

Les Services Web

A series of horizontal lines in teal and light blue colors, with varying lengths and offsets, creating a modern, layered effect across the middle of the slide.

Introduction Générale

Implémentations pour une SOA

Plusieurs technologies ont vu le jour pour développer des architectures orientées service.

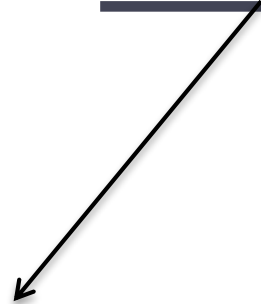
- ✓ **Services Web - WSOA**
- ✓ Framework asynchrones (Vert.x, Play, Node.js, ...)
- ✓ Broker de messages - MOM (ActiveMQ, Kafka, RabbitMQ...)

Des technologies spécifiques à des domaines

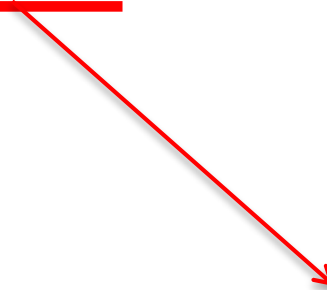
- ✓ Internet of Things (MQTT, UPnP, Jini) - SOAD
- ✓ DPWS (WS4D) - WSOAD
- ✓ OSGi (Equinox, Felix, ...)

Les technologies ne sont pas exclusives et peuvent être utilisées de manière complémentaire.

Service Web



Entité logicielle
autonome qui expose
un certain nombre
d'**opérations** offrant
un traitement de bout
en bout.

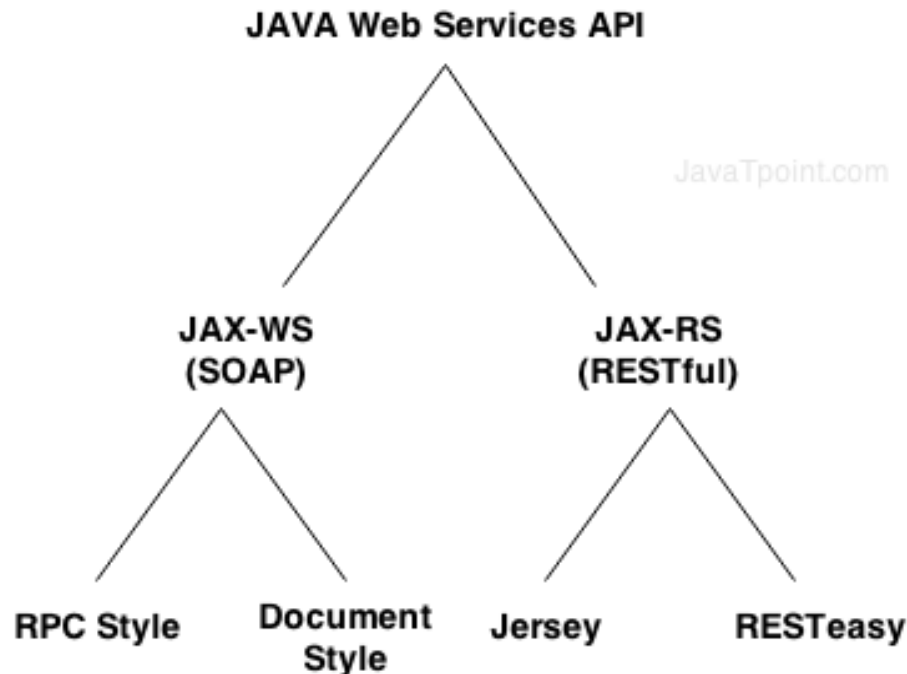


Basé sur les
protocoles et les
langages **standards** du
Web, ne nécessitant
pas une configuration
réseau particulière.

Services Web : implémentations possible

Deux familles de services web existent actuellement:

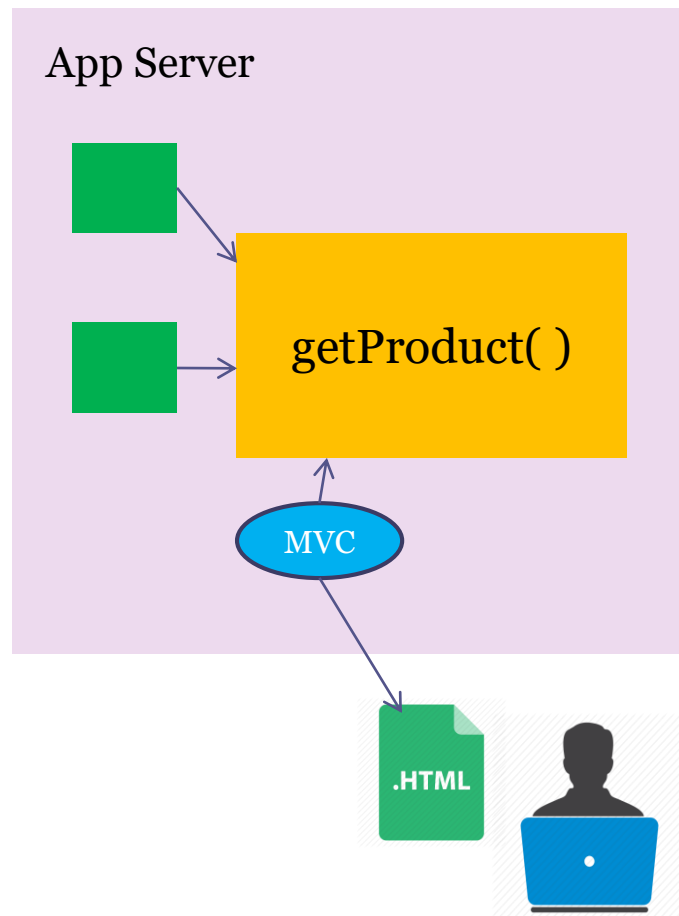
- ✓ **Services web étendus SOAP**
- ✓ **Services web REST**



