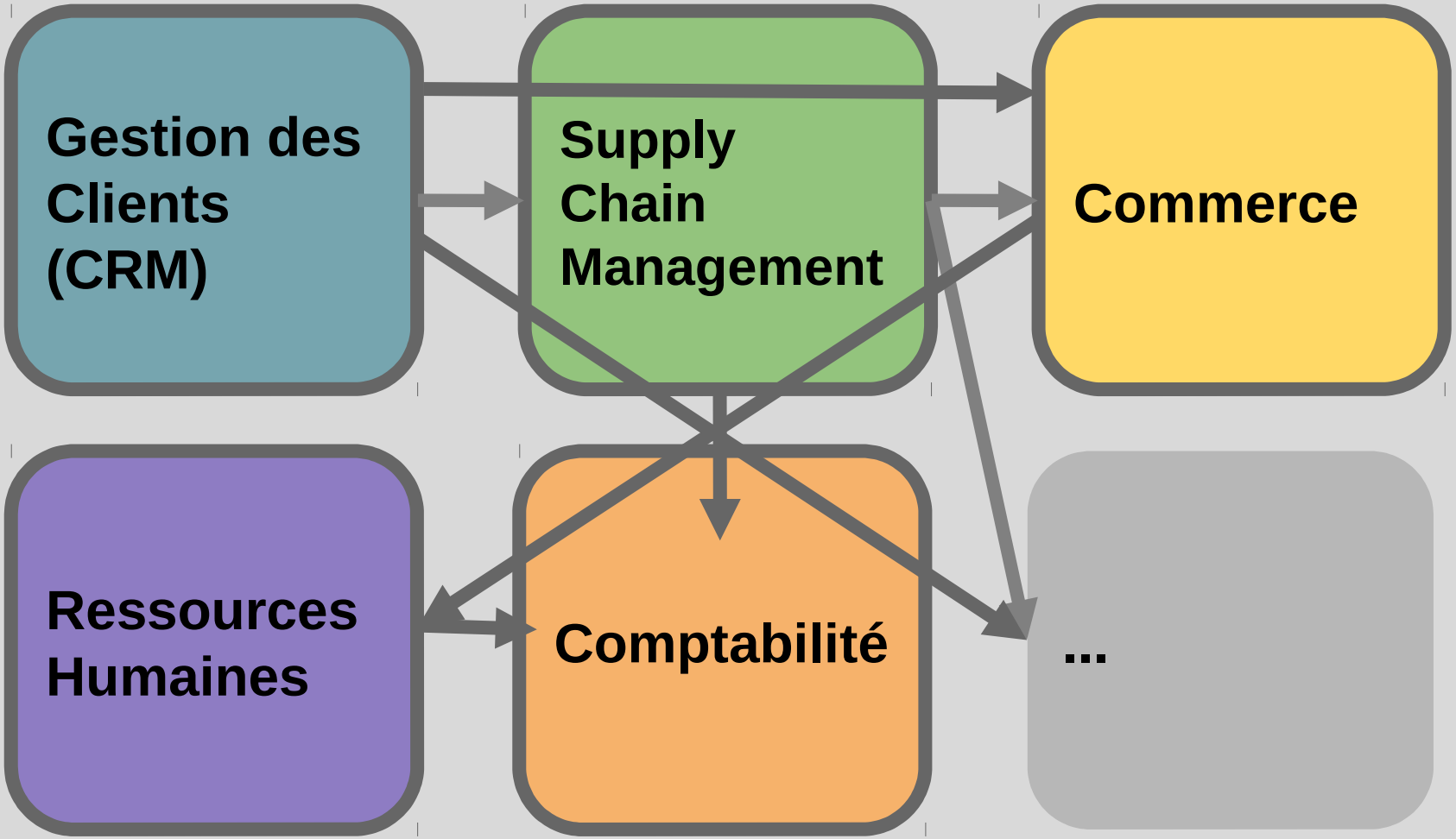
?

Un fichier?
ou
un batch?

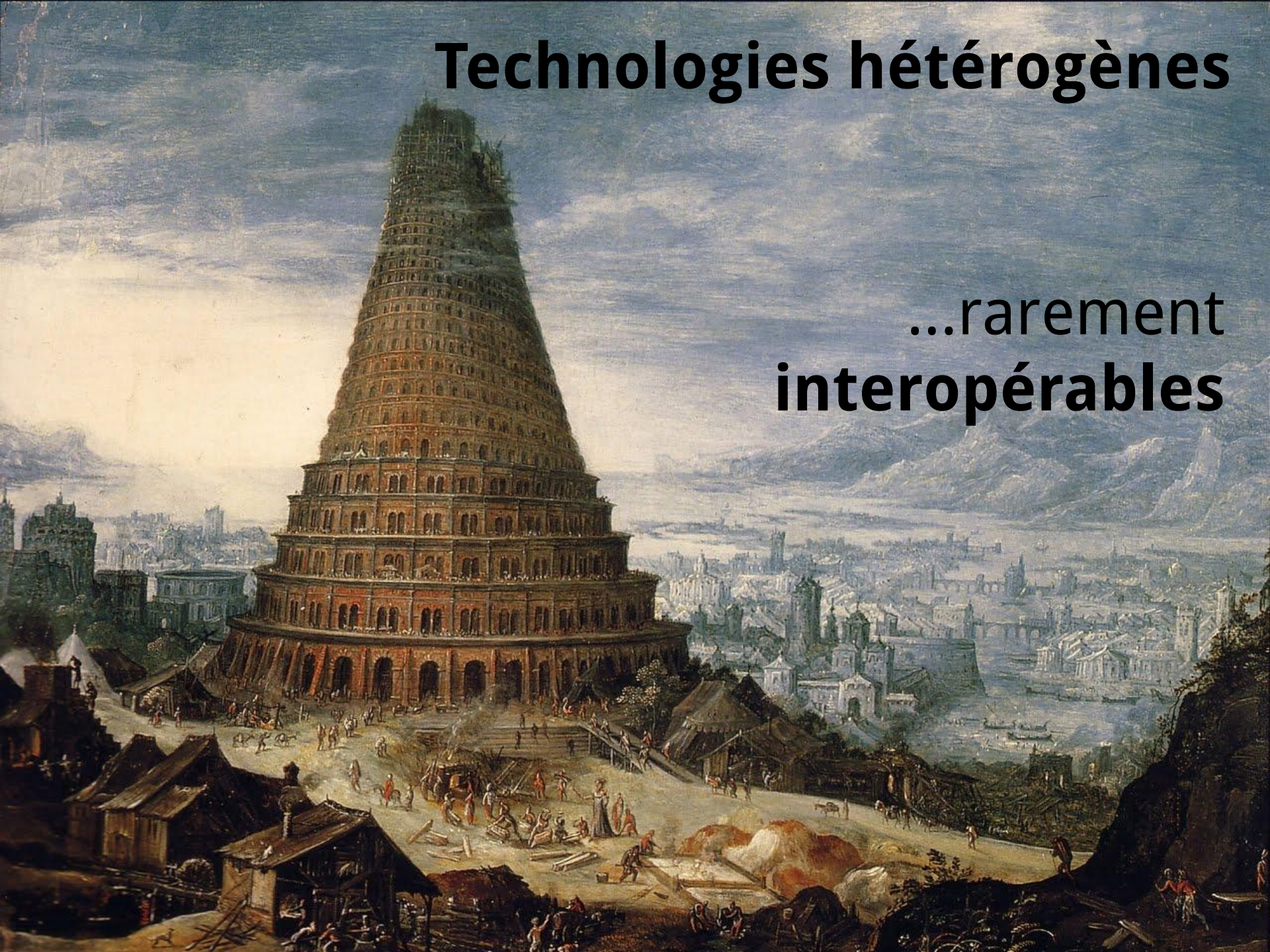
= interface point à point spécifique



Multiplications des interfaces!

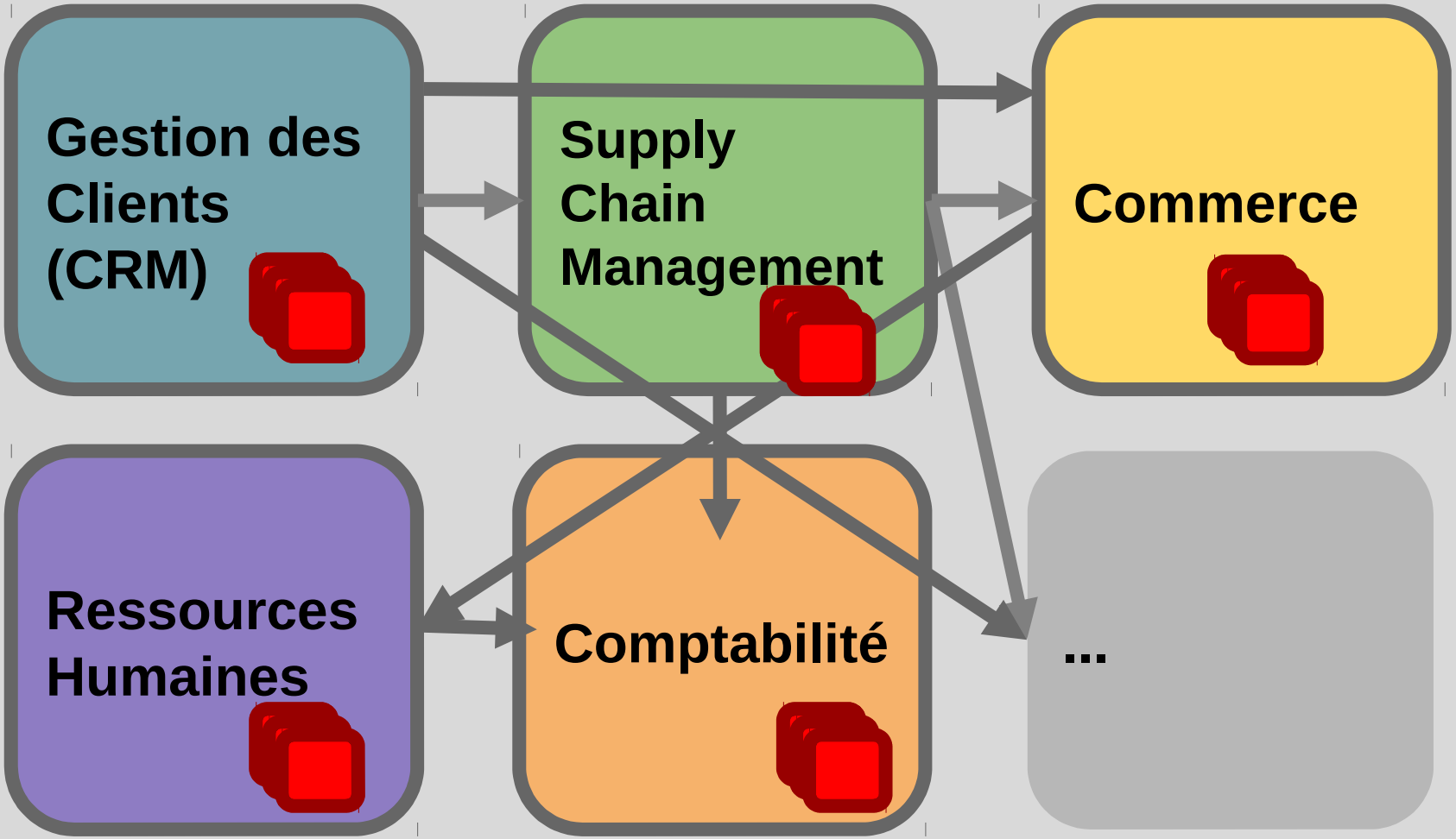


Multiplications des flux !


