

Web Services

Dr Thouraya Daouas



QU'EST CE QU'UN WEB SERVICE?

Dr Thouraya Daouas

2

fppt.com

WS: Qu'est ce que c'est ?



Les services web permettent à différentes applications écrites dans des langages de programmation différents de communiquer entre elles.

Applications modernes développées dans:
Java, Net, Angular JS, Node.js, etc.

Il peut être difficile d'assurer la communication entre ces applications.

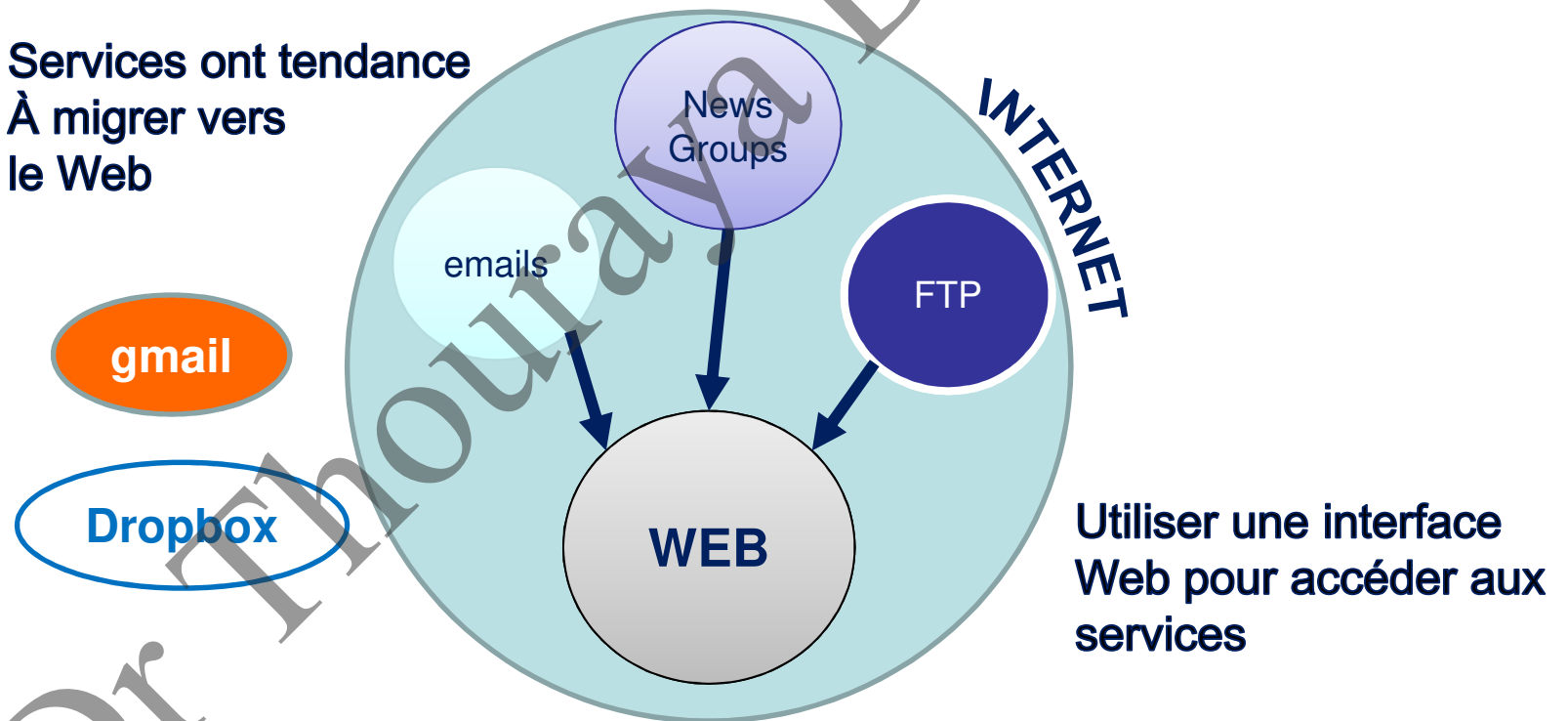
Utilisation des Web Services

Web et Internet: Confusion !

