



**Brevet de Technicien Supérieur**  
**Centre: Lycée Technique - Taza**



**Filière: Multimédia et Conception Web**

**Niveau: 2<sup>ème</sup> année**

**Module:**

# Réseaux Informatiques

**Réalisé par:**

**Pr. H. EL BOURAKKADI**

**hamid.elBourakkadi1@usmba.ac.ma**

A.F. 2021-2022

# Plan du cours

- **Chapitre 1: Introduction aux réseaux informatiques**
- **Chapitre 2: Modèle OSI**
- **Chapitre 3: Techniques d'adressage d'un réseau local**
- **Chapitre 4: Service DHCP**
- **Chapitre 5: Service DNS**
- **Chapitre 6: Service Web (IIS 8.0)**

# Sommaire

- Pourquoi un service Web?
- Qu'est ce qu'un service web?
- Protocole HTTP
- Service IIS

# Pourquoi un service Web?

- Il arrive fréquemment qu'une **application** ait **besoin** d'aller chercher des données (**dynamiques**) sur une autre machine, en passant à travers un réseau.

**Exemples:**

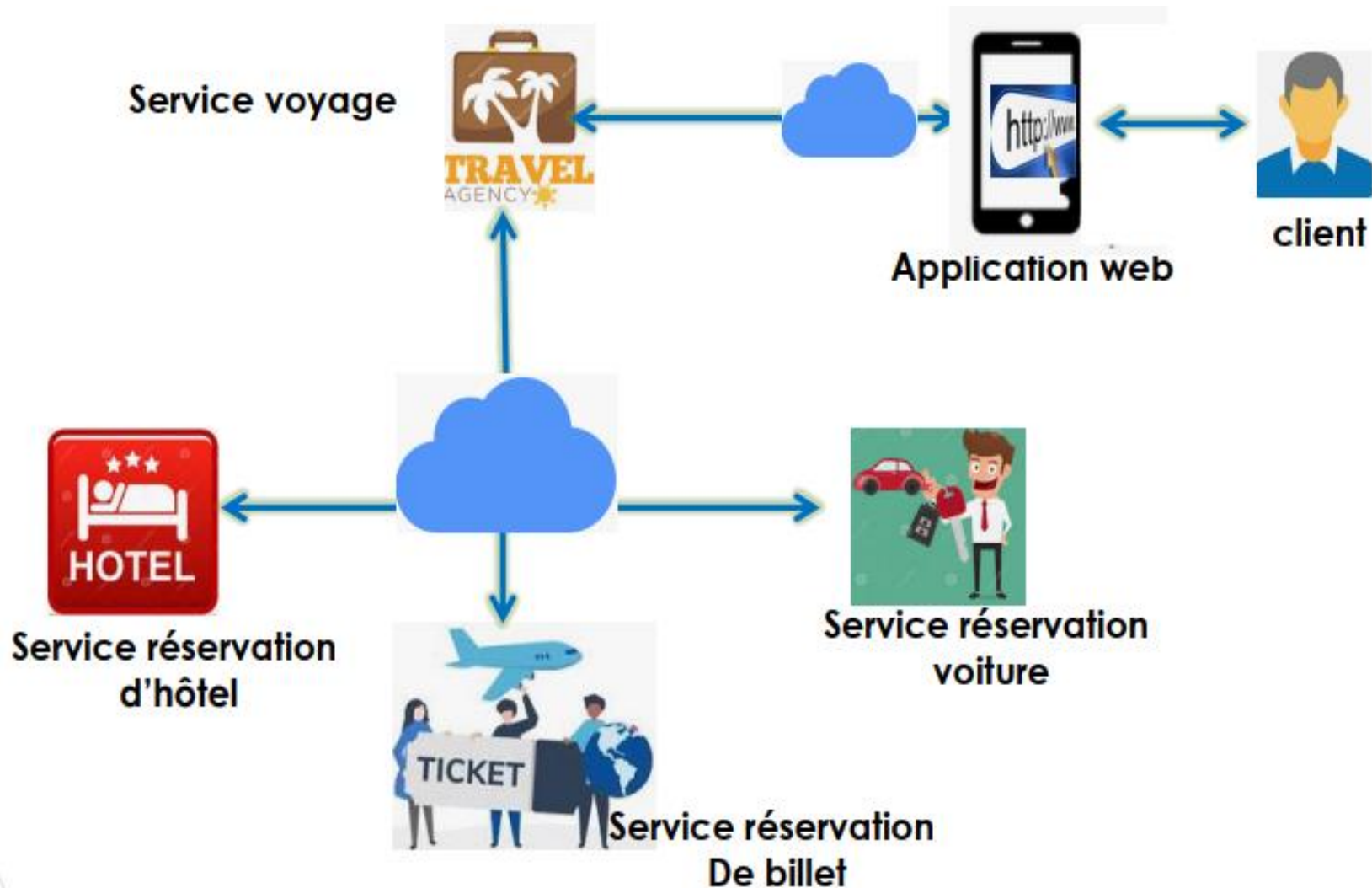
- données météorologiques d'une date précise,
- données de géolocalisation d'un endroit via son adresse.
- tout les **derniers** films qui sont disponibles au cinéma.

- Il arrive qu'un programme tournant sur une machine spécifique ait **besoin** de faire **exécuter une opération très coûteuse en temps** sur une autre machine plus puissante.

# Pourquoi un service Web?

- Application pour une agence de voyage, qui produit des voyages, un de ces derniers en réalité est une combinaison de plusieurs produits:
  1. Gestion de réservation des billets de transport
  2. Gestion de réservation des hôtels
  3. Gestion de réservation des voiture de location....
- L'élaboration d'un produit voyage est bien le résultat de données récupérées auprès de différents fournisseurs:
  1. Compagnies aérienne
  2. Chaîne hotellières
  3. Loueurs de véhicules...

# Pourquoi un service Web?



# Pourquoi un service Web?

- La réponse a ces besoins rentre dans le cadre des **applications distribuée**, avec une architecture où le traitement des données des applications est distribué sur plusieurs machines en réseau.
- Les technologies telles que ***RMI, DCOM et CORBA*** ont précédemment adopté ce style architectural mais ont généralement échoué en raison de:
  1. La diversité des plates-formes utilisées dans les organisations
  2. Leur usage n'était pas adapté à Internet (problème de passage à travers des FireWalls, etc.) d'où la lenteur, voire l'absence de réponses sur ce réseau.
  3. Les applications réparties fondées sur ces technologies offrent des solutions caractérisées par un couplage fort entre les objets.