Technologies hétérogènes
...rarement
interopérables

Données redondantes, dupliquées

Duplication des données



Faire évoluer le SI ?

Maintenir le SI ?

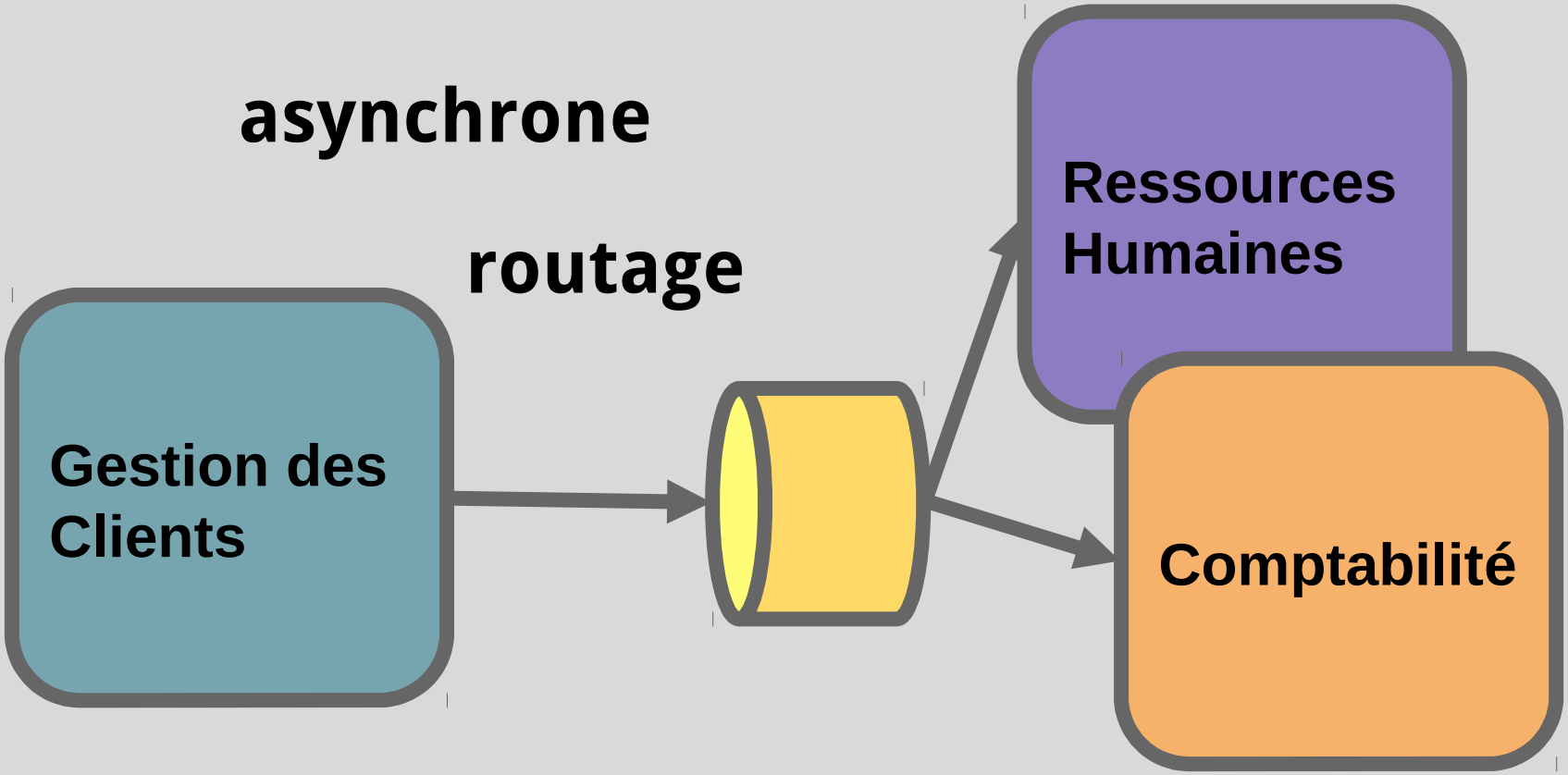
Premières Solutions d'intégration

Message
Oriented
Middleware

souvent
propriétaires

asynchrone

routage



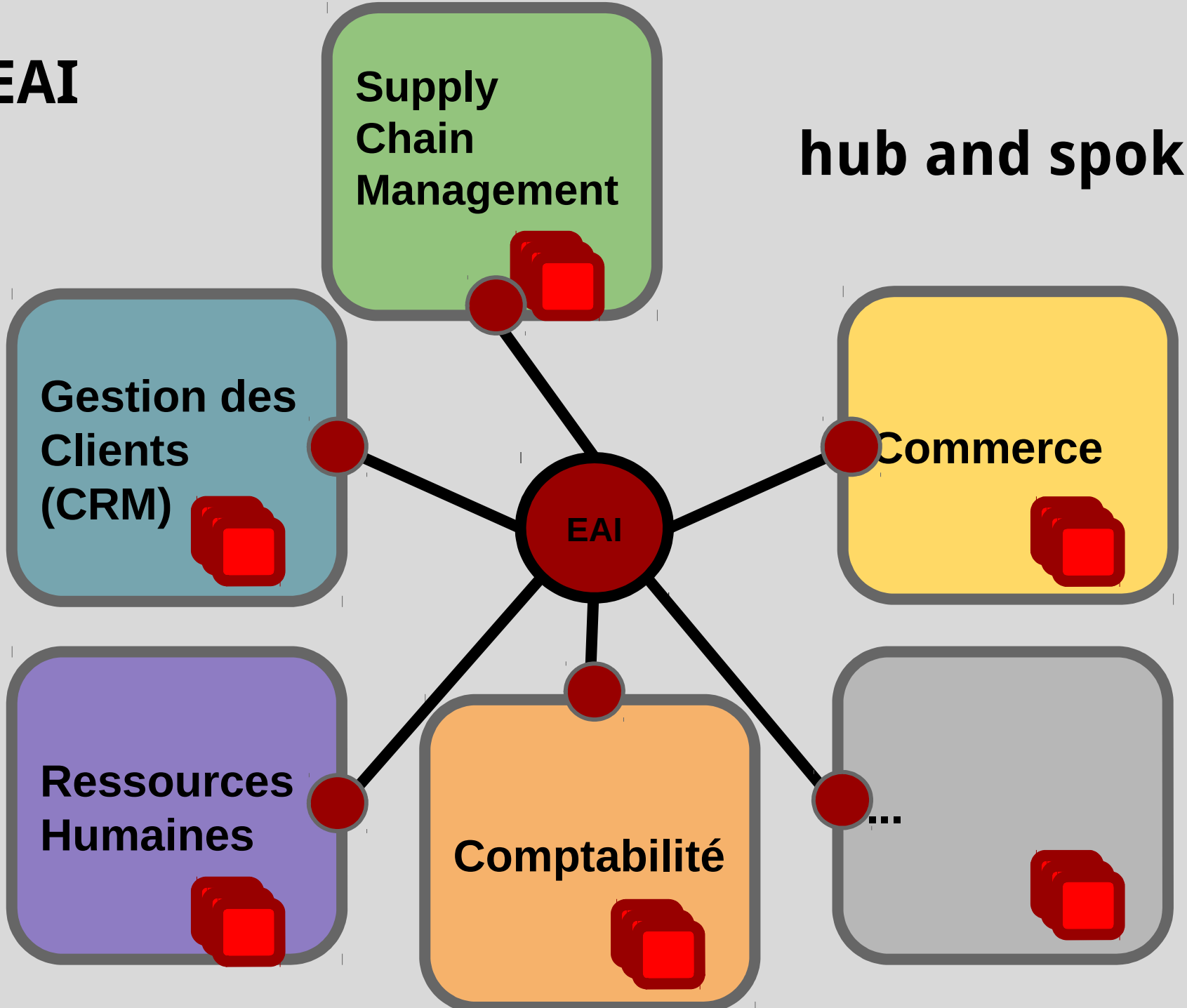
Premières Solutions d'intégration

… toujours propriétaires

Enterprise Application Integration (EAI)

EAI

hub and spoke



EAI, l’intégration magique?

Gère les problématiques techniques

Interopérabilité au moyen de connecteurs spécialisés

EAI, inconvénients

Complexe, couteux

Tout propriétaire :
- pour échanges et transport des messages
- connecteurs applicatifs spécifiques
- formats et encodages de données

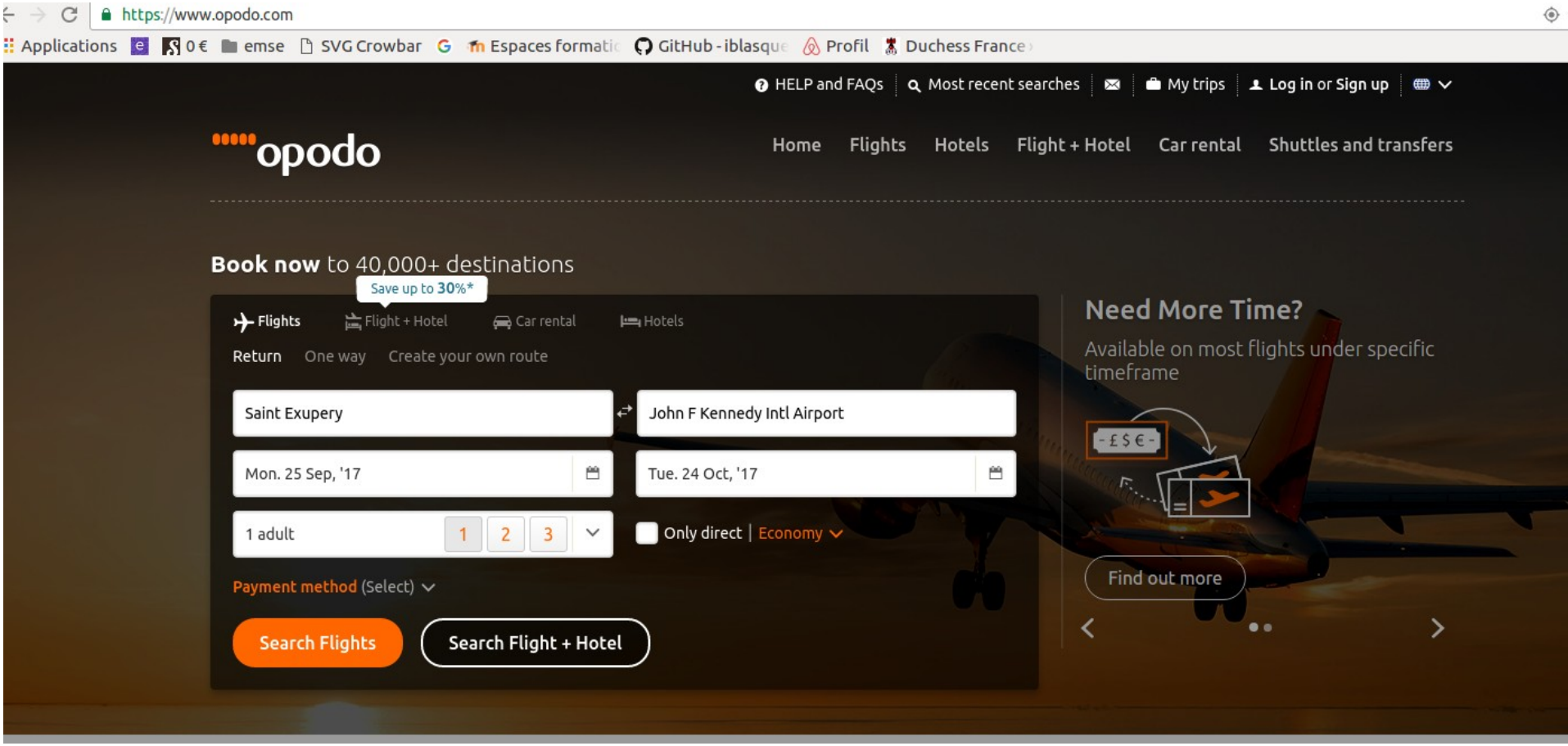
Une approche plus standardisée ?

