

Open the Box

Customer journey in an open Smart Home

Orange Open the Box team

Andre Bottaro, Sébastien Bolle, Jacques Pulou, Antonin Chazalet, Gregory Bonnardel, Cyrille Bareau, Orange Labs

Eclipse IoT Day, March, 30th, 2015



<http://openthebox.org>



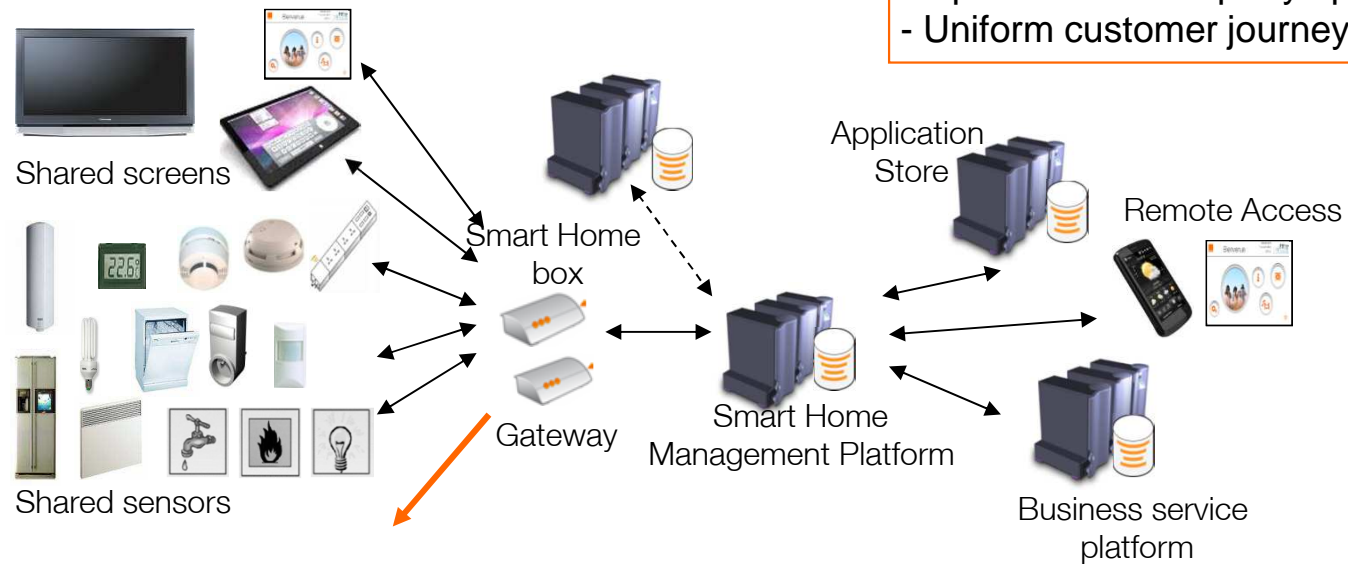
Summary

- A **new world of applications** is about to emerge at home thanks to the growing variety of available sensors and actuators
- To unleash service delivery, a telecom infrastructure is to be **open to third party applications** through cloud and embedded APIs and **open to any device** through local area networks.
- Today technology status enables to open the infrastructure to a set of trusted partners. **Remaining challenges are addressed in the project.**
- This presentation introduces the results of the project related to
 - the customer journey with an open smart home system
 - the device abstraction layer needed for the Smart Home

Open the Box project results

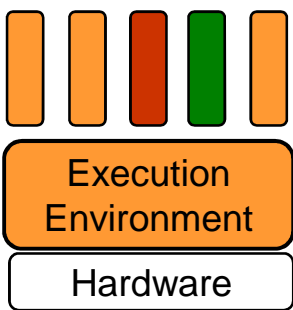


An open infrastructure with an ecosystem role model



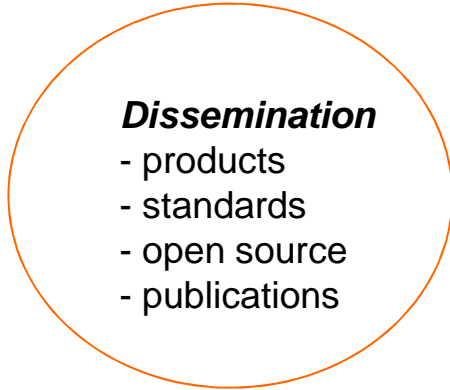
An end-to-end technical architecture

- Open APIs to 3rd party applications*
- Uniform customer journey*



Research issues solved on the embedded platform

- Security attacks and testing platform
- Hardware resource management
- Stale reference resolution for Java components
- Conflict management and application collaboration
- 3 embedded targets: openness vs hardware costs

