

Introduction

Module SOA
A.U 2019-2020

Objectifs du module

- Sensibiliser l'apprenant des défis de l'interopérabilité
- Maîtriser les concepts liés de services web et technologies liées
- Construire et déployer des services web et leurs clients
- Familiariser l'apprenant avec le style d'architecture SOA
- Maîtriser un outil de mise en œuvre SOA



Plan du module



Charge horaire: 30h

Pré-requis : Java, HTML, protocoles web



1ère période

- Services web
- XML/XSD
- WS REST
- WS SOAP



Chapitre 1

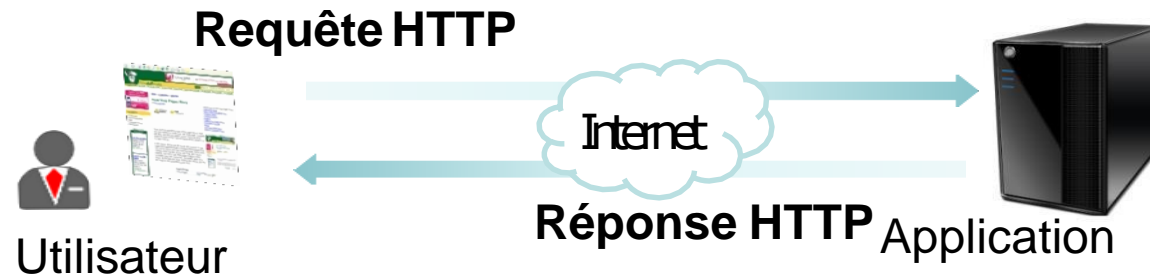
Introduction aux services web

Contexte

Human-centric web

- Le Web centré utilisateur implique que l'humain est l'acteur principal pour l'initialisation de l'ensemble des requêtes

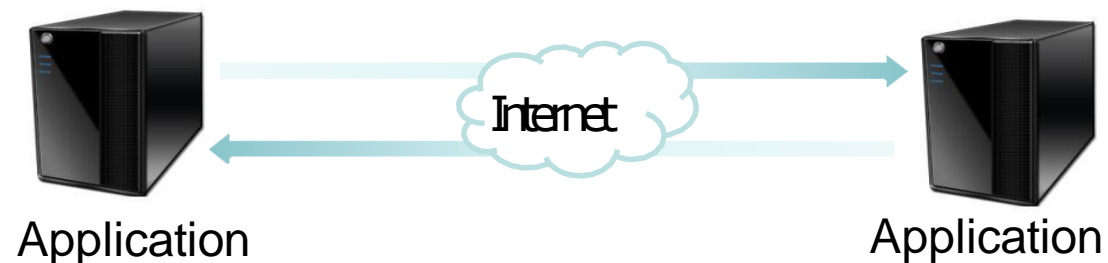
B2C
Business To Consumer



Application-centric web

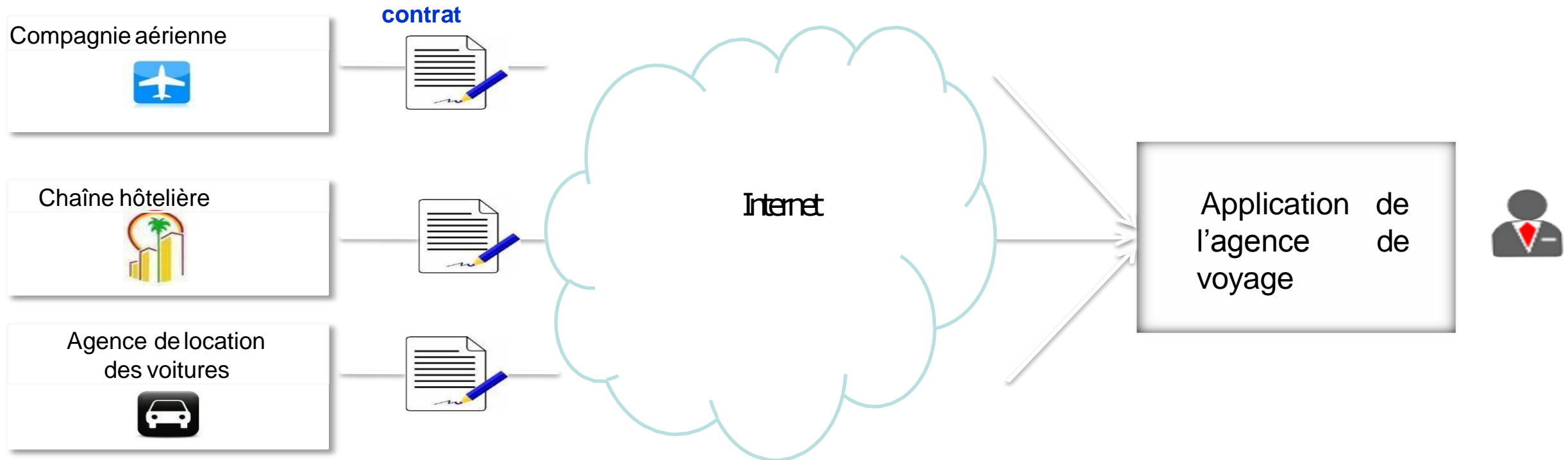
- Le Web centré application a pour objectif de permettre à des applications de différentes organisations de communiquer entre elles