Pour intégrer d'une façon standard
Le mode d'intégration des applications peut être standardisé
→ On parle de patterns d'Intégration d'Entreprise (EIP) : http://www.eaipatterns.com
Ex : splitter, content based router

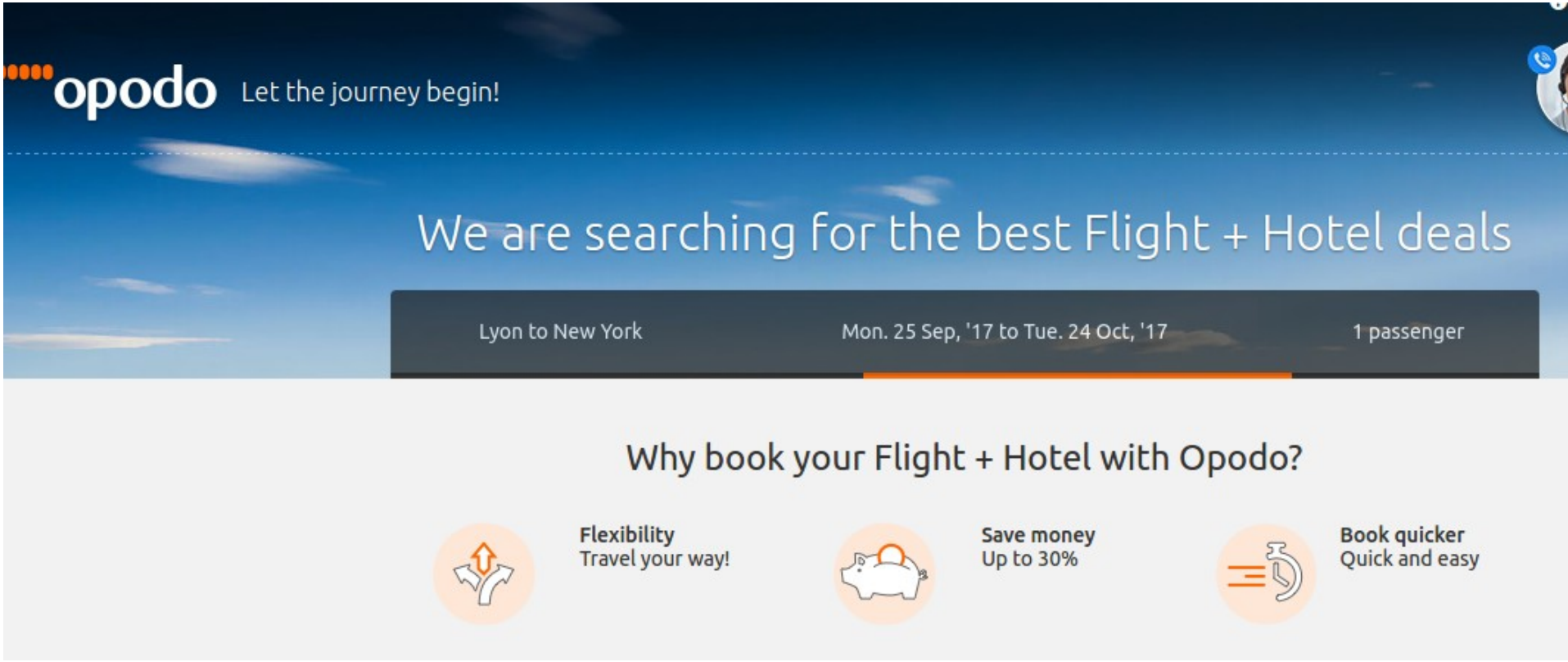
Et les applications définissent leurs services à exposer : API

* Advanced Message Queuing Protocol

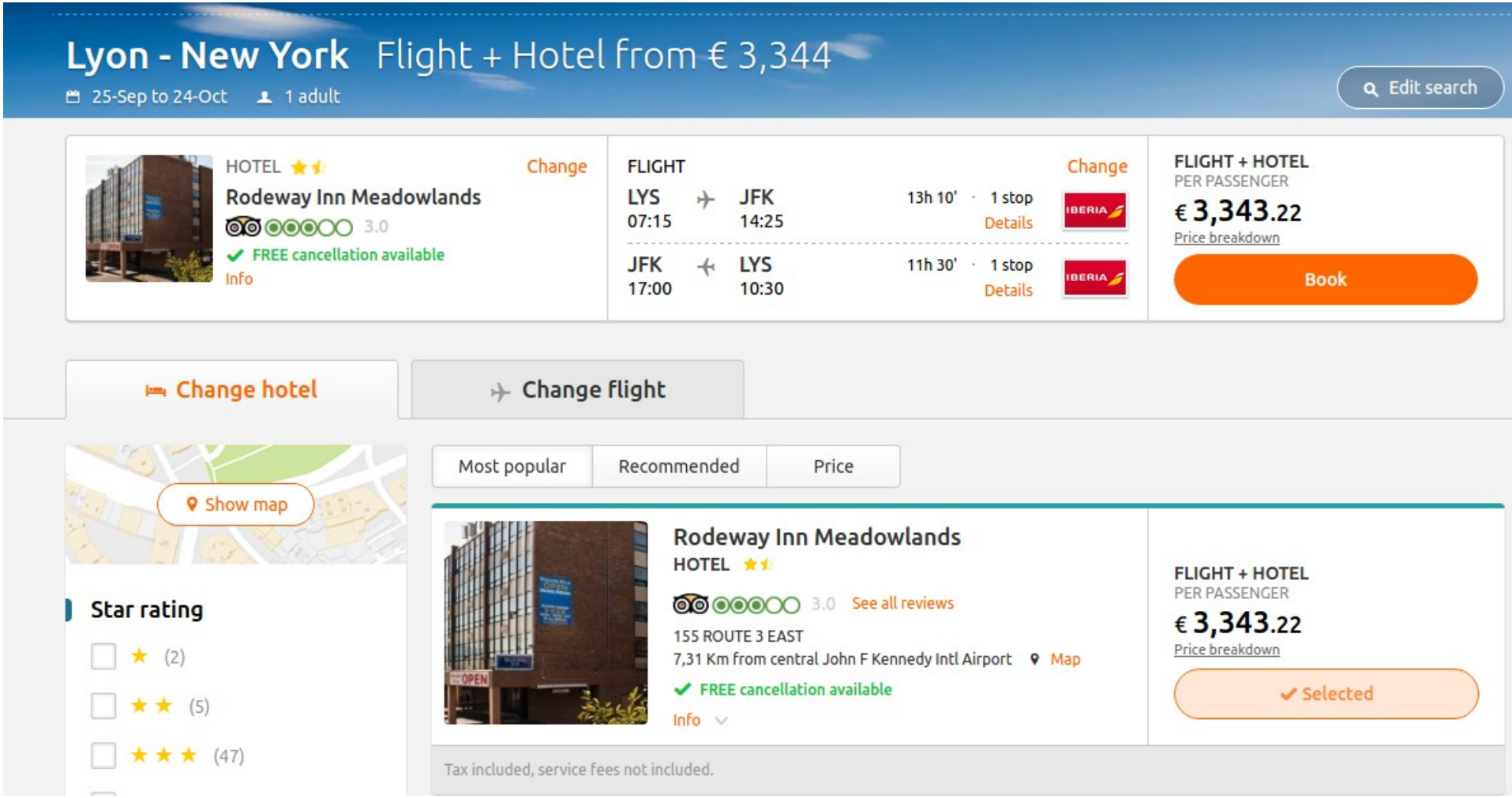
I want to Book my holidays !



I want to Book my holidays !



I want to Book my holidays !



