



**Energy@home**

## La Smart Home Hackathon & JEMMA

*La reference implementation di Energy@Home*

**Riccardo Tomasi,  
Ivan Grimaldi**

*Pervasive Technologies Area,  
Istituto Superiore Mario Boella*



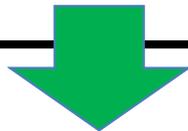
*Demo & Reference Implementation  
Working Group*

**Torino, November 21, 2014**

# Trend nella Smart Grid & nell'Home Automation

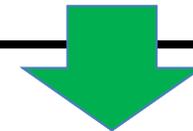
## Smart Grid

- Aumento della quota di energia proveniente da **fonti rinnovabili variabili non controllabili**
- Condizioni di mercato **dinamiche**
- Comparsa di nuovi **modelli di business and quadri contrattuali** in Europa
- Necessità crescente di sfruttare la **flessibilità degli utenti**



## Home Automation

- Numero crescente di **dispositivi eterogenei connessi** in casa
- **Frammentazione degli standard di comunicazione** in ambito domestico
- Necessità di scoprire in modo preciso le **funzionalità, le caratteristiche e il ruolo** dei dispositivi in casa per poterne sfruttare la flessibilità **senza ridurre il comfort** degli utenti



**Customer Energy Management System \***  
**(CEMS)**

# Customer Energy Management System (**CEMS**)

- Il CEMS è un **componente funzionale** nato per abilitare la gestione e l'ottimizzazione del **consumo** e della **produzione** in ambito domestico

L'approccio  **Energy@home** :

*Sviluppare il **CEMS** come parte di un **eco-sistema aperto***

- Raccolta di **informazioni** di consumo,
- Raccolta dei **profili** di consumo
- **Pianificazione** del consumo
- **Controllo diretto** dei dispositivi
- **Configurazione** del comportamento energetico dei dispositivi
- ...



# JEMMA: il CEMS di Energy@home



- **JEMMA: Java Energy Management Application** è la **reference implementation** del **CEMS** di Energy@home
- Può essere utilizzata per sviluppare rapidamente applicazioni e sistemi compatibili con l'ecosistema Energy@home
- Primo rilascio (v0.0.1) nell' **Ottobre 2013**, partendo da una soluzione CEMS già sviluppata e verificata in campo nei trial gestiti dall'associazione Energy@home
- Sviluppata e mantenuta tuttora dal gruppo di lavoro IMPL-WG di Energy@home, partendo da un contributo iniziale di codice da parte di Telecom Italia
- Versione corrente: **v0.9** (rilasciata a **Novembre 2014**)
- Mantenuta su GitHub



<http://jemma.energy-home.org>

<http://www.energy-home.it>

# JEMMA: Licenza



- La maggior parte del codice di JEMMA è rilasciato sotto licenza **LGPL** (*Lesser General Public License v3*)
- La licenza LGPL è stata scelta per permettere l'inclusione e la re-distribuzione di JEMMA anche per scopi commerciali ed integrando componenti di terze parti rilasciati sotto altre licenze
- Sono quindi **permesse**:
  - La condivisione di JEMMA e del suo codice sorgente
  - L'estensione e l'integrazione di JEMMA con altri prodotti o servizi (aperti o meno)



## Il Set-up di prova Energy@home

- Replica un completo set-up domestico
- E' normalmente mantenuto dai partner Energy@home nei **Laboratori ISMB** qui a **Torino**
- E' normalmente utilizzato per verificare in condizioni controllate nuovo hardware, applicazioni, ecc.