# Pourquoi un service Web?

- A contrario, le **web** impose des moyen de communication beaucoup plus sommaire:
  1. Débit plus faible
  2. Caractéristique de **http** bien inférieur à CORBA ou RMI
  3. Configuration client est très légère (un simple navigateur)
  4. Protocole **HTTP** (universellement mise en œuvre ) devient le moyen d'échange incontournable entre machines connectés sur le web.



# Pourquoi un service Web?

- Bien que l'architecture distribuée est devenu indispensable pour répondre au besoins des applications d'entreprise.

**Question:** Est-ce qu'on peut combiner:

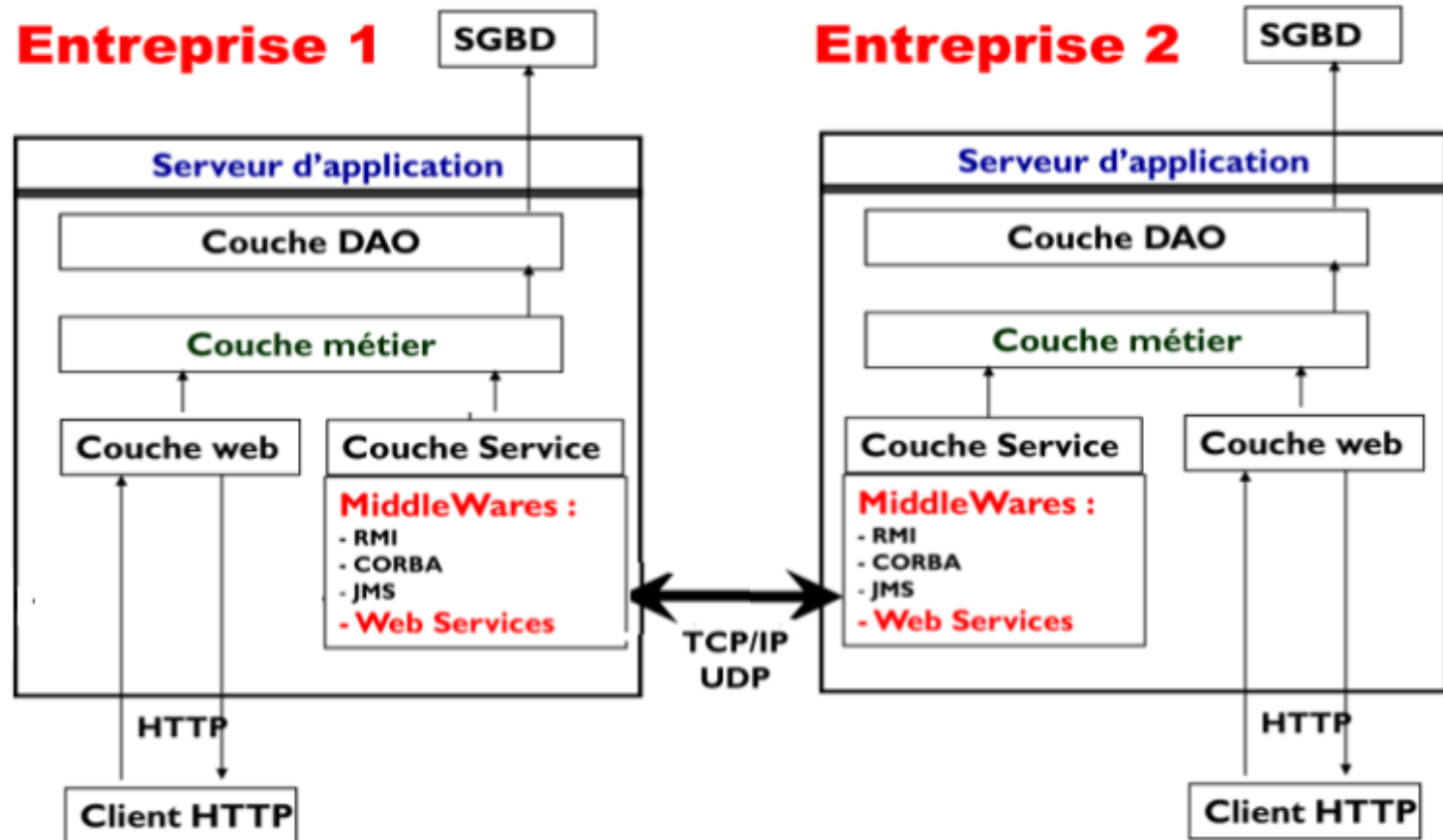
1. Les caractéristiques des technologies ( distribuer) comme CORBA ou RMI
2. Les contraintes imposées par le web à savoir (HTTP, XML)?



**Le service web permet d'atteindre cet objectif**

# Pourquoi un service Web?

La localisation de services web dans une Architecture distribuée



# C'est quoi un service Web?

Les services web matérialise deux perception dans l'évolution des architectures:

1. **Passage** du web client au web machines
2. **Adaptation** des architectures distribuées au monde web



1. **web utilisateur**; navigateur  
HTML + contenu dynamique sur  
le serveur  
2. Technologies distribuées:  
**CORBA, RMI, RPC, DCOM...**

1. **web machine**: communication  
directe entre applications  
2. Technologies web: **HTTP**, et  
**XML**

# Principe de base d'un service Web?

- Repose sur le transport d'une demande de service entre client et un serveur, le transport est assuré par la suite de protocoles internet (TCP/IP et **HTTP**)
- Les messages (texte) transportés (requête et réponse) entre clients et serveur sont au format **XML**. (Ce format est une représentation de base **portable sur toute plate forme**).



# Principe de base d'un service Web?

- Une service web est un **composant web** basé sur internet (**HTTP**) qui exécute des tâches précises et qui respecte un format spécifique (**XML**)
- Il permet aux applications de faire appel à des fonctionnalités à distances en simplifiant ainsi l'échange de données .
- Permettent aux applications de faire dialoguer à travers le réseau, **indépendamment de : Leurs plates forme d'exécution Et de leurs langages d'implémentation**
  - ☐ Permet d'accéder aux application à travers **les pare-feux**
  - ☐ Il sont adaptés pour l'assemblage de **composants faiblement couplés**.