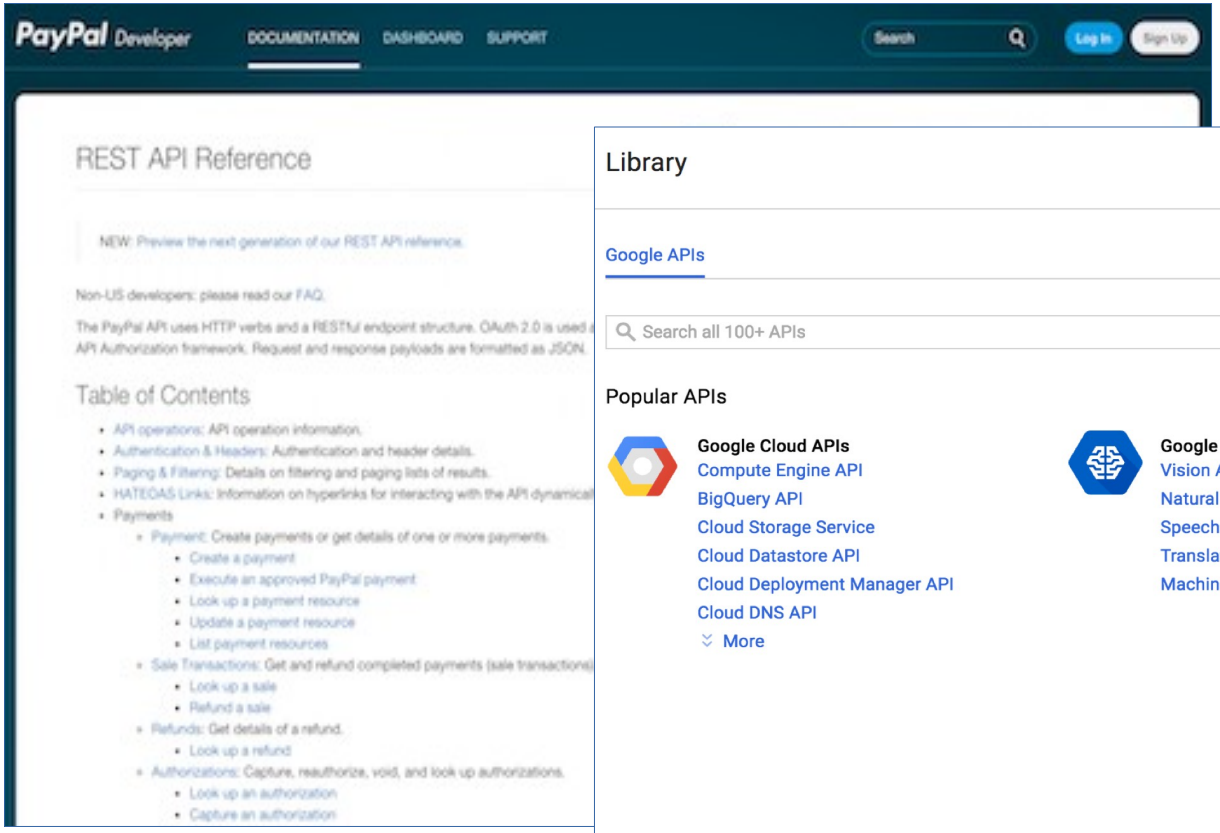
API

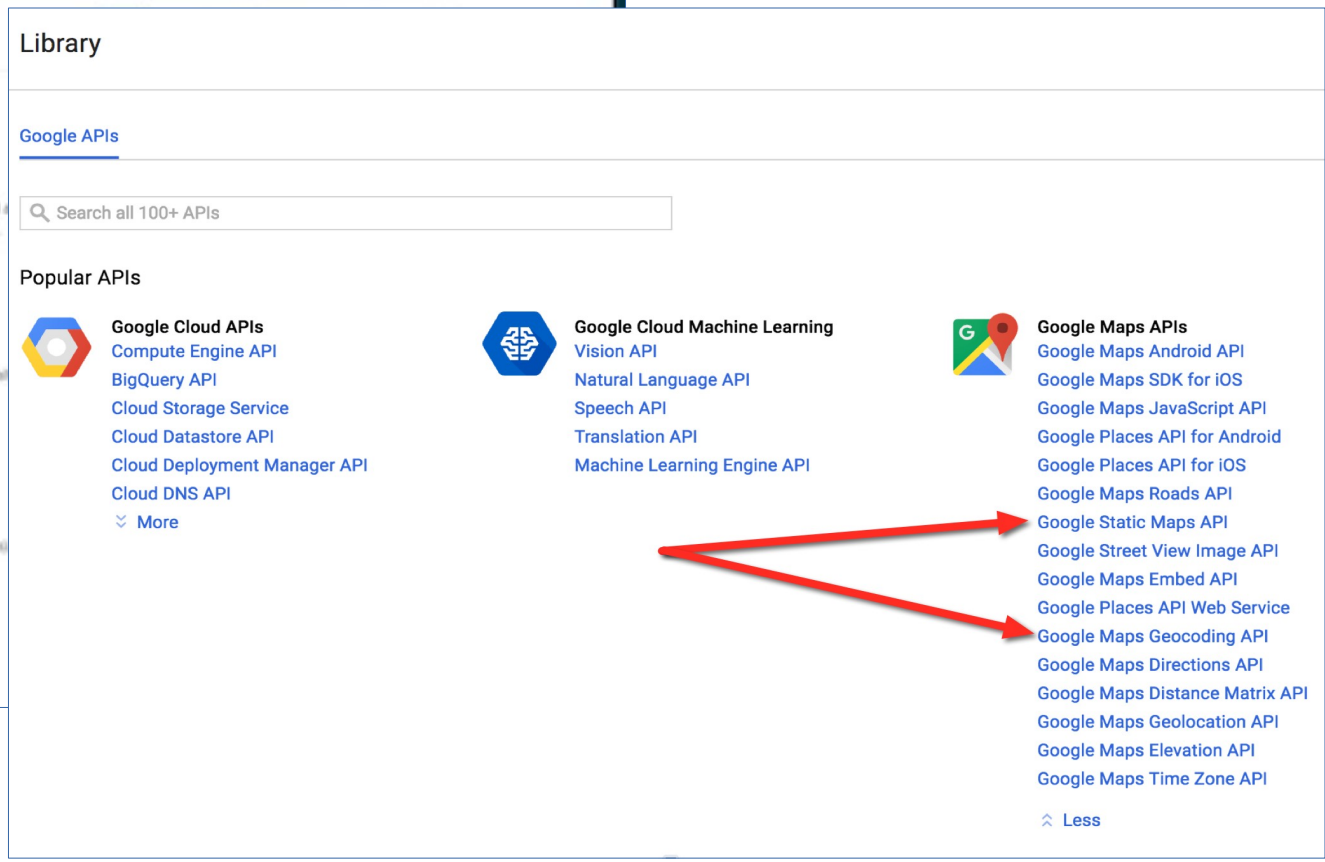
What does API mean?

Application Programming Interface

The world is full of APIs!

- Every app of your daily life has an API !





The world is full of APIs!

- URI (Uniform Resource Identifier) :

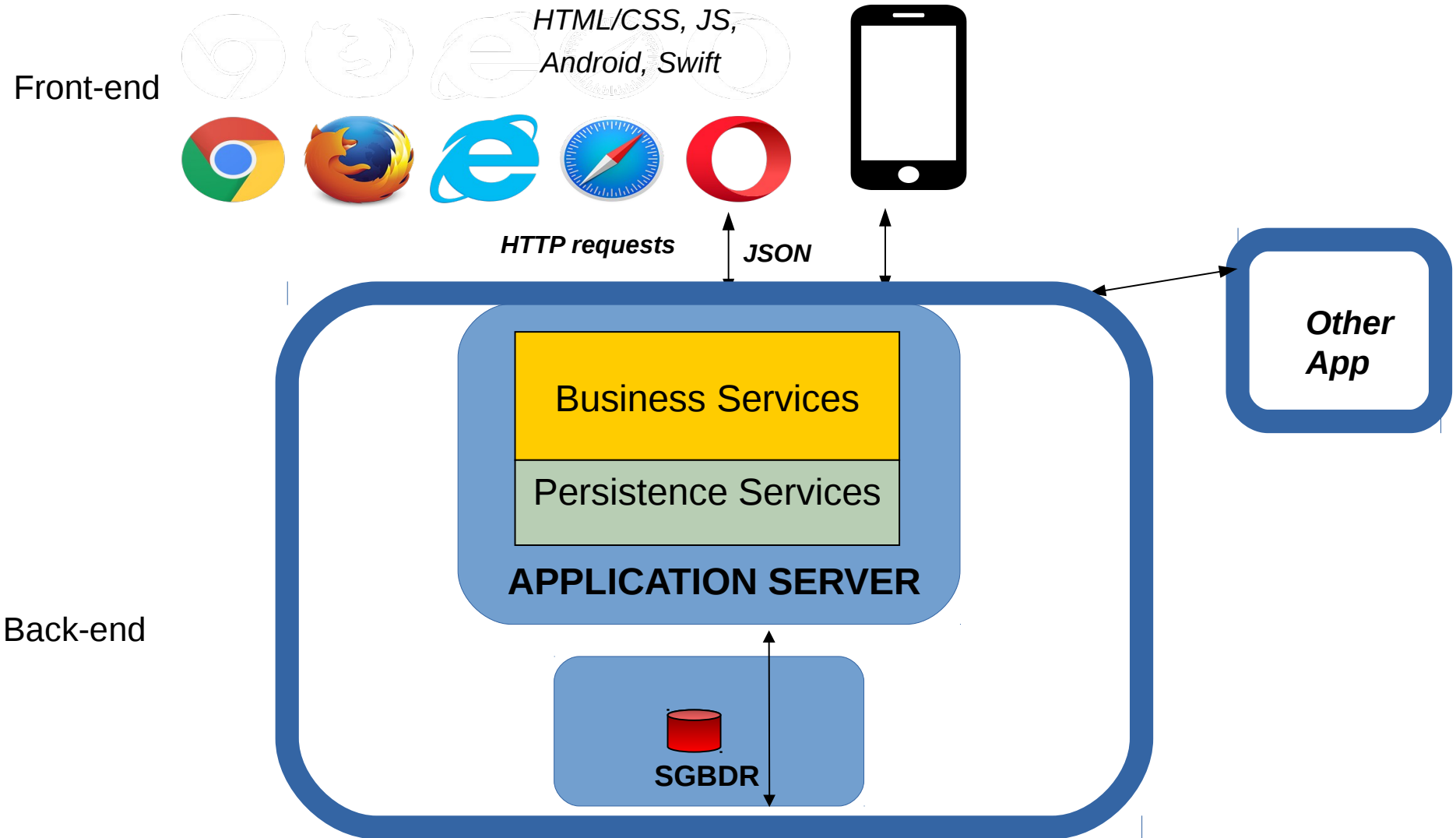
  http://mywebsite.com/books/87/comments

