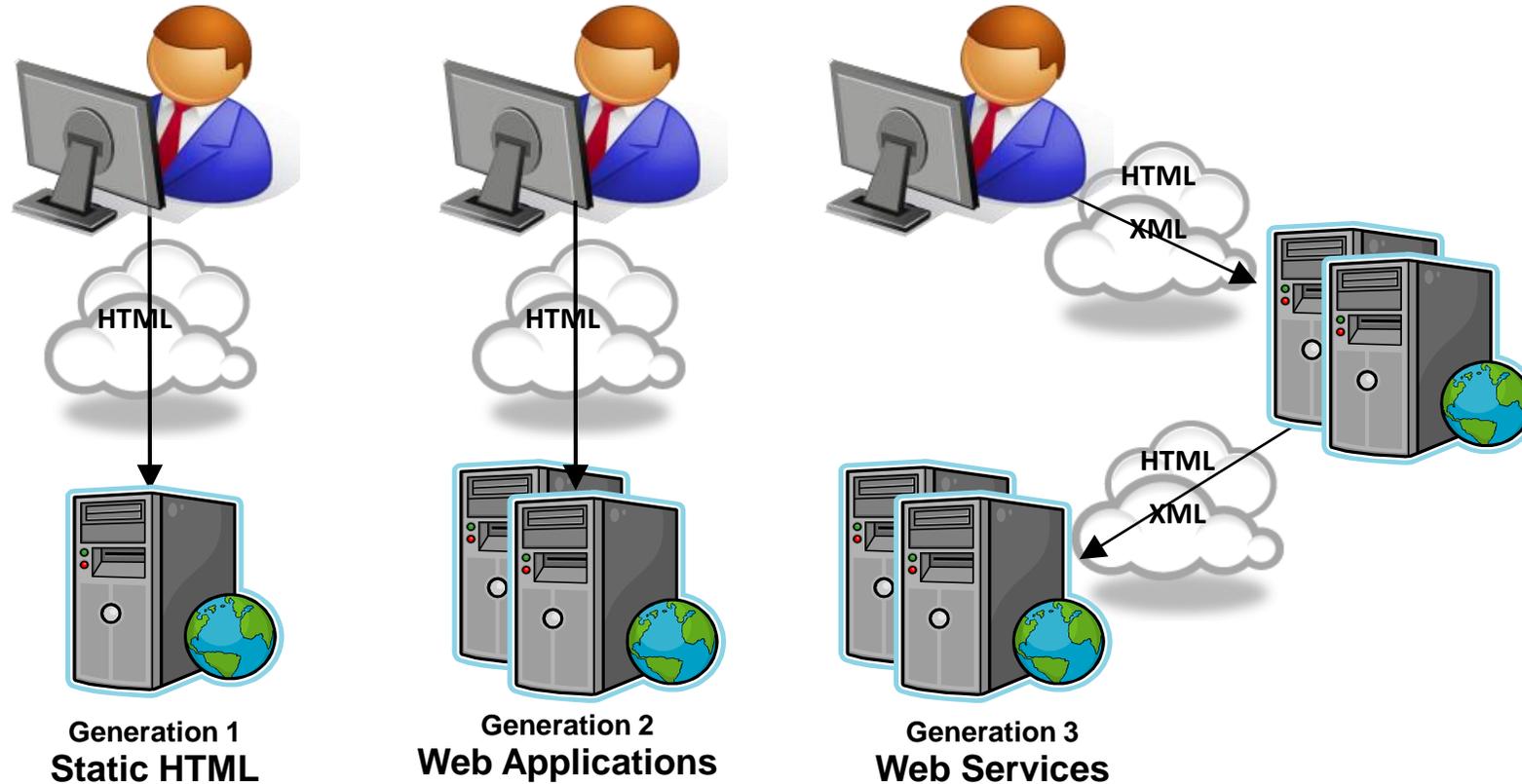


# Introduction aux Services Web

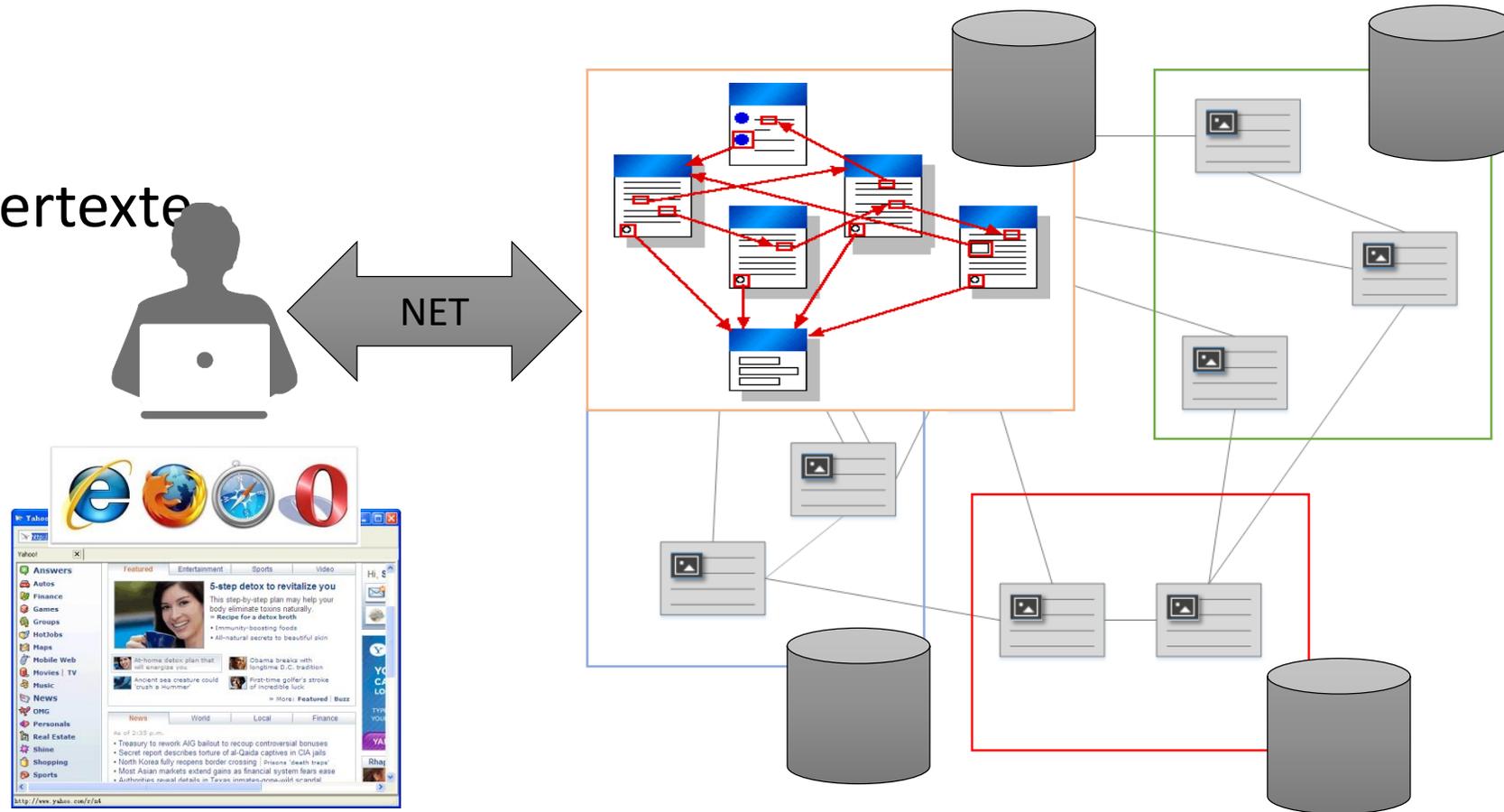
# Rappel : du Web H2M au M2M



**Machine to Machine (M2M)**

# Principes du Web Statique H2M

- Serveurs,
- Pages
- Liens hypertexte



# Adressage : URL - Uniform Resource Locator

- Une URL (Uniform Resource **Locator**) a au moins deux champs (protocole, adresse de la ressource)
  - **Le protocole**: schéma de représentation
  - **L'Adresse** : localisation complète de la ressource
  - Ex:
    - mailto:Quidam.no-spam@example.com
    - news:fr.comp.infosystemes.www.auteurs
- Une URL HTTP a au moins trois champs (protocole, adresse, emplacement) :
  - **Le protocole**: *http* suivi de :
  - **L'adresse**: le nom complet de la ressource // *login : password @ nom domaine : port*
  - **Emplacement**: Emplacement de la ressource à l'adresse
  - Données supplémentaires optionnelles transmises

Exemple :