# L'Architettura di riferimento per la Smart Home Hackathon

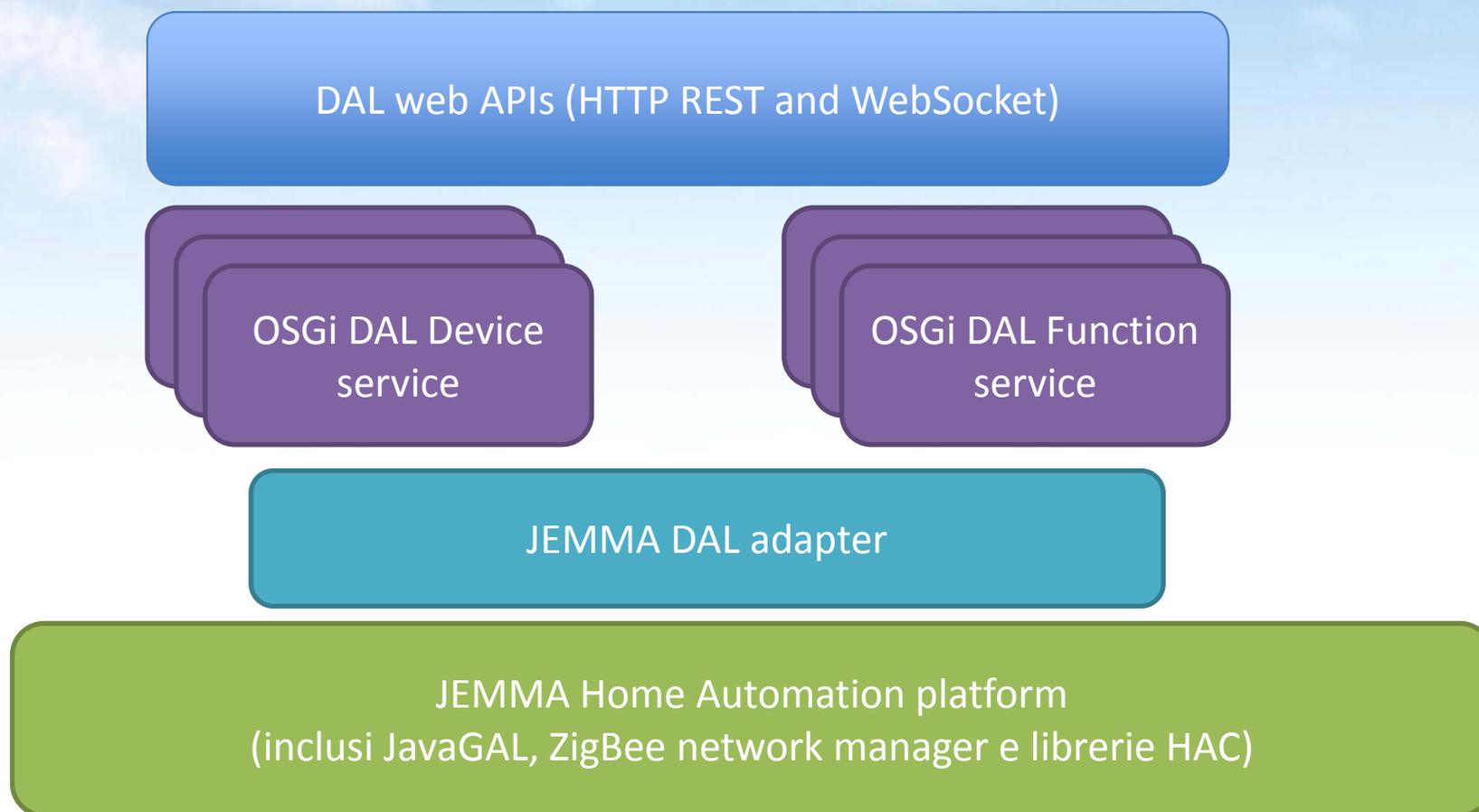
## Dispositivi accessibili via JEMMA

- Home Gateway Telecom Italia (su hardware di Flexgrid)
- Contatore Enel connesso tramite Smart Info con dongle ZigBee
- Suite Elettrodomestici connessi Indesit: lavatrice, forno, frigo.
- Flexgrid smart plug
- Inverter e smart plug per simulare un impianto locale di produzione di energia elettrica
- Lampade a led Philips hue e MAC
- Serratura (Door lock) ZigBee
- Sensore di temperatura

## Altri dispositivi ...

- Device NFC di ST
- TinkerKit Arduino
- Potete aggiungerne e usarne altri a vostro piacimento!

# Lo Stack Software di JEMMA



# Le JEMMA APIs & l' OSGi Device Abstraction Layer

---

- JEMMA include, a partire dalla versione 0.9, un nuovo layer di API per facilitare l'interazione con i dispositivi
- Le API sono basate sulle rfc0196 e rfc0210 dell'OSGi Alliance

Dall' RFC0196:



*“in order to provide a convenient programming model suitable for the realization of end-to-end services it is very useful to define and apply an abstraction layer which unifies the work with devices supporting different protocols”*

# Principali funzioni esposte dalle API

## *HTTP REST APIs*

**http://**

- Ottenere la lista dei dispositivi
- Ottenere la lista delle funzioni esposte da ciascun dispositivo
- Invocare le operazioni
- Modificare le proprietà dei dispositivi

## *WEBSOCKET APIs*

- Sottoscrivere agli eventi generati dai dispositivi e dal sistema JEMMA, es.
  - «attribute notifications» dei dispositivi
  - Eventi di overload (avvertimenti contrattuali, ecc.),
  - ...



WebSockets

# Suggerimenti ...

– Documentazione sul [Wiki di JEMMA](#)



– Per partire più velocemente ...

- conviene sfruttare l'ambiente «simulato» che include una serie di dispositivi «finti» che rispondono in modo predefinito, ma senza differenze dal punto di vista delle API

– Pronti a provare sul campo ?

- Sono prenotabili degli slot di test da 30 minuti sull'ecosistema vero contattando i tutor



– Per ricevere e condividere **aggiornamenti, suggerimenti, risposte a domande frequenti e frammenti di codice** utili durante l'evento:

- Mailing List di JEMMA [jemma-general@googlegroups.com](mailto:jemma-general@googlegroups.com)

– Domande? Dubbi ?

- Cercate i tutor !