# Exemples de services Web?

- **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)

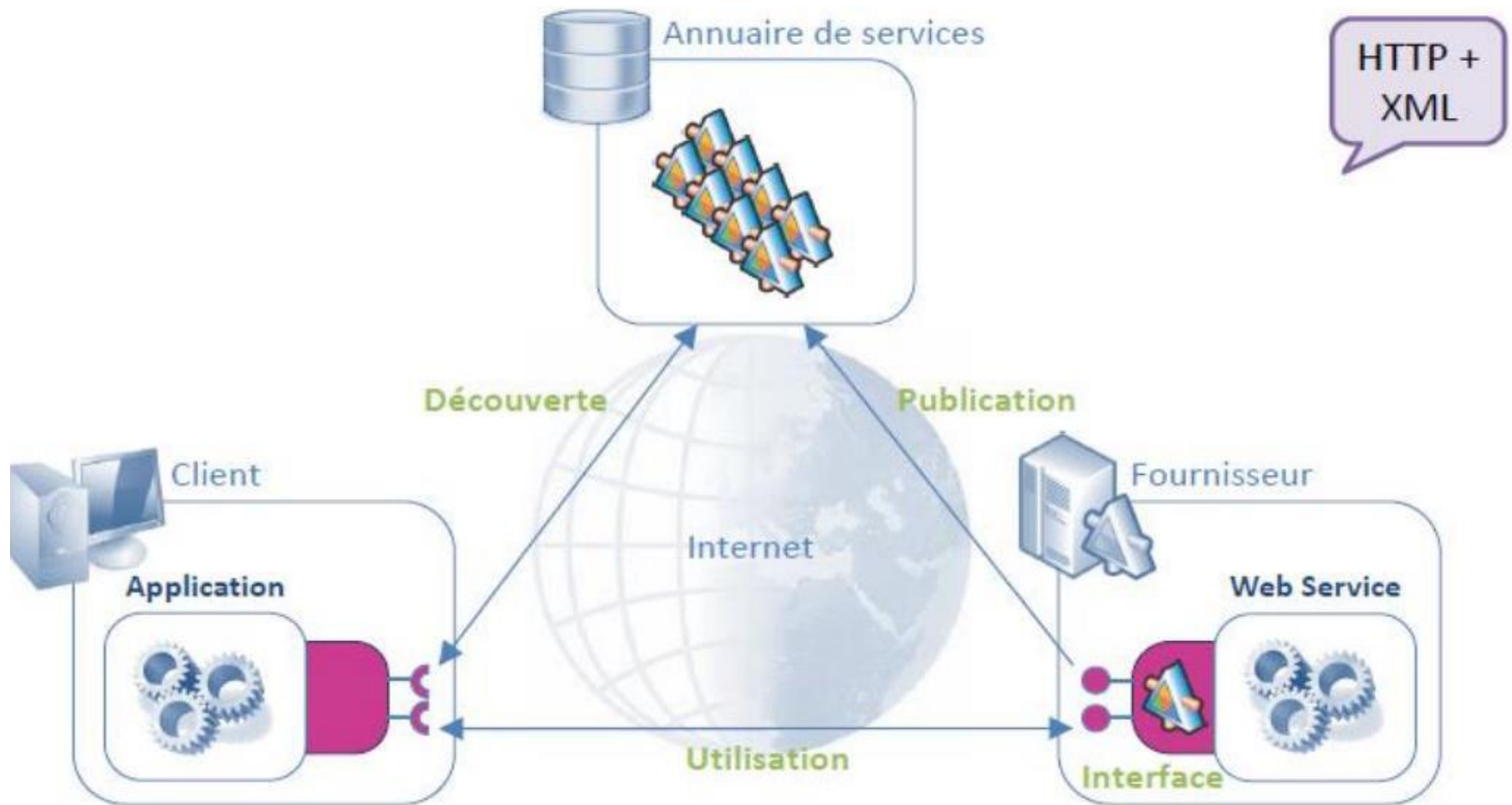


Bien d'autres ( <http://webservicex.net> par exemple)

# Qu'elles sont les acteurs du service Web?

- Les principaux **acteurs** dans la technologie des services web sont:
  - ✓ **Le fournisseur**: celui qui fournit le service web; il est représenté par un serveur d'application (exp: J2EE ). Il propose l'implémentation d'un service web, il fournit une description de ce dernier. Enfin, il publie cette description dans un annuaire.
  - ✓ **L'annuaire**: dans un serveur web, un registre (annuaire) qui détient les informations (la description) du service web
  - ✓ **Le client**: celui qui utilise, invoque le service, après une recherche faite dans l'annuaire.

# Architecture des services Web?





# Scénario complet

**Etape1: définition, description du service web:** Définir (en informatique) ce que fait le service, la description est faite en langage WSDL au sein du fournisseur.

**Etape2: publication du service:** Le service est déclaré dans un annuaire (UDDI), on parle alors de publication du service afin de le rendre accessible aux clients

**Etape3: Recherche du service:** Le client se connecte, sur un annuaire (UDDI) pour effectuer une recherche d'un service