B2B
Business To Business



Contexte

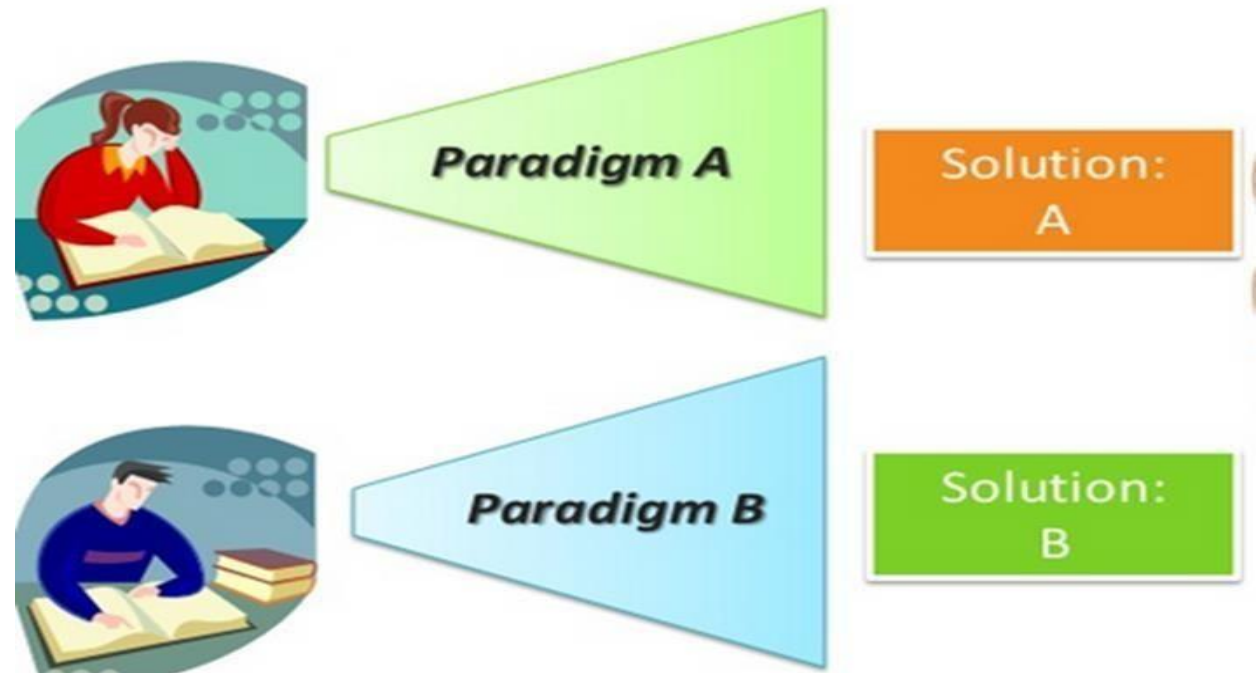
Exemple B2B



Evolution des paradigmes

Le terme de **paradigme** est employé pour exprimer la façon dont un système a été conçu et pensé dans ses grandes lignes. [1]

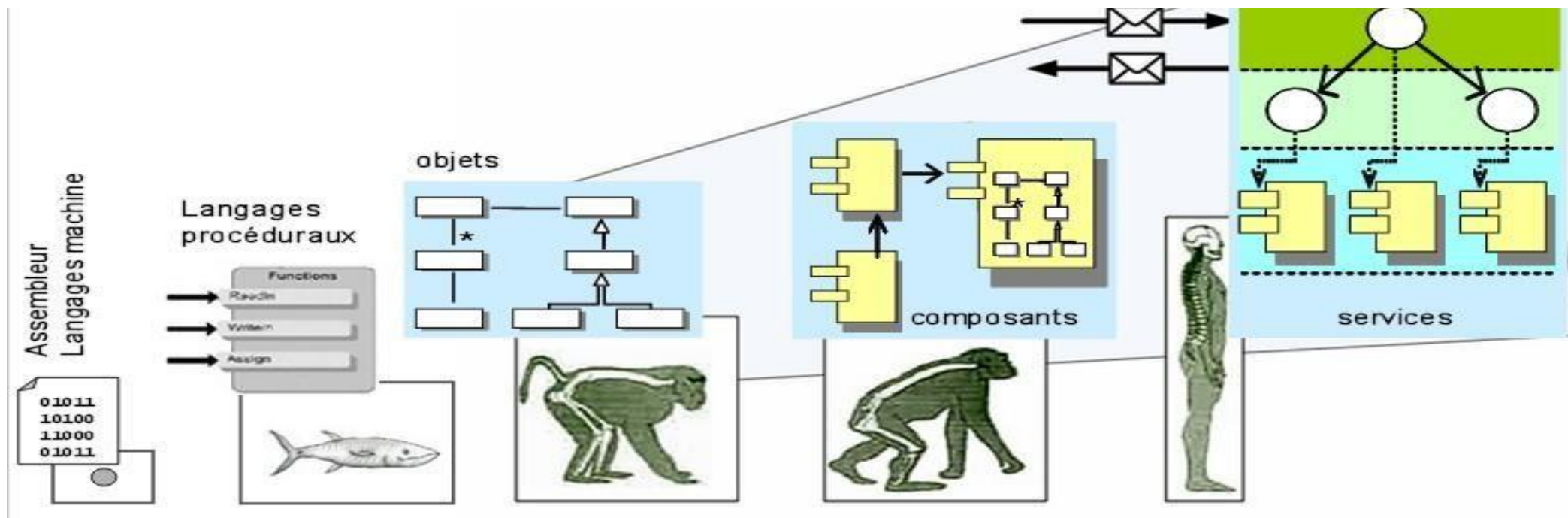
 **Objectif:**
Développer une application de facturation