- Demo

  https://api.github.com/search/repositories?q=java

  https://api.github.com/users/Ninja-Squad/repos

Under the hood
Software Architecture



ESB? Entreprise Service Bus

Apparus dans les années 2000
Evolution des EAI

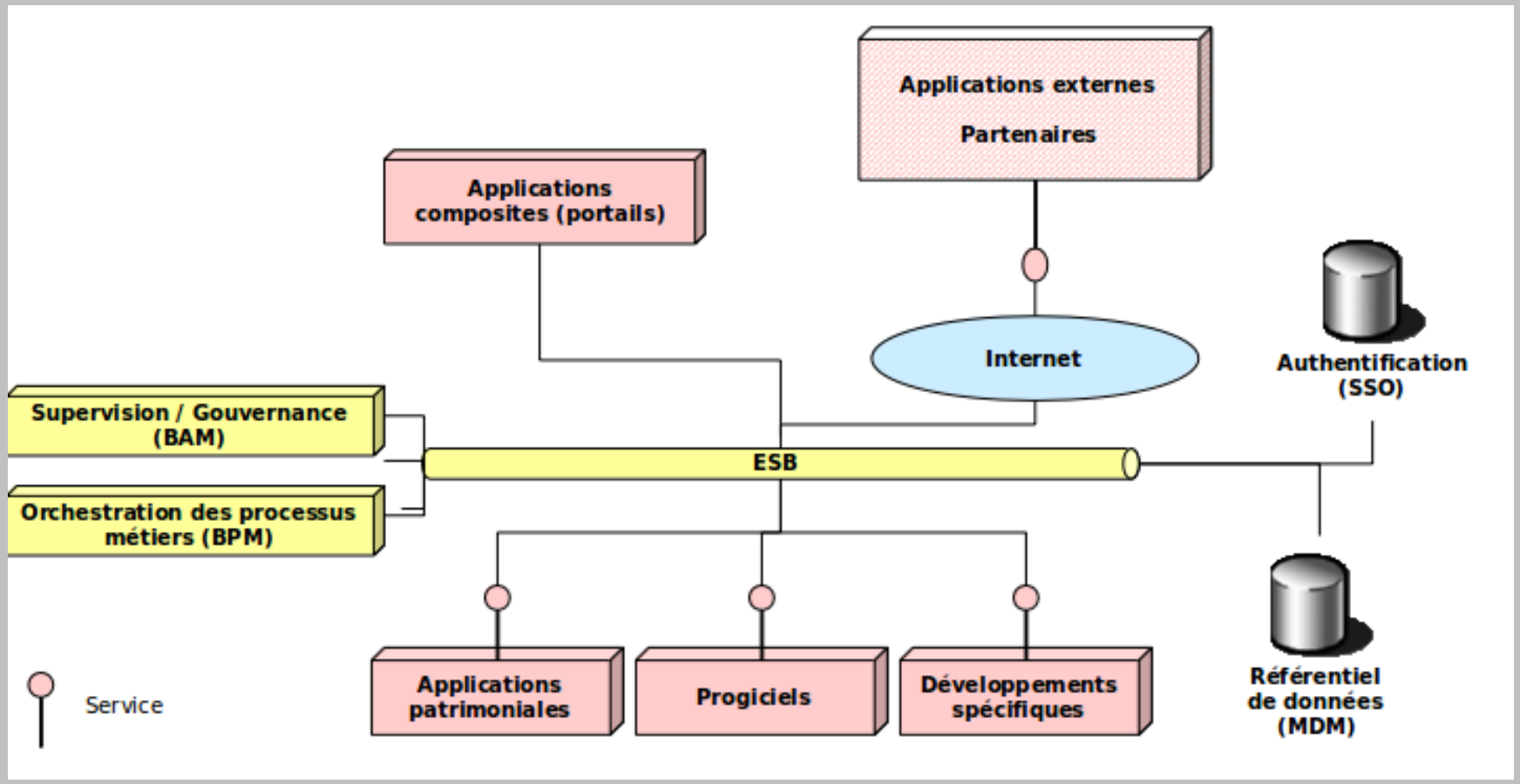
Construits sur des standards :

- Du web : XML/JSON, REST, webservices
- Technologie messaging : JMS, AMQP*

Des notions communes :
endpoints, producteurs,consommateurs,
souscription/publication de services...

Base des SOA (Service Oriented Architecture)

ESB? Entreprise Service Bus



ESB

- Réconcilier des mondes hétérogènes
- Exposer des services d’applications
- Découpler consommateurs et fournisseurs de services
- Agréger des services
- Tracer les messages qui transitent
- Limiter l'impact des changements,en "virtualisant" certains services, ou certaines versions de services, pendant les phases transitoires

ESB vs EAI

L'EAI mise sur l'interopérabilité et sur les échanges entre applications hétérogènes (via des connecteurs spécifiques à chaque technologie et souvent onéreux!)

Ne permet pas les réutilisations de briques d'une application

ESB vs EAI

l'ESB permet de :
- décharger les applications de code "technique" lié à l'implémentation des échanges (principe de couplage lâche)
- rationaliser les formats d'échanges
- rajouter/enlever facilement de nouveaux consommateurs (principe de souscription/publication)

Au coeur des ESB, des services

On identifie les services (logique métier) implémentés par les applications

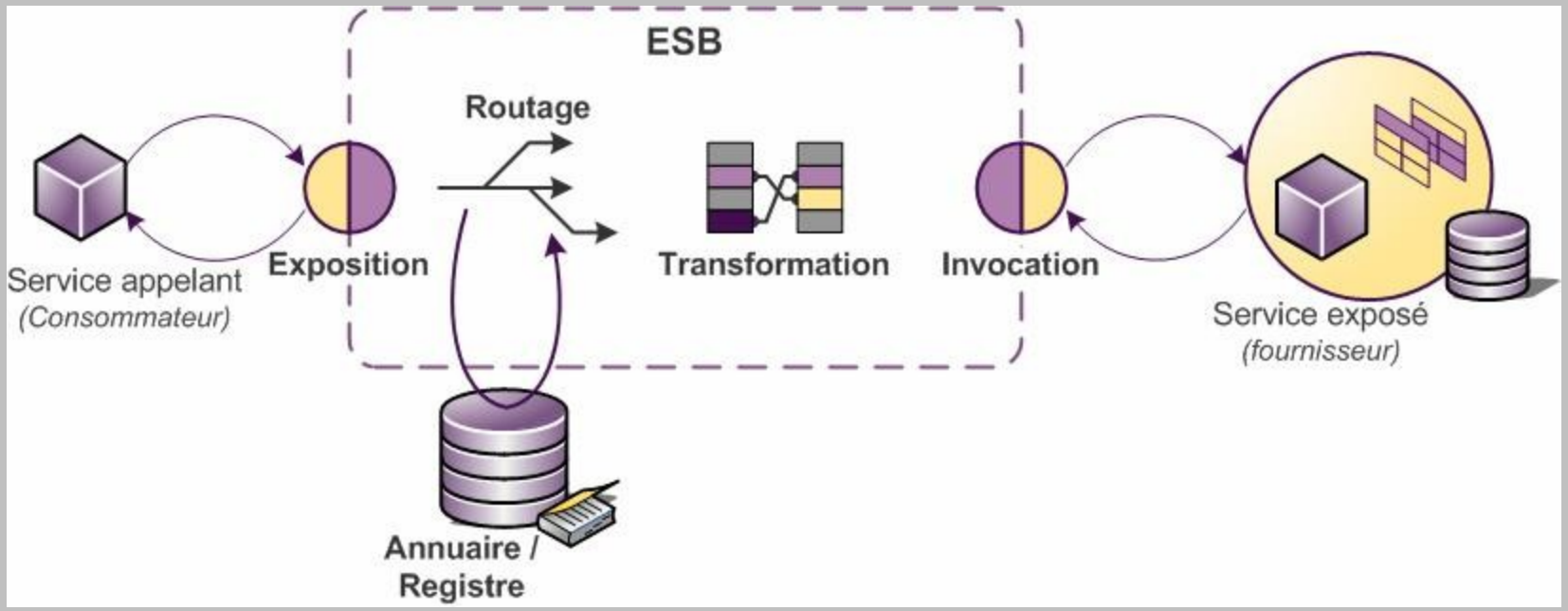
On les référence dans un catalogue de services

Un catalogue de service n'est qu'une vue logique (statique) des fonctionnalités métiers du SI.

ESB - Principes

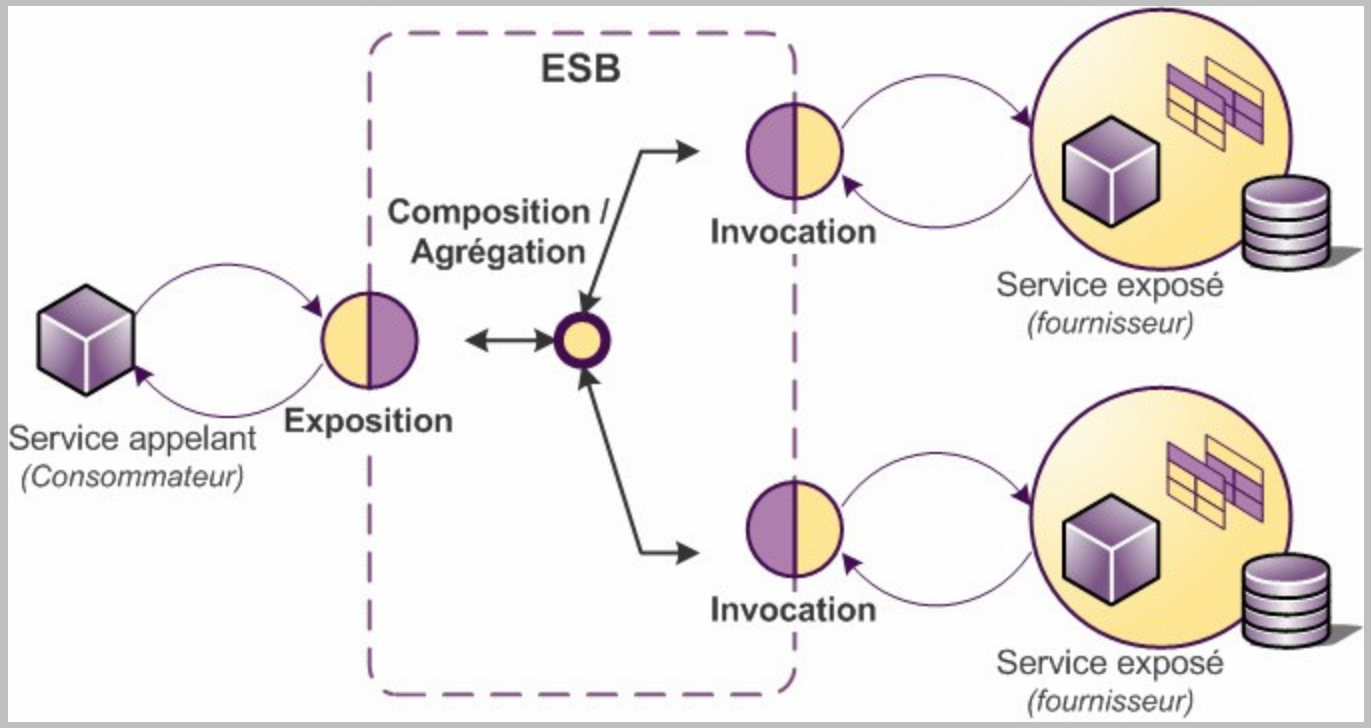
Fournissent des fonctions :