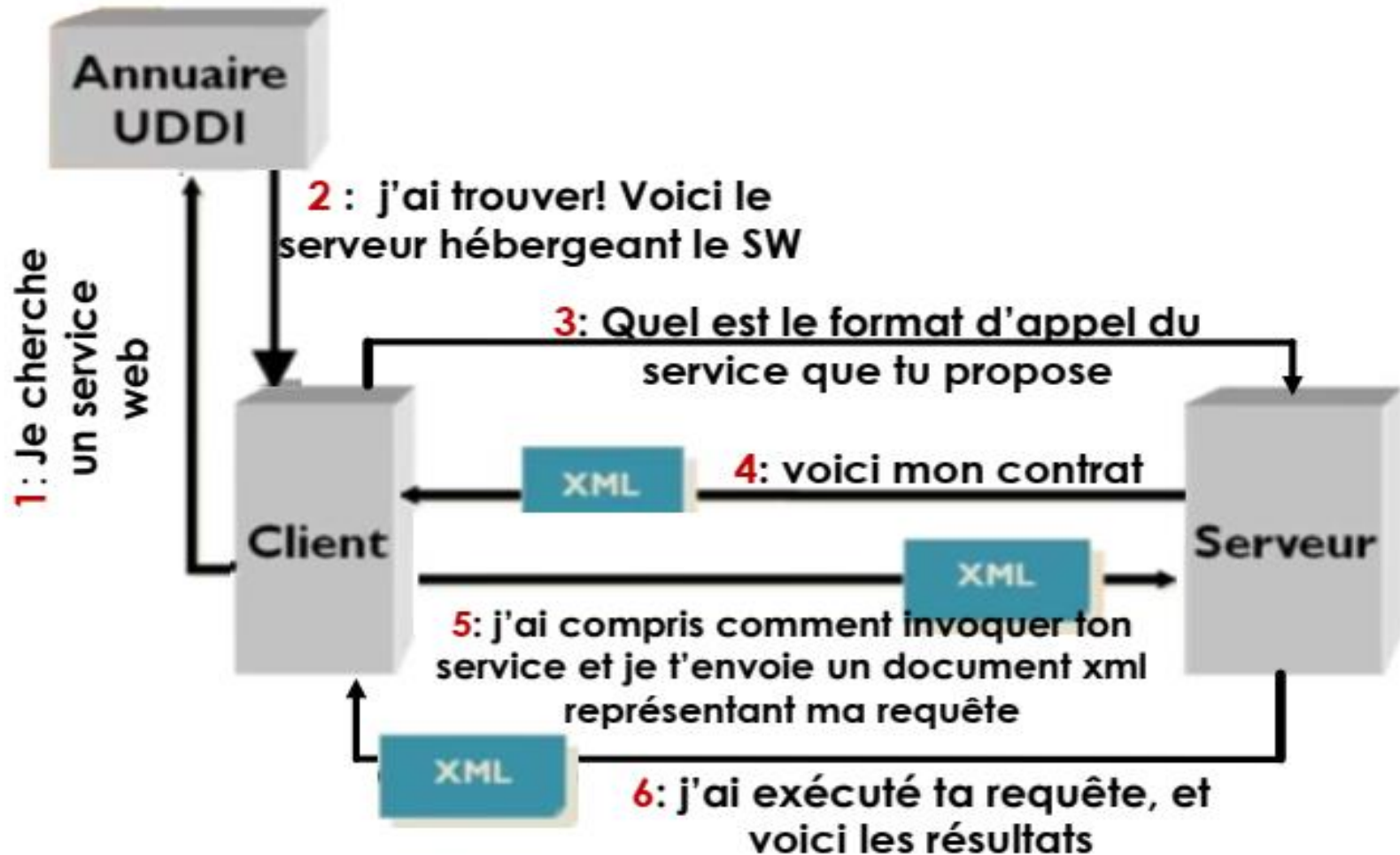
**Etape4: enregistrement au service web:** Une fois le service est trouvé par le client, ce dernier doit s'enregistrer auprès du fournisseur associé au service. Cet enregistrement indique au fournisseur l'intention du client d'utiliser le service suivants les conditions décrites dans la description WSDL.

**Etape5: invocation du service:** Le client peut invoquer le service suivant les conditions inscrites au sein du WSDL

**Etape6: composition du service:** C'est la possibilité de combiner plusieurs services. En fait, un service peut devenir le client d'un autre service

# Scénario complet

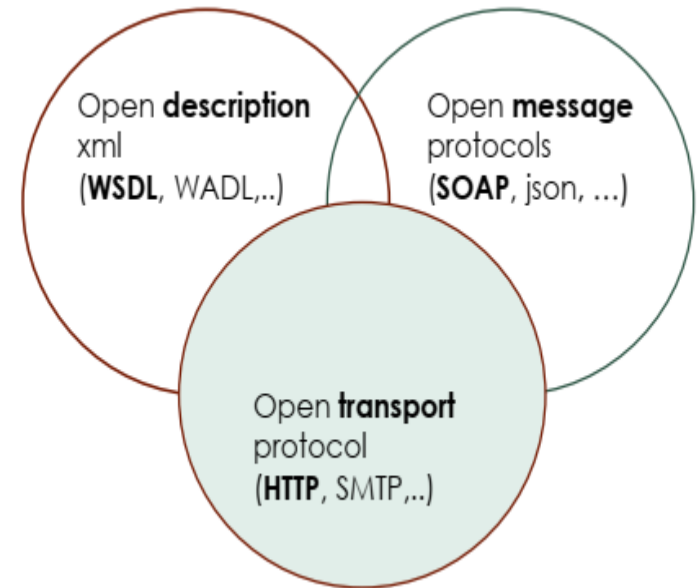
## Scénario des étapes 3, 4 et 5



# Composants d'un service web

Afin de mettre en œuvre un service web, trois composants sont nécessaires :

- **Un protocole pour décrire le service** : il permet –entre autres- d'énumérer les méthodes disponibles, ainsi que leurs signatures (nous étudierons plus tard WSDL et WADL) ;
- **Un protocole pour écrire les message**
- **Un protocole de transport**, afin de faire circuler les informations sur Internet.