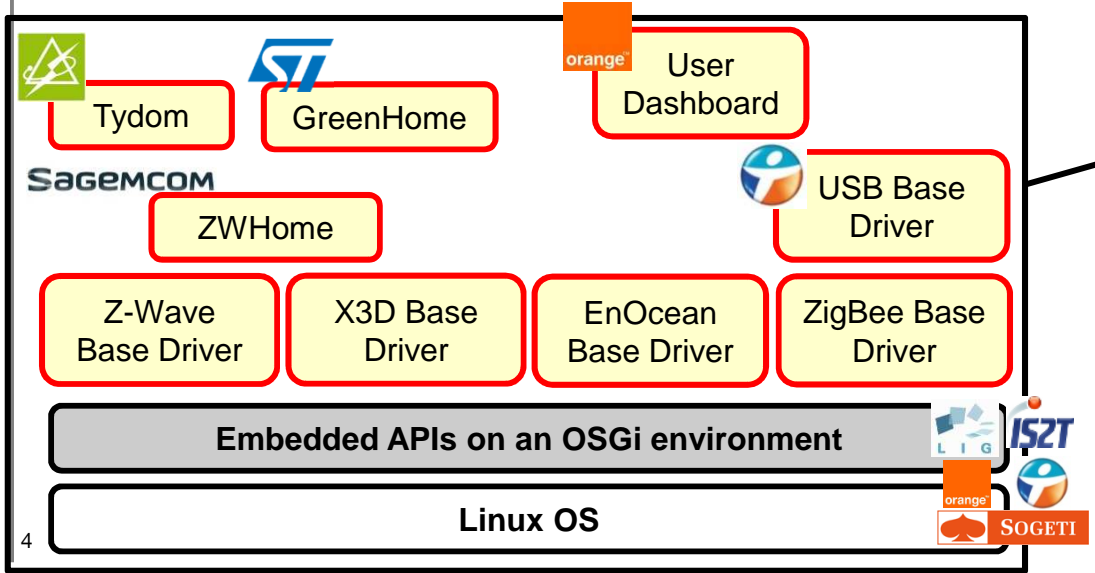
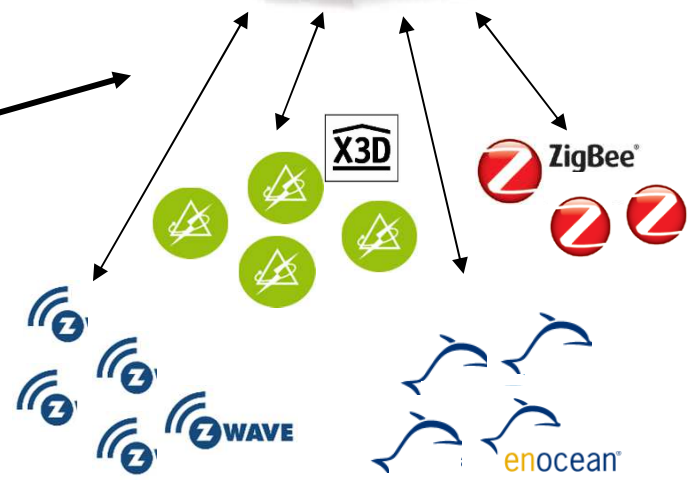
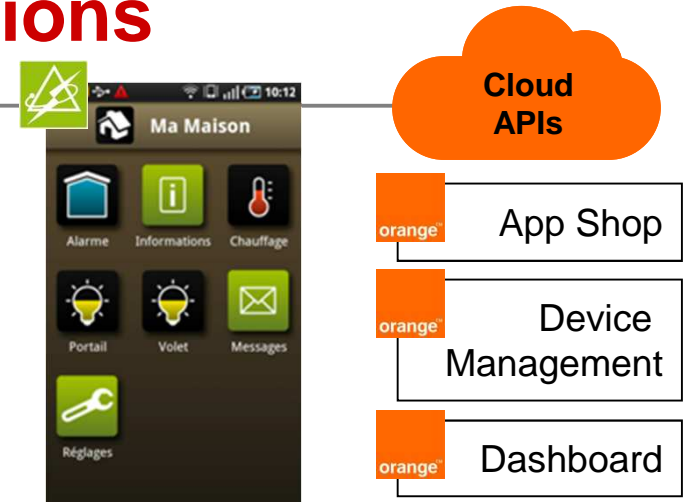


Dissemination

- products
- standards
- open source
- publications

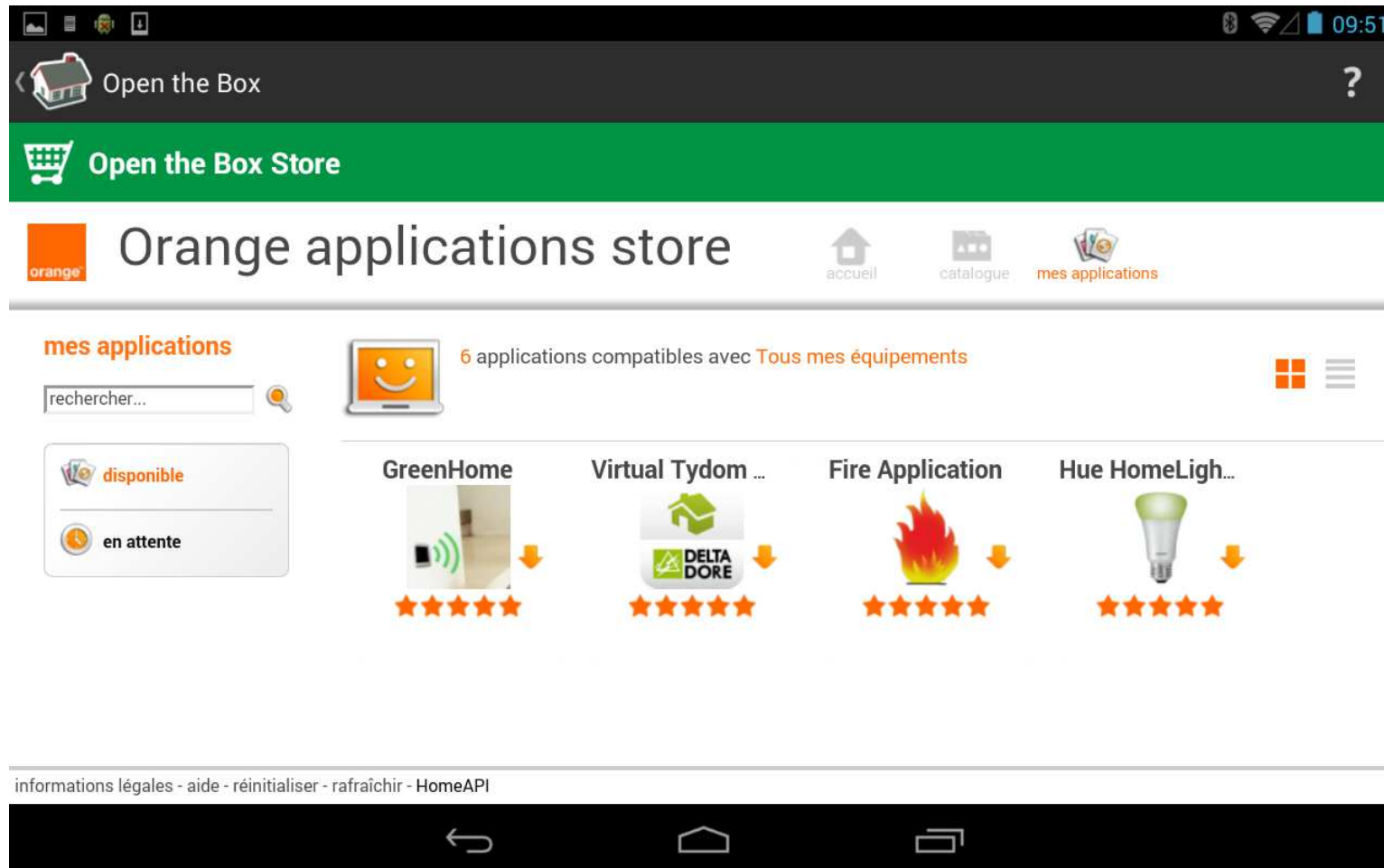
Open APIs to 3rd party applications

- The Smart Home operator provides
 - APIs for partners to access/manage locally and remotely apps and devices
 - an end-to-end infrastructure for the client to select, deploy, install, configure, use applications and devices
- Partners provide base drivers for the share of devices
 - e.g., Delta Dore, Sagemcom, STM, Orange, Bouygues T.
- Partners use APIs to deliver multiprotocol applications, e.g., Delta Dore, Sagemcom, STM, LIG, Orange



Customer journey

Application selection in an application store



- Presentation of eligible applications with respect to client home devices


Application store and device access rights

Open the Box

Open the Box Store

Orange Application Store

Home Catalog My applications


0,00 €

Access Permissions

This application requires the following services: Log (OSGi, Knopflerfish), HTTP (OSGi, Knopflerfish).