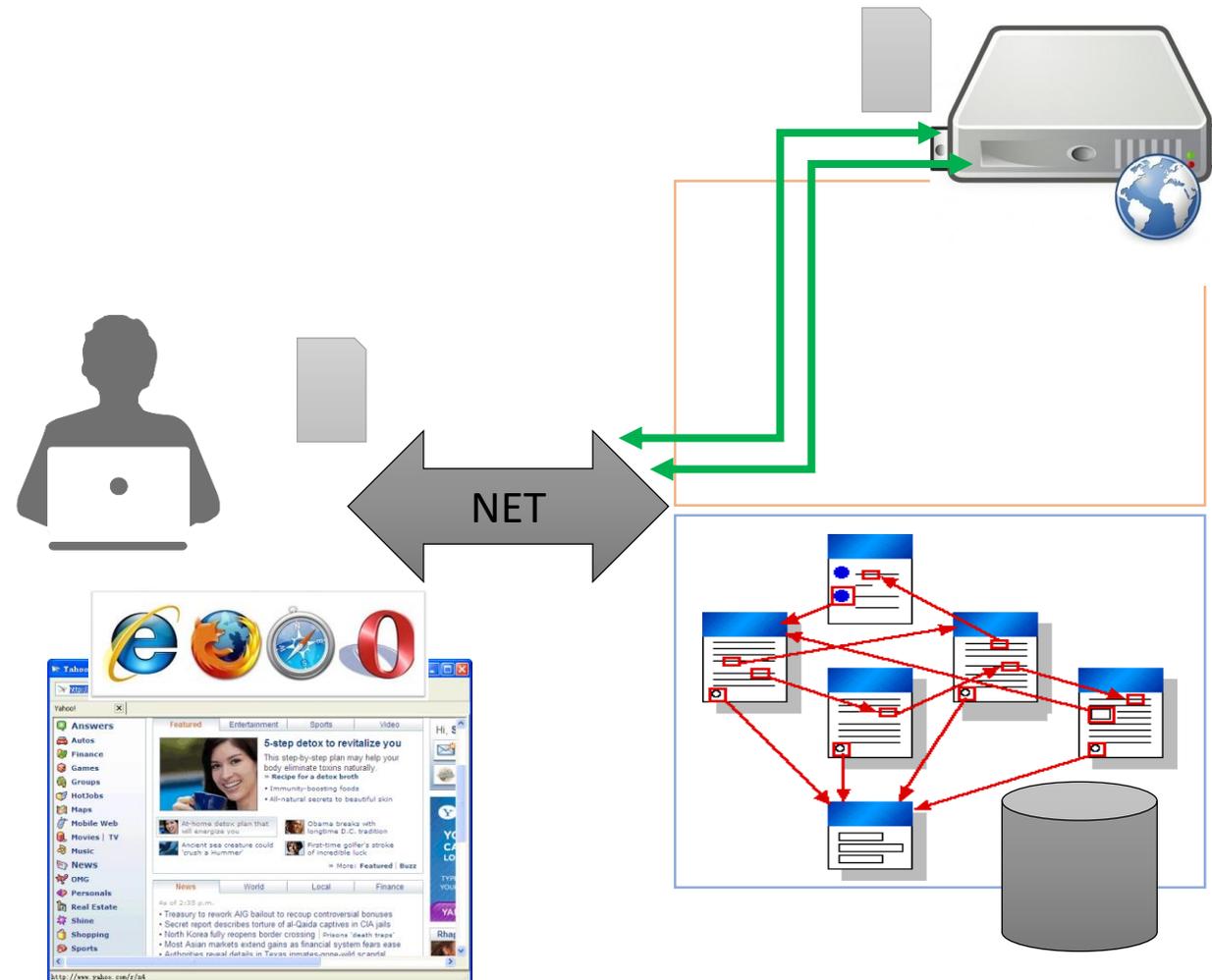
<http://Jojo:IApIn@www.example.com:8888/chemin/d/acc%C3%A8s.php?q=req&q2=req2#signet>

# Client / Serveur Particuliers

- Page Web:
  - Pointés par une URL
  - La plupart des pages WEB se composent de:
    - Une page HTML de base,
    - Différentes références à des « objets »
- L'agent utilisateur (client) pour le Web s'appelle un "**browser**" (navigateur en français)
  - Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari, Google Chrome, ...
- Un serveur pour le Web s'appelle un serveur Web :
  - Apache, Microsoft Internet Information Server (IIS), ...

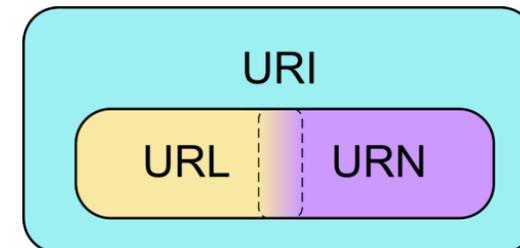
# Principes du Web Dynamique H2M

- Serveurs,
- Pages
- Liens hypertexte



# Adressage : Urx - Uniform Resource ...

- Un URN (Uniform Resource **Name**)
  - Identifie une ressource par un nom dans un espace de nommage (identifie la ressource et pas sa localisation)
  - urn:NID:NSS
  - Peu utilisé, voir [Plugins spécialisés](#)
- Plus généralement un URI (Uniform Resource Identifier)
  - Peut être une URL ou un URN



Référence : W3C , URIs, URLs, and URNs: Clarifications and Recommendations,  
<http://www.w3.org/TR/uri-clarification/>

# Pour quoi faire ?

- Les Services Web permettent d'interconnecter :
  - Différentes entreprises
  - Différentes applications
  - Différents clients
  - Différents matériels
  
- Utilisé dans différents cadres:
  - B2B (Business To Business)
  - EAI (Enterprise Application Integration)
  - ...

