

2^{ème} année Master Informatique

Option: **Génie Logiciel et Systèmes Distribués**

Université de Biskra

Année 2021/2022

Architectures Orientées Services

Dr. Mohamed Lamine KERDOUDI

Email : I.kerdoudi@univ-biskra.dz

1. Concept de Service :

- ❑ “A service is a **self-contained, self-describing and modular** piece of software that performs a specific business function such as validating a credit card or generating an invoice”
- ❑ « Un service est un logiciel **autonome, auto-décrivant** et **modulaire** qui exécute une fonction commerciale spécifique telle que la validation d'une carte de crédit ou la génération d'une facture »
 - «**Autonome**» implique des services comprenant tout ce qui est nécessaire pour les faire fonctionner.
 - « **Auto-décrivant** » qu'ils ont des interfaces qui décrivent leurs fonctionnalités d'affaires.
 - « **Modulaire** » : les services peuvent être agrégés pour former des applications plus complexes.
- ❑ C'est l'application de **principes de conception orientée services** qui permettent de distinguer une unité en tant que **service** parmi d'autres unités qui peuvent exister en tant qu'**objets** ou **composants**

2. Concept de Service Web :

Définition du W3C (<http://www.w3.org/TR/ws-gloss/>) :

Un service web est un système logiciel identifié par un URI, dont les **interfaces** publiques et les bindings sont définies et **décrites en XML**. Sa définition peut être **découverte** [dynamiquement] par d'autres systèmes logiciels. Ces autres systèmes peuvent ensuite interagir avec le service web d'une façon **décrite** par sa définition, en utilisant des messages XML transportés par des protocoles Internet.

Traduction

- ❑ **Système logiciel : programme**
- ❑ **URI : Uniform Resource Identifier** (chaîne de caractères identifiant une ressource sur un réseau)
 - **Exemples :**
 - URL : `http://www.w3.org/`
 - mail : `mailto:getlost@nospam.org`
 - FTP : `ftp://ftp.ufrmd.dauphine.fr/pub/docs/`
- ❑ **Interface : une description des opérations proposées par le composant logiciel** (même esprit que les interfaces Java)
- ❑ **Incarnation (binding) : spécification du protocole et du format des données utilisés pour échanger des messages en vue de l'utilisation d'une interface**
- ❑ **découverte (dynamique) : obtention de la description d'un service web**

Traduction (-cdt-)

❑ XML : eXtensible Markup Langage

- **pré-requis indispensable** pour faire des services web

❑ protocoles internet :

- bas niveau TCP/IP
- haut niveau (applicatif) :
 - HTTP (web)
 - SMTP (mail)
 - FTP (file)
 - etc.

2. Service Oriented Architecture

- Un service web est donc un programme
 - décrit en XML
 - identifié par un URI
- proposant diverses fonctionnalités que d'autres programmes peuvent
 - ❖ découvrir dynamiquement
 - ❖ et utiliser grâce à des protocoles décrits en XML
 - basés sur l'échange de messages
 - écrits en XML
 - transmis par HTTP, FTP, SMTP, etc.

Une vision plus simple

- ❑ Un service web est *un programme accessible par internet par l'intermédiaire de messages XML transmis par HTTP*
- ❑ Un standard pour les systèmes répartis basés sur XML+HTTP : XML-RPC (Remote Procedure Call):
 - XML-RPC est une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau.
 - XML-RPC permet d'appeler une fonction sur un serveur distant à partir de n'importe quel système (Windows, Mac OS X, GNU/Linux) et avec n'importe quel langage de programmation.
 - Le serveur est lui-même sur n'importe quel système et est programmé dans n'importe quel langage.
 - Les processus d'invocation à distance utilisent le protocole HTTP pour le transfert des données et la norme XML pour la structuration des données.

Exemples de services existants

Google

<http://www.google.com/apis/>

- Accès gratuit mais limité (1000 requêtes par jour après enregistrement)
- Propose trois opérations :
 - recherche
 - obtention d'une page depuis le cache
 - correction orthographique

Amazon

<http://associates.amazon.com/exec/panama/associates/join/developer/resources>

- Accès gratuit mais limité (1 requête par seconde après enregistrement)
- Propose recherche et gestion d'un panier d'achats

2. Service Oriented Architecture

- ❑ Les services sont définis comme étant basés sur des **standards** et ils sont **indépendants de la plate-forme et des protocoles** afin d'assurer des interactions dans des environnements hétérogènes
- ❑ Un **service** peut fournir une **collection de fonctionnalités**:
 - Elles sont regroupées parce qu'elles sont reliées à un **contexte fonctionnel** établi par le service et les besoins de l'entreprise.
 - Par exemple, expédition (**Shipment**).



2. Principes de conception orientée services

Huit principes sont définis par Thomas Erl comme suit:

- **Contrat de service standard:**

- Les services expriment leur but, leurs fonctionnalités, et leurs types de données via un contrat de service.
- Il adhère à un accord de communication standard, qui est défini collectivement par un ou plusieurs documents de description de service.

- **Couplage faible du service** (niveau de dépendance):

- Une relation spécifique à l'intérieur et à l'extérieur du service, en mettant l'accent sur la réduction des dépendances entre:

le **contrat de service**, son **implémentation** et ses **consommateurs**.

2. Principes de conception orientée services

- **Abstraction de service**: les contrats de service ne contiennent que des informations essentielles et l'information sur les services est limitée à ce qui est publié dans les contrats de services.
- **Réutilisation des services**: les services sont conçus pour être réutilisés. Ces services réutilisables sont conçus de manière à ce que leur solution logique soit **indépendante** d'un **processus métier** ou d'une **technologie** particulière.
- **Autonomie des services**: les services exercent un niveau élevé de contrôle sur leur environnement d'exécution sous-jacent.

2. Service Oriented Architecture

- **Services sans états**: La gestion d'informations de l'état peut compromettre la disponibilité d'un service et compromettre son potentiel d'évolution. Les services sont donc idéalement conçus pour prendre un état sauf si cela est vraiment nécessaire.
- ➔ réduction des ressources consommées par un service.
- **Possibilité de découverte de service**: les services sont complétés par des méta-données communicatives par lesquelles ils peuvent être efficacement découverts et interprétés.
 - **Composabilité de service**: les services sont des participants efficaces de la composition, quelle que soit la taille et la complexité de la composition.

But fondamental de SOA

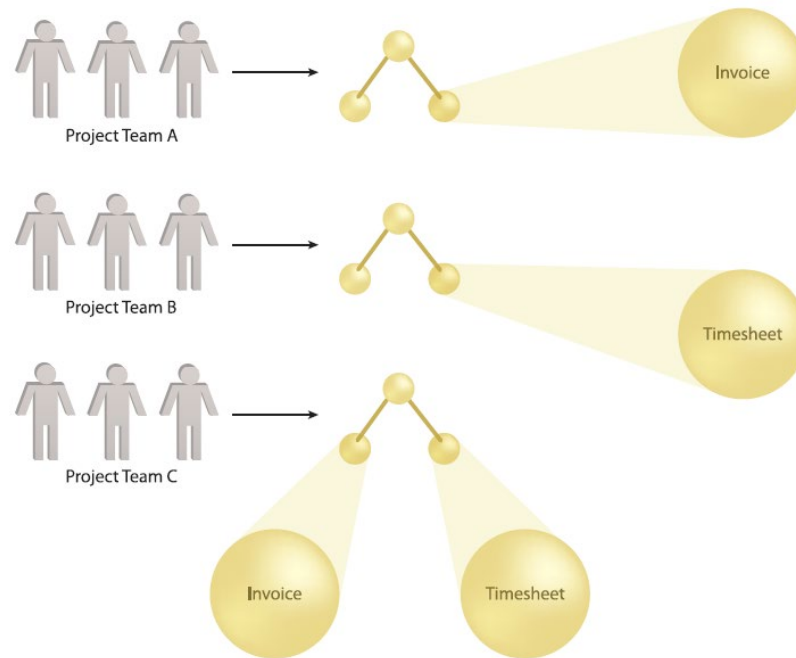
➔ **Augmentation de l'interopérabilité :**

❑ Elle représente un objectif fondamental de l'approche orientée services qui établit une base pour la réalisation d'autres **objectifs et bénéfices stratégiques**.

- La **standardisation des contrats**, **l'évolutivité**, et la **fiabilité** sont quelques-unes des caractéristiques de conception nécessaires pour faciliter l'interopérabilité.
- L'interopérabilité se réfère à l'échange transparent d'informations entre des services disparates.
- L'intégration peut être considérée comme un processus qui permet l'interopérabilité.

2. Service Oriented Architecture

Exemple



- ➔ Les services sont conçus pour être intrinsèquement interopérables quel que soit le moment et le but pour lequel ils sont livrés.
- ➔ L'interopérabilité des '**Services de Facture**' et de '**Emploi de Temps**' fournis par les équipes de **projet A et B**, leur permet de se combiner dans une nouvelle composition de service par l'équipe de **projet C**.

Questions ??