INTERNET \neq WEB

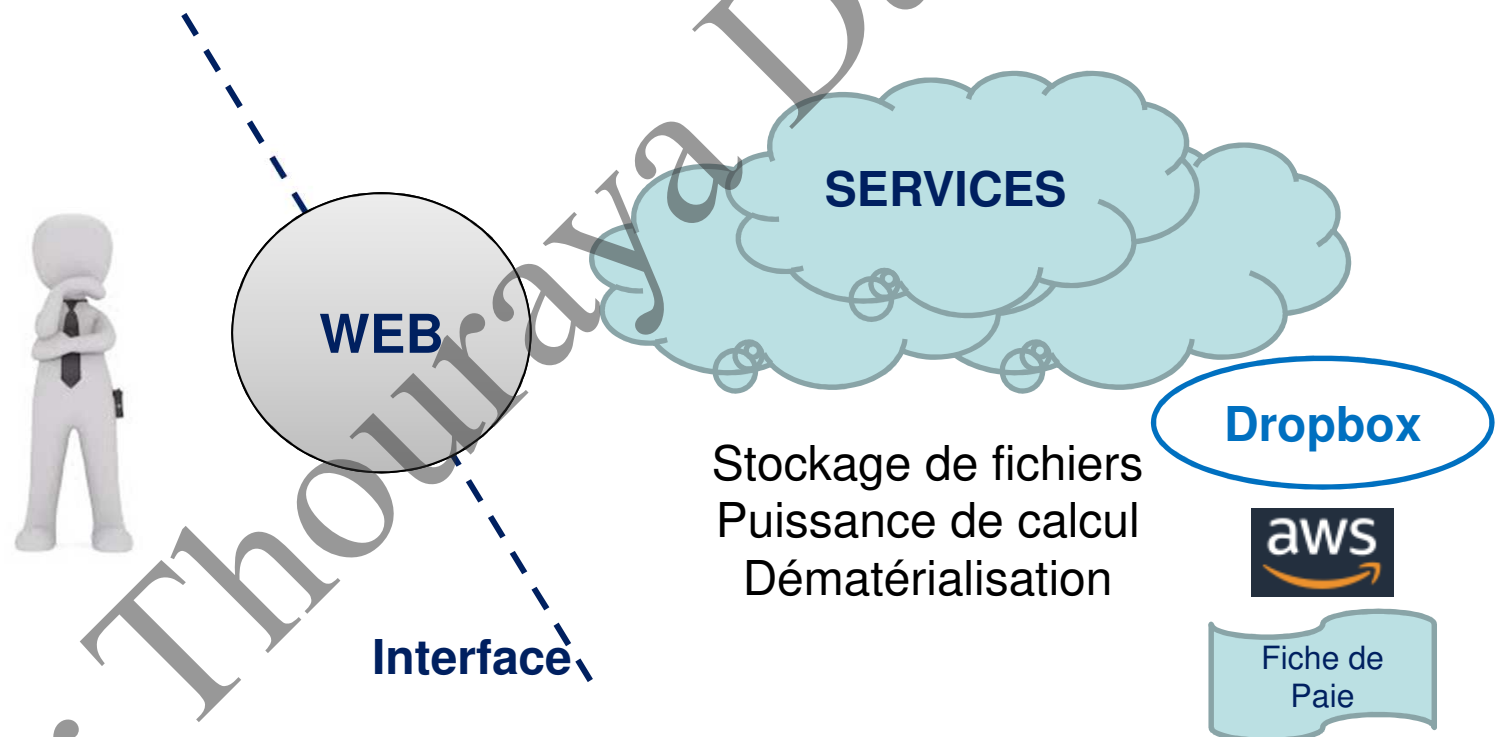
Web une partie d'Internet

Services ont tendance
À migrer vers
le Web



Et le Cloud ?

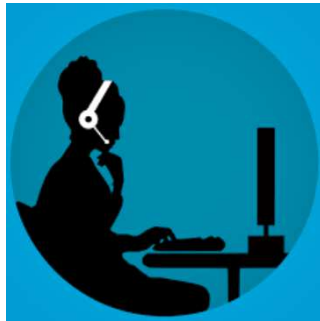
Accès à des services via
une interface Web.



Service Web ?

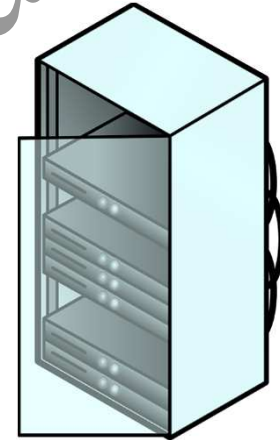
- Fournit une plateforme commune permettant à de multiples applications développées avec différents langages de programmation **de communiquer entre elles.**
- Médium standardisé permettant la **communication entre les applications clients et serveur sur le World Wide Web.**
- Module logiciel conçu pour effectuer certaines tâches.