# Deux types de Services Web

## WS-RESTFuL & WS-SOAP

# Les principes REST

- Le Web aujourd'hui : extraction des ressources / ressources
- Les ressources sont identifiées par des URL
- Dénominations
  - Architectures Orientées Données (DOA)
  - Architectures Orientées Ressources (ROA)

# Les principes REST

- **REpresentational State Transfer**
- Style architectural pas seulement dédié aux architectures orientées services et aux communication entre machines.
- Aucune hypothèse sur les protocoles impliqués, seulement des contraintes
- Les systèmes qui suivent les principes de l'architecture REST sont souvent appelés *RESTful* et s'appuient sur le Web

*Chapitre 5 de la thèse de doctorat "Representational State Transfer (REST)". de Roy Fielding , 2000*

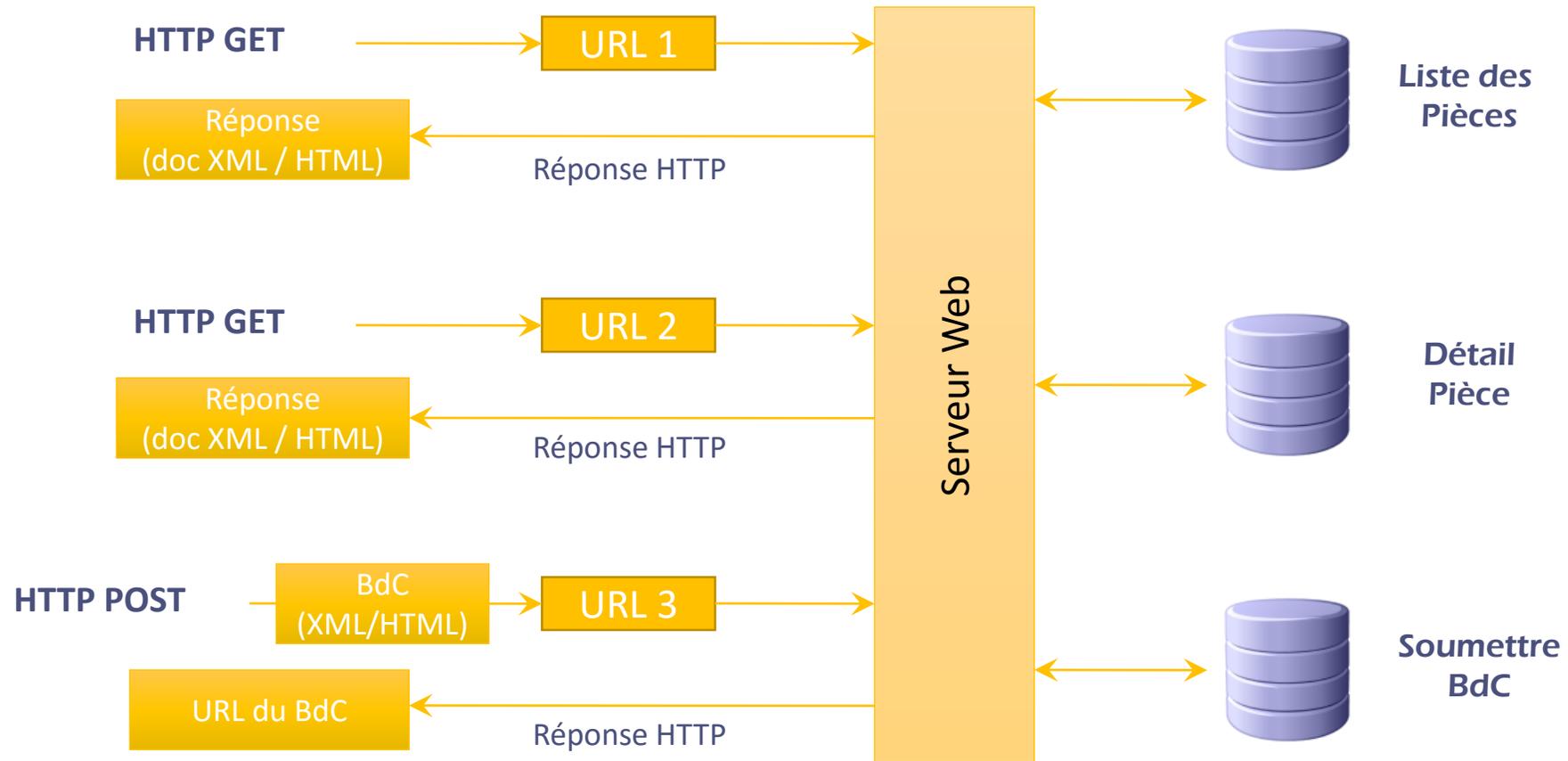
# Les principes REST ou ROA

- Ressources (Identifiant)
  - Entité identifiable dans le système (livre, agenda ...)
  - URI et donc possiblement URL
  - Une URI identifie une seule Ressource
  - Une Ressource peut avoir plusieurs URI
  - Exemple :
    - Emploi du temps de tigli : /edt/prof/tigli/lundi
- Méthodes (Verbes)
  - Quatre opérations de base « CRUD » : Create (créer), Retrieve (lire), Update (mettre à jour), Delete (Supprimer)
  - Exemple méthodes HTTP : GET, POST, PUT, DELETE
  - Déjà adaptées à la manipulation de Ressources
- Représentation (Vue de l'état)
  - Informations transférées entre client et serveur
  - Exemple : XML, JSON, XHTML, CSV ....

