

Services et Web Services, modèles et implémentations

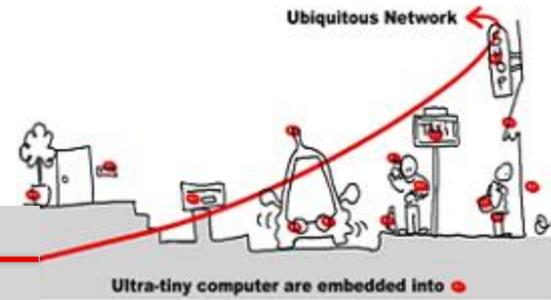
4 cours / 4 TDs

Jean-Yves Tigli

<http://www.tigli.fr>

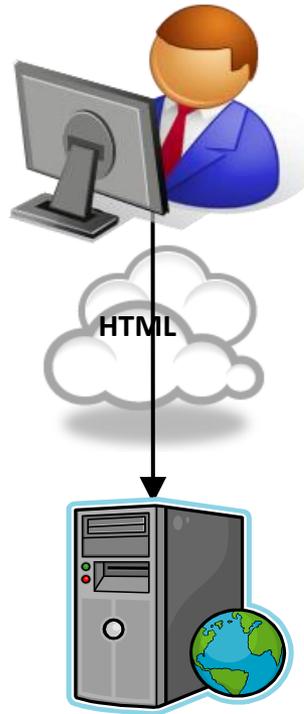
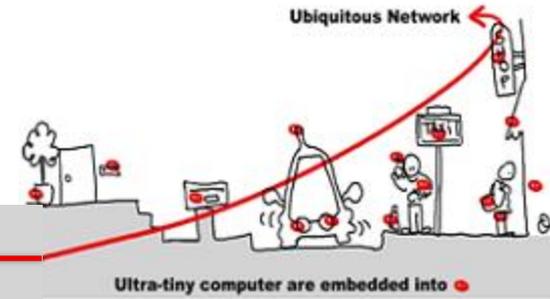
Polytech of Nice - Sophia Antipolis University

[Email : tigli@polytech.unice.fr](mailto:tigli@polytech.unice.fr)