Evolution des paradigmes



- Les révolutions informatiques coïncident généralement avec un changement de paradigme
- Niveau d'abstraction grandissant avec l'évolution des paradigme



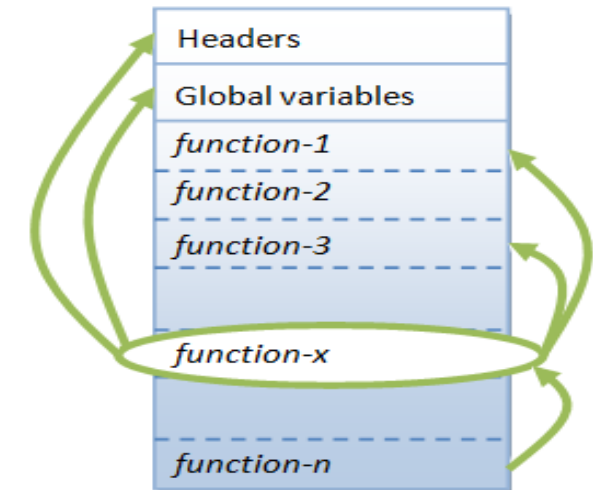
Paradigme procédural



- Le programme est une liste des tâches et des opérations à exécuter.

Limites

- ⊖ Tend à générer du code "Spaghetti "
- ⊖ Maintenance complexe
- ⊖ Modularité et abstraction absente
- ⊖ Réutilisation ardue



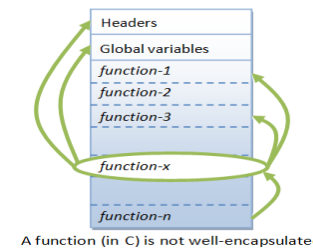
A function (in C) is not well-encapsulated

Paradigme objet

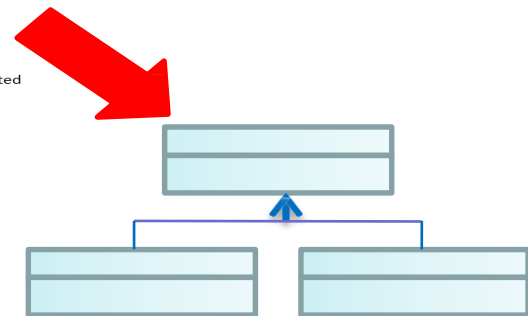
- L'idée est de concevoir les programmes non plus comme des lignes de codes qui s'exécutent séquentiellement, mais comme des **objets qui dialoguent**
- Ses principes incluent l'abstraction encapsulation, de données, polymorphisme et héritage.

Limites

- ⊖ Réutilisation difficile
- ⊖ Couplage fort -> rend difficile la maintenance



A function (in C) is not well-encapsulated



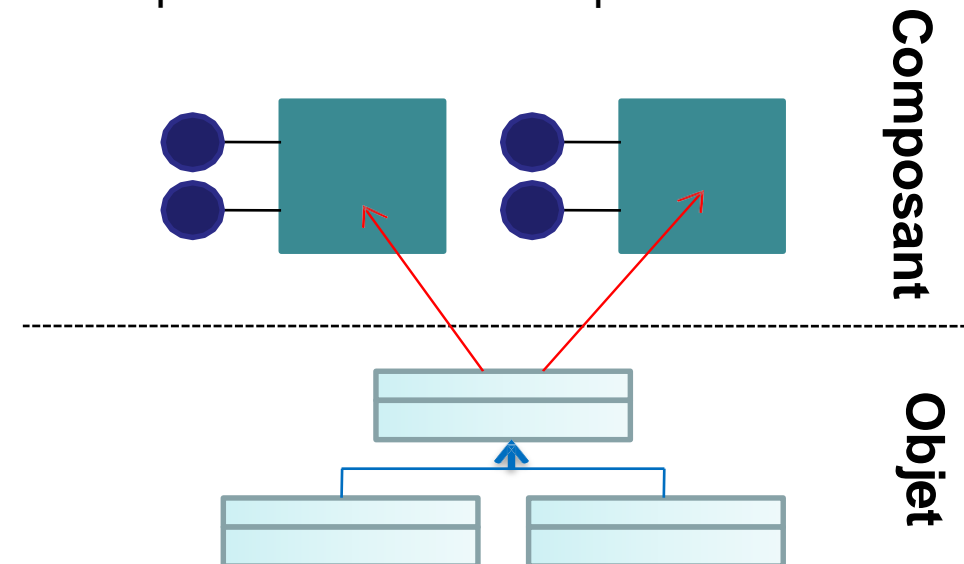
Paradigme composant

- Construire une application composée par un ensemble de briques de base **configurables**
- Il s'agit d'externaliser le code fonctionnel d'une application afin de le rendre **réutilisable** dans d'autres applications.

focaliser l'expertise sur les problèmes "métiers" plutôt que "techniques"