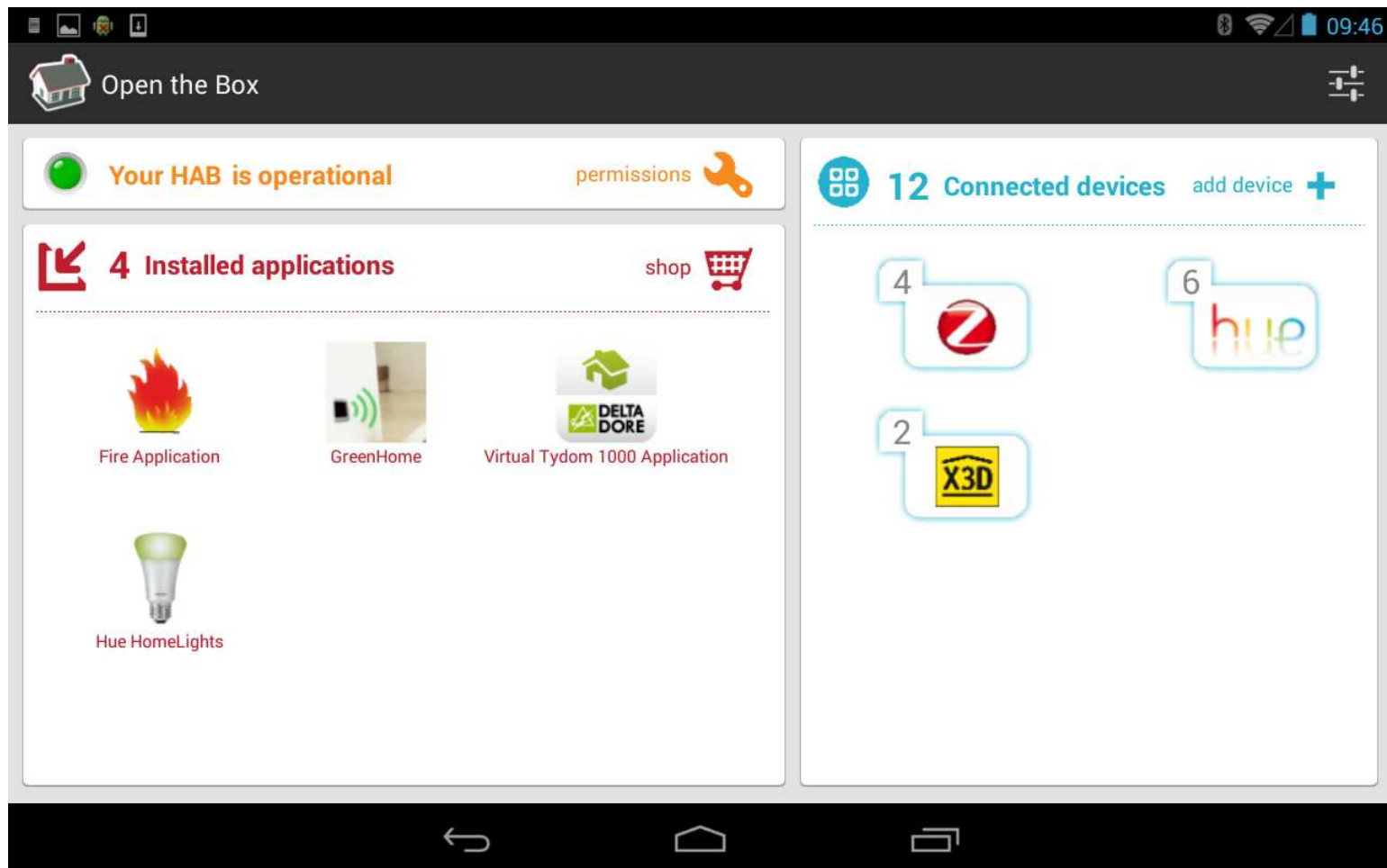
This application requires access permission to the following devices: IASZoneFire (ZigBee Smoke Detector) *optional, multiple*, IASWDFire (ZigBee Warning Device) *optional, multiple*, Pump (ZigBee Gas Valve) *optional, multiple*, x3dLightDimmer (X3D Light Dimmer) *optional, multiple*, hueLight (Hue Lamp) *optional, multiple*, greennetSwitch *optional, multiple*, zwavesmokedetector (Smoke Detector Alarm) *optional, multiple*.

Version: 1.0.0.SNAPSHOT
Editor: Orange
Used devices: IASZoneFire (ZigBee Smoke Detector), IASWDFire (ZigBee Warning Device), Pump (ZigBee Gas Valve), x3dLightDimmer (X3D Light Dimmer), hueLight (Hue Lamp), greennetSwitch, zwavesmokedetector (Smoke Detector Alarm).

Cancel Validate

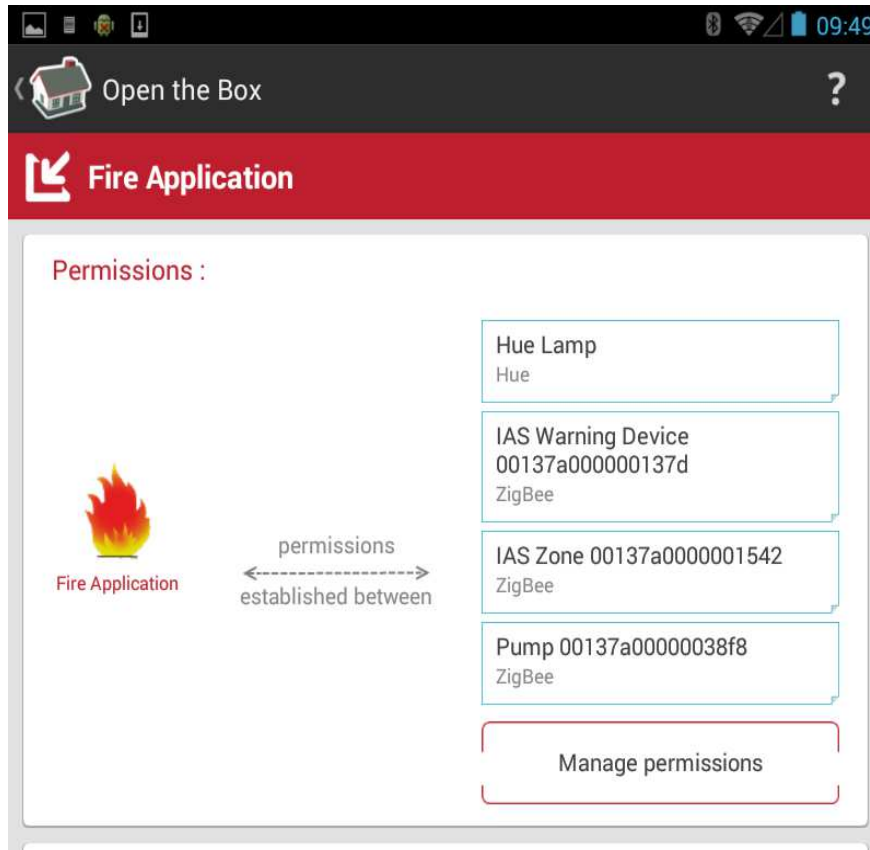
- Device access rights are presented to be validated for the selected application

User dashboard

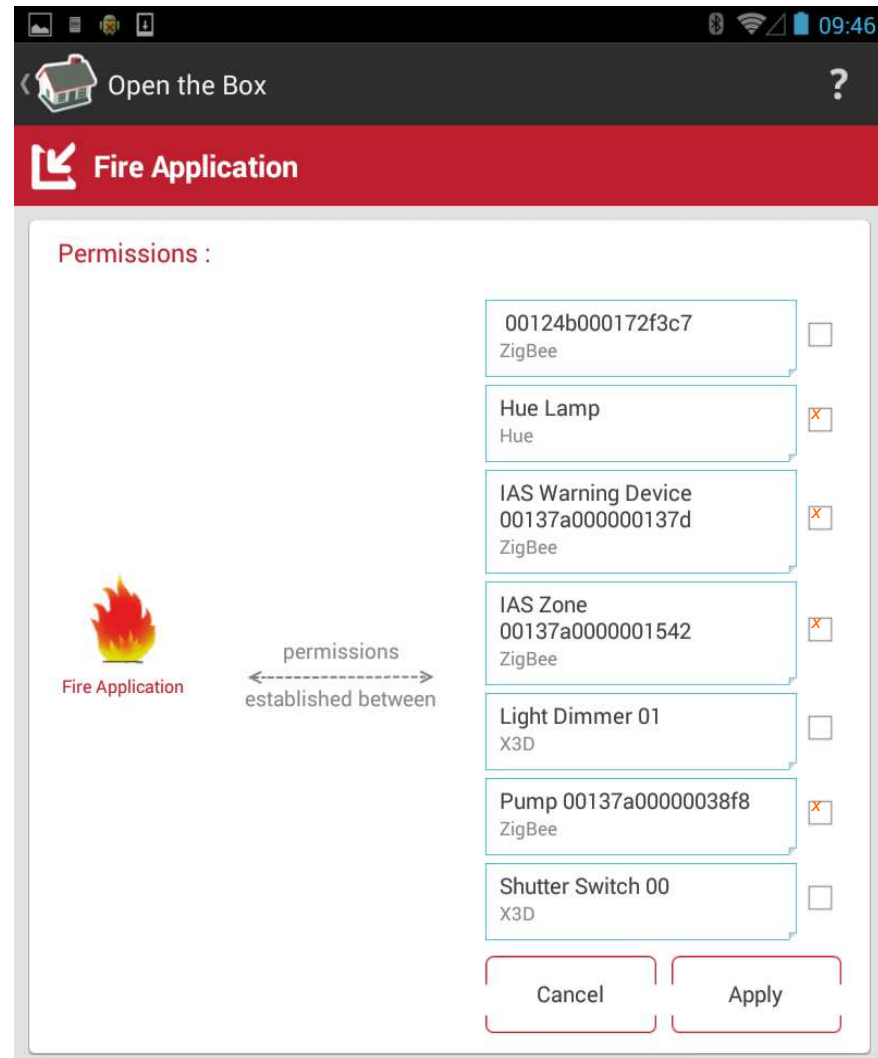


- Dashboard presenting deployed applications and connected devices

Management of device access rights



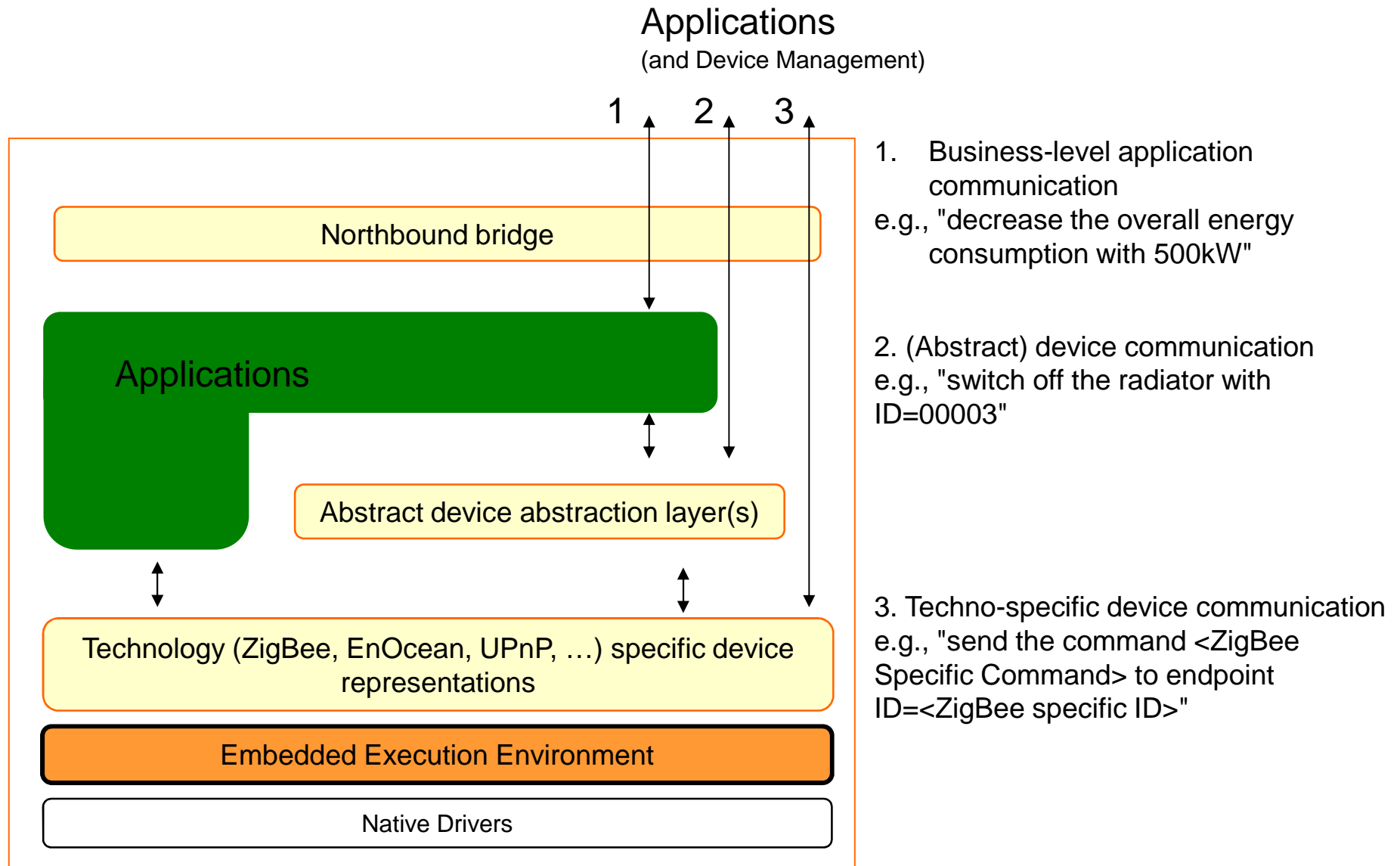
- Connected things visible by the 'fire' app.



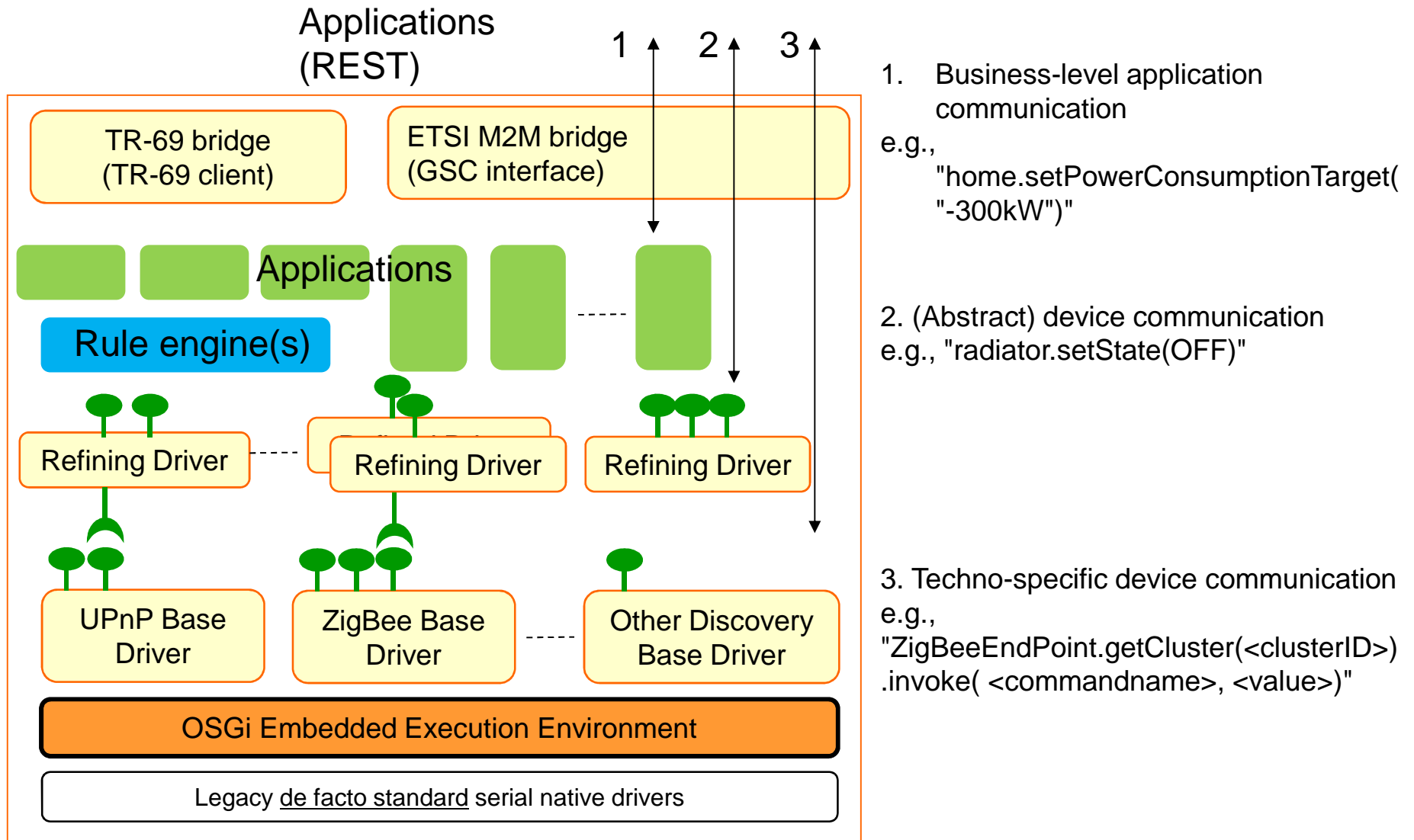
- Configuration of device access rights for 'fire' app.
- Only devices of relevant types are presented.

Device abstraction layer

3 levels to represent local home devices



3 levels to represent home connected things



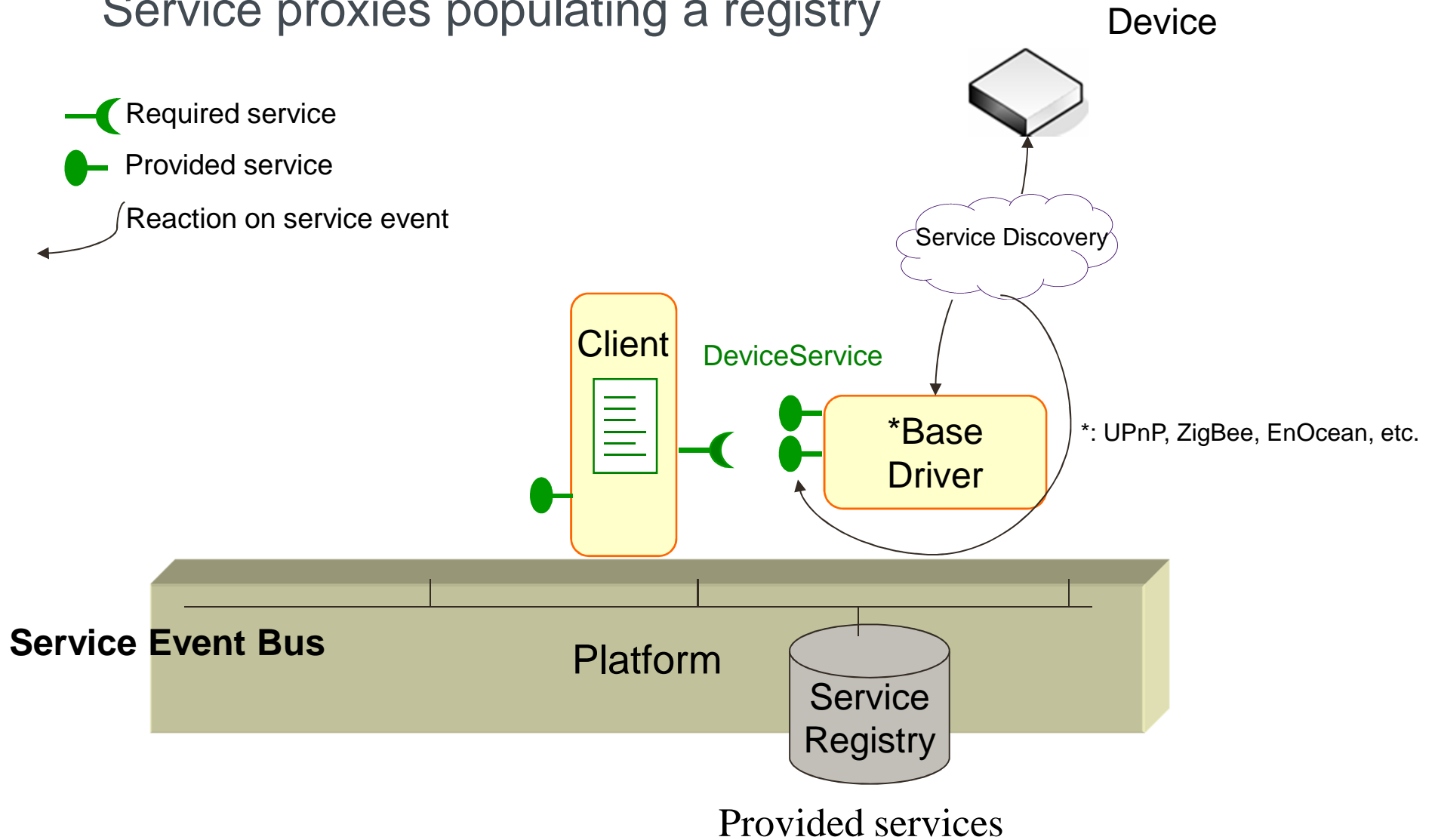
1. Business-level application communication
 e.g., `home.setPowerConsumptionTarget("-300kW")`

2. (Abstract) device communication
 e.g., `radiator.setState(OFF)`

3. Techno-specific device communication
 e.g., `ZigBeeEndPoint.getCluster(<clusterID>).invoke(<commandname>, <value>)`

Handling distribution

Service proxies populating a registry



The role of the Base Driver (BD)

1. Making distributed computing transparent

- Networked devices represented as programmed proxies
- Network management and message parsing by the BD
- Import
 - Method calls are relayed into device control messages
 - Listeners registration in the registry replaces event subscription
 - Network events are relayed into programmed listener notification
- Export
 - Platform-based services accessible through BD network interfaces
 - Event subscription of networked subscribers relayed into listener registration by the BD
 - Event notification is relayed on the network by the BD.

The role of the Base Driver (BD)

2. Mirroring the network dynamicity

■ Import: Mirroring network dynamicity on the platform

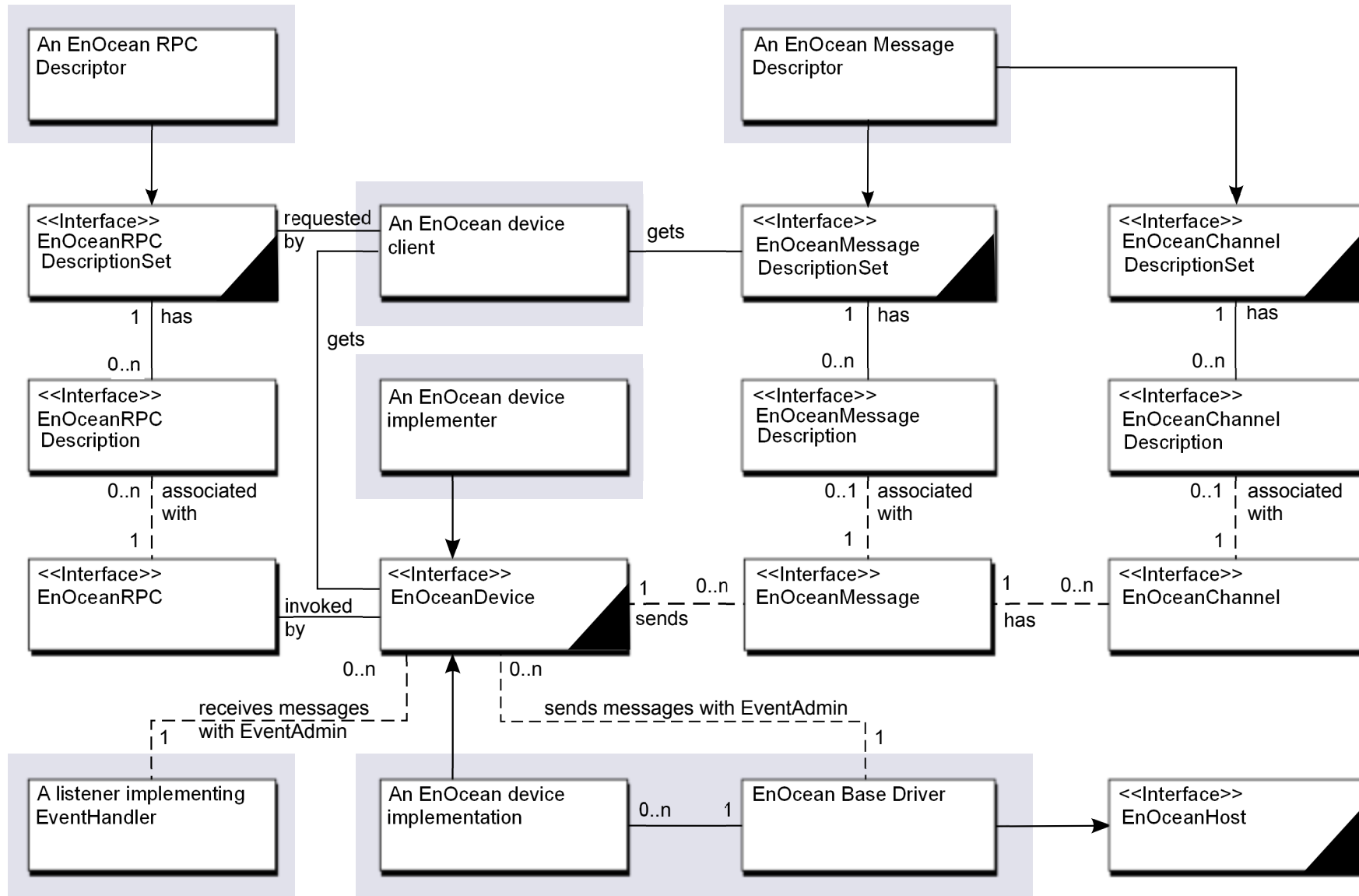
- At starting time
 - The BD searches for available devices
 - Every found device is registered in the platform service registry
 - The platform service registration generates a platform service event
- At runtime
 - The BD listens to device registration, modification, unregistration
 - The BD answers with platform service registration, modification, unregistration
 - Each platform service operation generates an event on the platform

■ Export: Mirroring platform dynamicity on the network

- Platform service registration, modification, unregistration relayed on the network
 - Discovery messages sent by the BD on the network

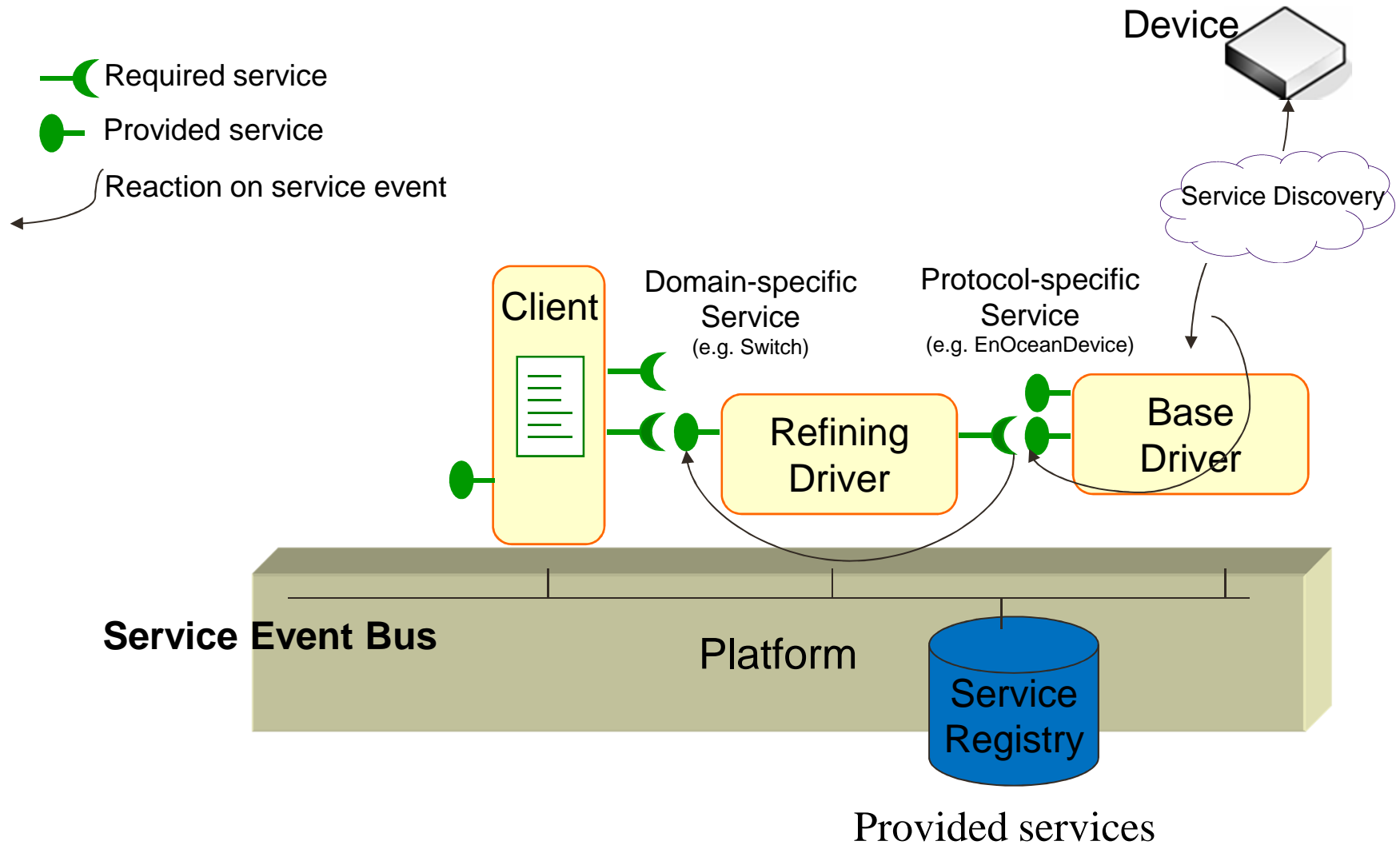
Example: The EnOcean Base Driver

with OSGi EnOcean Device service standard specification



Handling heterogeneity

Refined proxies populating a registry



The role of the Refining Driver (RD)

1. Masking protocol heterogeneity

Mediating protocol specific proxies into domain specific objects

- Service interfaces are domain-specific, e.g., Media Sources and Sinks
- Events are domain-specific, e.g., playing status

2. Mirroring service dynamicity

- At starting time
 - The RD searches for available protocol-specific services
 - Every found service is adapted and registered in the platform service registry
 - The platform service registration generates a platform service event
- At runtime
 - The RD listens to service registration, modification, unregistration
 - The RD answers with platform service registration, modification, unregistration
 - Each platform service operation generates an event on the platform

Open source, standardization and research perspectives

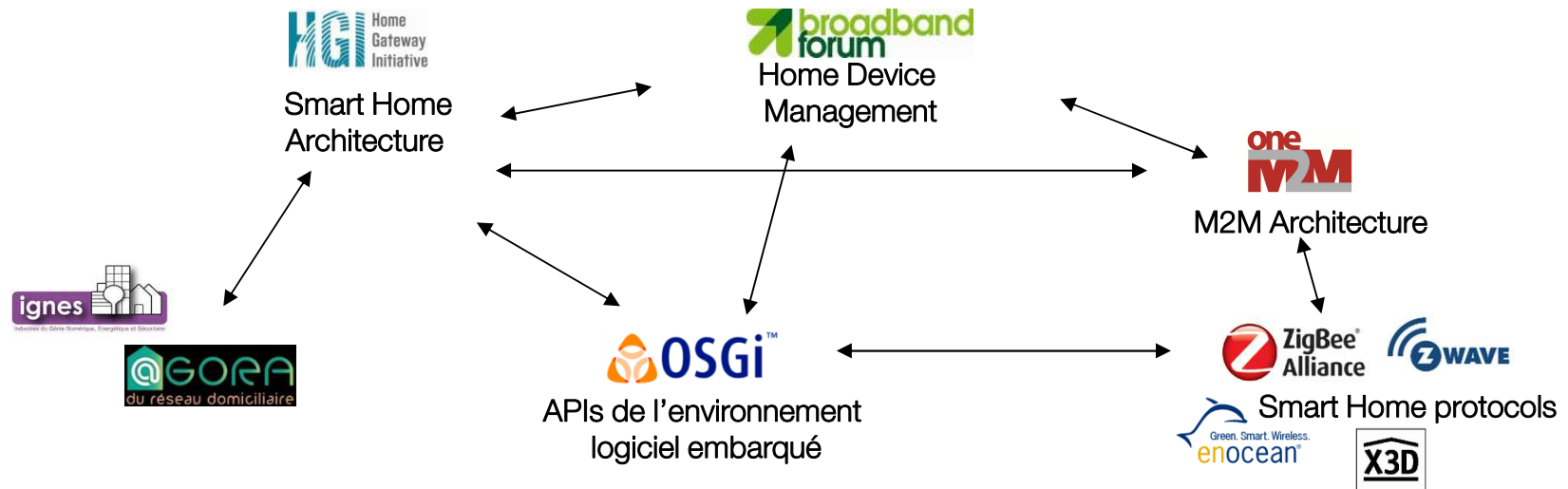
- Open source and freeware software
 - [[EnOcean](#)] Orange EnOcean base driver, under transfer to eclipse Smart Home.
 - [[Modus](#)] TR-069 device management client, Apache license.
 - [[SimpleBee](#)] SimpleBee base driver and Arduino client for fablabs, Apache license.
 - <http://openthebox.org> Orange freeware environment available with tutorials
 - Future work: integrate eclipse Smart Home rule engine and eclipse OM2M cloud APIs
- Standardization
 - Base driver layer now in HGI reference architecture and OSGi specifications
 - ZigBee, EnOcean, USB device service specification under finalization at OSGi Alliance
 - Remaining efforts: Standardize Home and IoT device abstraction layer
- Research perspectives
 - How to detect and handle device access conflicts between applications?
 - How to share data for the better aggregation, management and mining?
 - How to manage an IoT infrastructure as cloud resources with optimal deployment of application parts?

Thanks



***Standardization,
open source results,
and scientific communications***

Contributions to the standardization ecosystem



- **Contributions to HGI reference architecture**

- *technical requirements for resource management on an open gateway*
- *a Base Driver sublayer in the full abstraction layer*

- **Specifications in the next OSGi Residential Specification release – March 2015**

- *ZigBee Device Service (RFC 192)*
- *EnOcean Device Service (RFC 199)*
- *Resource Monitoring (RFC 200)*

Open source results

EnOcean Base Driver	Orange
SimpleBee Base Driver	Orange, LIG
GreenNet Base Driver	STMicroelectronics
INCINERATOR – JVM for stale reference free OSGi platforms	Orange
Secure coding good practices (French)	Sogeti High Tech
Attacks defined and implemented on a testing platform targeting OSGi technology (French – to be translated)	Sogeti High Tech
MODUS – TR-069 Device Management client	Orange
RoSE – Distributed OSGi framework and sensor interaction	LIG
APAM – Framework for applications collaboration	LIG

Technical and scientific communications

Two topics

- **Software Engineering of adaptive applications**

Open the Box project has contributed to the state of the art the functionalities of an embedded service platform: dynamic dependency management, applications modeling, automatic deployment, autonomic application management.

- **Software engineering of multi-tenant service platforms**

Open The Box project envisions a software platform shared by a dynamically extensible set of applications from distinct providers. This objective raises challenges to enable fair resource sharing and isolation, and to address security issues with respect to bugged and malicious applications.

Software engineering of adaptive applications

■ Publications

- Jacky Estublier, German Vega, Thibaud Flury. Completeness for Adaptable Applications. Submitted to 37th International Conference on Software Engineering. ICSE 2015, May 2015, Florence, Italy.
- Jacky Estublier, German Vega. Causality Control in Dynamic Platforms. Proceedings of the 9th International Conference on Software Engineering Advances, ICSEA 2014, October 2014, Nice, France.
- Jacky Estublier, German Vega. Building Reliable Dynamic Applications for Ubiquitous Computing. Proceedings of the 5th International Workshop on Principles of Engineering Service-Oriented Systems (PESOS), San Francisco, May 2013.
- Jacky Estublier, German Vega. Reconciling Components and Services: The Apam Component-Service Platform. International Conference on Service Computing, Jun 2012, Honolulu, HI, United States.
- Diana Moreno, Jacky Estublier. Model-Driven Design, Development, Execution and Management of Service-Based Applications. International Conference on Service Computing, Jun 2012, Honolulu, HI, United States
- Diana Moreno, Elmehdi Damou. Model-driven execution of service-based applications. Journées sur l'Ingénierie Dirigée par les Modèles, Jun 2011, Lille, France.

■ PhD theses

- J. Bardin (LIG) RoSe : a framework to design and execute dynamic distributed heterogeneous applications. Oct. 2012. (French)
- Diana Moreno-Garcia. Modèles, outils et plate-forme d'exécution pour les applications à service dynamiques. Grenoble University, Feb. 2013. (French)

Software engineering of multi-tenant service platforms

■ Publications

- K. Attouchi, G. Thomas, A. Bottaro, J. Lawall, G. Muller. Memory Monitoring in a Multi-tenant OSGi Execution Environment. Proceedings of the 17th symposium on Component Based Software Engineering (CBSE), June 2014, Marcq-en-Baroeul, France.
- K. Attouchi, G. Thomas, A. Bottaro, G. Muller. Incinerator – Eliminating stale references in Dynamic OSGi applications. INRIA – Research Report N° 8485, February 2014
- Y. Maurel, A. Bottaro, R. Kopetz, K. Attouchi. Adaptive Monitoring of End-user OSGi-based Home Boxes. Proceedings of the 15th symposium on Component Based Software Engineering (CBSE), June 2012, Bertinoro, Italy.
- M. Anne, K. Attouchi, D. Henry-de-Villeneuve, J Pulou. Jasmin: an alternative for secure modularity inside the digital home. Proceedings of the 15th symposium on Component Based Software Engineering (CBSE), June 2012, Bertinoro, Italy.
- Jacky Estublier, German Vega, Elmehdi Damou. Resource Management for Pervasive Systems. International Workshop on Engineering Service-Oriented Applications, WESOA'12, Nov 2012, Shanghai, China.
- Jacky Estublier, German Vega. Managing Multiple Applications in a Service Platform. International Workshop on Principles of Engineering Service-Oriented Systems, PESOS 2012, Jun 2012, Zurich, Switzerland.

■ PhD theses

- Koutheir Attouchi. Managing Resource Sharing Conflicts in an Open Embedded Software Environment, Pierre et Marie Curie University, July 2014. (English)
- Elmehdi Damou. ApAM : An Environment to develop and execute Ubiquitous Applications. Grenoble University, Oct. 2013. (French)