Service Web => Code consumption or M2M communication

Exemple

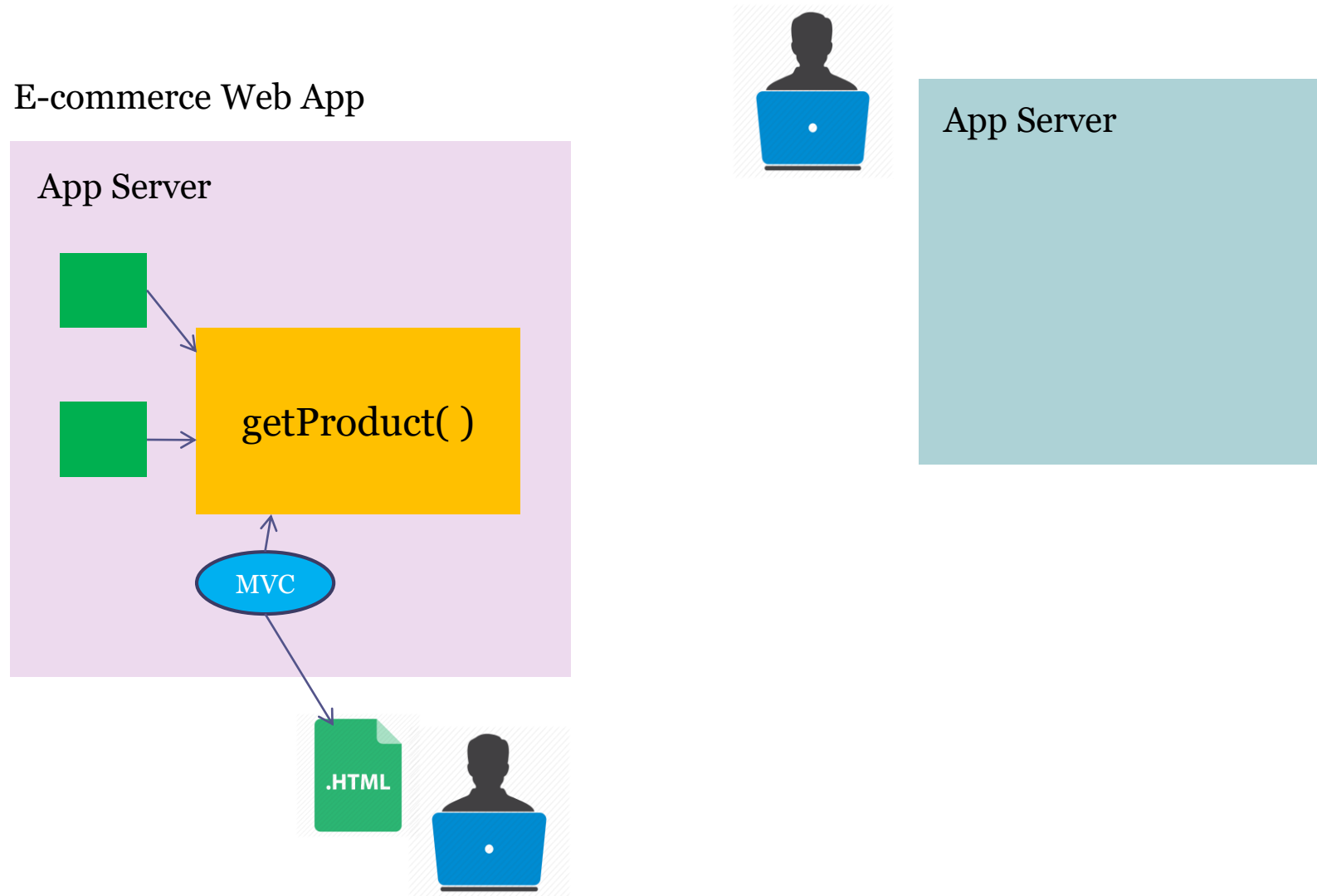
E-commerce Web App



Service Web => Code consumption or M2M communication

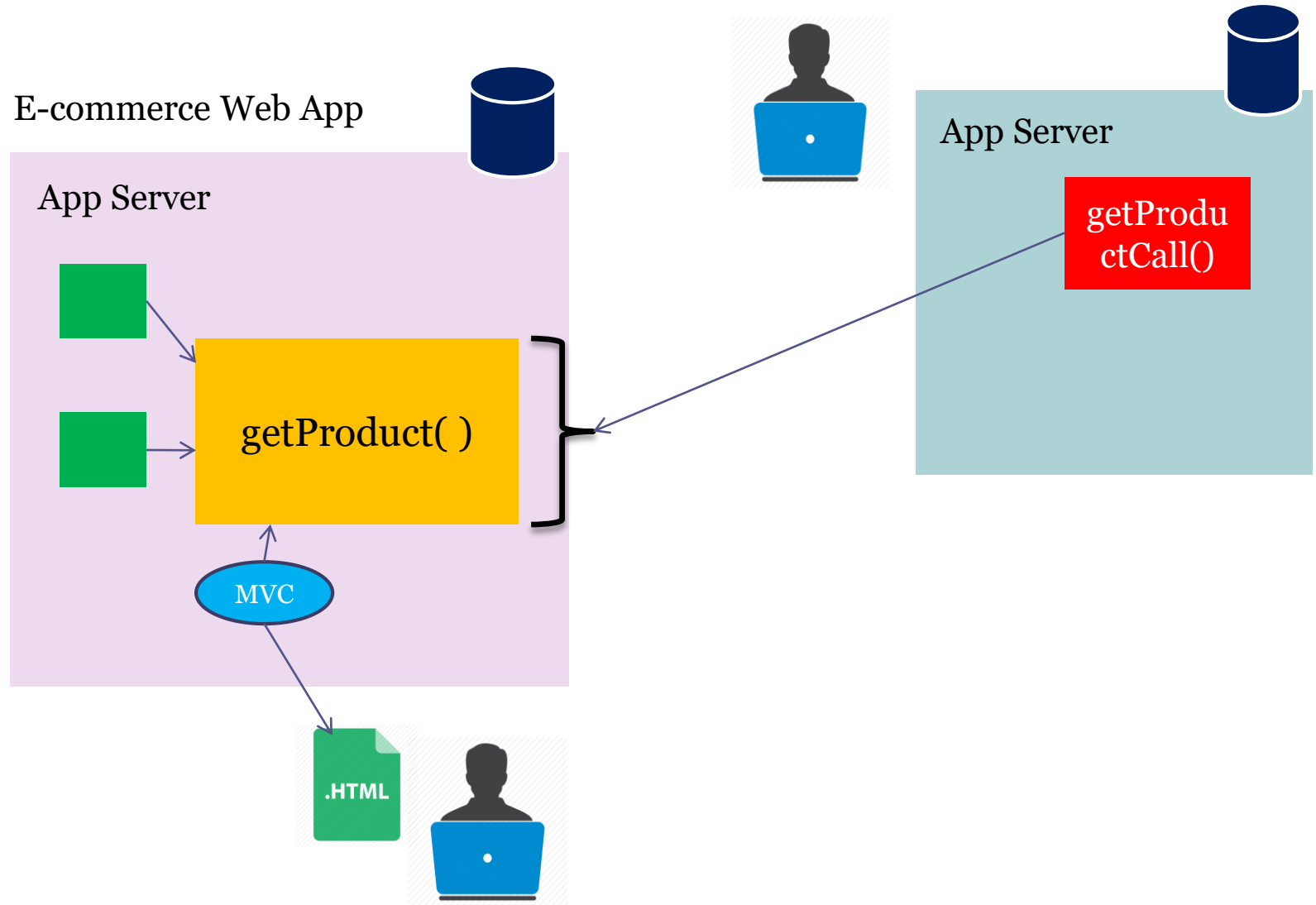
Exemple

E-commerce Web App



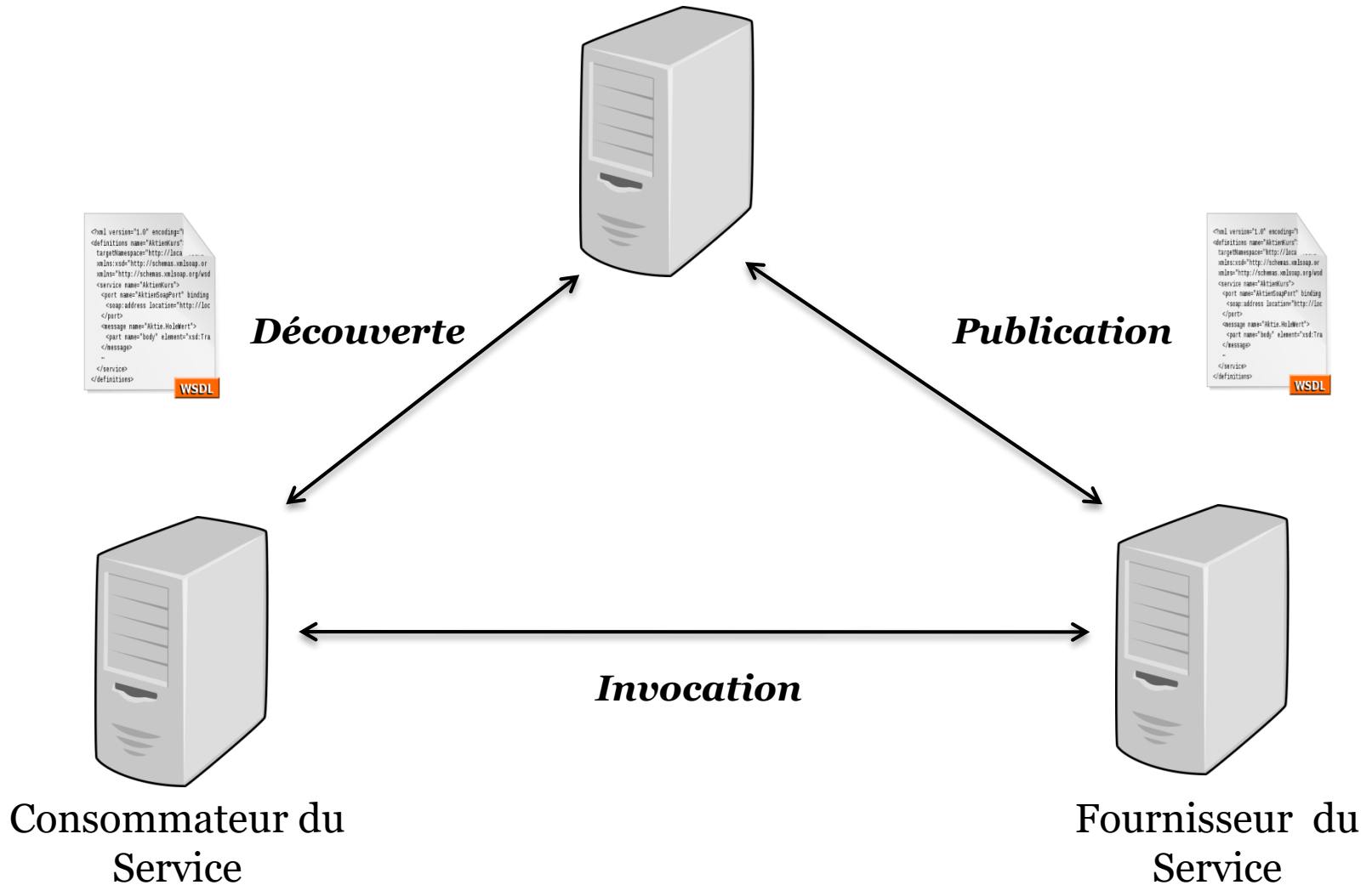
Préférence

Exemple