# Protocole HTTP

## Principe de Base de HTTP?

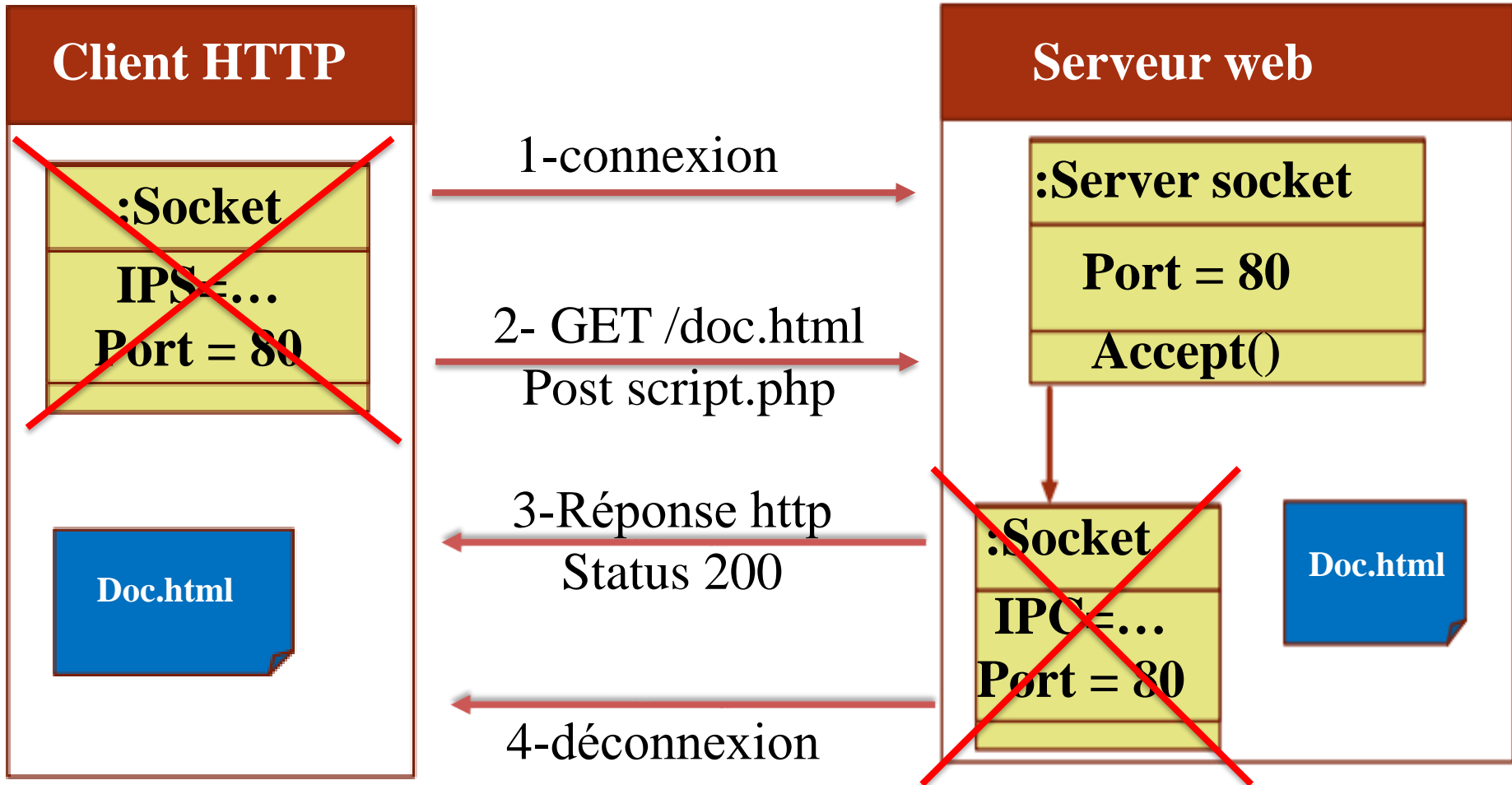
- Quelles sont les méthodes http?
- Exemple Requête HTTP, réponse HTTP
- Les Entêtes HTTP

## Comment fonctionne le protocole HTTP

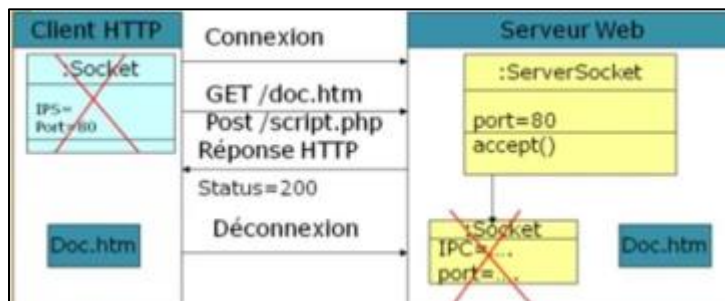
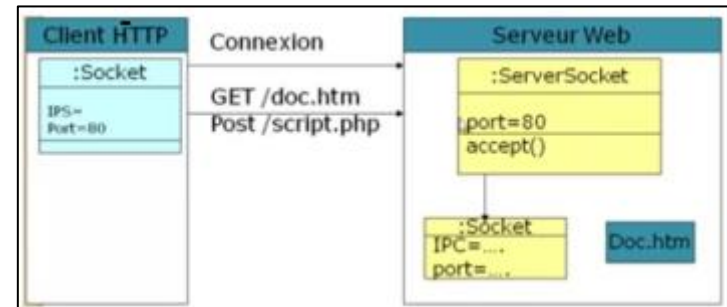
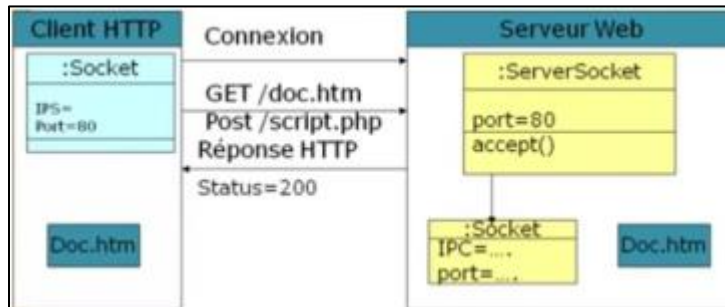
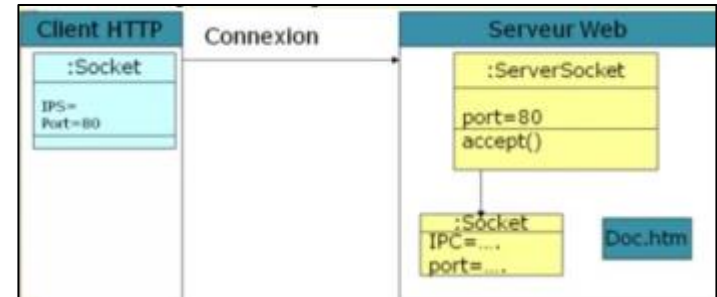
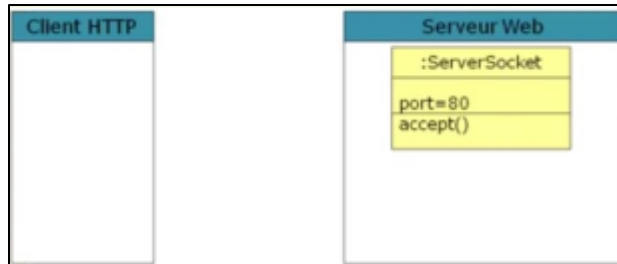
# Comment fonctionne le protocole HTTP

- **HTTP : Hyper Text Transfert Protocol**
  1. Permet au client de récupérer des document du serveur
  2. Ces document peuvent être statique ou dynamique
  3. Permet de soumissionner des formulaires
- **Fonctionnement**
  1. Le client se connecte au serveur (créer une socket)
  2. Le client demande au serveur un document: requête http
  3. Le serveur renvoi au client le document (status 200) ou d'une erreur (document n'existe pas)
  4. déconnexion

# Comment fonctionne le protocole HTTP



# Comment fonctionne le protocole HTTP



# Qu'elles sont les méthodes de HTTP

- Une requête http peut être envoyée en utilisant les méthodes suivants:
  1. **GET** : pour récupérer le contenu d'un document
  2. **POST**: pour soumissionner des formulaires (envoyer les données saisies par l'utilisateur)
  3. **PUT**: envoyer un document d'un utilisateur vers le serveur
  4. **DELETE**: demander au serveur de supprimer un document
  5. ....