Comment ça fonctionne ?



Requête appel de SW
Appel de procédure distante

Remote Procedure Call



.NET
JAVA

Fichier XML: Extensible
Markup Language

**Message SOAP: Simple
Object Access Protocol**




Prix produits
vendus en ligne

**Interface
Frontend**

Pour quels avantages ?

- Permettre aux applications en différents langages de **communiquer entre elles**.
- Permettre une **interopérabilité entre les applications**.
- Permettre **d'accéder à des fonctionnalités** via internet.
- Utilisation d'un **protocole industriel standardisé** pour la communication.
- Réduire les **coûts des communications**



ARCHITECTURE D'UN WS?

Dr Thouraya Daouas

Architecture du WS ?

- WS reprennent la plupart des idées et des principes du Web (HTTP, XML) les appliquent à des interactions entre machines.
- WS communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune.
- WS conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante.
- Technologies utilisées : HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

Architecture du WS : REST

- ***Representational State Transfer.***
- Élaborée en 2000 par *Roy Fielding*: Un des créateurs du protocole HTTP, du serveur Apache HTTP entre autres.
- REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web.

Architecture du WS : XML-RPC

- ***Remote Procedure Call.***
- Protocole simple utilisant XML effectue des messages RPC.
- Requêtes écrites en XML et envoyées via HTTP POST.
- Requêtes intégrées dans le corps de la réponse HTTP.
- XML-RPC indépendant de la plate-forme, donc communique avec diverses applications.
- **Ex:** *Client Java peut parler de XML-RPC à un PerlServer.*

Architecture du WS : SOAP

- ***Simple object Access Protocol.***
- Protocole standard de communication.
- Protocole décrit en XML et standardisé par le W3C.
- Présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes.
- Circule sur le protocole HTTP et permet d'effectuer des appels de méthodes à distance.

Architecture du WS : WSDL

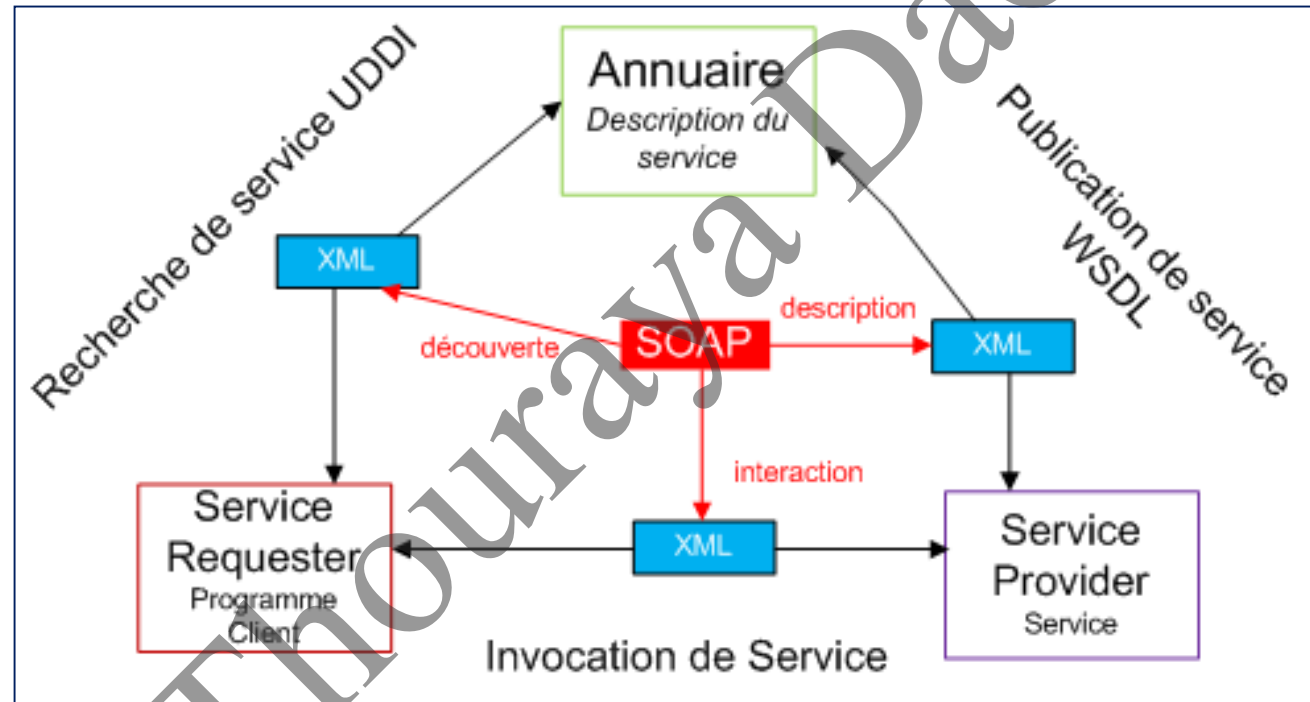
- ***Web Services Description Language.***
- Langage de description standard.
- Interface présentée aux utilisateurs.
- Indique comment utiliser le service Web et comment interagir avec lui.
- Basé sur XML et permet de décrire de façon précise les détails concernant le service Web :
 - Protocoles, Ports utilisés, Opérations pouvant être effectuées, Formats des messages d'entrée et de sortie et Exceptions pouvant être envoyées.

