



Institut  
Mines-Télécom

# Protocole domotique xAAL



# Intéropérabilité : solutions existantes

- Revue (non exhaustive) des solutions existantes : projets européens [7, 3, 1], projet open source [10, 4, 6, 9], standards [2, 5], autre [8].

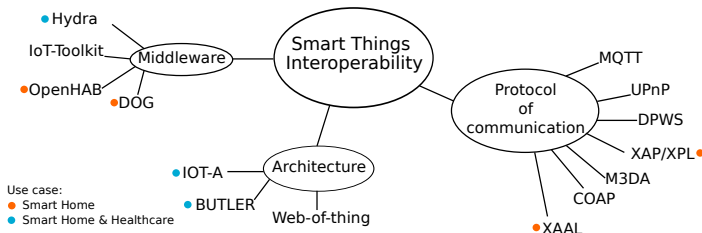
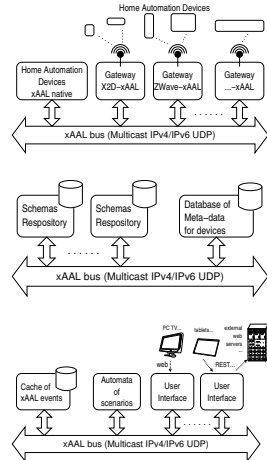


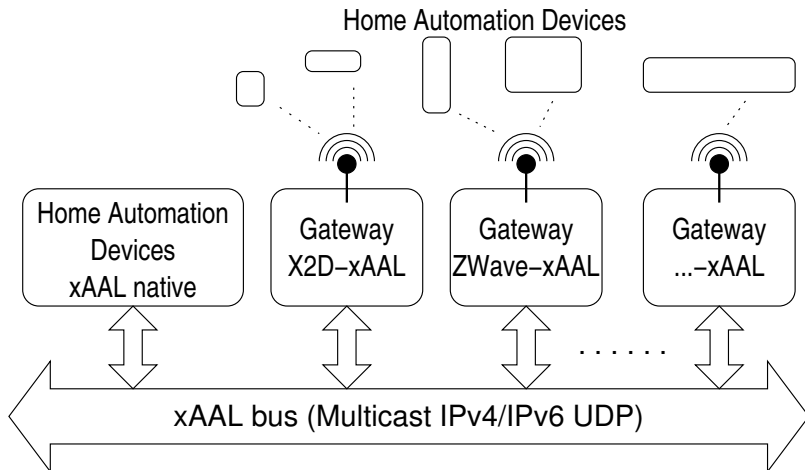
FIGURE : Carte heuristique

# Principes généraux d'xAAL

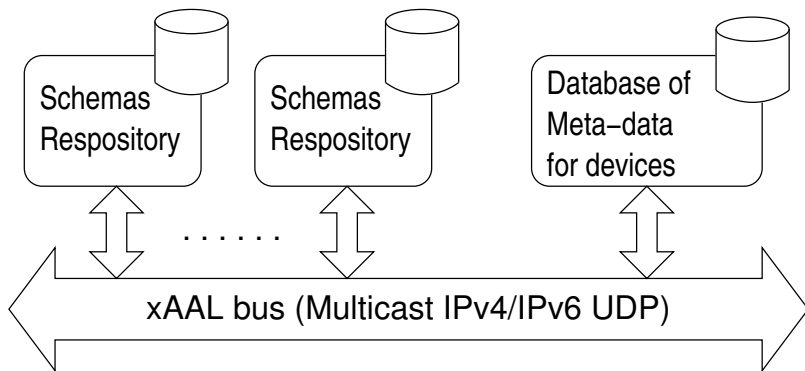
- Nom : xAAL
- Objectif : interopérabilité
- Bus IP multicast
- Fortement distribué
- Mécanisme de découverte
- Description par schémas
- Messages Json



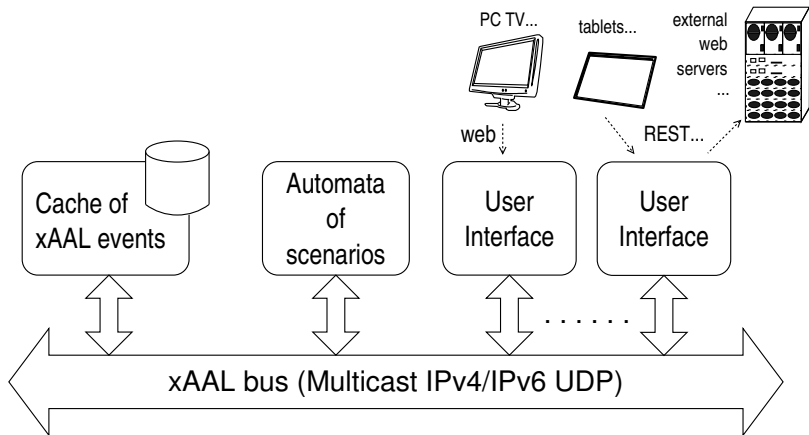
## Bus - Les devices



## Bus - Bases de données



## Bus - Les activateurs



# Les schémas

- Un pour chaque entité
- *méthodes(paramètres), notifications, variables d'état*
- Hérarchie :
  - Générique
    - ▶ Variables : devtype address parent
    - ▶ Notifs : Alive StateChange Error
    - ▶ Méthodes : IsAlive GetDescription GetState SetBusConfig
  - Classe de type
    - ▶ lamp.any ...
  - Type spécifique
    - ▶ lamp.basic lamp.dim shutter.simple door.locker...