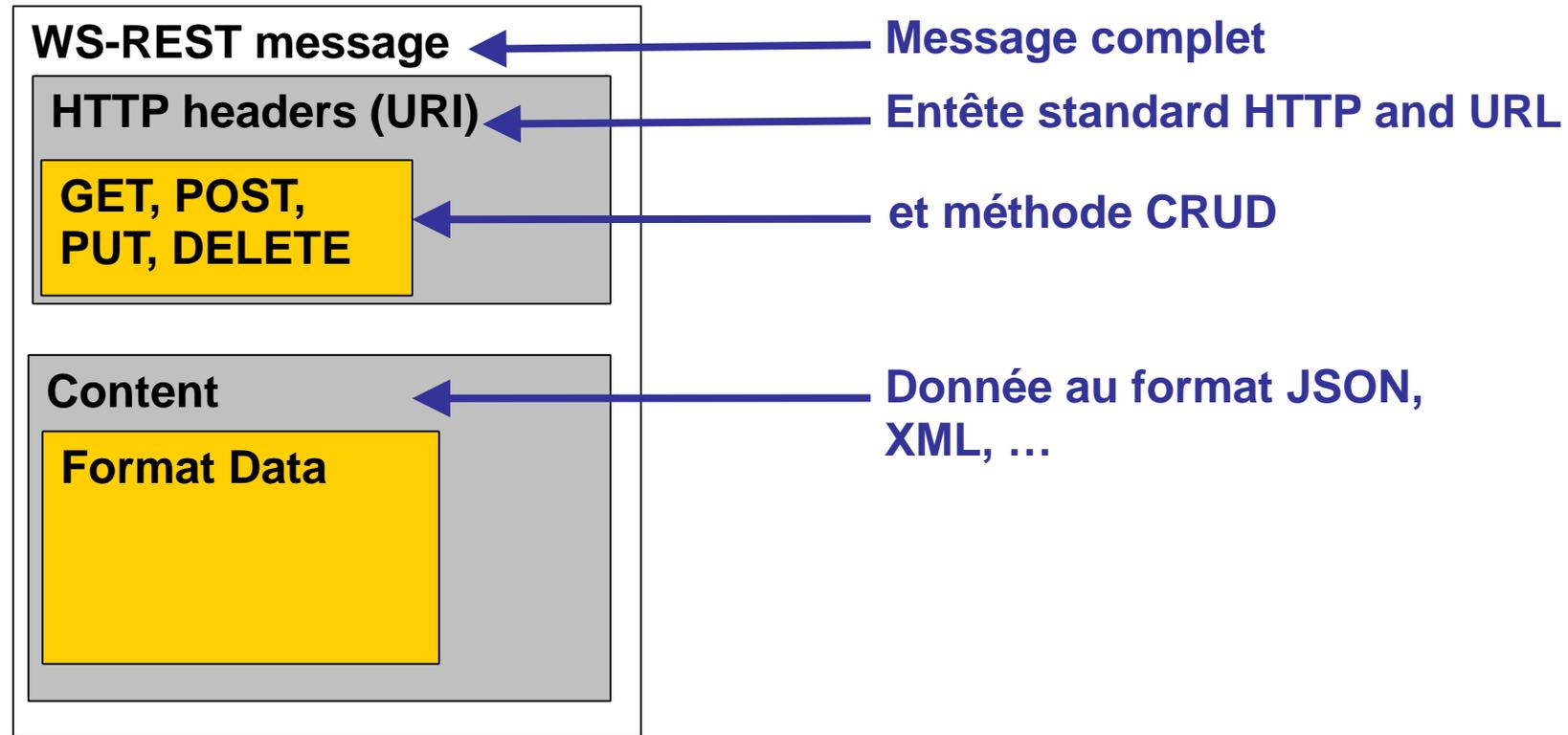
# Cycle de Vie REST

- Plus récemment REST s'est doté d'un langage de description de service : WADL (Web Application Description Language)
- Soumis en 2009 au W3C, il n'est toujours pas standardisé
- Le WADL est un format de fichier basé sur XML qui permet de décrire des applications REST.
- Cette spécification se heurte néanmoins à **la spécification WSDL 2.0, qui elle aussi permet la description de web services REST.**
- De plus, WADL est encore très mal supporté par l'ensemble des frameworks existants ce qui limite son utilisation.

# REST pour une approche ROA



# La structure des messages WS-REST



# Exemple RESTFuL/XML

- Exemple de message HTTP RESTFuL

```
POST http://MyService/Person/
Host: MyService
Content-Type: text/xml; charset=utf-8
Content-Length: 123
<?xml version="1.0" encoding="utf-8"?>
```

*HTTP Header  
Commande POST*

```
<Person>
  <ID>1</ID>
  <Name>M Vaqqas</Name>
  <Email>m.vaqqas@gmail.com</Email>
  <Country>India</Country>
</Person>
```

*HTTP Body  
XML representation  