# Exemple requête/réponse HTTP

- Requête http envoyé par client avec : **Get**

**Get/ nom\_script? login= valeur1 &  
pass=....**

**http/1.0**

**Accept :text/html**

**Accept langage: fr**

**User-agent : Mozilla 6.0**



Entête de la  
requête

The diagram shows three overlapping light green rectangular boxes with blue borders. The front-most box is slightly offset to the left and bottom, revealing the ones behind it. The text 'Entête de la requête' is centered in red on the front box.



Corps de la  
requête vide

The diagram shows three overlapping light pink rectangular boxes with red borders. The front-most box is slightly offset to the left and bottom, revealing the ones behind it. The text 'Corps de la requête vide' is centered in red on the front box.

# Exemple requête/réponse HTTP

- Requête http envoyé par client avec : **post**

**Post/** nom\_script http/1.0

**Accept :text/html**

**Accept langage: fr**

**User-agent : Mozilla 6.0**

\*\*\* saut de ligne\*\*\*

**login = valeur1 & pass = valeur2 & val3  
=valeur3**



Entête de la  
requête

The diagram shows three overlapping light green rectangular boxes with blue borders. The front-most box has a wavy bottom edge and contains the text 'Entête de la requête' in red.



Corps de la  
requête

The diagram shows three overlapping light red rectangular boxes with red borders. The front-most box has a wavy bottom edge and contains the text 'Corps de la requête' in red.

# Exemple requête/réponse HTTP

- Le serveur retourne la réponse **http**

```
Http/1.0 200 ok
Date: Wed, 05Feb02 15:02:01 GMT
server: Apache/1.3.24
Last-Modified: Web 02Oct01 24:05:01 GMT
Content-type: Text/Html
Content-length: 4205
```



Entête de la  
réponse

A diagram showing three overlapping light green rectangular cards with blue borders. The top card is slightly offset to the right and top, showing the text 'Entête de la réponse' in red.

\*\*\* saut de ligne\*\*\*

```
<html> <head>
.....
</body></html>
```



Corps de la  
réponse:

A diagram showing three overlapping light pink rectangular cards with red borders. The top card is slightly offset to the right and top, showing the text 'Corps de la réponse:' in red.

*document à afficher*

# Les Entêtes HTTP

Il y a quatre types d'entêtes HTTP à savoir :

- **Général**: commun au serveur, au client, au http
- **Requête client**: format de documents et paramètres pour le serveur
- **Réponse serveur**: information concernant le serveur.
- **Entité**: informations concernant les données échangés

# Les Entêtes HTTP

## 1- Entêtes généraux:

- **Cache-Control** --> contrôle au cache
- **Connection** → listes d'option
- **Date** → date actuelle
- **MIME-version** → version MIME utilisé
- **Transfer-encoding** → comment le donéne sont envoyées
- .....

# Les Entêtes HTTP

## Entête Requête client:

- Accept → type MIME visualisable par l'agent
- Accept-Encoding → méthode de codage acceptées.
- Accept-Charset → jeu de caractères préféré du client
- Accept-Language → liste de langues
- User-agent → modèle du navigateur
- Référer → URL d'origine
- If-Modified-Since → envoyer la ressource seulement si elle est modifiée.
- ...