## Exemple de messages

```
{
  "header":
  {
    "version": "0.2",
    "source": 39595606,
    "target": 16778244
    "devtype": "thermometer.queryable",
    "msgtype": "request",
    "action": "getstate",
    "cipher": "none",
    "signature": ""
  }
}
```

```
{
  "header":
  {
    "version": "0.2",
    "source": 16778244,
    "target": 39595606,
    "msgtype": "reply",
    "devtype": "thermometer.queryable",
    "action": "getstate",
    "cipher": "none",
    "signature": ""
  },
  "body":
  {
    "temperature": 9.3000000000000007
  }
}
```





# Planning

- xAAL version 0.1-bin : mars 2012, sur le papier
- xAAL version 0.1-json : mars 2012, développé
- xAAL version 0.2 : fin juin, SDK (Python), Gateway KNX, IHM web

# References |



Butler project.

<http://www.iot-butler.eu/>.



Coap : draft ietf core coap.

<https://datatracker.ietf.org/doc/draft-ietf-core-coap/>.



Internet of things - architecture — iot-a : Internet of things architecture.

<http://www.iot-a.eu/public>.



lot toolkit.

<http://iot-toolkit.com/>.



Mqtt.

[https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=mqtt](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt).



Openhab project.

<http://www.openhab.org/>.



Project hydra middleware.

<http://www.hydramiddleware.eu>.

## References II



Xap prot.

[http ://www.xapautomation.org/](http://www.xapautomation.org/).



Dario Bonino, Emiliano Castellina, and Fulvio Corno.

Dog : an ontology-powered osgi domotic gateway.  
2008.



Dominique Guinard.

*A Web of Things Application Architecture – Integrating the Real-World into the Web.*  
PhD thesis, ETH Zurich, 2011.

## Annexe : solutions existantes

L'interopérabilité OUI mais plusieurs façons de l'atteindre :

- Un protocole de communication commun à tous les objets connectés
- Un intergiciel (Middleware) permettant une interopérabilité au niveau des services (Ressource exposés comme des services).
- Une architecture : serveurs, passerelles, objets connectés



# Protocole de communication

## Avantages :

- Permet une architecture distribuée
- Permet extensibilité, flexibilité
- N'impose pas un langage

## Inconvénients :

- Effort en développement de "Bridges" pour intégrer la pile protocolaire
- Centré sur une technologie (doit être adopté pour garantir succès)



# Intergiciel

## Avantages :

- C'est du logiciel : dynamicité, déploiement, modularité
- Intergiciel dirigé par les modèles
- Objets/ressources exposé comme des services

## Inconvénients :

- Impose généralement l'utilisation d'OSGI (tout concentré dans une Gateway)
- Communication entre intergiciel ou via un protocole de communication (REST/Json, RPC, ...)
- La Gateway doit supporter l'intergiciel (taille mémoire, vitesse processeur)



# Architecture

## Avantages :

- Permet une interopération horizontale entre les domaines comme la domotique, la santé, le transport, ... [3, 1]
- Choix utilisation : centré autour d'un protocole, utilisation d'un intergiciel ou hybride

## Inconvénients :

- Infrastructure lourde à mettre en place
- Connaissance de plusieurs technologies (serveur, gateway, standards,...), protocoles de communication



# Synthèse

- Protocole de communication ou intergiciel (un choix qui se discute)
- Définir une architecture → une nécessité pour mettre en place l'interopérabilité des objets & services



Quels sont les éléments nécessaires à une architecture visant l'interopérabilité entre les objets connectés ?

- Identification : nommage, adressage, relation, localisation (Naming, addressing, identification, location)
- Découverte des ressources/objets/appareils (Resource discovery)
- Résolution (Resolution, "Linked data principle")
- Recherche des ressources (Search, indexing)
- Sauvegarde, mise en cache (storage, repository)
- Représentation des données (Data model)
- Performances (Performances : Simplicity, Integration, Portability, Speed, ...)
- Management des ressources/objets/appareils (Device management)



## Synthèse II

- Protocole de communication, interopération (Communication protocol, interoperation)
- Extraction du contexte, intelligence ambiante (Context awareness, ambient intelligence)
- Distribué (Distributed)
- Sécurité & intimité (Security & Privacy)
- Ontologies, Web Sémantique (Ontology, Web semantic)