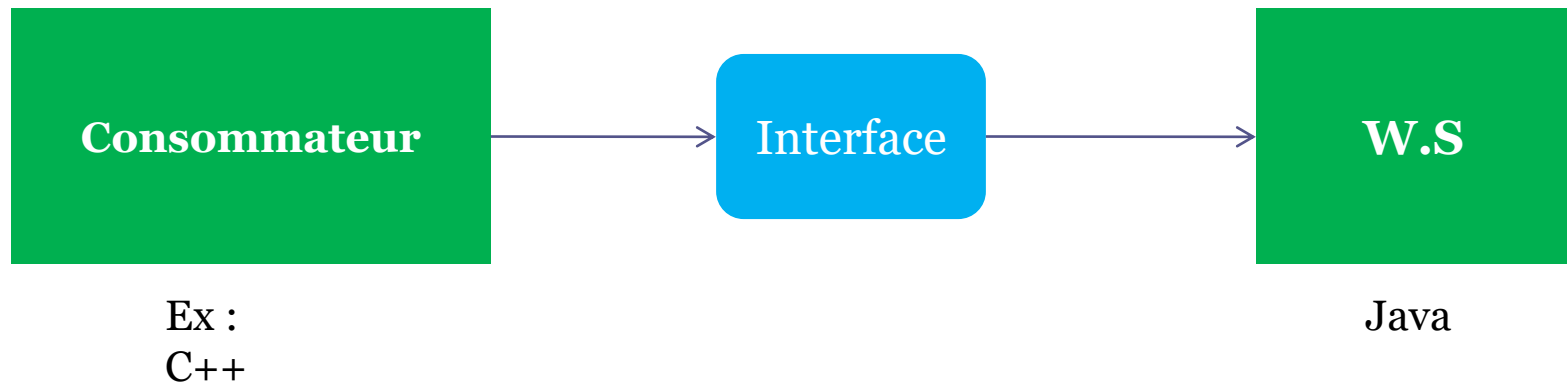
Style architectural - Acteurs

Annuaire UDDI



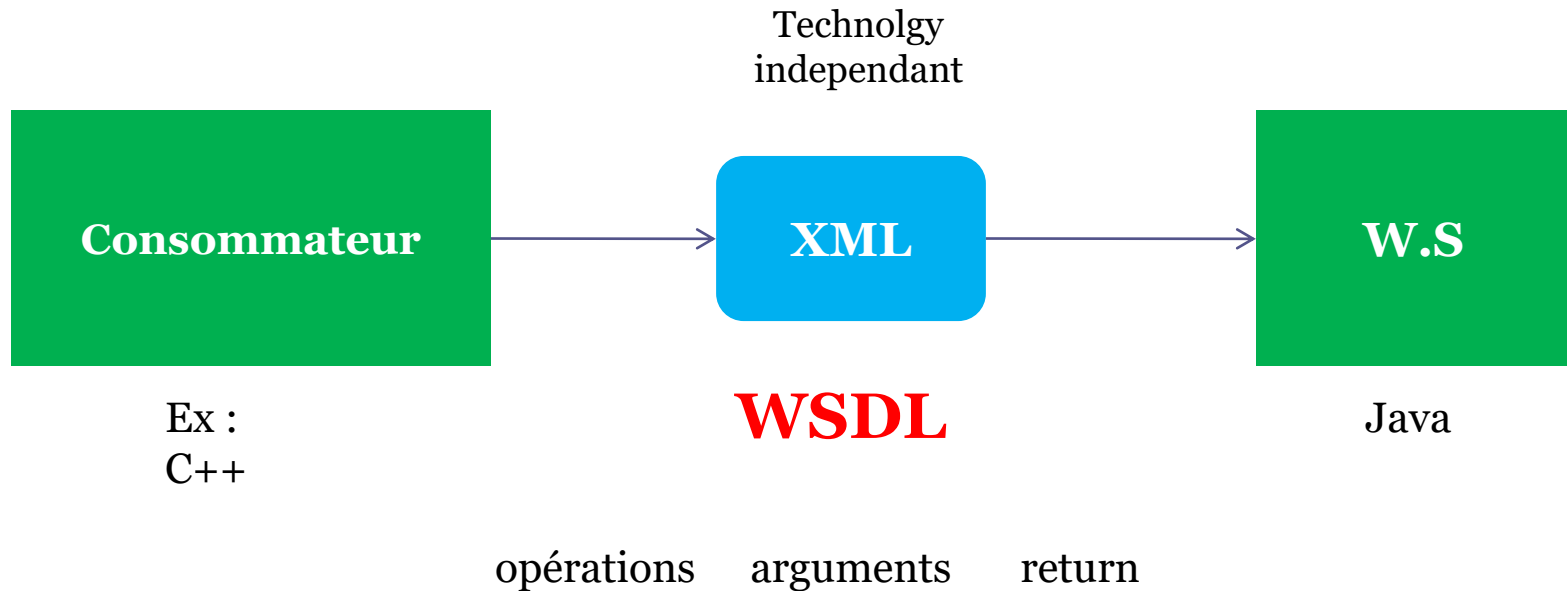
Services Web SOAP : Concepts

Exemple : Interface



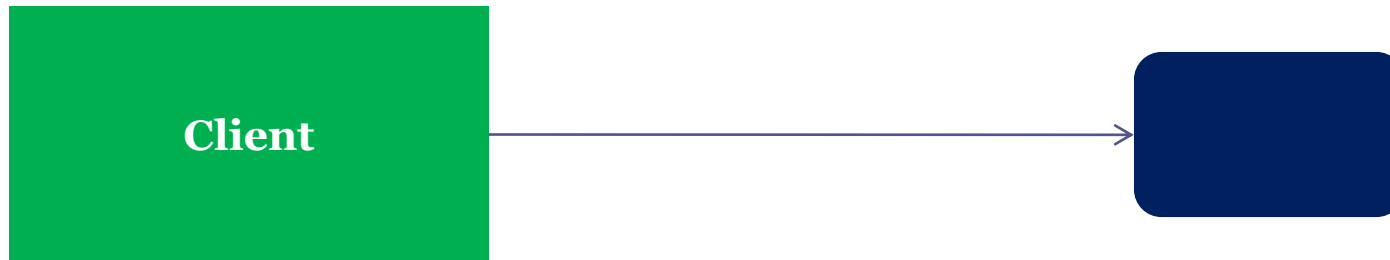
Services Web SOAP : Concepts

Exemple : Contrat



Services Web SOAP : Concepts

Exemple: **où trouver le WSDL ?**

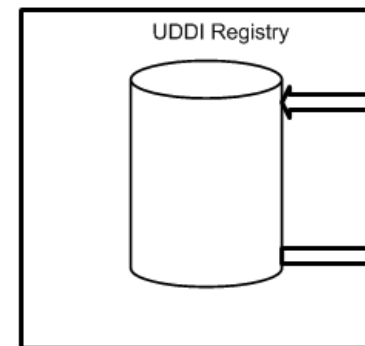


Services Web SOAP : Concepts

Exemple: où trouver le WSDL ?

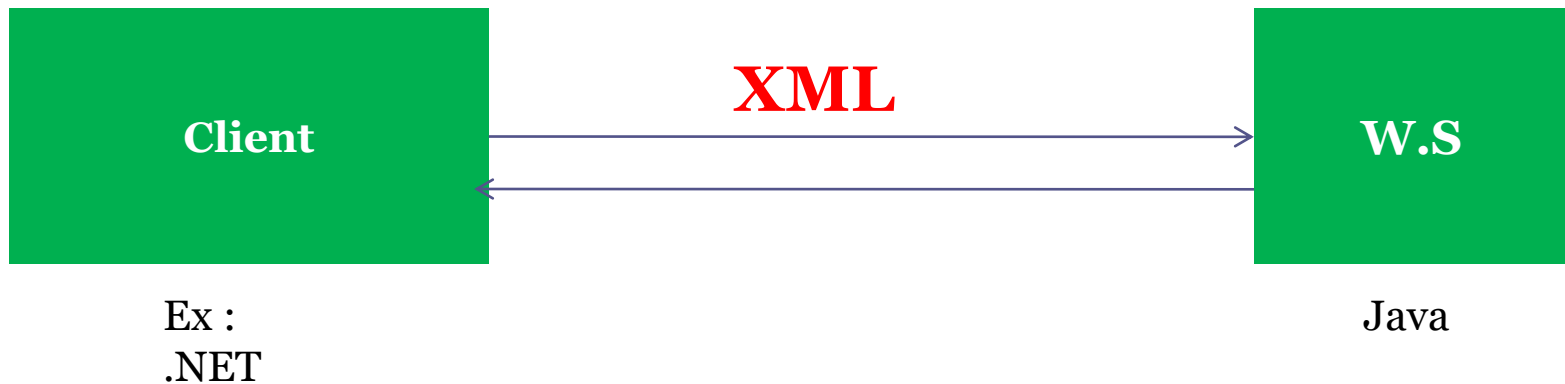


Universal Description Discovery and Integration



Services Web SOAP : Concepts

Exemple: En quel **format les données (messages)** sont-elles échangées ?



Services Web SOAP : Concepts

Exemple: En quel format les données sont-elles échangées ?



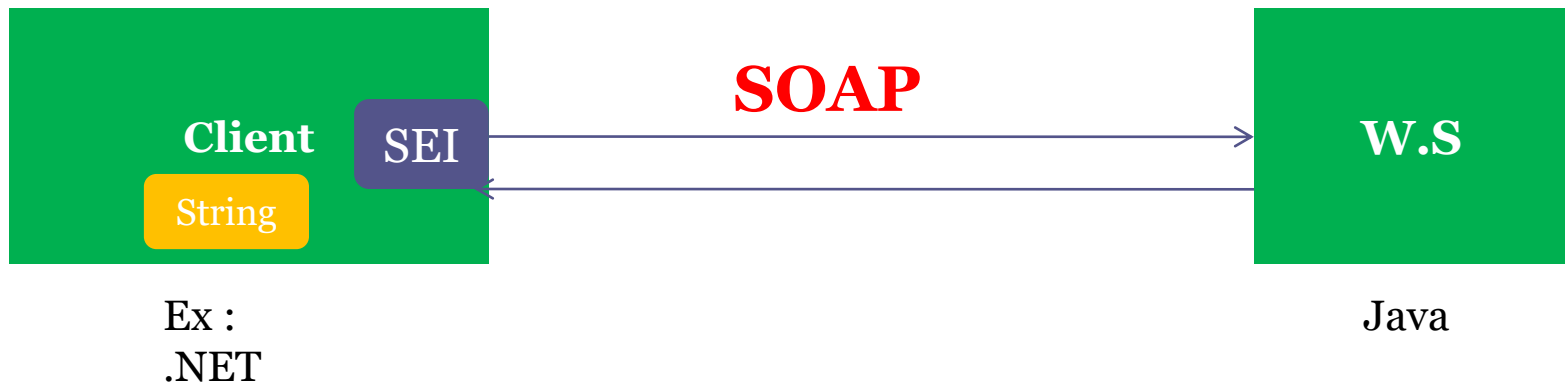
Services Web SOAP : Concepts

Exemple: **Conversion**



Services Web SOAP : Concepts

Exemple: Conversion – **Service Endpoint Interface**



Services Web SOAP : Concepts

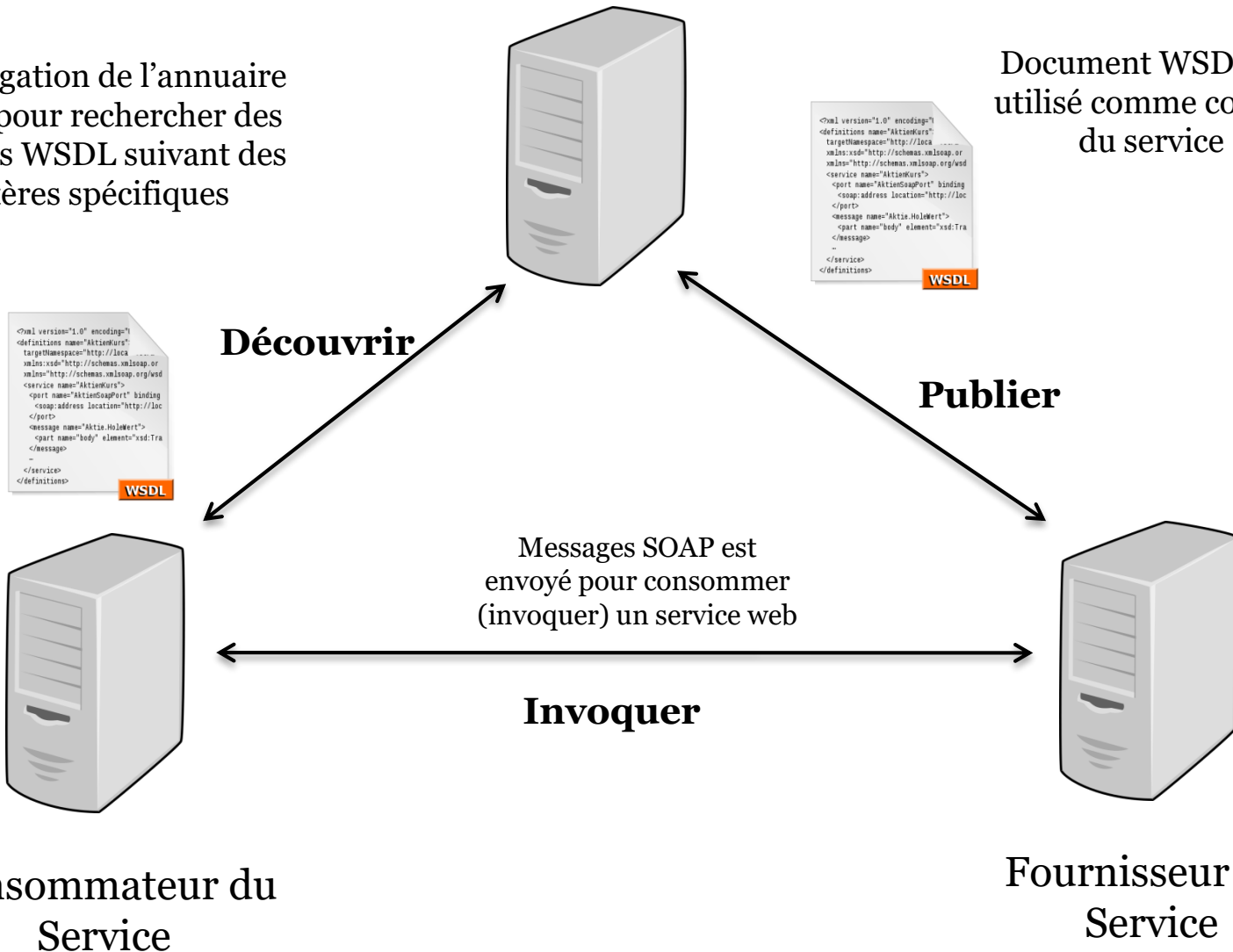
- ✓ WSDL – Web Service Description/Definition Language
- ✓ SEI – Service Endpoint Interface
- ✓ SOAP – Simple Object Access Protocol
- ✓ UDDI – Universal Description Discovery and Integration

Style architectural - Acteurs

Annuaire UDDI

Interrogation de l'annuaire UDDI pour rechercher des contrats WSDL suivant des critères spécifiques

Document WSDL est utilisé comme contrat du service



Les Services Web

Implémentation : Service Web et Client consommateur

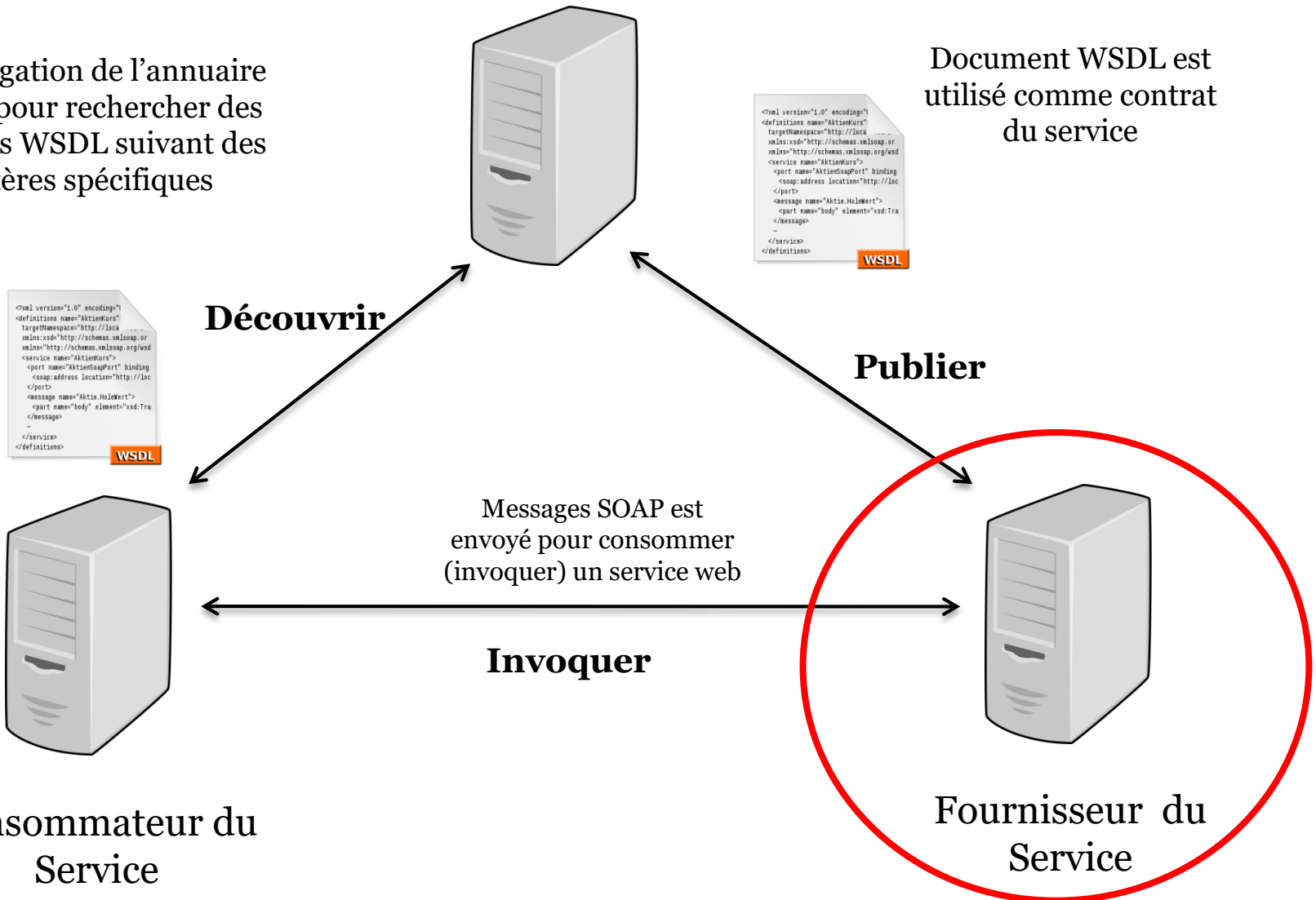
Aicha AID - 2021
mailto: aid_aicha@hotmail.fr

Style architectural - Acteurs

<http://localhost:4848/helloworldws?wsdl>

Interrogation de l'annuaire
UDDI pour rechercher des
contrats WSDL suivant des
critères spécifiques

Document WSDL est
utilisé comme contrat
du service



Implémentation: JAX-WS API

Deux façons permettant de **développer** un service web SOAP :

- ✓ Approche Top-Down (*Contract First*) : à partir du WSDL
- ✓ **Approche Bottom-Up (Code First)** : à partir des POJOs annotés

Trois façons permettant de **tester** un service web SOAP:

- ✓ Tester avec le Tester de Glassfish Server
- ✓ Tester avec SoapUI
- ✓ Tester via une application cliente JSP (Eclipse)

Consommer un service web SOAP via un Client

Service Web SOAP

Approche Bottom-Up (*Code First*) :

Création d'un service web SOAP HelloWorld qui fournit deux opérations :

1. Une opération **simpleHello** sans paramètre en entrée et retourne la chaîne de caractères *Hello World*.
2. Une opération **makeHello** qui prend en paramètre une chaîne de caractères *name* et retourne la chaîne de caractères *Hello World, name*.

Service Web SOAP

SW - Hello World

Opérations

simpleHello ()

makeHello()

Pas d'args

« Hello World »

Nom

« Hello World, Nom »



Service Web SOAP

```
public String simpleHello() {  
    return "Hello World";  
}
```

```
public String makeHello(String name) {  
    return "Hello World, " + name;  
}
```


Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

1. Création de **l'interface**
2. Création de la **classe** implémentant l'interface
3. Création de la **classe** endpoint publiant le service web

Service Web SOAP - JAX-WS

Pour implémenter **Hello World WS** :

Interface	HelloWorld
Classe implements	HelloWorldImpl
Classe Main Publish	HelloWorldPublisher

Opération 1	String simpleHello()
Opération 2	String makeHello(String nom)

Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

1. Création de l'interface :

```
public interface HelloWorld {  
    public String simpleHello();  
    public String makeHello(String name);  
}
```

Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

1. Création de l'interface :

```
@WebService
public interface HelloWorld {

    @WebMethod
    public String simpleHello();

    @WebMethod
    public String makeHello(String name);

}
```

Service Web SOAP - JAX-WS

2. Création de la **classe** implémentant l'interface :

```
@WebService(endpointInterface = "org.soa.ws.tp.HelloWorld")
public class HelloWorldImpl implements HelloWorld{
    @Override
    @WebMethod
    public String simpleHello() {
        return "Hello World";
    }

    @Override
    @WebMethod
    public String makeHello(String name) {
        return "Hello World, " + name;
    }
}
```

Service Web SOAP - JAX-WS

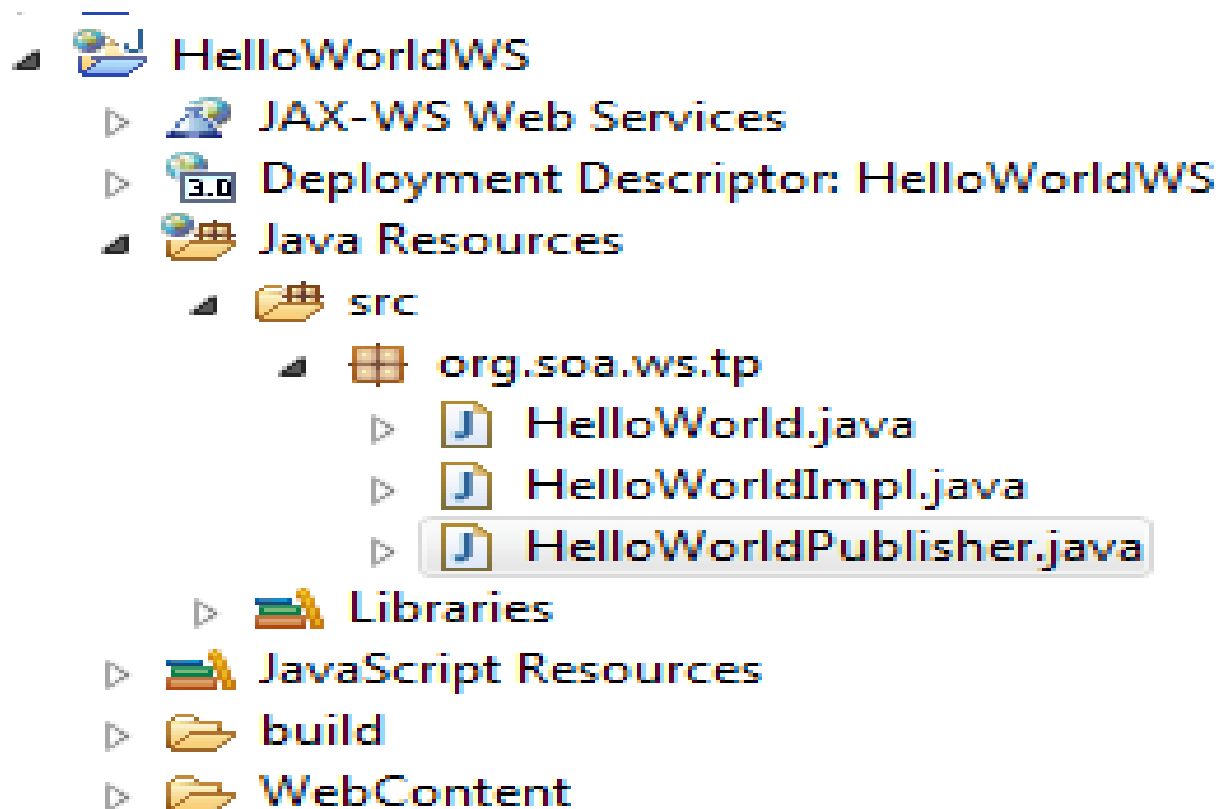
3. Création de la **classe** endpoint publiant le service web:

```
public class HelloWorldPublisher {  
  
    public static void main(String[] args) {  
  
        Endpoint.publish("http://localhost:4848/helloworldws",  
                           new HelloWorldImpl());  
    }  
  
}
```

Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

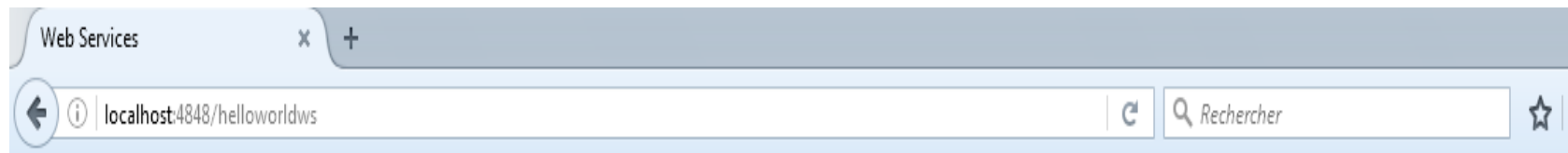
Projet de type : Dynamic Web Project



Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

Exécution de la classe HelloWorldPublisher:



Web Services

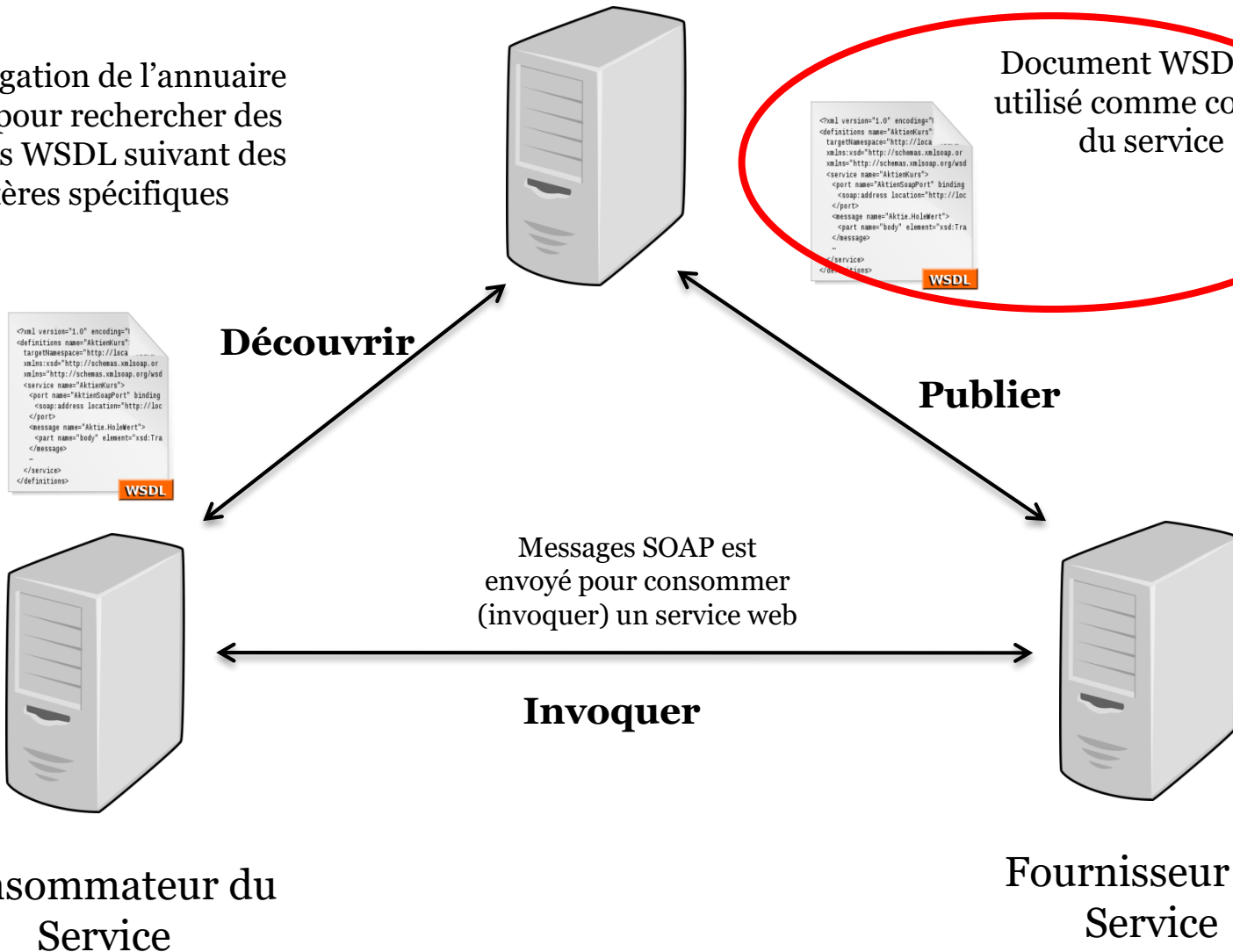
Endpoint	Information
Service Name: {http://tp.ws.soa.org/}HelloWorldImplService	Address: http://localhost:4848/helloworldws
Port Name: {http://tp.ws.soa.org/}HelloWorldImplPort	WSDL: http://localhost:4848/helloworldws?wsdl
	Implementation class: org.soa.ws.tp.HelloWorldImpl

Style architectural - Acteurs

<http://localhost:4848/helloworldws?wsdl>

Interrogation de l'annuaire
UDDI pour rechercher des
contrats WSDL suivant des
critères spécifiques

Document WSDL est
utilisé comme contrat
du service



Service Web SOAP

Exercice 2: Série TD 2

Ecrire un service web SOAP Random fournissant deux opérations :

1. Une opération ***getRandomValue*** : pour récupérer une valeur aléatoire entre 0 et 1.
2. Une opération ***getSinusValue*** : pour récupérer le sinus d'une valeur passée en paramètre.

Ecrire les classes de ce service web, à savoir, **RandomWS** (interface), **RandomWSImpl**, et **RandomWSPublisher**.

Service Web SOAP

SW - RandomWS

Opérations

Pas d'args

**Valeur aléatoire
entre 0 et 1 :
double**

double : val

double : Sinus(val)

getRandomValue()

getSinusValue()



Service Web SOAP

```
public double getRandomValue() {  
    return Math.random();  
}
```

```
public double getSinusValue(double val) {  
    return Math.sin(val);  
}
```

Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

1. Création de **l'interface**
2. Création de la **classe** implémentant l'interface
3. Création de la **classe** endpoint publiant le service web

Service Web SOAP - JAX-WS

Pour implémenter **RandomWS** :

Interface	RandomWS
Classe implements	RandomWSImpl
Classe Main Publish	RandomWSPublisher

Opération 1	double getRandomValue ()
Opération 2	double getSinusValue (double val)

Service Web SOAP - JAX-WS

Pour implémenter un Service Web SOAP en JAVA:

1. Création de l'interface :

```
@WebService
public interface RandomWS {

    @WebMethod
    public double getRandomValue();

    @WebMethod
    public double getSinusValue(double val);

}
```

Service Web SOAP - JAX-WS

2. Création de la **classe** implémentant l'interface :

```
@WebService(endpointInterface = "org.soa.ws.tp.RandomWS")
public class RandomWSImpl implements RandomWS{
    @Override
    @WebMethod
    public double getRadomValue() {
        return Math.random();
    }

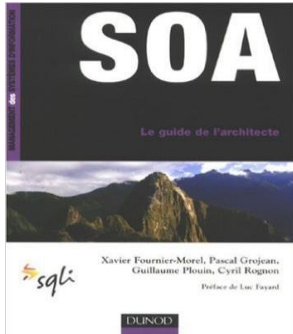
    @Override
    @WebMethod
    public double getSinusValue(double val) {
        return Math.sin(val);
    }
}
```


Service Web SOAP - JAX-WS

3. Création de la **classe** endpoint publiant le service web:

[illegible]

Ressources



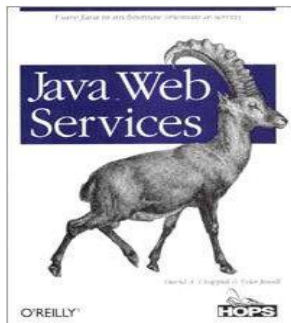
Le guide de l'architecte du SI

- ✓ 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

Cours – Koushik Kothagal - Developing SOAP Web Services with JAX-WS

- ✓ https://javabrainz.io/courses/javaee_jaxws/lessons/Introduction-to-Web-Services