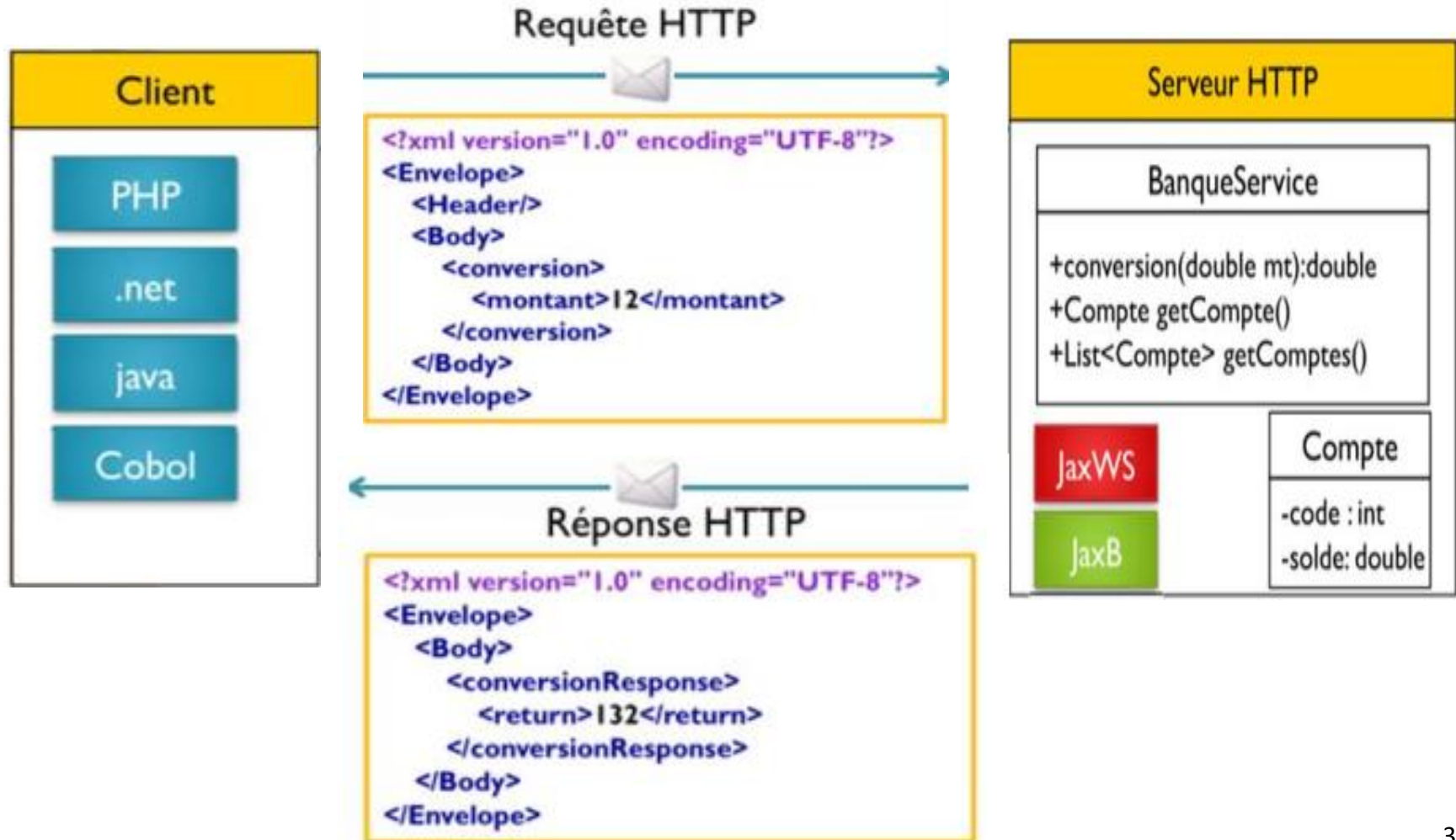
# Les Entêtes HTTP

## Entêtes d'entités:

- Content-Base → URI de base
- Last-Modified → date de dernière modification du doc
- Content-Length → taille du document en octet
- Content-Encoding → type encodage du document envoyé
- Content-Language → langage du document retourné
- ...

# L'idée du web service

L'idée est que le corps de la requête http est codé en **xml** (ou autre format de donnée tel que **json**,...)





# 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,

# Services d'information sur Internet (IIS)

- Internet Information Services (IIS) est un serveur Web flexible et **polyvalent de** Microsoft qui fonctionne sur les systèmes Windows pour servir les pages ou **fichiers HTML** demandés.
- Un serveur Web IIS accepte les demandes provenant d'ordinateurs clients distants et renvoie la réponse appropriée. Cette fonctionnalité de base permet aux serveurs Web de partager et de fournir des informations à travers les réseaux locaux (**LAN**), tels que **les intranets d'entreprise**, et les réseaux de grande surface (**WAN**), tels que **l'Internet**.
- Un serveur Web peut fournir des informations aux utilisateurs sous plusieurs formes, telles que des pages Web statiques codées en HTML; par le biais d'échanges de fichiers sous forme de téléchargements et de téléchargements; et des documents texte, des fichiers d'images et plus encore.

# Services d'information sur Internet (IIS)

## Les serveurs Web fournissent des portails

Les serveurs Web modernes peuvent fournir beaucoup plus de fonctionnalités pour une entreprise et ses utilisateurs. Les serveurs Web sont souvent utilisés comme portails pour des applications web **sophistiquées, hautement interactives**, qui lient les applications **intermédiaires d'entreprise** et back-end pour créer des systèmes de classe entreprise. Par exemple, Amazon Web Services permet aux utilisateurs d'administrer des **ressources cloud** publiques via un portail Web. Pendant ce temps, les services de médias en continu, tels que Spotify pour la musique et Netflix pour les films, fournissent du contenu en streaming en temps réel via des serveurs Web.

# Fonctionnement de l'IIS

- IIS fonctionne à travers une variété de langues et protocoles standard. HTML est utilisé pour créer des éléments tels que le texte, les boutons, les emplacements d'image, les interactions/comportements directs et les hyperliens. Le protocole de transfert Hypertext (**HTTP**) est le protocole de communication de base utilisé pour échanger des informations entre les serveurs Web et les utilisateurs. **HTTPS** -- HTTP over Secure Sockets Layer (SSL) -- utilise **Transport Layer Security** ou **SSL** pour chiffrer la communication pour plus de sécurité des données. Le protocole de transfert de fichiers(**FTP**),ou sa variante sécurisée, FTPS, peut transférer des fichiers.
- D'autres protocoles pris en charge incluent le protocole de transfert de courrier simple (**SMTP**), pour envoyer et recevoir des e-mails, et le protocole de transfert de nouvelles réseau, pour livrer des articles **sur Usenet**.

# Fonctionnement de l'IIS

## IIS travaille avec ASP.NET Core

Le **ASP.NET** Core est la dernière génération de Active Server Page(**ASP**), un moteur de script côté serveur qui produit des pages Web interactives. Une demande arrive sur le serveur IIS à partir du web, qui envoie la demande à l'application ASP.NET Core, qui traite la demande et renvoie sa réponse au serveur IIS et au client à l'origine de la demande. Des exemples d'applications écrites sur ASP.NET Core incluent les plateformes **de blog** et les systèmes de gestion de contenu (**CMS**).

Les développeurs peuvent produire des sites Web IIS avec un certain nombre d'outils, **y compris WebDAV**, qui peut créer et publier du contenu Web. Les développeurs peuvent également utiliser des outils de développement intégrés, tels que Microsoft Visual Studio.

# Versions d'IIS

IIS a évolué avec Microsoft Windows. Les premières versions de l'IIS sont **arrivées avec Windows NT**. IIS 1.0 est apparu avec Windows NT 3.51, et a évolué à travers IIS 4.0 avec Windows NT 4.0. IIS 5.0 expédiés avec **Windows 2000**. Microsoft a ajouté IIS 6.0 à Windows Server 2003. IIS 7.0 a offert une refonte majeure **avec Windows Server 2008** (IIS 7.5 est dans Windows Server 2008 R2). IIS 8.0 est venu **avec Windows Server 2012** (Windows Server 2012 R2 utilise IIS 8.5). Et IIS 10 est arrivé **avec Win Server 2016** et **Windows 10**. Avec chaque itération d'IIS, Microsoft a ajouté de nouvelles fonctionnalités et mis à jour les fonctionnalités existantes. Par exemple, IIS 3.0 a ajouté ASP pour le script dynamique; IIS 6.0 a ajouté une prise en charge de **l'IPv6** et amélioré la sécurité et la fiabilité; et l'IIS 8.0 a apporté une mise à l'échelle multicœur **sur le matériel d'accès à** la mémoire non uniforme, le support centralisé du certificat SSL et l'indication du nom du serveur.