Architecture du WS : UDDI

- ***Universal Description, Discovery and Integration.***
- Annuaire de services fournit l'infrastructure de base publication et découverte des services Web.
- Permet aux fournisseurs de présenter leurs WS aux clients.
- Contient 3 types d'informations :
 - Pages blanches : incluent adresse, contact et identifiants relatifs au WS.
 - Pages jaunes : identifient secteurs d'affaires relatifs au service Web.
 - Pages vertes : donnent informations techniques.

Fonctionnement des WS ?

Registre fournit un endroit central où les programmeurs peuvent publier de nouveaux services ou en trouver.



Consommateur du WS. Utilise un WS existant en ouvrant une connexion réseau et en envoyant une demande en XML.

Met en application le service Web et le rend disponible sur Internet.

WS en couches ?

Découverte des services	UDDI	Centraliser les services dans un registre commun. Simplifier les fonctionnalités de recherche et de publication.
Description du service	WSDL	Description de l'interface publique du WS avec WSDL
Communication	SOAP	Formatage des données échangées. D'un côté protocole RPC de l'autre côté REST.
Transport	HTTP	Transport des messages XML échangés. Inclut HTTP, SMTP, FTP, et un nouveau tel que BEEP.



SOAP PROTOCOLE DE COMMUNICATION ?

Dr Thouraya Daouas

18

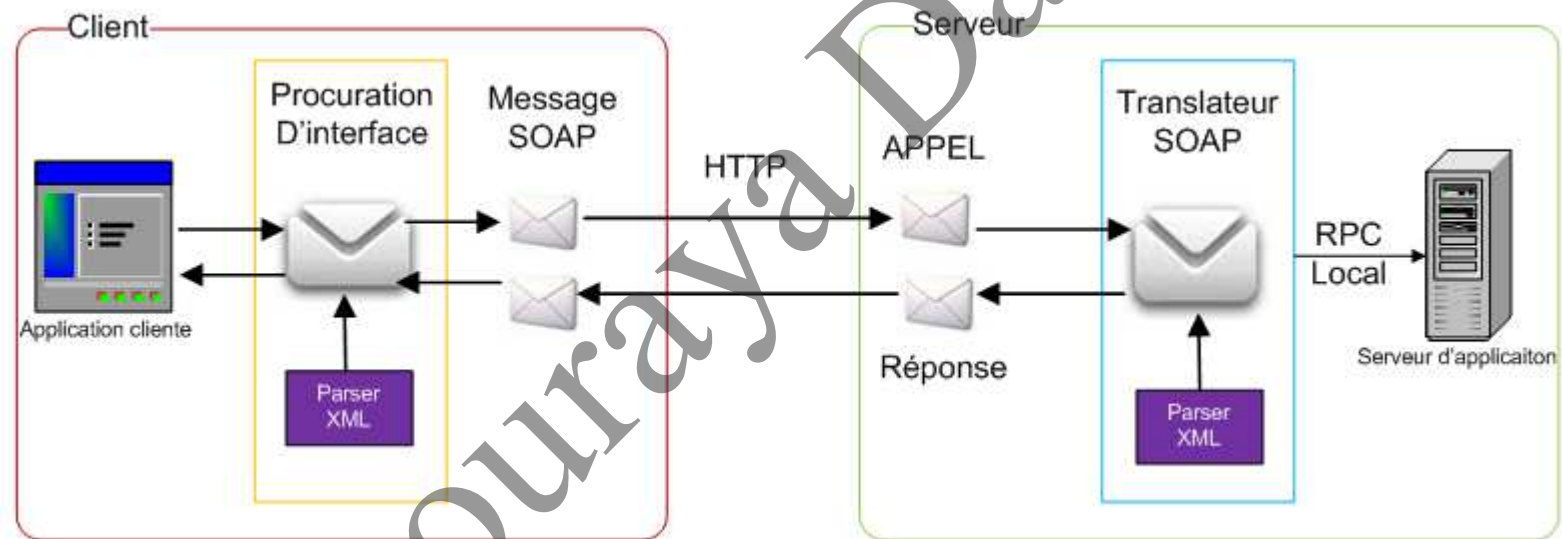
fppt.com

SOAP : Définition ?