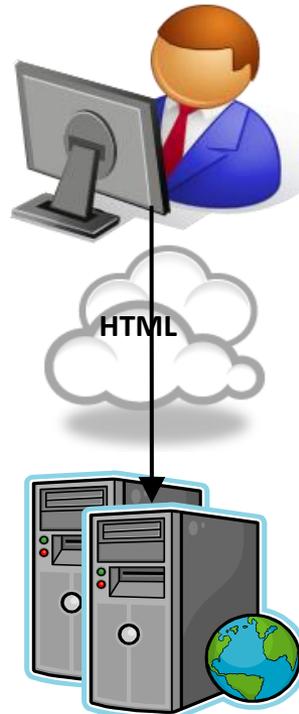


Introduction aux Web Services

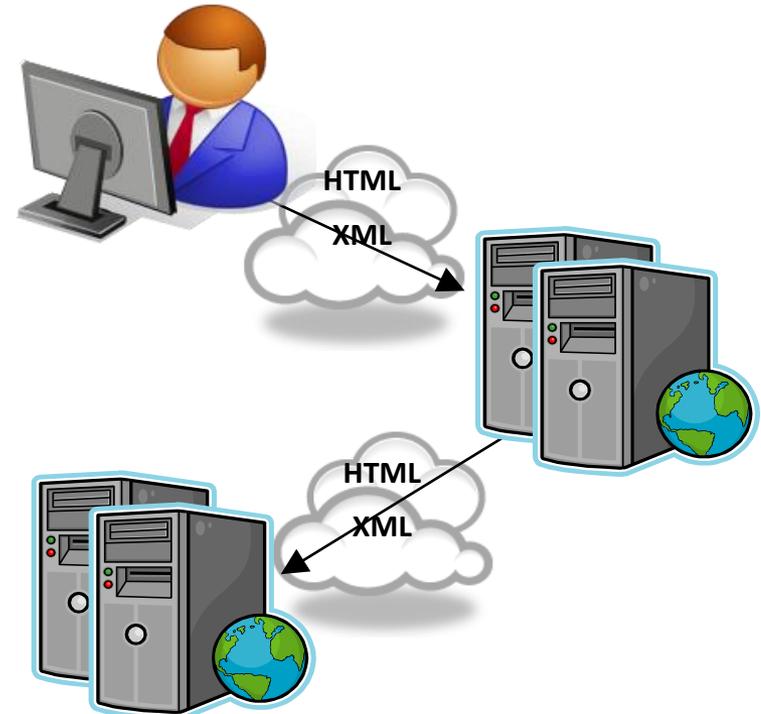
Evolution du Web : du H2M au M2M



**Generation 1
Static HTML**



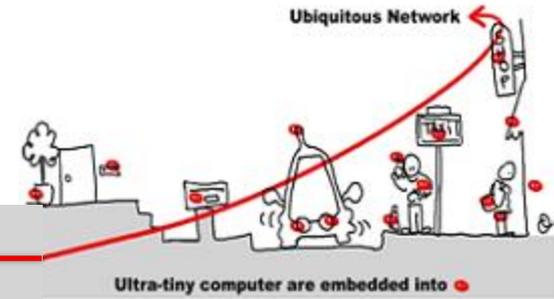
**Generation 2
Web Applications**



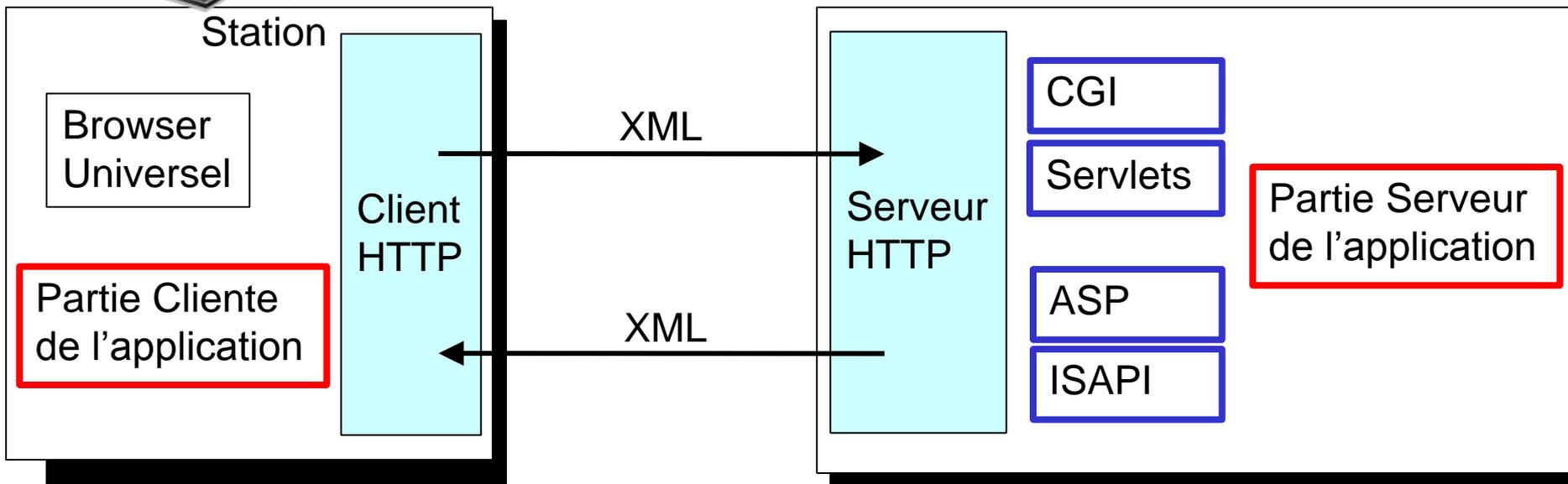
**Generation 3
Web Services**

Machine to Machine (M2M)