# Caractéristiques de l'IIS 10

- IIS 10 ajoute également un certain nombre de nouvelles fonctionnalités et fonctionnalités.
- IIS 10 ajoute une prise en charge du protocole HTTP/2, afin d'offrir une utilisation plus efficace des ressources et une latence plus faible que HTTP 1.1. IIS 10 fonctionne sur le modèle de déploiement minimal **du serveur Nano Server** sous Windows Server 2016, et peut exécuter des charges de travail ASP.NET Core, Apache **Tomcat et PHP** sur IIS sur le Nano Server.
- IIS 10 fonctionne dans un conteneur et **une machine virtuelle**, de sorte que les développeurs et les administrateurs ont plus de flexibilité dans les choix de déploiement, ainsi que la densité pour accueillir un large éventail d'applications Web.

# Installation et de configuration de l'IIS

## En utilisant le gestionnaire de serveur:

- Appuyez sur la clé [Windows] et sélectionnez Server Manager.
- Dans le tableau de bord Server Manager, cliquez sur gérer > ajouter des rôles et des fonctionnalités.
- Cliquez sur le type d'installation.
- Sélectionnez l'option d'installation basée sur les fonctions ou basée sur les fonctionnalités et cliquez sur Suivant.
- Sélectionnez le serveur sur lequel l'IIS sera installé et cliquez sur Suivant.
- Activez le rôle du serveur Web (IIS).



# Installation et de configuration de l'IIS

- Cliquez sur Ajouter des fonctionnalités pour ajouter la console de gestion IIS.
- Cliquez ensuite : la fenêtre Select Features s'ouvrira.
- Cliquez ensuite : la fenêtre Rôle du serveur Web (IIS) s'ouvrira.
- Cliquez ensuite : la fenêtre Sélectionner les services de rôle s'ouvrira.
- Sélectionnez les services de rôle requis et cliquez sur Suivant.
- Cliquez sur Installer pour installer les rôles, les services de rôle et les fonctionnalités sélectionnés.
- Cliquez sur Fermer pour terminer l'installation.

# Installation et de configuration de l'IIS

Voici les étapes pour configurer IIS à l'aide du gestionnaire de serveur :

- Revenez au gestionnaire de serveur et sélectionnez gestionnaire des services d'information Internet (IIS) à partir du menu Outils.
- Cliquez sur Ajouter le site Web
- Spécifiez au moins le nom et le chemin du site. Cliquez sur OK.
- Le premier site est prêt à être consulté.
- **Lorsque l'IIS est installé, il est préconfiguré pour servir de site Web par défaut.**

# Installation et de configuration de l'IIS

pour modifier les paramètres de base du site Web :

- Connectez-vous à l'ordinateur du serveur Web en tant qu'administrateur.
- Cliquez sur Démarrer, pointer vers les paramètres, puis cliquez sur Panneau de contrôle.
- Double-cliquez sur outils administratifs, puis double-cliquez sur Internet Services Manager.
- Cliquez à droite sur le site web pour le configurer dans le volet gauche, puis cliquez sur Propriétés.
- Cliquez sur l'onglet site Web.

# Installation et de configuration de l'IIS

- Tapez une description pour le site Web dans la boîte de description.
- Tapez l'adresse IP (Protocole Internet) pour l'utiliser pour le site Web ou laissez le paramètre par défaut All (Unassigned).
- Modifiez le port du Protocole de contrôle de la transmission, le cas échéant.
- Cliquez sur l'onglet Home Directory.
- Pour utiliser un dossier sur l'ordinateur local, cliquez sur Un répertoire sur cet ordinateur, puis cliquez sur Parcourir pour localiser le dossier que vous souhaitez utiliser.
- Pour utiliser un dossier qui a été partagé à partir d'un autre ordinateur sur le réseau, cliquez sur Une part située sur un autre ordinateur, puis tapez le chemin réseau ou cliquez sur Parcourir pour sélectionner le dossier partagé.
- Cliquez sur Lire pour accorder l'accès lu au dossier (requis).
- Cliquez sur OK pour accepter les propriétés du site Web.

# Installation et de configuration de l'IIS

Suivez ces étapes pour créer un nouveau site Web dans IIS :

- Connectez-vous à l'ordinateur du serveur Web en tant qu'administrateur.
- Cliquez sur Démarrer, pointer vers paramètres, puis cliquez sur Panneau de contrôle.
- Double-cliquez sur outils administratifs, puis double-cliquez sur Internet Services Manager.
- Cliquez sur Action, pointez vers Nouveau, puis cliquez sur le site Web.
- Après le début de l'assistant de création du site Web, cliquez sur Suivant.
- Tapez une description pour le site Web. (Cette description est utilisée à l'interne pour identifier le site Web dans Internet Services Manager uniquement.)

# Installation et de configuration de l'IIS

- Sélectionnez l'adresse IP à utiliser pour le site. La sélection de tous (non attribués) rendra le site accessible sur toutes les interfaces et toutes les adresses IP configurées.
- Tapez le numéro de port TCP pour publier le site.
- Tapez le nom de l'en-tête hôte (le vrai nom qui est utilisé pour accéder à ce site).
- Cliquez ensuite.
- Soit tapez le chemin vers le dossier qui contient les documents du site Web ou cliquez sur Parcourir pour sélectionner le dossier, puis cliquez sur Suivant.
- Sélectionnez les autorisations d'accès pour le site Web, puis cliquez sur Suivant.
- Cliquez sur Finition.

# IIS vs Apache

Les différences entre IIS et Apache incluent :

- IIS est emballé avec Windows tandis qu'Apache est gratuit **et open source**.
- Alors que l'IIS ne fonctionne que sous Windows, Apache peut fonctionner sur presque n'importe quel système d'exploitation, y compris macOS, **UNIX** et Linux (il est le mieux adapté à Linux).
- IIS s'intègre à d'autres offres Microsoft, telles que **.NET** et le langage de script ASPX.
- IIS dispose d'un service d'assistance pour gérer la plupart des problèmes tandis que la prise en charge d'Apache provient de la communauté des utilisateurs.
- Les caractéristiques de sécurité de l'IIS en font une option plus sûre qu'Apache.
- La technologie sous-jacente d'IIS est compatible avec les interfaces Web standard dans le monde entier.

**Voir TP12**