Protocole d'invocation de méthodes sur des services distants. Basé sur XML, SOAP a pour principal objectif d'assurer la communication entre machines.

- XML-RPC et REST ses ancêtres.
- SOAP léger le plus implémenté actuellement recommandé par W3C.
- Très bien adapté à l'utilisation des WS : Permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers.
- Aisément porté sur toutes les plates-formes et les technologies existantes.

SOAP : Fonctionnement ?



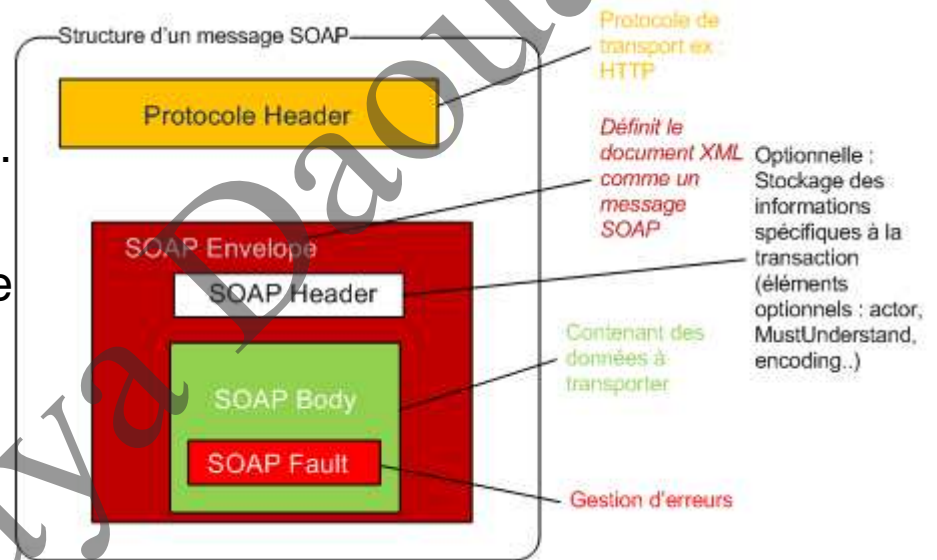
SOAP: Structure ?


Enveloppe: Élément de base.
contient spécifications espaces
désignation, codage de données.

Header: Facultative. Indique si le
message est mandataire ou
optionnel. Utile quand message
Traité par plusieurs.

Body: *container* pour les info. mandataires à l'intention du récepteur.
Contient méthodes et paramètres qui seront exécutés par le destinataire
final.

Fault: Facultatif. Utilisé pour reporter les erreurs.