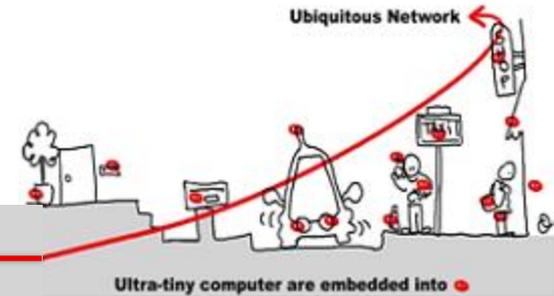
Evolution du Web : du H2M au M2M



- Exemple HTTP + XML

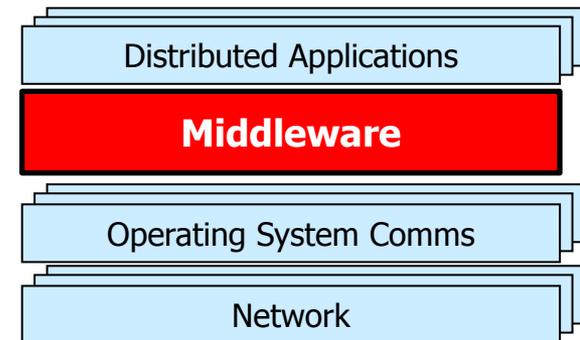
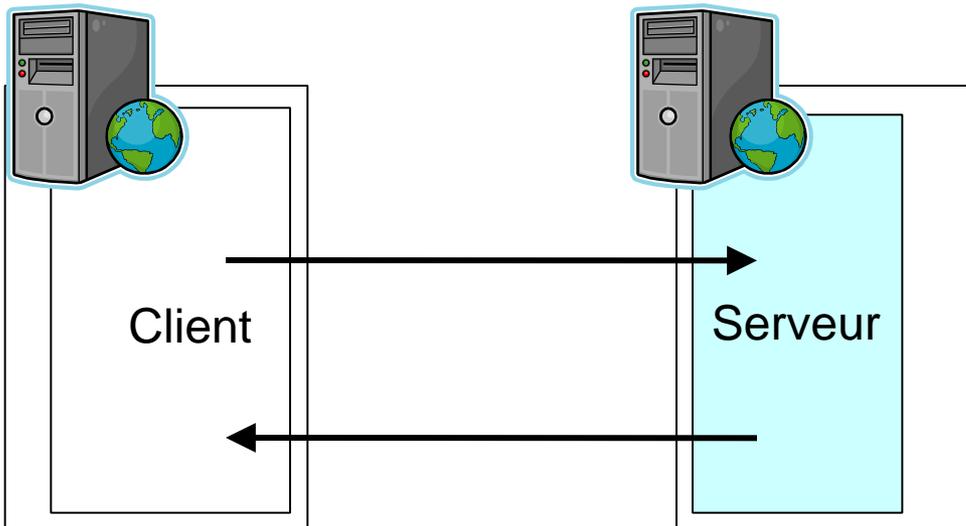


Evolution des intergiciels

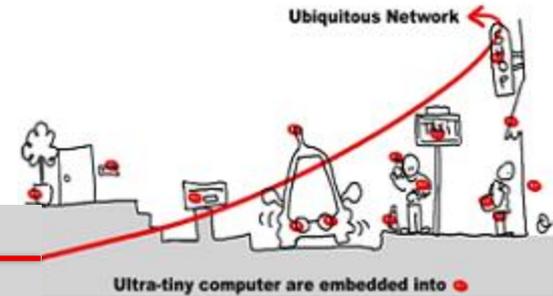


- *“The intersection of the stuff that network engineers don’t want to do with the stuff that applications developers don’t want to do.”*

[Kenneth J. Klingenstein ('99)]

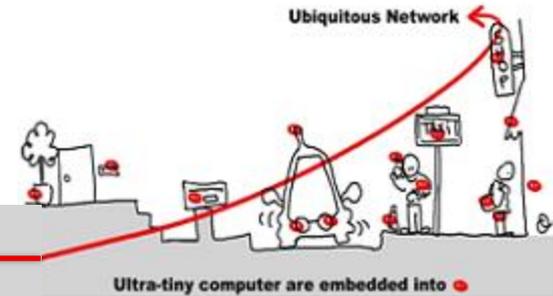


Caractéristiques des intergiciels orientés Service



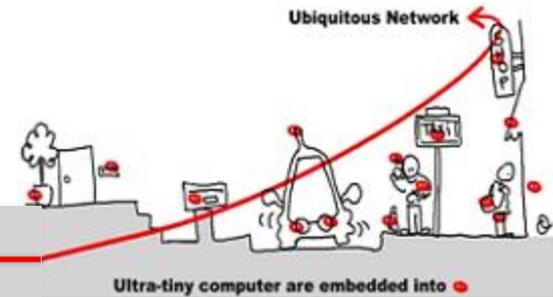
- Caractéristiques
 - Architecture de type Client / Serveur
 - Le serveur rend des services à un client
 - Réutilisable
 - Par plusieurs clients (simultanément ou pas)
 - **Indépendamment** de
 - la plate-forme (UNIX, Windows, ...)
 - du langage pour l'implémentation (VB, C#, Java, ...)
 - la plate-forme de développement sous-jacente (.NET, J2EE, Axis...)

Quels objectifs pour les Services Web ?



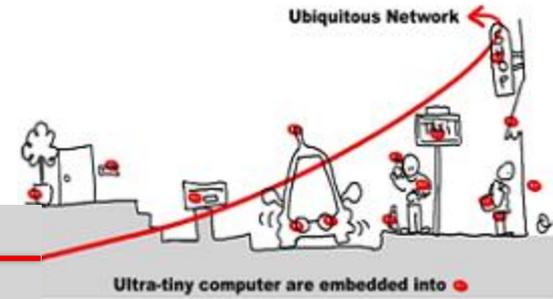
- Remplacer les outils (RPC, DCOM, CORBA, RMI) par une approche entièrement ouverte et interopérable, basée sur la généralisation des serveurs Web avec scripts CGI.
- Faire interagir des composants hétérogènes, distants, et indépendants avec un protocole standard (ex. SOAP).
- Passer les politiques de sécurité grâce en grande partie à une couche session basée sur HTTP (port 80).