of a resource « Person »*

# Exemple RESTFul/JSON

- Exemple de message HTTP RESTFul

```
POST http://MyService/Person/
Host: MyService
Content-Type: text/xml; charset=utf-8
Content-Length: 123
<?xml version="1.0" encoding="utf-8"?>
```

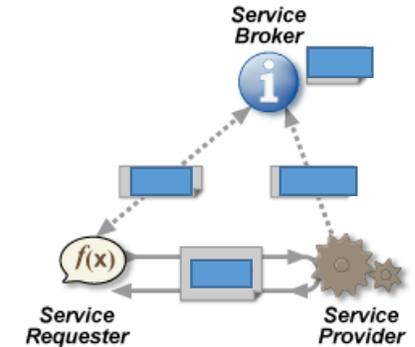
*HTTP Header  
Commande POST*

```
{
  "ID": "1",
  "Name": "M Vaqqas",
  "Email": "m.vaqqas@gmail.com",
  "Country": "India"
}
```

*HTTP Body  
JSON representation  
of a resource*

# Cycle de Vie REST

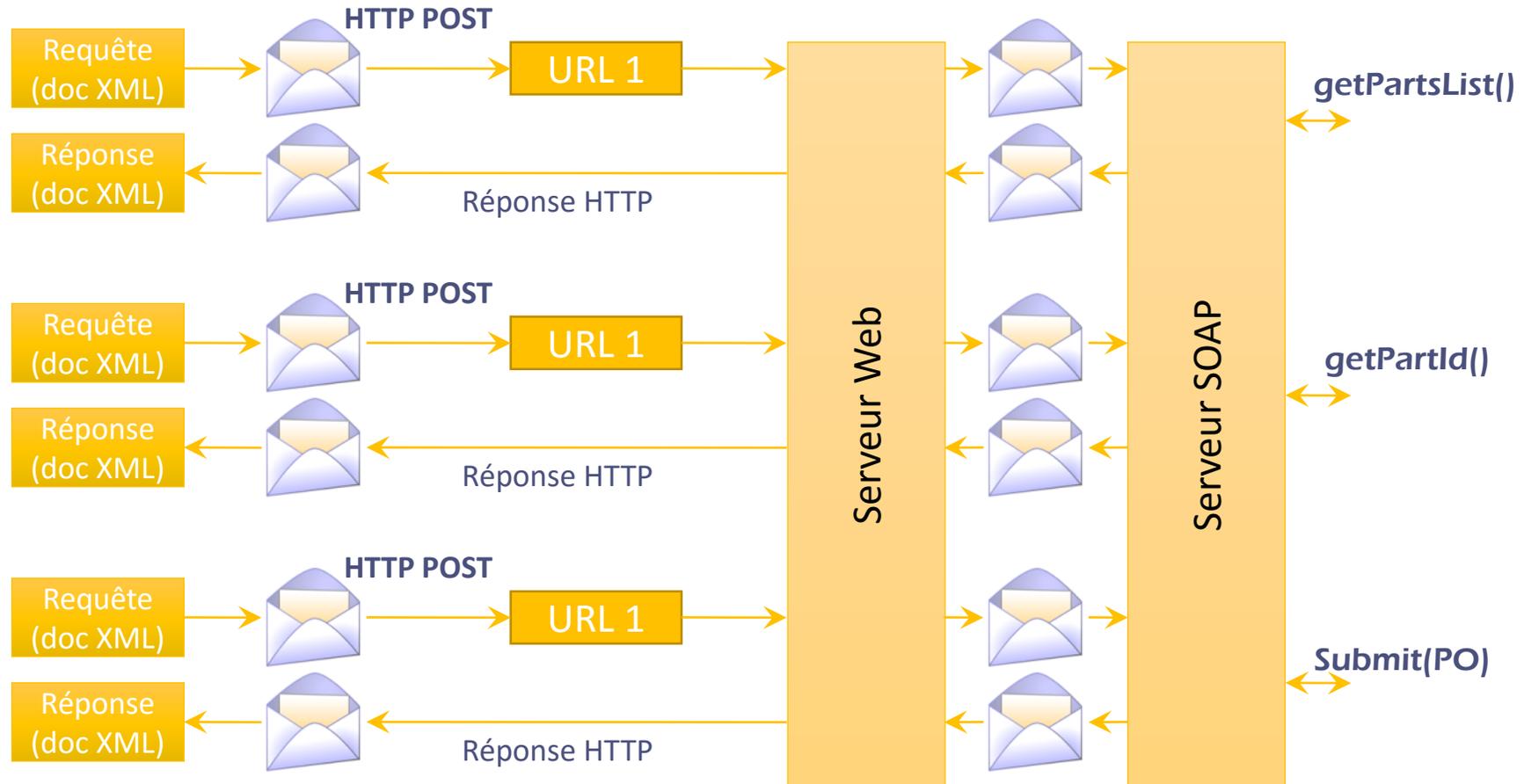
- Plus récemment REST s'est doté d'un langage de description de service : WADL (Web Application Description Language)
- Soumis en 2009 au W3C, il n'est toujours pas standardisé
- Le WADL est un format de fichier basé sur XML qui permet de décrire des applications REST.
- Cette spécification se heurte néanmoins à **la spécification WSDL 2.0, qui elle aussi permet la description de web services REST.**
- De plus, WADL est encore très mal supporté par l'ensemble des frameworks existants ce qui limite son utilisation.