- de **transformation**
- et de **routage** des flux



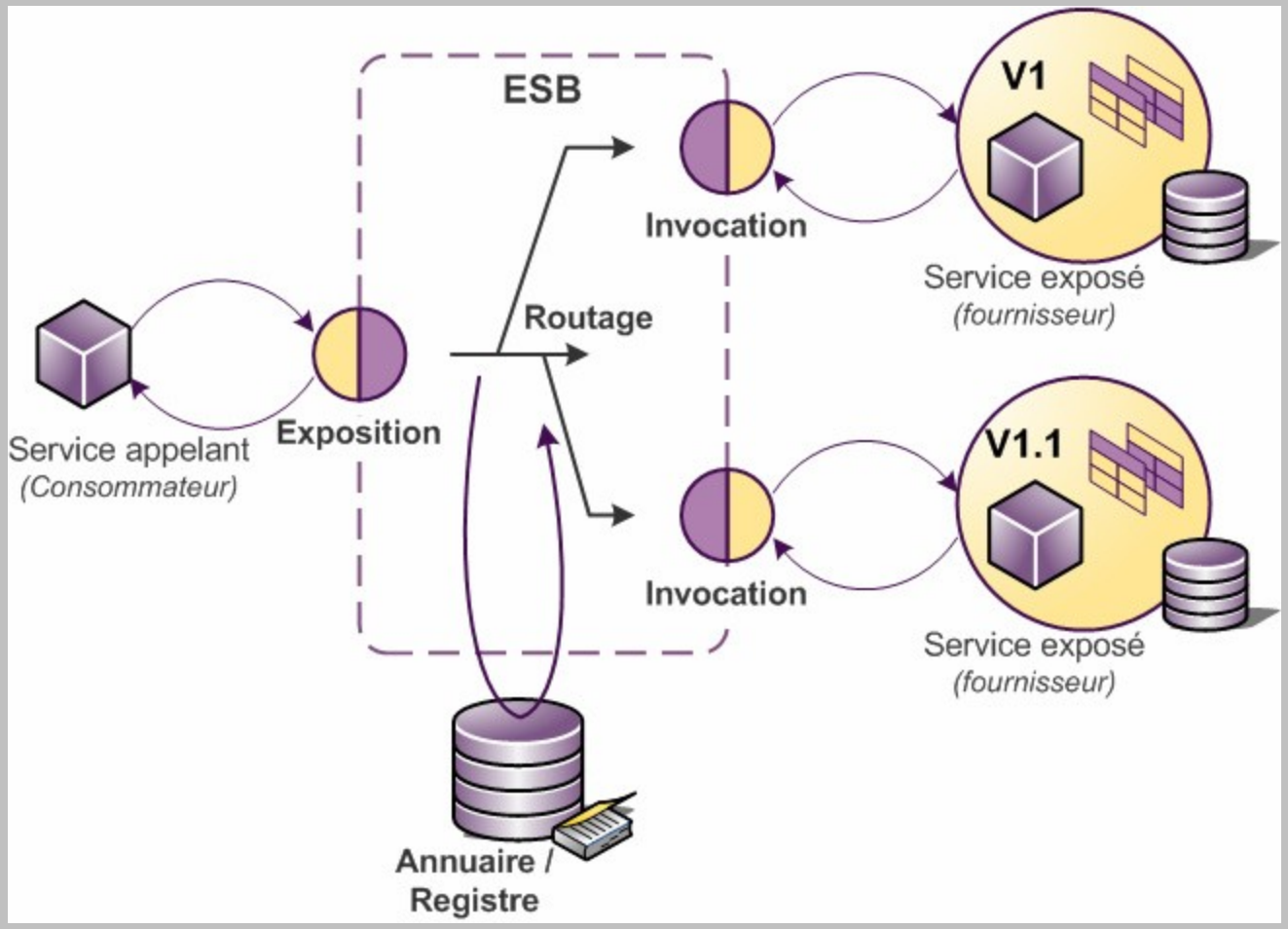
ESB – exemple cas d'utilisation

Composition de services



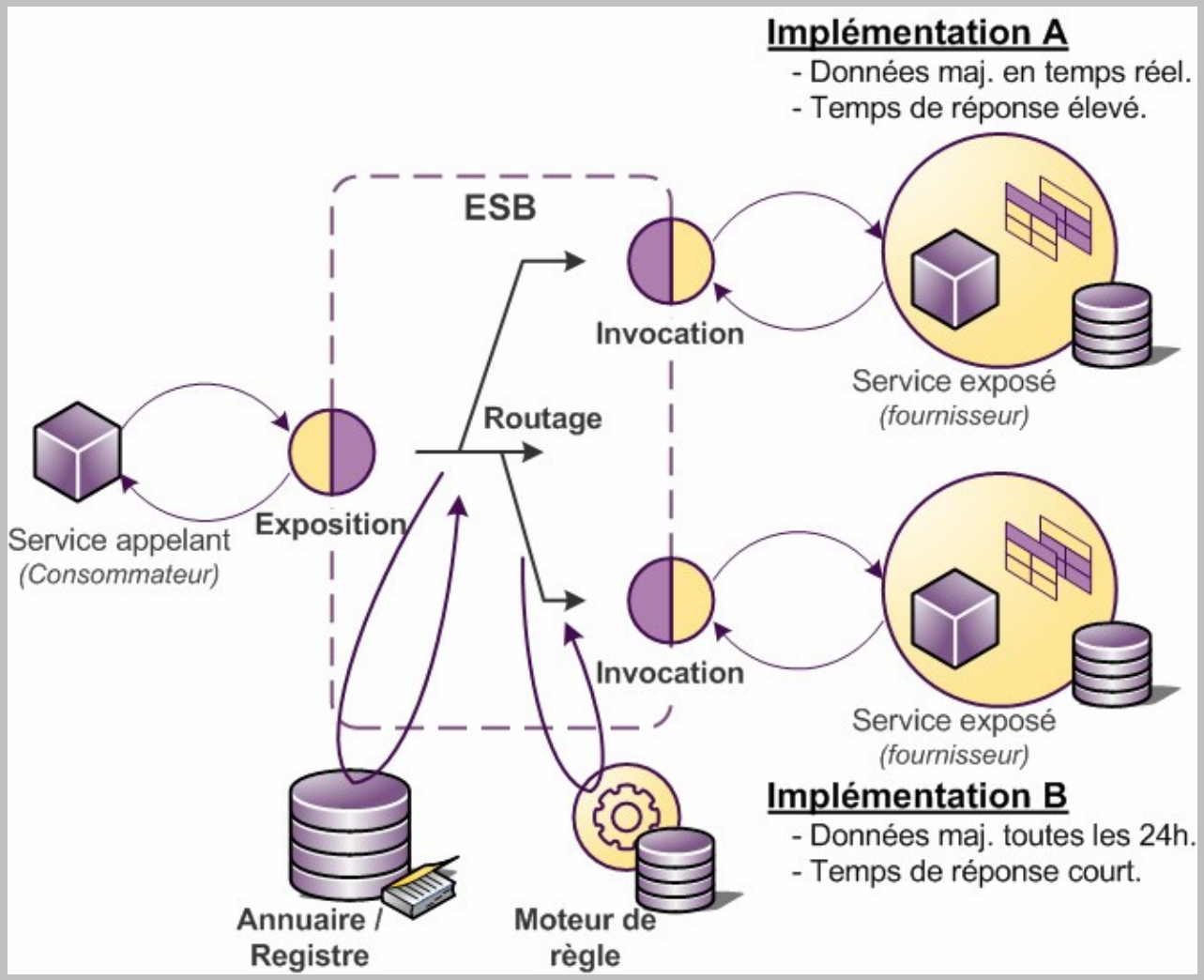
ESB – exemple cas d'utilisation

Gestion de version



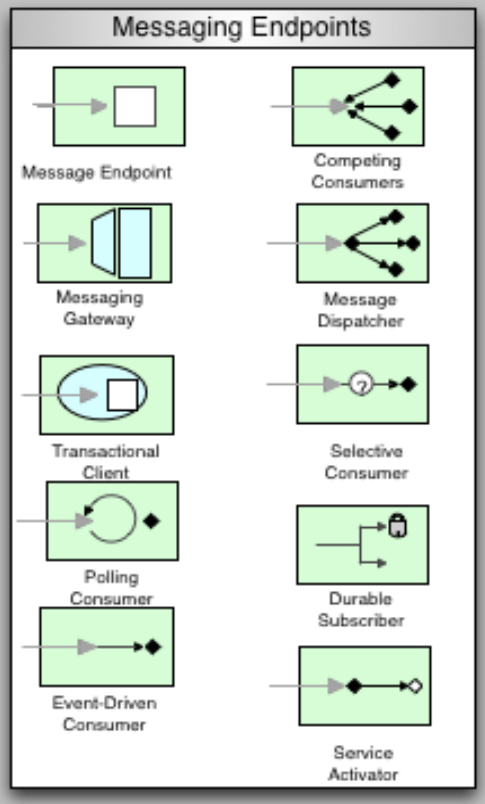
ESB – exemple cas d'utilisation

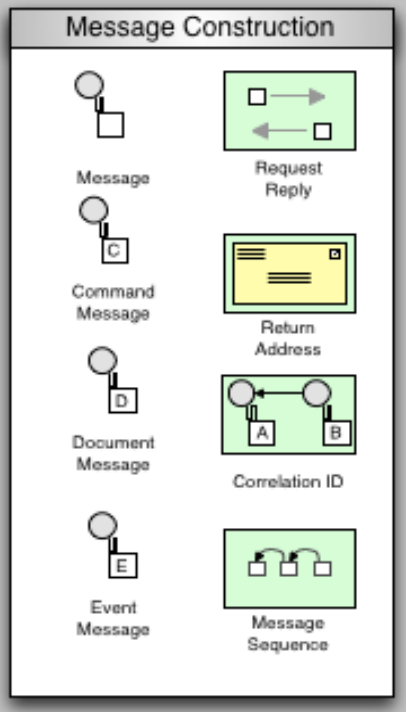
Gestion de la qualité de service (QoS)