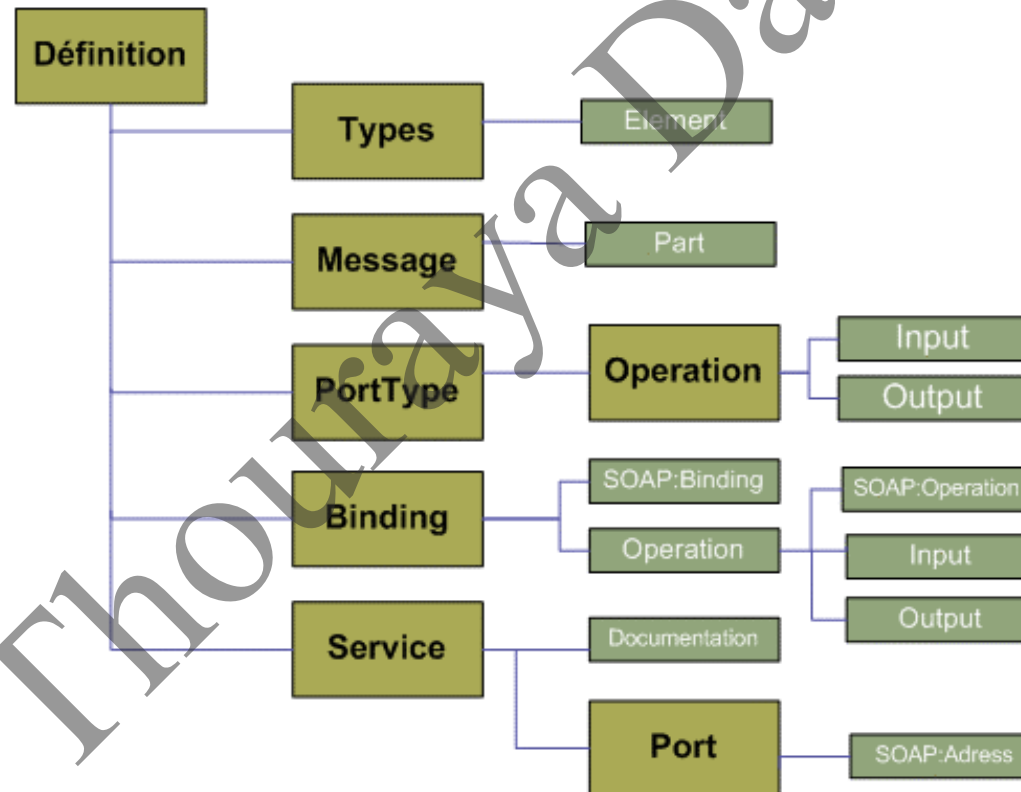




WSDL LANGUAGE DE DESCRIPTION ?

WSDL : Structure ?

Un document WSDL se compose d'un ensemble d'éléments décrivant les types de données utilisés par le service, les messages que le service peut recevoir ainsi que les liaisons SOAP associées à chaque message.



WSDL : Éléments de structure ?

- **Types** : fournit la définition de types de données utilisés pour décrire les messages échangés.
- **Messages** : représente une définition abstraite (noms et types) des données en cours de transmission.
- **PortTypes** : décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
 - **Opération** : c'est la description d'une action exposée dans le port.
- **Binding** : spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP, etc.).
- **Service** : indique les adresses de port de chaque liaison.
 - **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.

Structure de base d'un fichier XML

- On parle de document XML
- Fichier books.xml

```
XML Copier

<?xml version="1.0"?>
<Catalog>
  <Book id="bk101">
    <Author>Garghentini, Davide</Author>
    <Title>XML Developer's Guide</Title>
    <Genre>Computer</Genre>
    <Price>44.95</Price>
    <PublishDate>2000-10-01</PublishDate>
    <Description>An in-depth look at creating applications
    with XML.</Description>
  </Book>
  <Book id="bk102">
    <Author>Garcia, Debra</Author>
    <Title>Midnight Rain</Title>
    <Genre>Fantasy</Genre>
    <Price>5.95</Price>
    <PublishDate>2000-12-16</PublishDate>
    <Description>A former architect battles corporate zombies,
    an evil sorceress, and her own childhood to become queen
    of the world.</Description>
  </Book>
</Catalog>
```


Explications fichier XML

- 1^{ère} ligne facultative. Se trouve par défaut: le XML correspond à la version 1.0 de la norme W3C.
- On peut mettre des commentaires : <!-- commentaire -->
- Un élément XML : une balise ouvrante et une fermante.
- Contient des sous-éléments.
- Une règle: il ne doit posséder qu'un seul nœud racine.
- Il est possible de créer des attributs sur un élément. Pas de limite au nombre d'attributs.
- Un attribut possède un nom et une valeur.

Comparaison XML et JSON

```
<?xml version="1.0" ?>
<root>
| <fruits>
  <item>
    <kiwis>3</kiwis>
    <mangues>4</mangues>
    <pommes></pommes>
  </item>
  <item>
    <panier>true</panier>
  </item>
</fruits>
<legumes>
  <patates>amandine</patates>
  <figues>de barbarie</figues>
  <poireaux>>false</poireaux>
</legumes>
</root>
```

```
var courses = {
  "fruits": [
    { "kiwis": 3,
      "mangues": 4,
      "pommes": null
    },
    { "panier": true },
  ],
  "legumes":
    { "patates": "amandine",
      "figues": "de barbarie",
      "poireaux": false
    }
};
```



POURQUOI UTILISER DES SERVICES WEB XML ?

Principales raisons

- Gagner une interopérabilité entre différentes applications distribuées hébergées par différents serveurs programmées dans des langages différents.
- Rendre accessible des applications à travers firewall en utilisant des protocoles Internet.
- Offrir un langage de modélisation XML indépendant de la plateforme ou de la technologie utilisée qui facilite le développement.