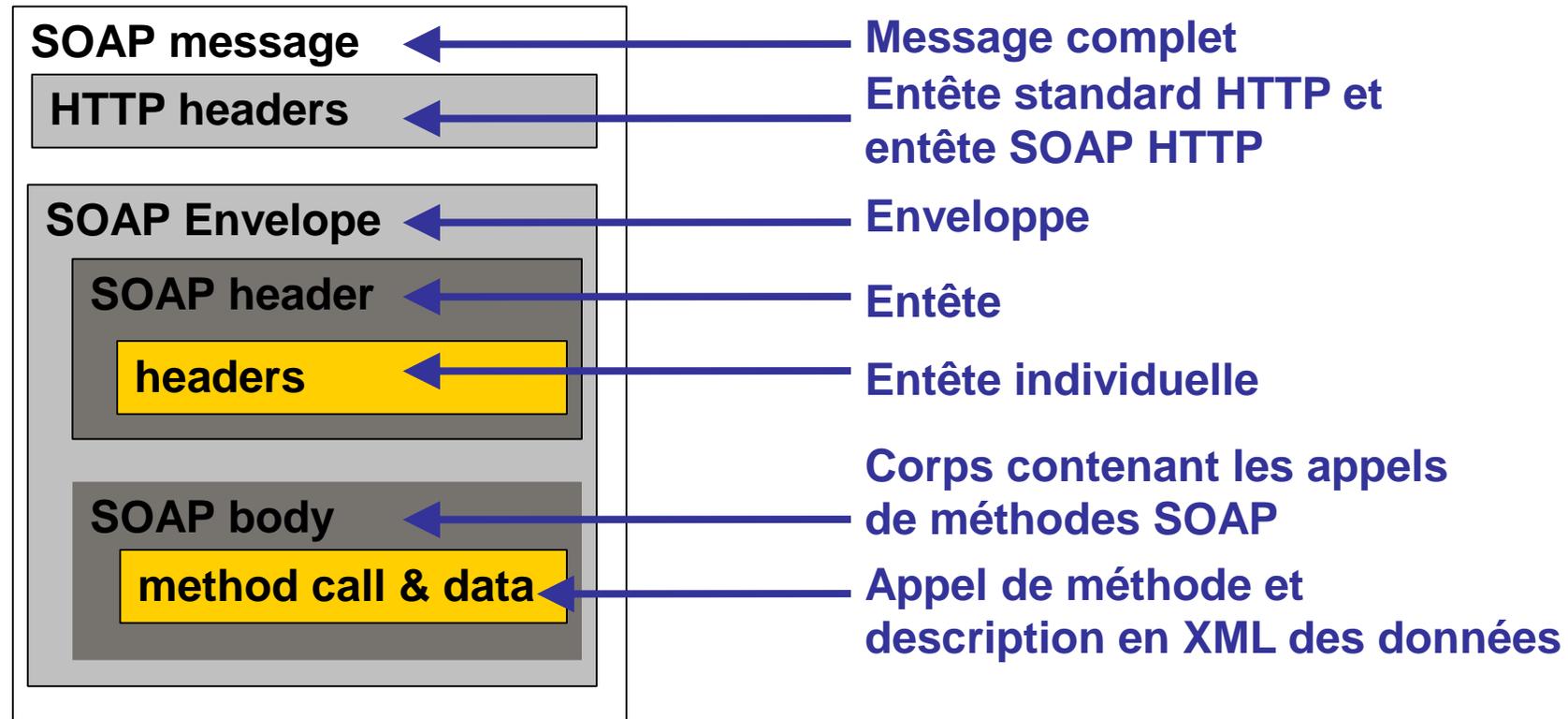
# Services Web

## WS-SOAP

# SOAP pour une approche SOA



# La structure des messages WS-SOAP



# Exemple de requête SOAP utilisant HTTP

- Demande de cotation à un serveur :

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAP-Action: "Some-URI"
```

Une seule commande  
HTTP/POST  
« envoi de message SOAP »

---

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/enveloppe/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# Exemple de réponse SOAP utilisant HTTP

- Réponse du serveur

HTTP/1.1 200 OK

Content-Type: text/xml; charset="utf-8"

Content-Length: nnnn

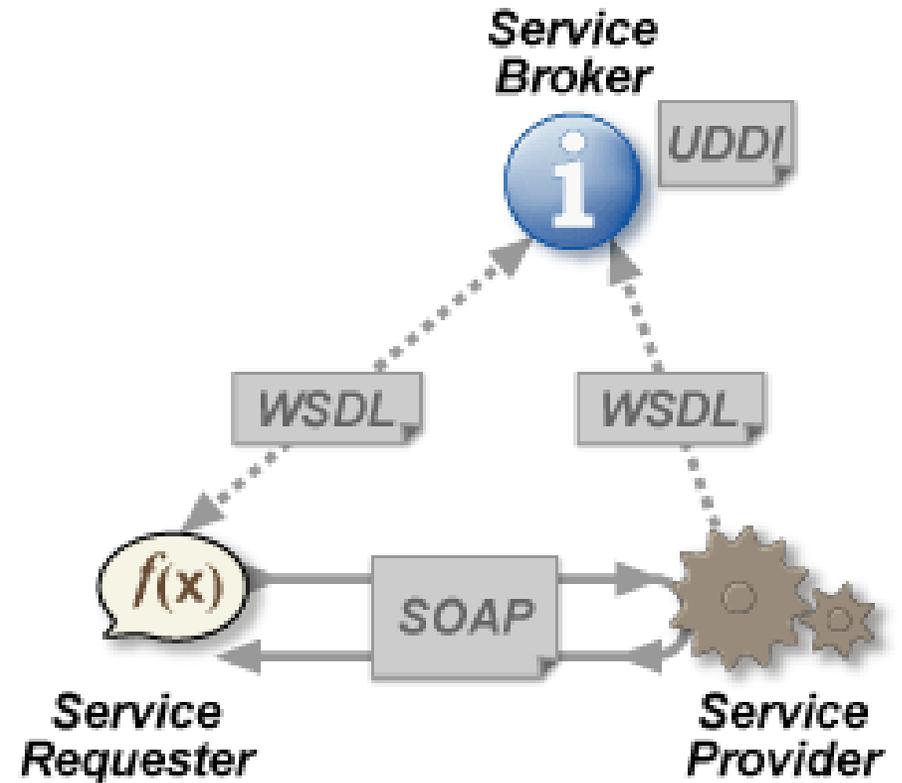
Réponse au HTTP/POST

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



# Cycle de Vie WS-SOAP

- Etape 1 : **Déploiement** du service Web
  - Dépendant de la plate-forme
- Etape 2 : **Enregistrement** du service Web
  - WSDL : description du service
  - Référentiels : DISCO (local), UDDI (global)
- Etape 3 : **Découverte** du service Web
- Etape 4 : **Invocation** du service Web par le client



# Applications Domotiques – GTB ...

- Une autre façon de voir les capteurs / actionneurs et équipements déployés dans un domicile / immeuble / quartier / ville
- Ex . WIT et son DataCenter
- Ex. HDSN en visite le 21 Mars matin et ses services REST
- Ex. Nice Côte d'Azur – ville du numérique et « big data » - Véolia / IBM