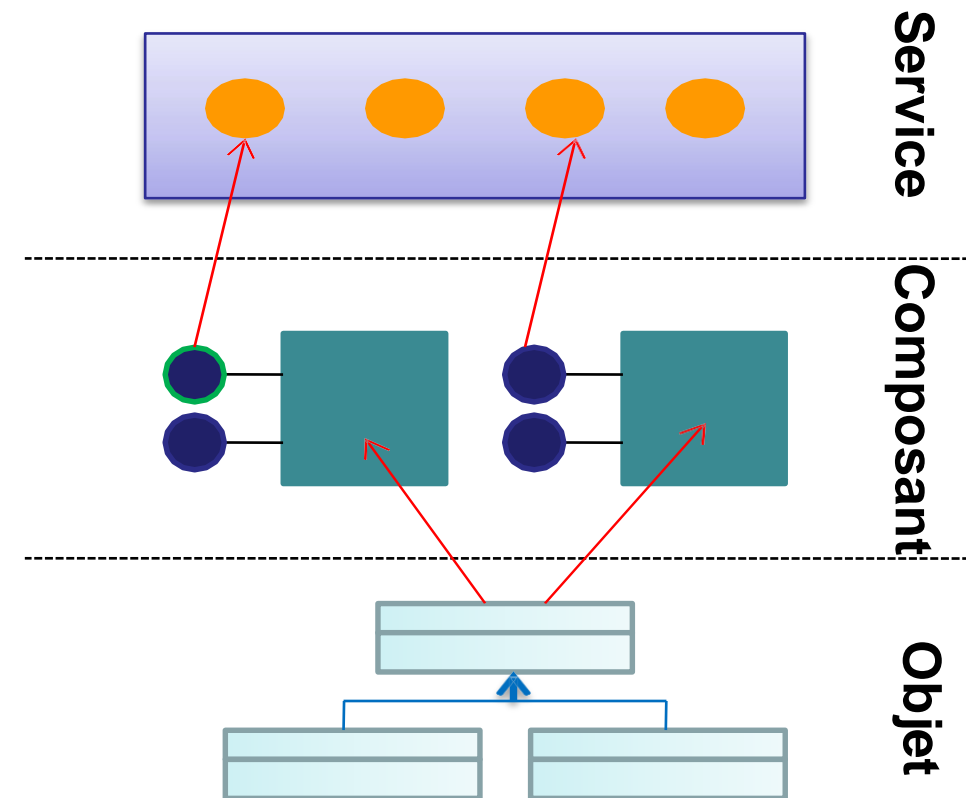
Limites

- ⊖ Interopérabilité entre composants hétérogènes

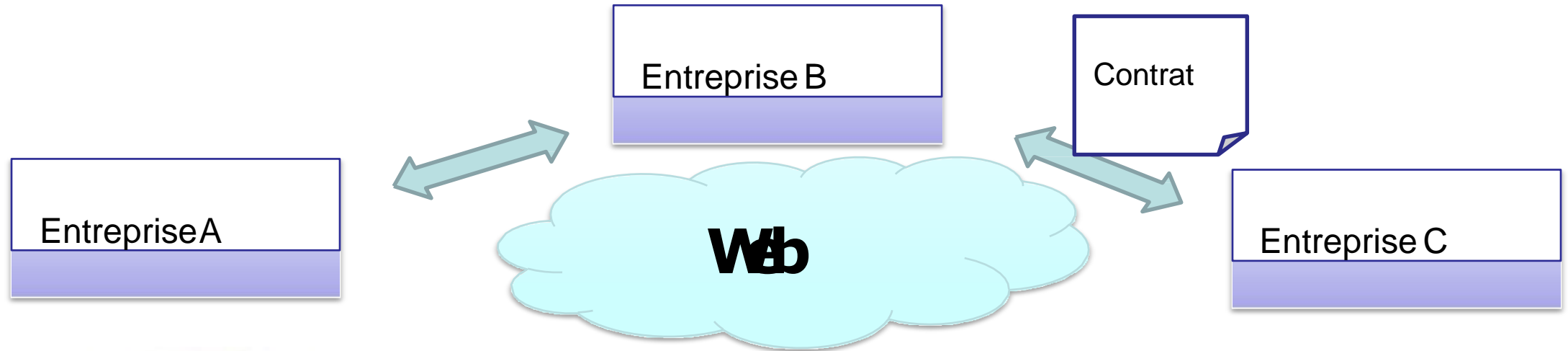


Paradigme service

- Prise en charge de la diversité et de l'hétérogénéité des systèmes logiciels, en termes de langages de programmation, de technologies de conception (et de réalisation) ou de plates-formes d'exécution
- Le paradigme service permet de:
 - réduire le **couplage**
 - améliorer la **réutilisation**
 - augmenter l'**abstraction**



Besoins...