Propriétés

Les services Web XML doivent respecter les propriétés suivantes :

- Être accessible via Internet.
- Se décrire et décrire les services qu'ils proposent par un descriptif de type XML.
- Communiquer avec un client sous forme de message XML transmis sur Internet via le protocole HTTP.



CARACTÉRISTIQUES DES SERVICES WEB XML ?

Caractéristiques générales

- Technologie basée sur XML.
- Supportée par la majorité des entreprises en TI.
- XML langage neutre pour représenter les données. 2volution du langage SGML (Standard Generalized Markup Language).
- Le support de XML par les entreprises assure que chaque technologie logicielle aura une stratégie de service XML dans les années à venir.

Caractéristiques Service Web XML


- Représentation des données basée sur XML.
- Interface flexible.
- Habileté d'interagir de manière synchrone ou asynchrone.
- Support des appels de fonctions distantes.
- Support d'échange de documents.

Frameworks disponibles

- De nombreux langages sont en cours d'adaptation pour permettre l'utilisation de services Web.
- Adaptation: Développement librairie d'accès au services Web qui se base sur d'autres librairies (librairies de sérialisation XML).

Exemples concrets

- MS SOAP toolkit: Permet de créer des applications ou modifier des existantes développées par ex. avec Visual Studio.
- Framework .NET: Complètement orienté SW XML. Indiqué pour des entreprises débutant le dév. de SW XML.
- Communauté Java: Très productive des services Web XML.
- Borland, MS, IBM et SUN : chacun a développé sa propre librairie payante pour l'accès aux SW XML.
- Apache et Jboss: développement gratuit. Solution gratuite sont encore complexes à utiliser.
- Python ou PHP: Dév. et accès: Premiers pas!



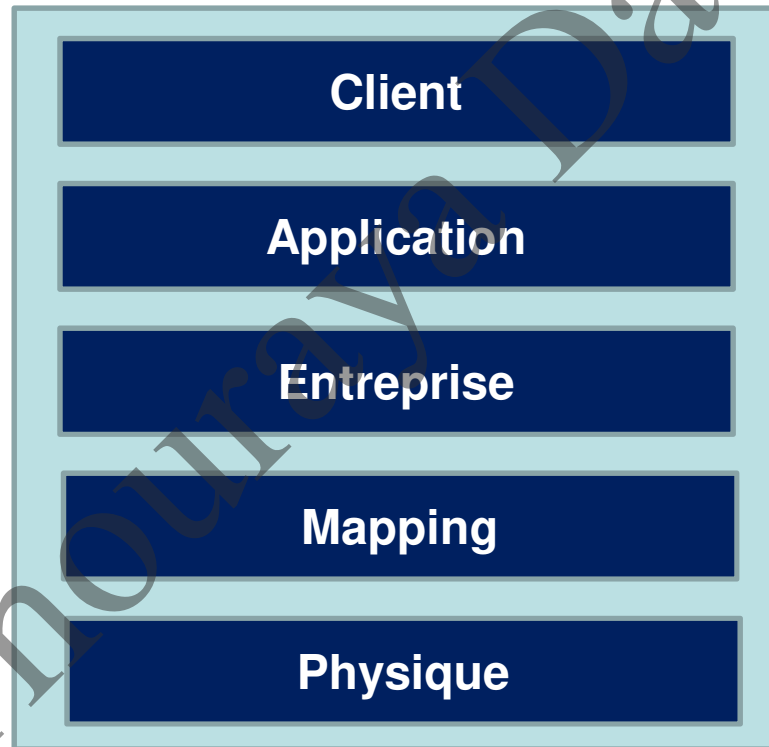
ARCHITECTURE 5 COUCHES

Dr Thouraya Daouas

36

5 Couches

- Proposé par Stève Sfartz pour bien coder les applications.



Les couches

- **Physique** : Structure physique des données (SGBD, Annuaire LDAP, etc.)
- **Mapping** : Réalise les accès vers la couche Physique (cas de SGBDR c'est un outil mapping relationnel/objet).
- **Entreprise** : Objets structurants qui n'intègrent aucune notion fonctionnelle. Objets transversaux à toutes les applications. Propose des services d'accès à ces méthodes à travers de méthodes de règles de gestion (création, recherche, modifications et suppressions).

Les couches

- **Application** : Regroupe la logique fonctionnelle d'une application. Utilise les services de la couche Entreprise pour réaliser le fonctionnel spécifié et le présente sous services.
- **Client** : Interface utilisateur. Dans le cas d'une application Web elle change fréquemment.