Services Web et Interopérabilité

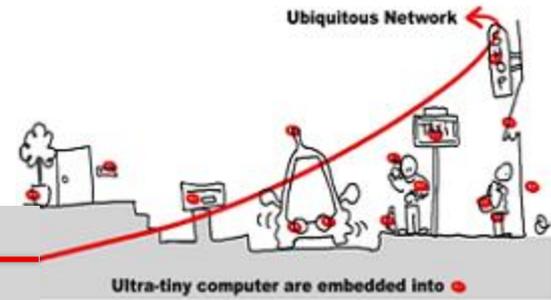


- **Platform independent** : A l'instar de Corba (précurseur historique), les Services Web gèrent l'interopérabilité au niveau du protocole d'échange.
- **Platform dependent**: d'autres approches gèrent l'interopérabilité par le portage de la plate-forme d'exécution (ex. OSGi, RMI sur Java et historiquement COM/DCOM)

Pour quoi faire ?

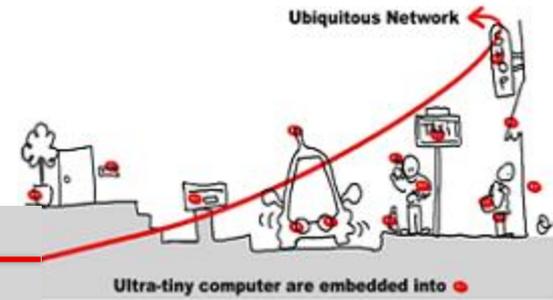


- Les Services Web permettent d'interconnecter :
 - Différentes entreprises
 - Différentes applications
 - Différents clients
 - Différents matériels
- Utilisé dans différents cadres:
 - B2B (Business To Business)
 - EAI (Enterprise Application Integration)
 - ...



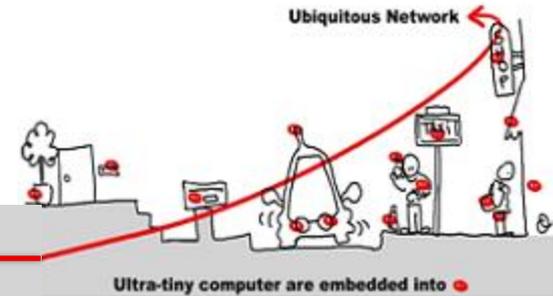
Architectures ROA - SOA

Architecture orientées ressources ou données



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

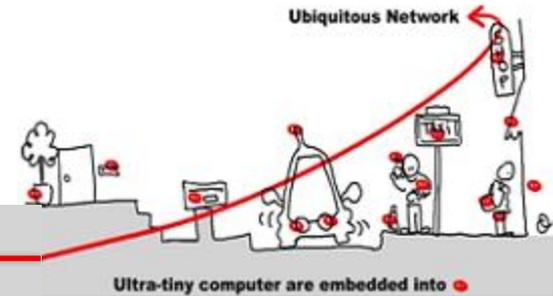
Les principes REST



- **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

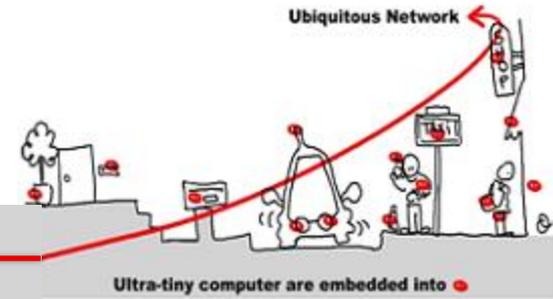
Chapitre 5 de la thèse de doctorat "Representational State Transfer (REST)". de Roy Fielding , 2000

Les principes REST ou ROA



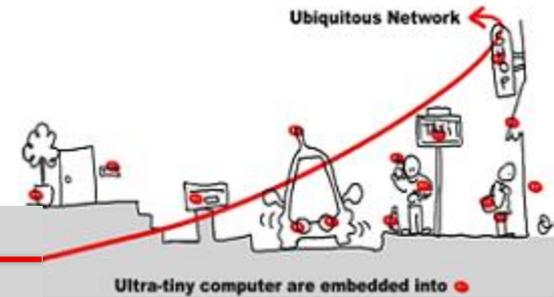
- Ressources (Identifiant)
 - Entité identifiable dans le système (livre, agenda ...)
 - URI et donc possiblement URL
 - Une URI identifie une seule Ressource
 - Une Ressource peut avoir plusieurs URI
 - Exemple :
 - Emploi du temps de tigli : /edt/prof/tigli/lundi
- Méthodes (Verbes)
 - Quatre opérations 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

Architectures orientées services



- Service Oriented Architecture (SOA)
- Un paradigme qui ne s'appuie pas forcément sur le Web
- Lorsque l'architecture SOA s'appuie sur des web services, on parle alors de WSOA, pour Web Services Oriented Architecture).

Cycle de Vie WSOA



- 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