- Langage commun
- Protocole commun
- Contrat
- Middelware



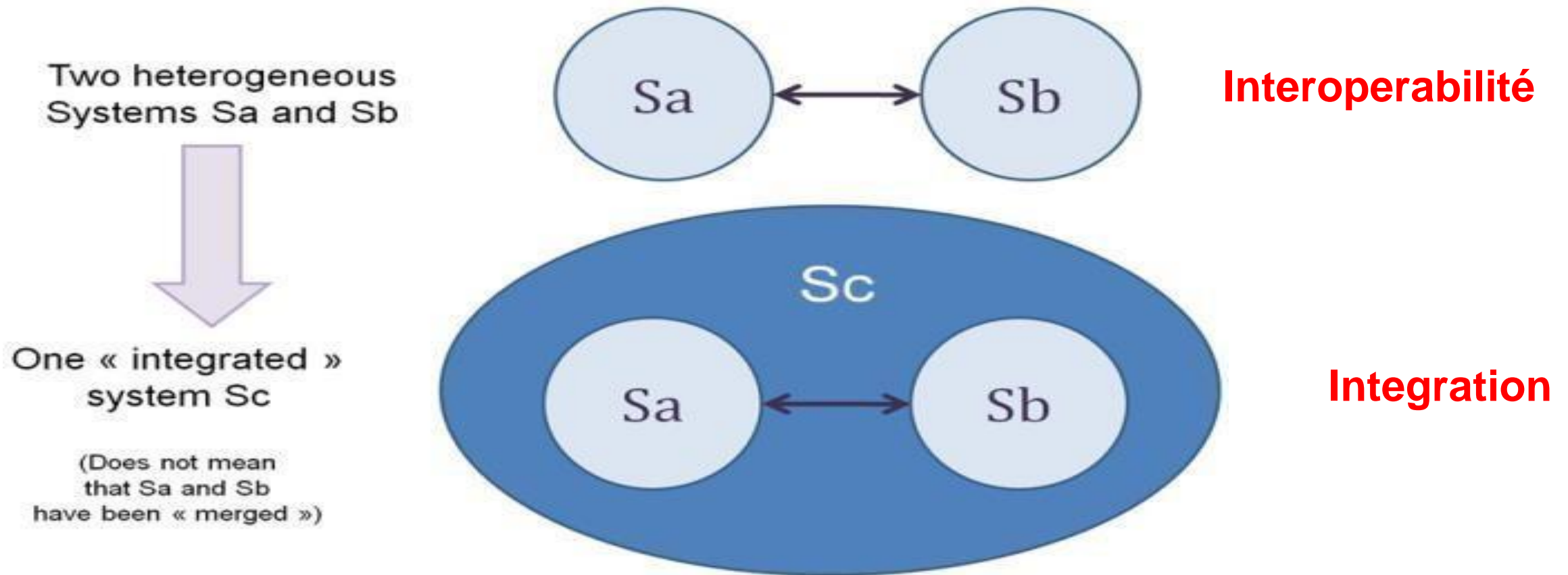


Intégration, Interopérabilité - Quoi?



- **Interoperability** means that two (or more) systems work together unchanged even though they weren't necessarily designed to work together.
- **Integration** means that you've written some custom code to connect two (or more) systems together.
 - *[Bobby Wolf – IBM Architect]*

Intégration, Interopérabilité - Quoi?

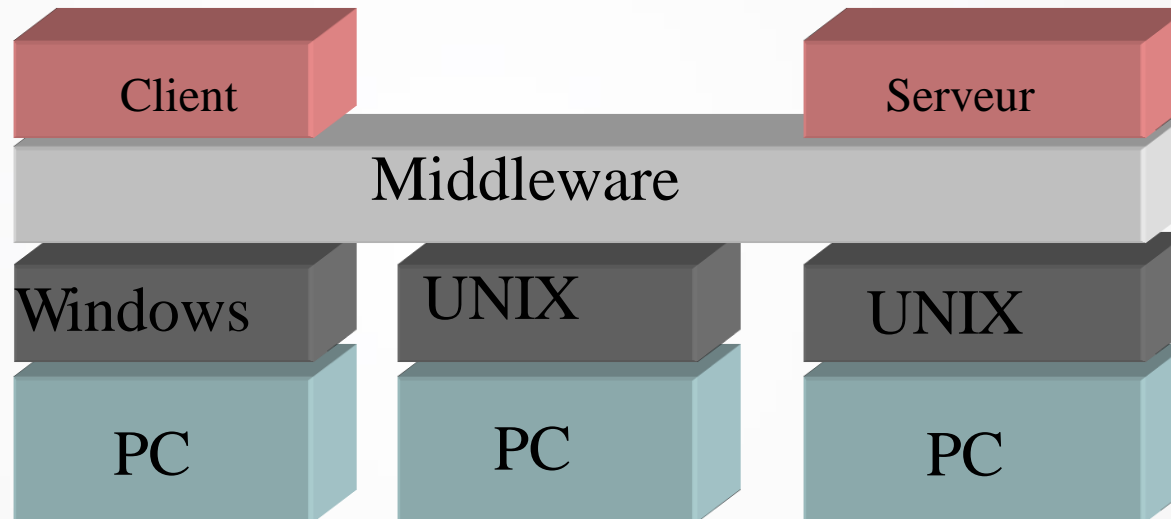


Source: <http://modelseverywhere.wordpress.com/2010/11/04/model-driven-integration/>

Intégration, Interopérabilité - Comment?

Middleware (intergiciel)

- Les logiciels servant d'intermédiaire entre d'autres logiciels; ou
- Un intermédiaire de communication entre des applications complexes et distribuées [3]



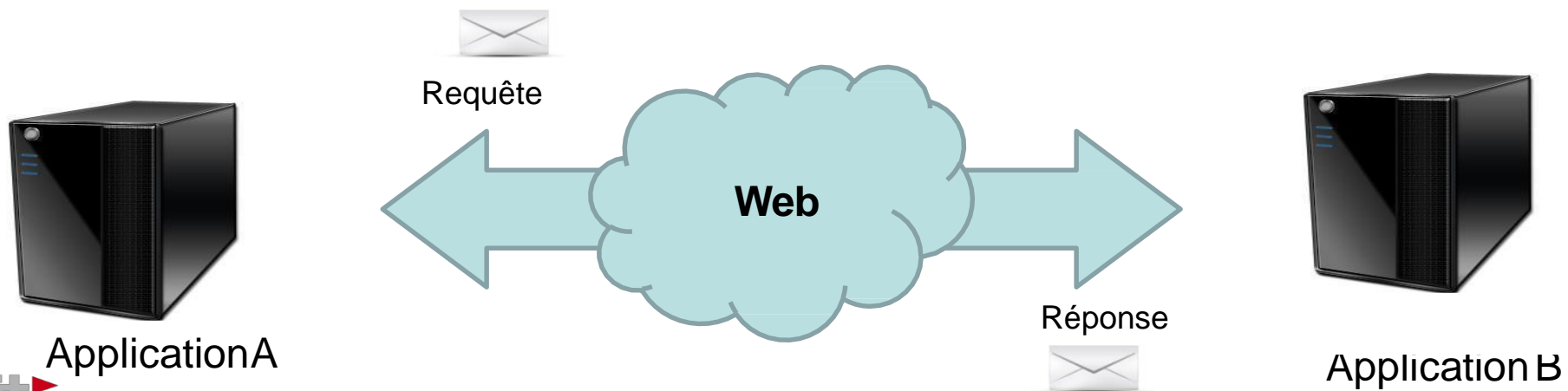
Rôles de base d'un middleware:

- Résoudre l'**interopérabilité** : Unifier l'accès à des machines distantes
- Résoudre l'**hétérogénéité** : Etre indépendant des systèmes d'exploitation et du langage de programmation des applications

Middleware

- Solutions existantes :
 - DCOM,
 - .NET Remoting
 - RMI
 - CORBA
 - ?

Quels sont les atouts d'une meilleure solution ?





Services web



- Service web = service + web
- “A Web service is a software system designed to support interoperable **machine-to-machine** interaction over a **network**”.

W3C – 2004

- Les services Web interagissent à travers l'échanges de messages
- Il existe deux grandes familles de services web:
 - Les services web étendus (SOAP/WSDL)
 - Les services web REST



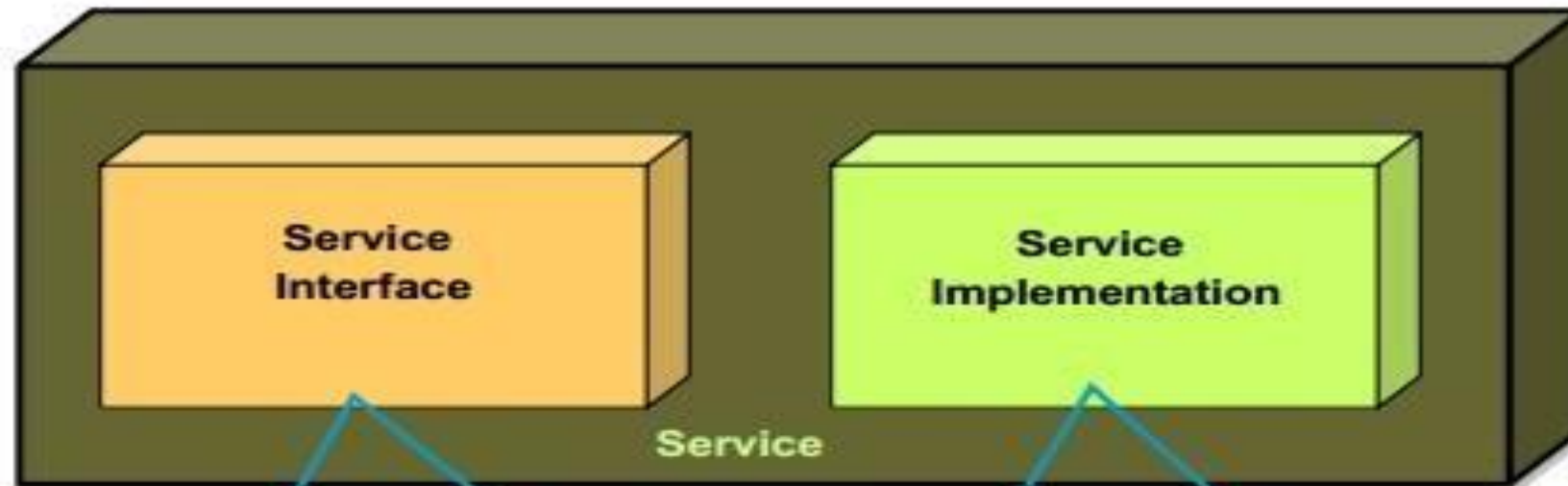
Présentation SOA



- “ L’architecture orientée service constitue un **style d’architecture** basée sur le principe de séparation de l’activité métier en une série **de services**”.
- “ Ces services peuvent **être assemblés et liés** entre eux selon le principe de couplage lâche pour exécuter l’application désirée. ”

Gartner - Septembre 2005

Anatomie d'un service

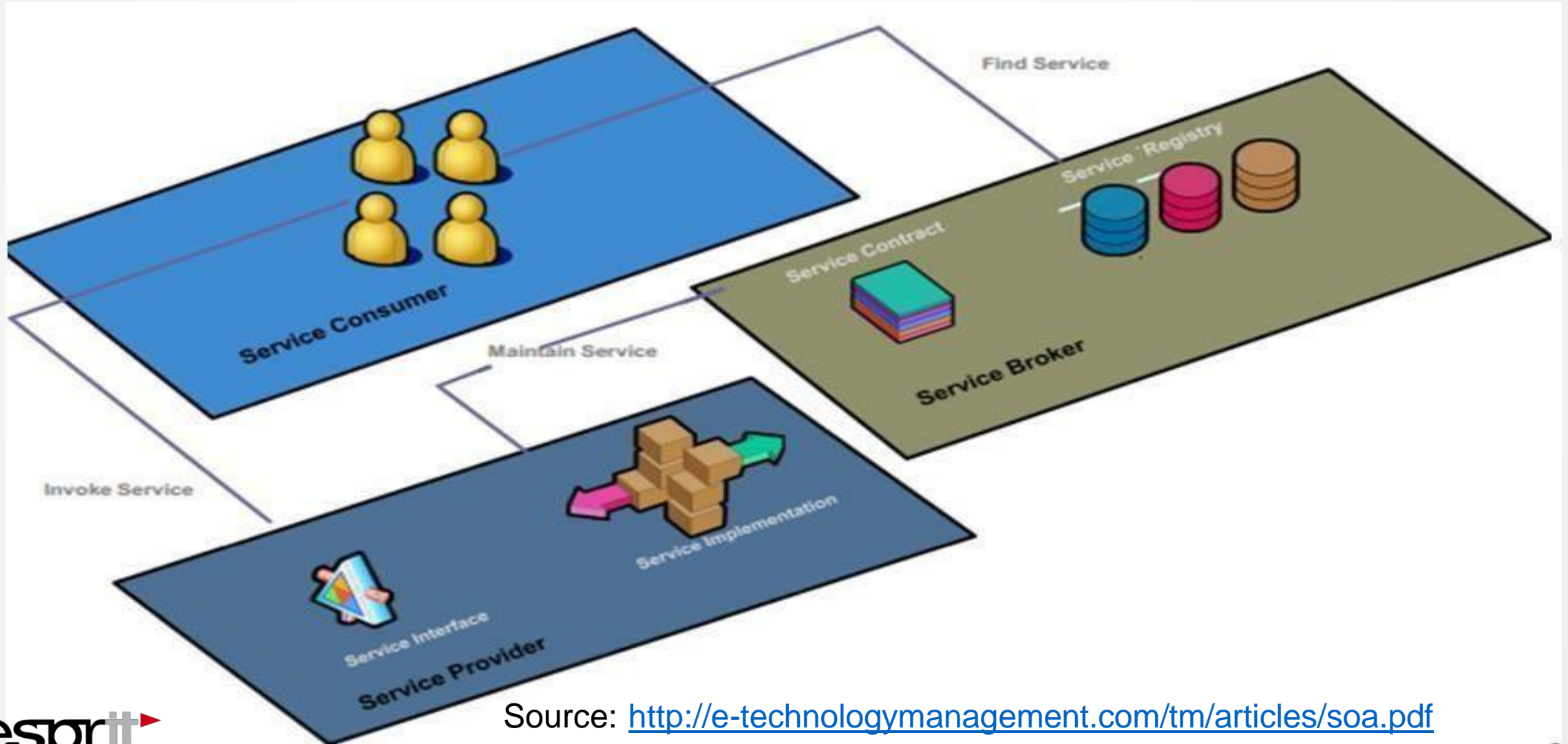


Access layer between the service consumer and service provider. It contains,

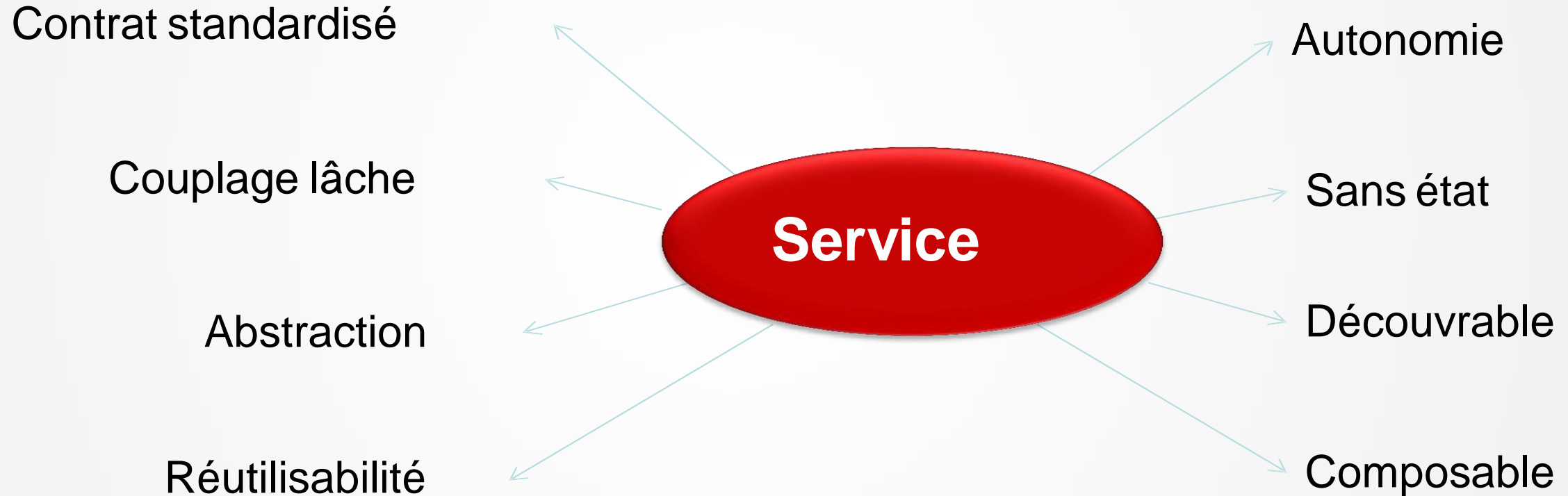
- Service Identity
- Service Input & Output data information
- Service Purpose & Function Metadata

- Contains the core functional or business logic of the service.
- The implementation should be totally transparent to the service consumer, with no knowledge necessary about the implementation specifics.

Eléments SOA



Caractéristiques d'un service

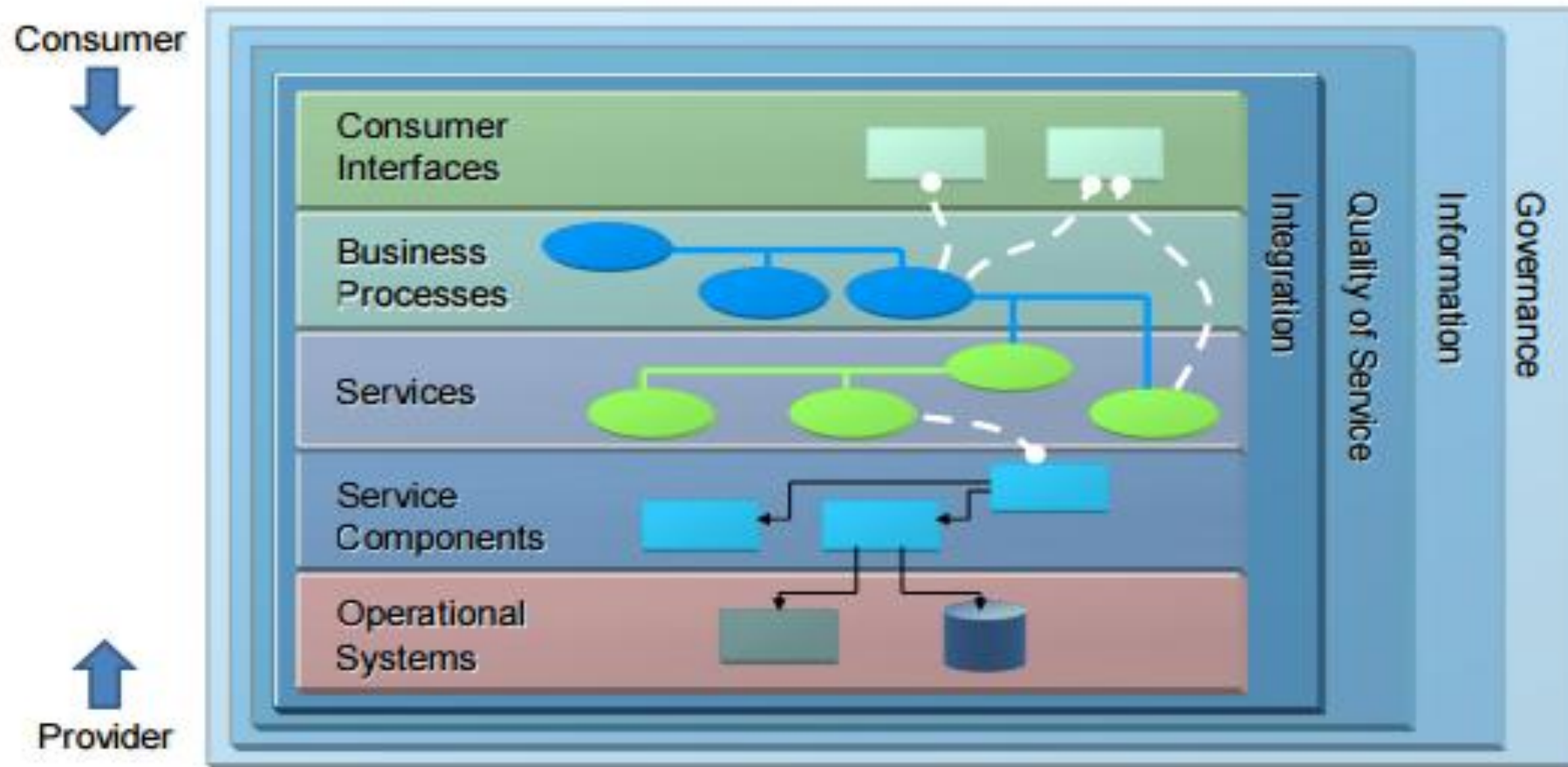


Caractéristiques d'un service

- Contrat standardisé : L'ensemble des services d'un même Système Technique sont exposés au travers de contrats respectant les mêmes règles de standardisation.
- Couplage lâche : Le contrat d'un service doit imposer un couplage lâche de ses clients.
- Abstraction : Le contrat d'un service ne doit contenir que les informations essentielles à son invocation. Un service est vu comme une boîte noire.
- Réutilisabilité : Un service exprime une logique agnostique et peut ainsi être positionné comme une ressource réutilisable.
- Autonomie : Un service ne doit être dépendant d'aucun contexte ou service externe
- Stateless (*sans état*) : Un service doit minimiser la consommation de ressources en déléguant la gestion des informations d'état quand cela est nécessaire.
- Découvrabilité : Un service est complété par un ensemble de métas données de communication au travers desquelles il peut être découvert et interprété de façon effective.
- Composabilité : Un service doit être conçu de façon à participer à des compositions de services.[4]

L'Architecture de référence SOA (SOA-RA)

- **Open Group SOA Reference Architecture (SOA RA)** Standard IBM's advantage
- Crée une abstraction avec une approche indépendante de la technologie
- Aide à définir et exécuter une feuille de route
- Définir un vocabulaire commun pour l'architecture





Références



- 1 <http://fr.wikipedia.org/wiki/Paradigme>
- 2 <http://design-patterns.fr/introduction-a-la-programmation-orientee-objet>
- 3 <http://fr.wikipedia.org/wiki/Middleware>
- 4 <http://blog.xebia.fr/2009/04/29/soa-du-composant-au-service-lautonomie>