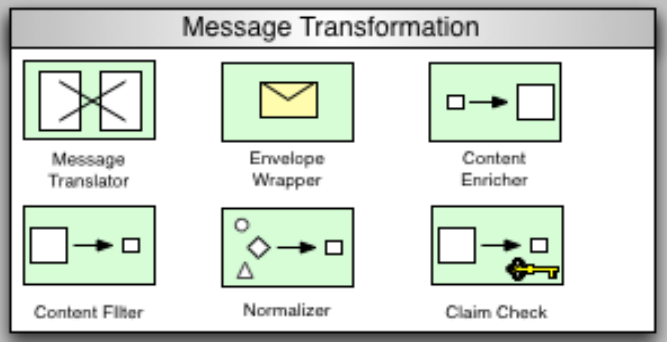


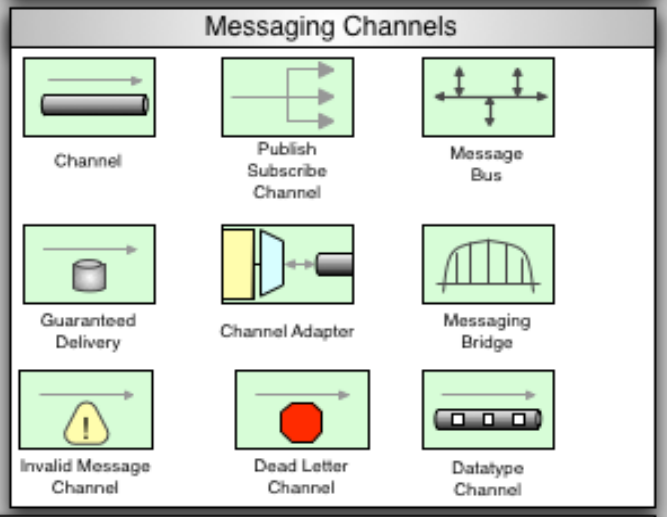
Entreprise Integration Pattern

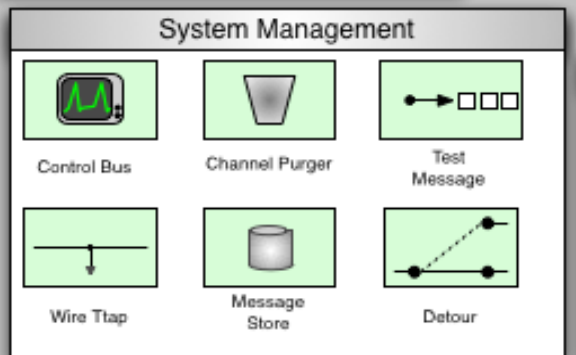
Entreprise Integration Pattern

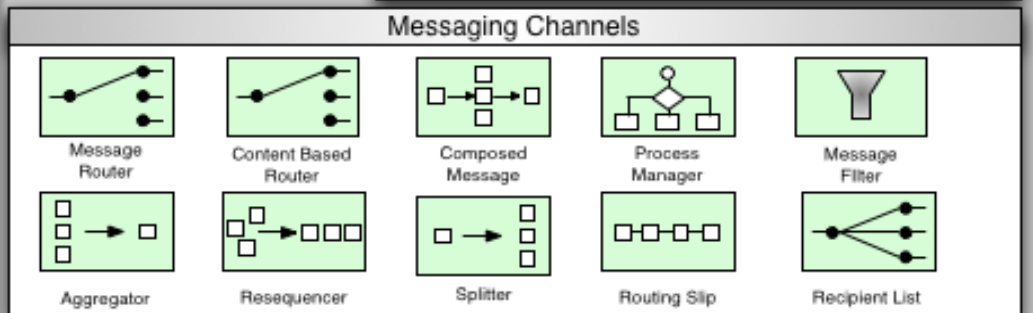












ESB Open source

Au delà de l'ESB

Frameworks d'intégration, encore plus légers que les ESBs

- Exemple : Apache Camel et Spring Integration

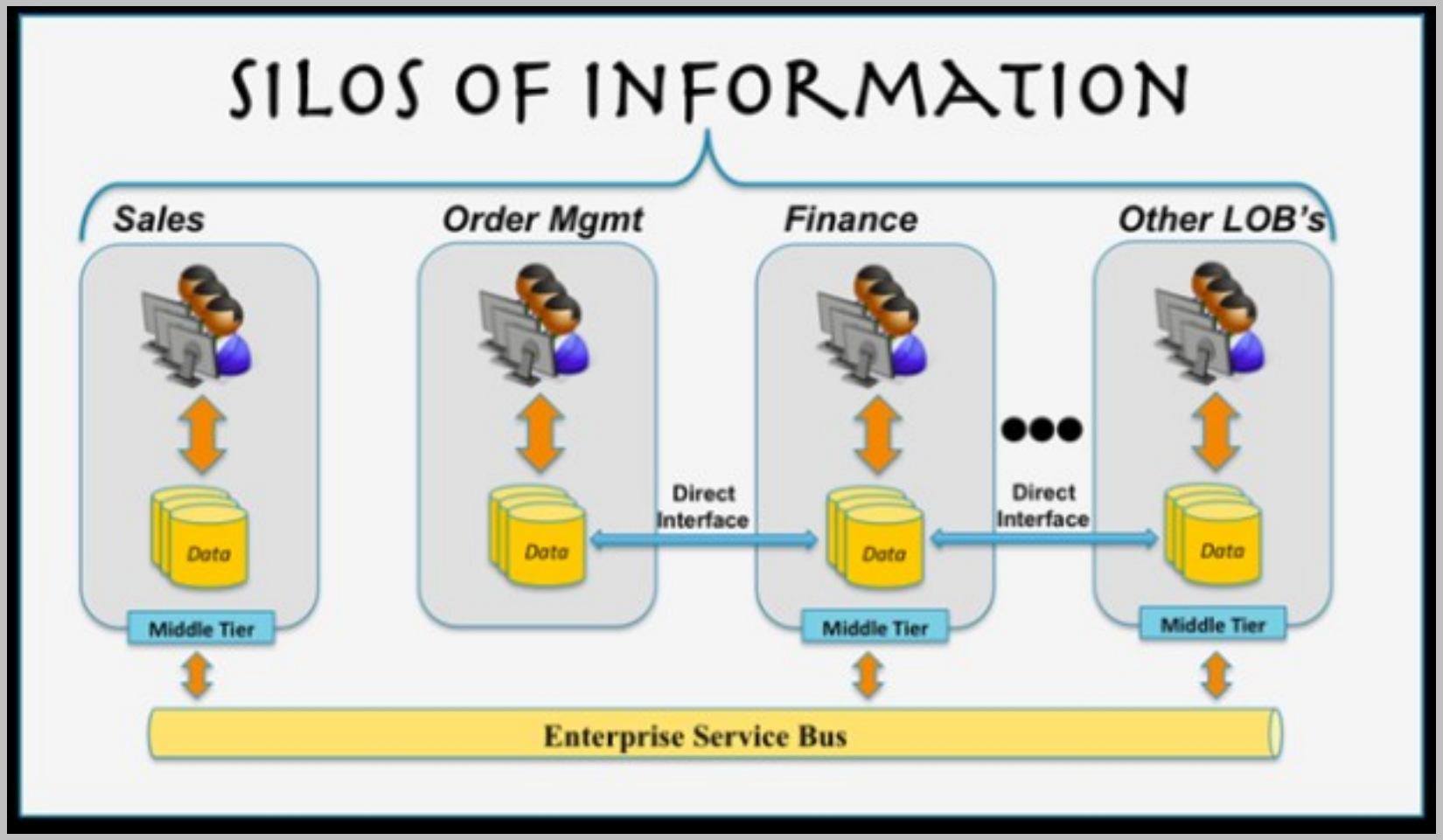
Le web peut être votre middleware d'intégration!

Data ?

Quid des données?

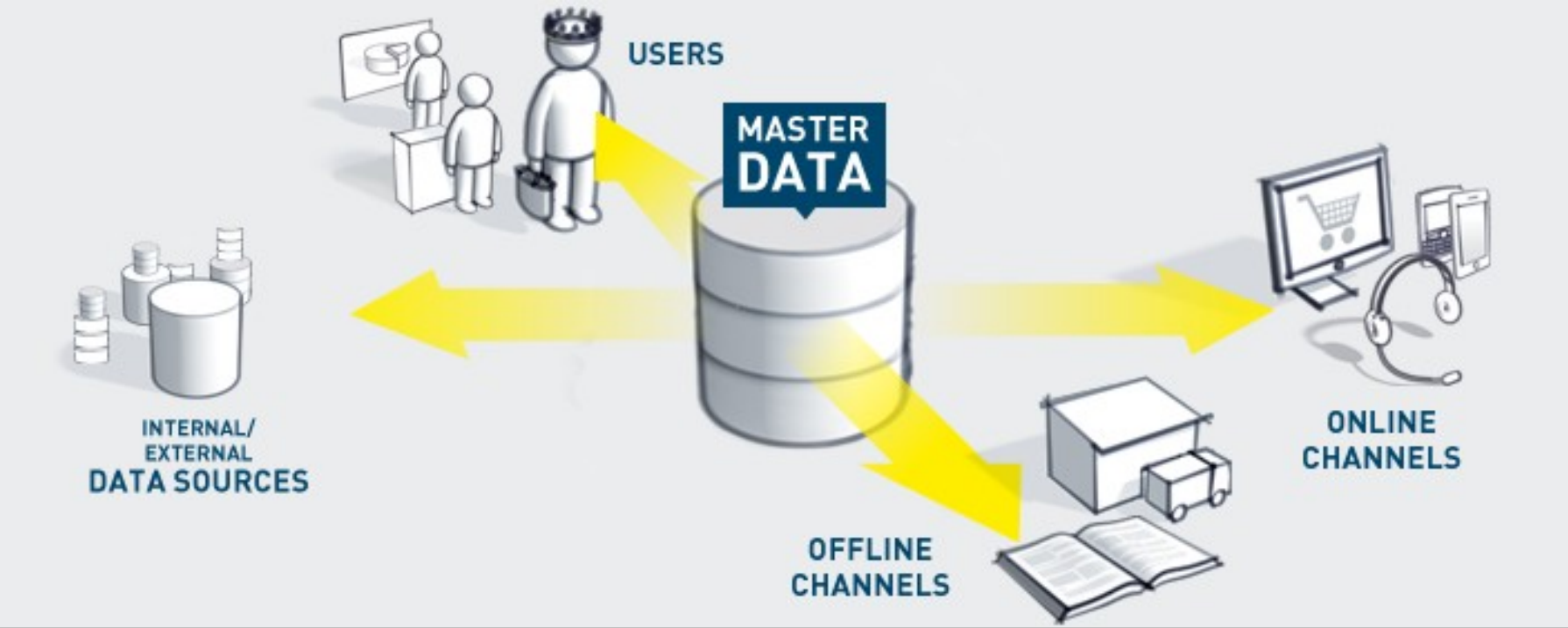
A éviter :
- Duplication
- Interfaces directes entre les données des applications

Data ?

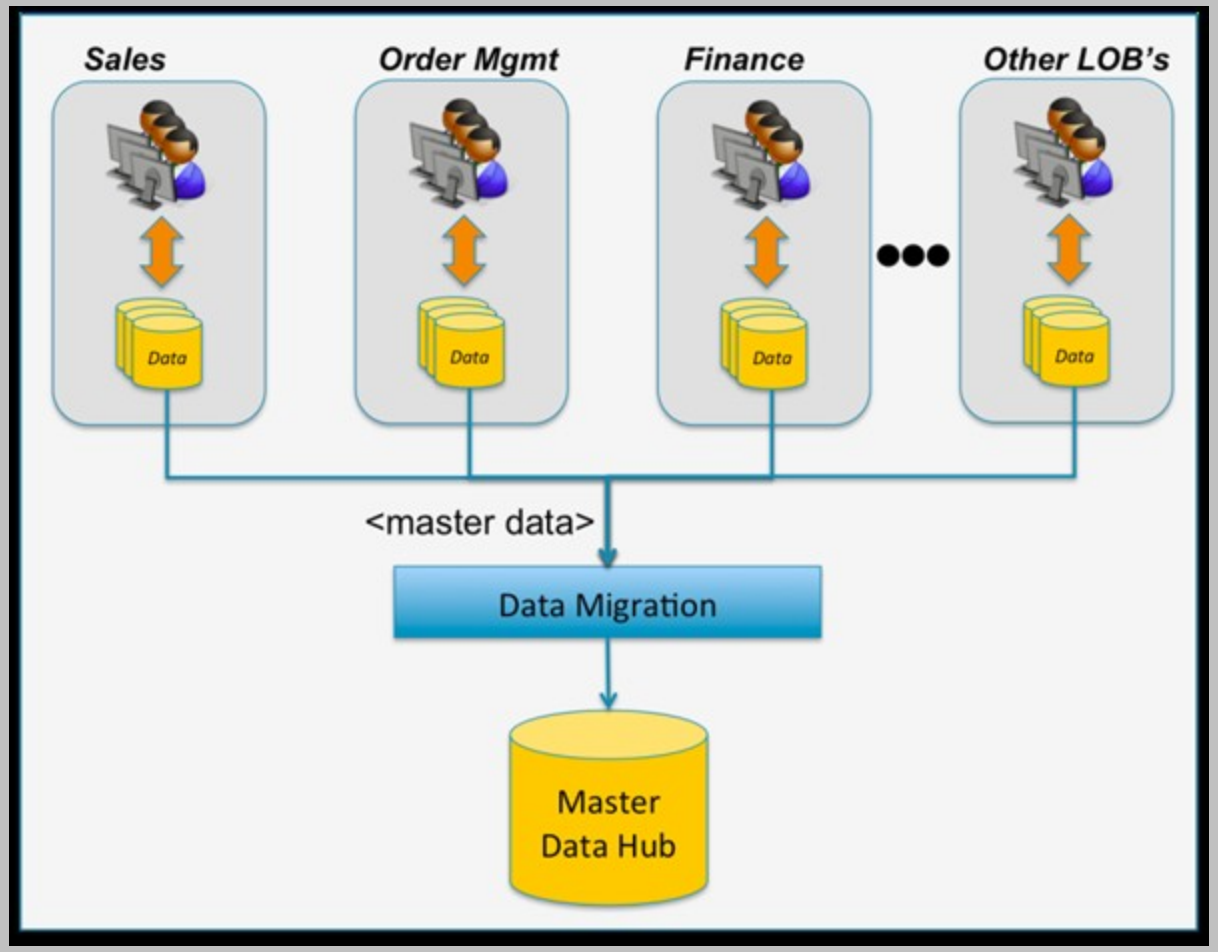

SILOS OF INFORMATION
Sales
Order Mgmt
Finance
Other LOB's
Data
Data
Data
Data
Direct Interface
Direct Interface
Middle Tier
Middle Tier
Middle Tier
Enterprise Service Bus

Les MDM

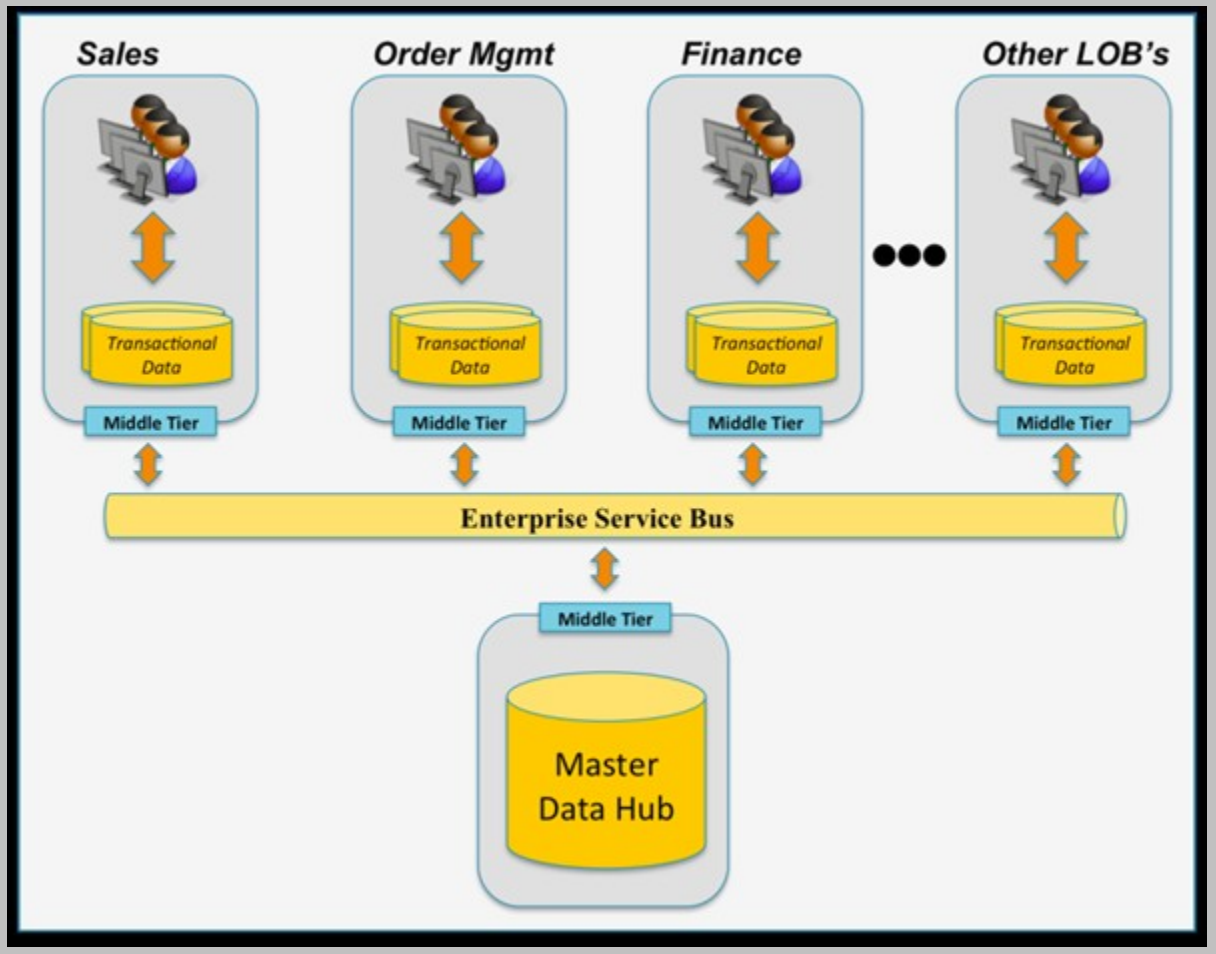
Master Data Management
ou Master Data Hub
Pour gérer des données référentielles



Pour aller vers le MDM



Pour aller vers le MDM



SOA

Objectif : répondre à l'instabilité du SI
En offrant :

- **Agilité** (maintenabilité)
- **Flexibilité** des processus métiers (évolutivité)
- **Rationalisation** des référentiels de données (stabilité, cohérence)

SOA

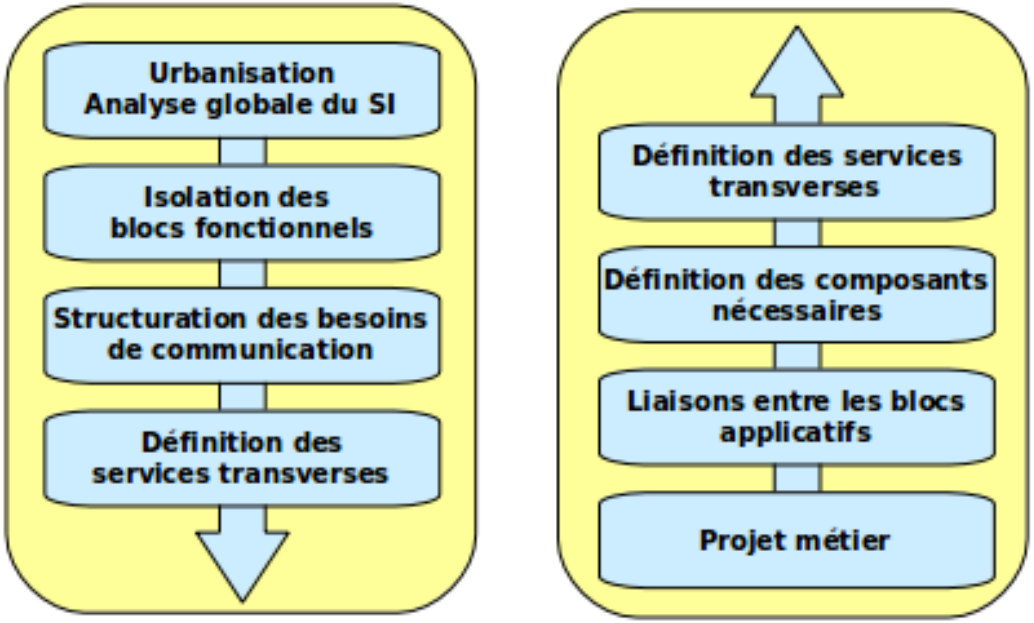
Pourquoi une architecture SOA réussirait-elle là où d'autres ont échoué ?

Les technologies web sont arrivées à maturité
Web Services

Méthodologie / SOA

Approche Top-Down

- Le SI est appréhendé dans sa globalité
- Les services sont identifiés à priori à partir d'une démarche cartographique (processus métiers)



Approche Bottom-Up

- Le SI est vu comme constitué de blocs autonomes et hétérogènes et appréhendé bloc par bloc
- Les services sont identifiés en fonction des besoins projets

Conclusion

Design évolutif

Approche bottom-up en continue, top-down éventuellement au départ

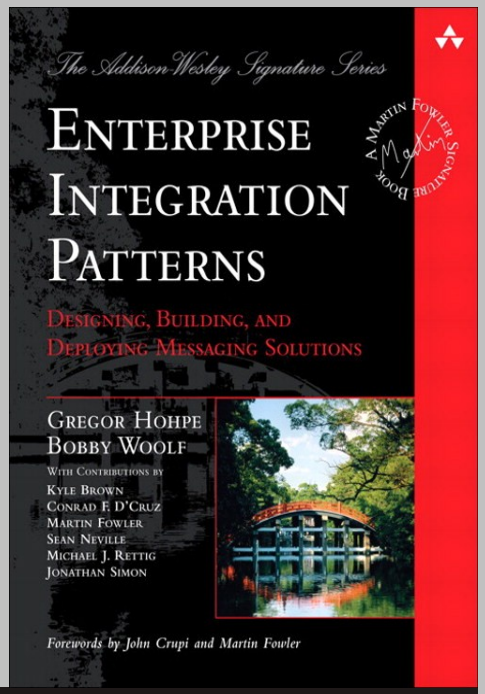
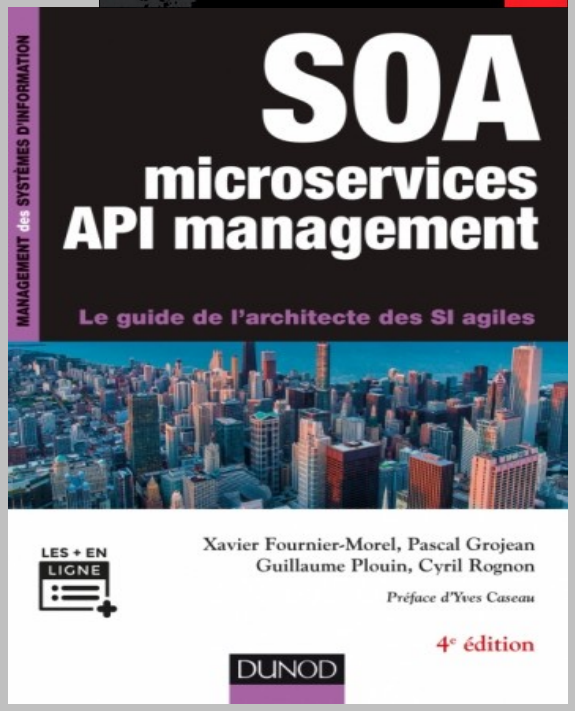
Le contenu d’un ESB doit être en perpétuelle évolution

il sert à donner de l’agilité et de la souplesse au SI pour "éponger" les changements et modérer les impacts

Merci!

Credit photo:
Romain Guy
Cyril Lacote
William Forsythe

Patterns of Enterprise Application Architecture - Martin Fowler – Hardcover



SOA microservices API management
X Fournier Morel - Dunod