Chaque sous système peut être modélisé de façon indépendante. Ce modèle garanti de hauts niveaux :

- D'évolutivité : Fonctionnel et transactionnel dissociés.
- De maintenabilité : Responsabilités par couche.
- De réutilisabilité : Faible couplage entre les couches.

Utilisation des services

- Se fait naturellement : Développer l'accès à une application via les SW XML revient à développer une nouvelle couche Client.
- Le développement de la couche Client « Services Web » se fera en important la façade de la couche application dans un outil de développement SW XML.



XML

Définition

- XML (*Extensible Markup Language*, ou Langage Extensible de Balisage).
- Langage destiné à succéder à HTML (*Hypertext Markup Language*) c'est un langage de balisage.
- Différence : HTML possède un ensemble de balises de présentation prédéfinies / XML permet d'inventer de nouvelles balises d'isolement d'informations ou d'agrégats élémentaires que peut contenir une page Web.
- XML et HTML ont un ancêtre commun : le SGML (Standard Generalized Markup Language).
- Une de ses caractéristiques la séparation du formatage et du contenu.

Exemple HTML vers XML

```
<H2>Bibliotheque</H2>
<UL>
  <LI>Victor Hugo,
    <I>Les miserables</I>,
    1995,Dupond,Paris
  </LI>
  <LI>Frederic Beigbeider,
    <I>Windows on the world</I>,
    2004,Fayard,Paris
  </LI>
</UL>
```

Les balises concernent
la présentation finale.

Bibliotheque

- Victor Hugo, *Les miserables*, 1995, Dupond, Paris
- Frederic Beigbeider, *Windows on the world*, 2004, Fayard, Paris

Exemple HTML vers XML

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<BIBLIO SUBJECT='XML'>
  <LIVRE ISBN='9742212030811' LANG='fr' SUBJECT='applications'>
    <AUTEUR>
      <PRENOM>Victor</PRENOM>
      <NOM>Hugo</NOM>
    </AUTEUR>
    <TITRE>Les miserables</TITRE>
    <EDITEUR>
      <NOM>Dupond</NOM>
      <LIEU>Paris</LIEU>
    </EDITEUR>
    <DATEPUB>1999</DATEPUB>
  </LIVRE>
  <LIVRE ISBN='3782242090420' LANG='fr' SUBJECT='général'>
    <AUTEUR>
      <PRENOM>Frederic</PRENOM>
      <NOM>Beigbeider</NOM>
    </AUTEUR>
    <TITRE>Windows on the world</TITRE>
    <EDITEUR>
      <NOM>Fayard</NOM>
      <LIEU>Paris</LIEU>
    </EDITEUR>
    <DATEPUB>2004</DATEPUB>
  </LIVRE>
</BIBLIO>
```

Les balises ont un sens
et une hiérarchie et ne
concernent pas la
Présentation finale.

Règles d'écriture

Informations doivent être :

- Encadrés par des balises ouvrantes et fermantes. Ces élt ne doivent pas se chevaucher. Les élt vides sont permis, selon le format <ELEMENTVIDE/>.
- Soit incluses à l'intérieur même des balises : *attributs*.
Ex : <LIVRE SUBJECT='XML'>. attribut *SUBJECT* de l'élément *LIVRE* a la valeur 'XML'.
- Soit encore définies sous forme d'*entités*. Les entités sont des abréviations. ex ; si 'Extensible Markup Language' déclaré comme entité associée à la notation 'xml' ; cette chaîne de caractères pourra être abrégée en '&xml ;' dans tout le fichier XML.

Règles d'écriture

- Une entité peut représenter un fichier XML externe tout entier. Donc un même fichier XML (ex: bibliographie) pourra être utilisé par plusieurs pages XML différentes.
- La structure arborescente du document XML (intitulé des balises, imbrications des balises, etc.) peut être déclarée formellement dans le corps du document XML ou dans un fichier à part. Cette déclaration s'appelle Définition de Type de Document (DTD).
- En XML cette déclaration est *facultative*, ce qui donne une grande souplesse aux développeurs.

Règles d'écriture

- Lorsqu'un document XML possède une DTD associée et la respecte, on dit qu'il est *valide*.
- Lorsqu'il respecte seulement les règles de la grammaire XML (balises fermées, correctement imbriquées, etc.) on dit qu'il est *bien formé*.
- La spécification XML se trouve à l'adresse

[http ://www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml)