

Module : Services Web



Cours 03 : Services Web

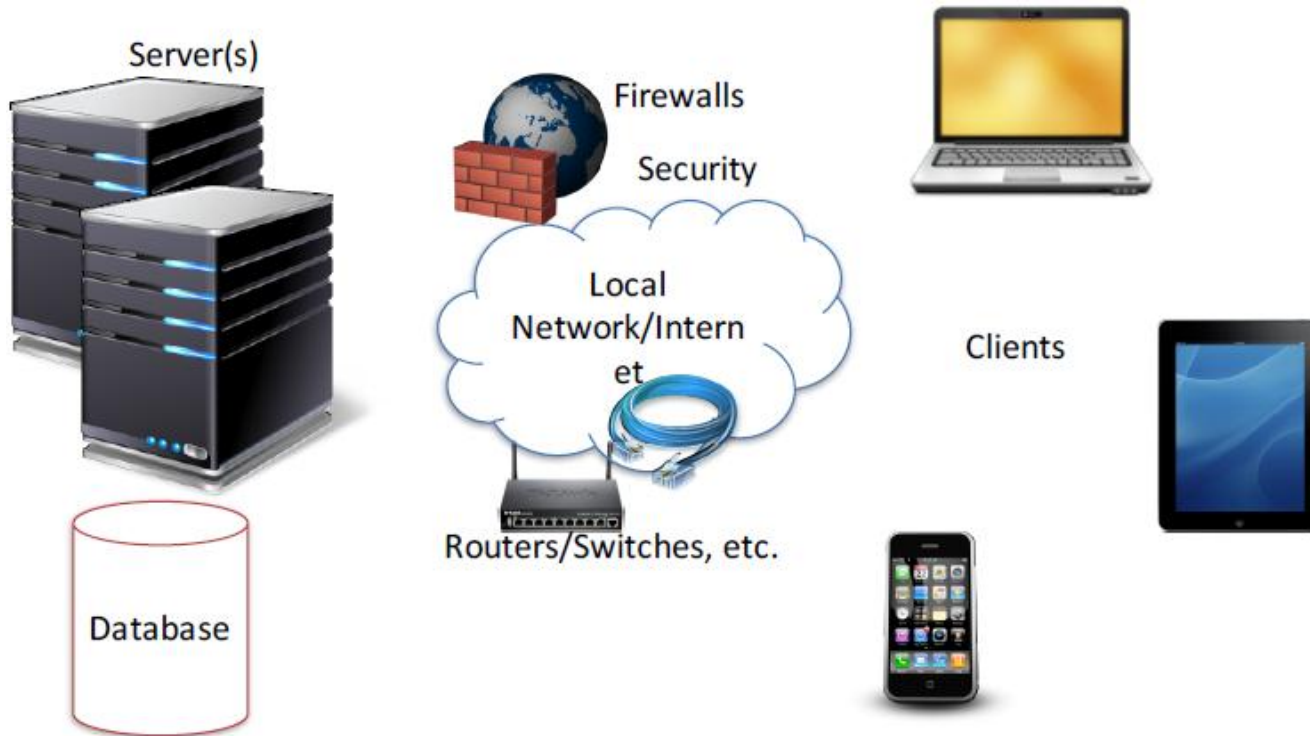
Présenté par Dr BRAHAMI EL BOUHISSI H.

2017-2018

M2 (IA – GL – ReSyD)

Architecture distribuée : Problématique

Architecture où le traitement des données des applications est distribué sur plusieurs machines en réseau : Architectures client-serveur, N-Tiers, Web



SI d'une entreprise souvent confronté à des changements :

Interopérabilité ?

Réutilisabilité ?

Communication entre applications hétérogènes ? (Java/C#)

L'interopérabilité exprime le besoin d'échanger des données (d'1 facture par ex) entre 2 systèmes distribués et **éventuellement hétérogènes** pour un besoin **d'intégration**

Solution : SOA (Service Oriented Architecture)

SOA, proposée par Gartner Group (2003-2008), se prête comme solution au problème d'interopérabilité, en répondant aux questions :

- Quel **transport de communication** (TCP/IP ? quelle version?) ?
- Quel **protocole applicatif** (HTTP, FTP, SFTP...) ?
- Quel type **d'encodage de données** ?
- Quel est le **format de données** à adopter ? (bureautique, XML)

Enjeux de la SOA

Solution incontournable pour gagner d'autres enjeux :

○ **Métier :**

Produire des SI ouverts (à la fois interopérables + évolutifs)

Proposer une solution utilisant des standards et assurant un couplage faible.

Réduire le Time to Market (le temps entre le besoin et la mise en production)

Proposer des moyens pour réduire le temps du cycle projet

○ **Technique :**

Fédérer les technologies

Proposer des moyens pour rendre les solutions techniques réutilisables,

○ **Financier :**

Maîtriser les coûts et les délais

Coût important

Architecture SOA

Architecture orientée service (Service Oriented Architecture)

- **Style d'architecture distribuée** qui permet de fournir ou consommer un processus métier en tant que **service**
- **Offre des services réutilisables et interopérables** via des interfaces standards (construites autour de XML)
- Plusieurs partenaires peuvent communiquer et échanger des données dans le contexte de SOA **indépendamment des Plateformes et langages**

Acteurs du SOA

○ Fournisseur de service :

- Fournit un service accessible via une adresse
- publie son contrat dans le registre de services
- et exécute les requêtes des consommateurs (un Proxy et un cache peuvent être utilisés du côté consommateur pour délocaliser le traitement et réduire le nombre d'appels réseau)

○ Consommateur de service : application, service...

- Cherche le service dans le registre (son adresse)
- Se lie dynamiquement au service (binding)
- Invoque le service via une requête conforme au contrat

○ Registre de services : Annuaire des contrats de services

- Le Contrat décrit le format d'échange (format des requête/réponse, les pré et post conditions du service et sa QoS, ex: temps de réponse)
- Le contrat est renouvelable par demande de nouveau bail à partir du registre

Technologies d'implémentation de SOA

L'architecture SOA peut être implémentée par différentes Technologies :

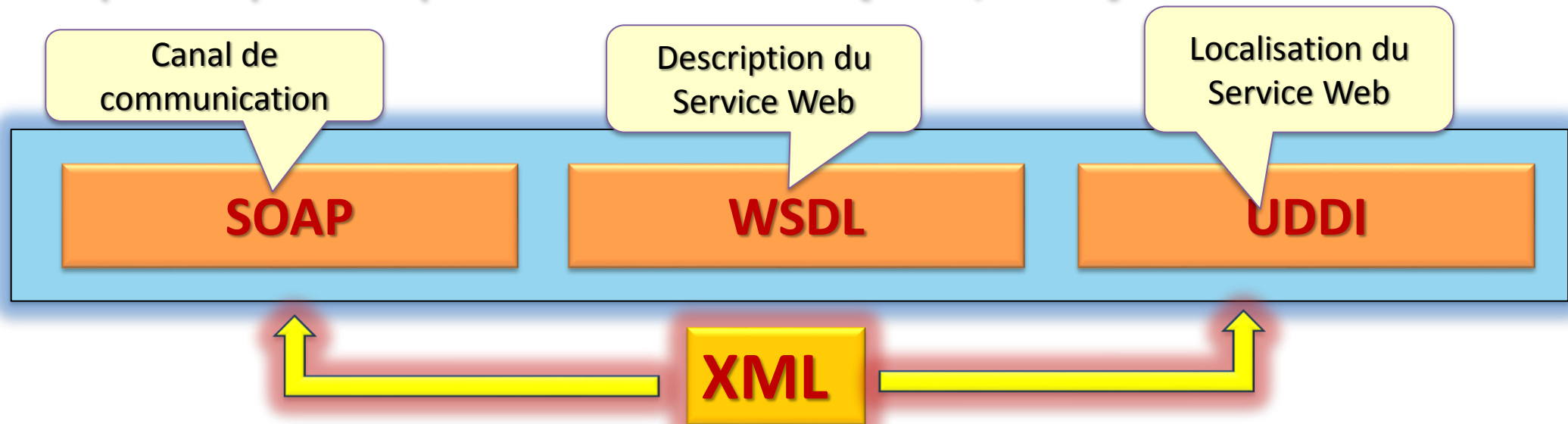
- **CORBA** : Multi-langage , Multiplateforme, installation coûteuse
- **DCOM** : Multi-langage , Mono-plateforme, faible diffusion (non disponible pour certains OS)
- **RMI** : Multi-langage , Multiplateforme, problème de performance et de sécurité
- **Services Web** (protocole SOAP)

Les **Services Web** demeurent la technologie émergente pour l'implémentation de l'architecture SOA :

- Déployés sur n'importe quelle plateforme
- Modulaires : implémenté dans n'importe quel langage
- Enveloppés dans une couche de standards dérivés du XML
- Ne nécessitant pas une configuration réseau particulière
- Publié, localisé et invoqué de n'importe quel point du Web
- Accessible via des protocoles standards Internet

Service Web : Définition

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. [W3C, 2004]



Web Service = Service + Web ?

O Service = fonctionnalité mise à disposition et exécutée par un fournisseur , elle est invoquée par un consommateur , **réutilisable** + **composable** + **indépendant**

O Interface :

Définit l'usage du service (syntaxe, sémantique, qualité) ➡ **contrat**

Masque l'implémentation du service pour un couplage consommateur/fournisseur faible

O Format pivot : Langage commun pour décrire et échanger les données

O Web Service = Service mis à disposition sur Internet

- ✓ Associé à une URL sur le web
- ✓ Accessible via des protocoles internet standard
- ✓ Accessible indépendamment des technologies d'implémentation
- ✓ Auto-descriptif



HTTP



XML

Exemples de Services Web existants

Google (<http://www.google.com/apis/>) :

- Accès gratuit mais limité (1000 requêtes par jour après enregistrement)

Amazon (<http://aws.amazon.com/fr/>)

- accès gratuit mais limité (1 requête par seconde après enregistrement)

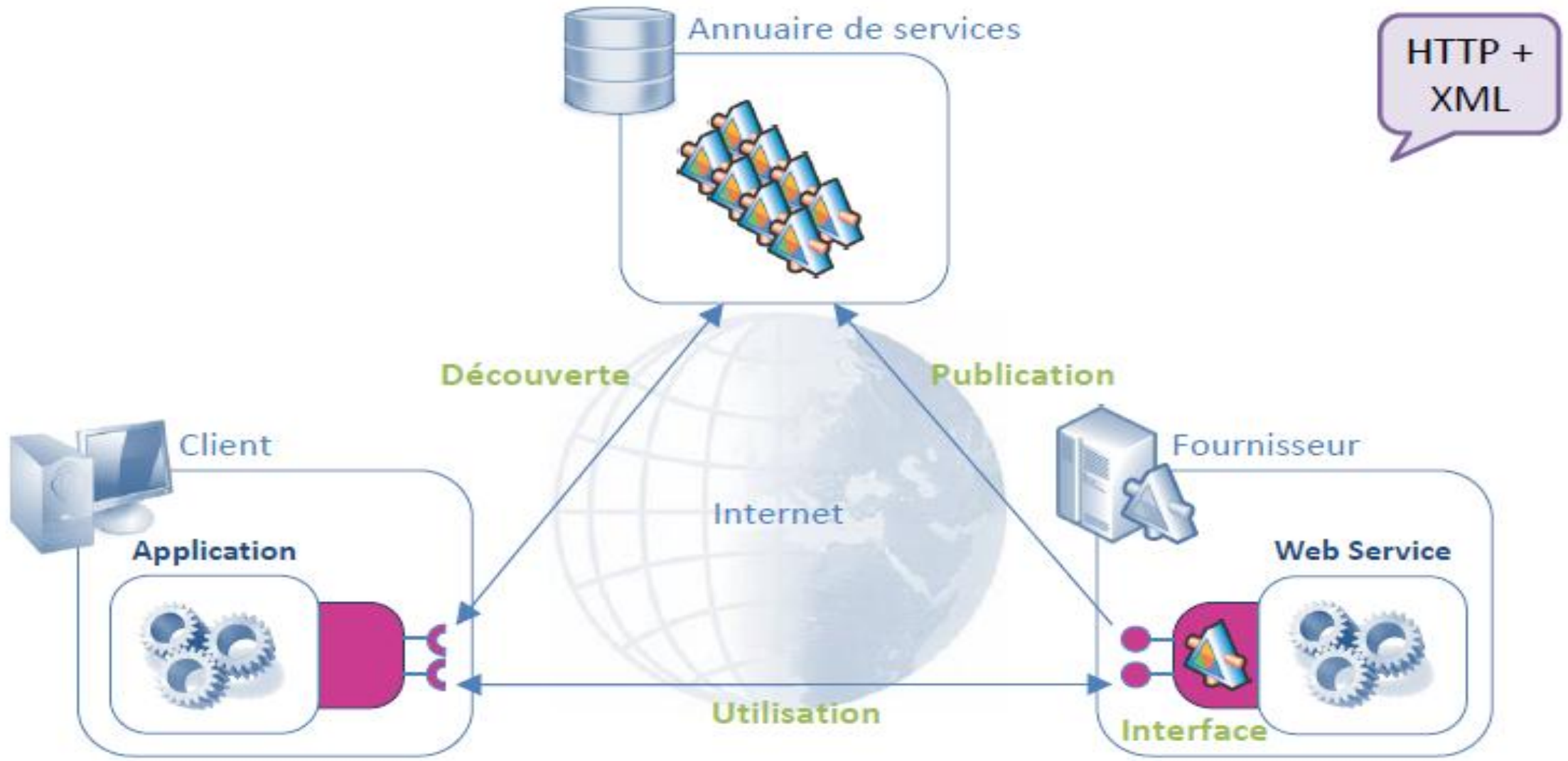


Bien d'autres (<http://webservicex.net> par exemple)

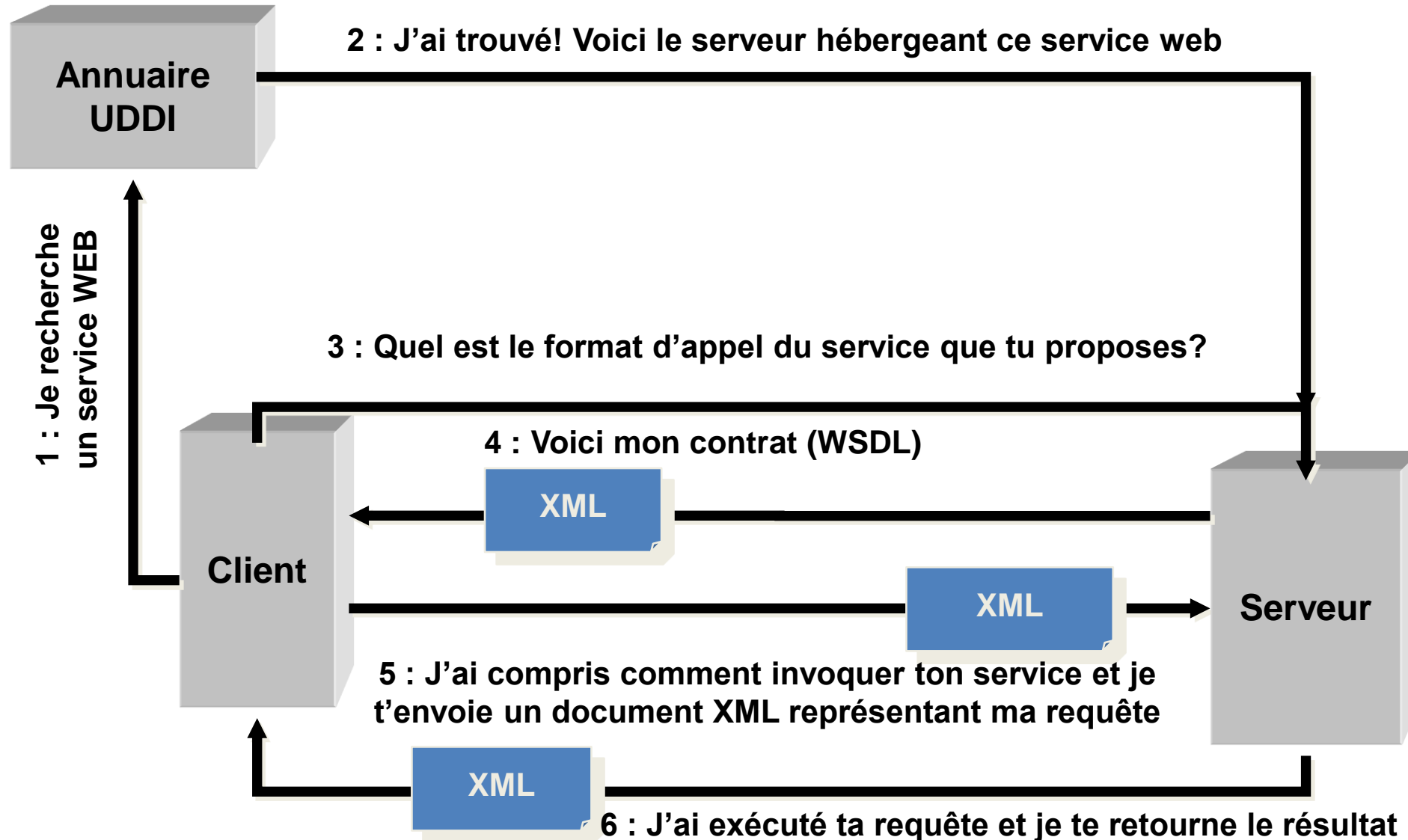
Infrastructure des Services Web (1)

- **SOAP** (Simple Object Access Protocol) : assure la communication avec et inter-Services Web ;
- **WSDL** (Web Services Description Language) : offre un schéma formel de description des Services Web ;
- **UDDI** (Universal Description, Discovery and Integration) : offre une manière uniforme de définir des registres des Services Web et un schéma uniformément extensible de descriptions des Services Web.

Infrastructure des Services Web (2)



Cycle de vie d'utilisation d'un Service Web

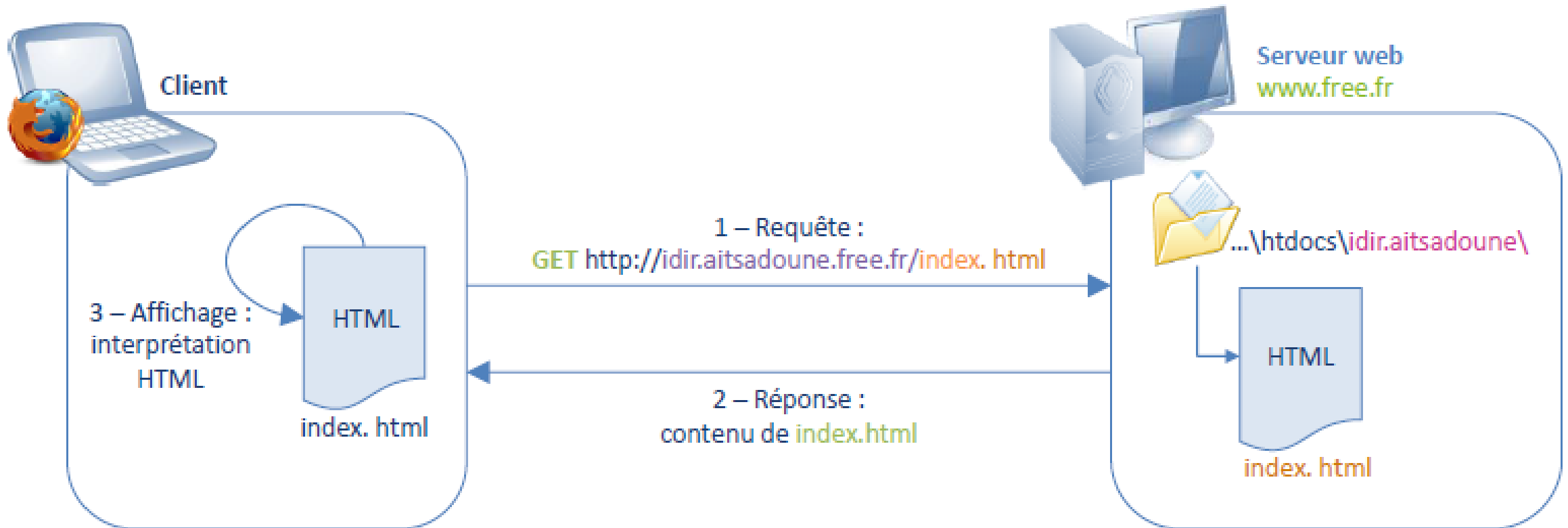


HTTP (Hyper Text Transfer Protocol)

Protocole de communication dédié au web

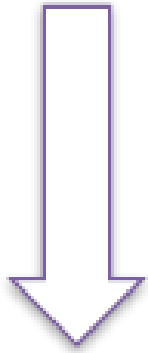
0 Chaque ressource du web est identifiée par une URL

0 Mode de communication = requête / réponse



Web Services : Types

Web Services 1.0



Web Services 2.0

“SOAP Web Services”

- Utilisent le **protocol SOAP** (Simple Object Access Protocol)
- **XML** (Extensible Markup Language)

“RESTful Web Services”

- Utilisent le **protocol REST** (Representational State Transfer)
- Utilisent les méthodes standards du **HTTP** (**GET, PUT, POST, DELETE**) (HTTP: Hypertext Transfer Protocol)
- Utilisent **JSON** (JavaScript Object Notation) or **XML**

Implémentation des Services Web (1)

Côté fournisseur

Pour créer un Web Service :

1. Définir le contrat du service
2. Développer le service
3. Développer la couche de traitement XML
4. Déployer sur le serveur
5. Publier dans l'annuaire



Suivant les technologies,
certaines tâches
sont
automatisées

Implémentation des Services Web (2)

Côté client

Pour créer une application cliente :

1. Rechercher le service dans l'annuaire
2. Récupérer le contrat du service
3. Développer la couche de traitement XML
4. Utiliser le service et présenter les résultats (rendu)



Suivant les technologies, certaines tâches sont automatisées

Tâches relatives aux Services Web

- **Invocation** : vise à établir la communication entre le client et le fournisseur en décrivant la structure des messages échangés.
- **Découverte** : permet de localiser un service web particulier dans un annuaire de services décrivant les fournisseurs ainsi les services fournis.
- **Composition** : consiste à combiner des services pour former un nouveau service dit composé ou composite. Le but de la composition est avant tout la réutilisation de services (simples ou composés) et de préférence sans aucune modification de ces derniers.
- **Sélection** : choisir parmi les services web découverts, ceux qui répondent au mieux aux exigences de l'utilisateur sur la base des besoins fonctionnels et/ou non fonctionnels,