# A Dynamic Distributed Fedrated Approach for the Internet of Things

February 12, 2011

# University of Alcalá - UAH Thales-TAI

## A Dynamic Distributed Fedrated Approach for the Internet of Things

Diego Casado Mansilla
Juan Ramón Velasco Pérez
Mario López-Ramos

Madrid - February 12, 2011

**Abstract** In this position paper we discuss the some integration and Internetworking challenges in user-generated Internet of Things applications as well as some outcomings, as being worked on among the large consortium of the DiY Smart Experiences project (DiYSE, ITEA2 08005). The main objective on the project, is to allow people to control their smart environment at home and in the urban area as part of an open Internet-of-Things world. DiY tools, objects, services and application templates together with an Internet-of-Things application-creation environment will lower the barrier to application creation and distribution for non-technical people;

# 1 Introduction

As technology advances and becomes more pervasive, the DiY (Do-it-Yourself) paradigm that emerged on the furniture & home decoration market in the 70's is now experiencing a second birth in the digital realm. Continuing from the prosumer paradigm, where people are allowed not only to surf a network obtaining content and information, but also (co-)create such elements themselves, the user-centered participation is expected to further increase beyond the Web 2.0 as we know it. The work here presented aims at applying the freedom of creativity of Web 2.0 to the IoT, it is essential that non-expert users are easily enabled to search for public devices or share their own, privately bought or DiY-built devices, and as such, personalize their physical environment by aggregate, combining and mashing-up the device's functionalities. In this context, users should thus be capable of using any kind of existing device in order to integrate them in the DiYSE ecosystem, regardless of whether these were or were not previously known by the system.
To allow this dream of simply and easy application creation on pervasive spaces which are populated for a disparate number of smart-objects (with different communication protocols, addressing schemas, device and service description and discovery, etc.), a set of common standars for interoperability purposes among different smart spaces are needed. In this short paper we outline some parts of the research work done in the IoT context by briefly starting from the object and isolated networks integration on the Internet (discovery and unique identifiying), following with the automatic object's functionality description (sintactic and sematic) to create true smart-objects, and finally by introducing some top level concepts like presence, object ownership, event-based and data-based architectures and object mobility.

# 2 Work Description

## 2.1 Discovery and Addressing on the Intranet

The first challenge requires a flexible layer which abstracts from the underlying, heterogeneous sensor network technologies and supports fast and simple deployment and addition of new platforms, facilitates ecient distributed query processing and combination of sensor data, provides support for object mobility, and enables the dynamic adaption of the system configuration during runtime with minimal programming effort. To this aim we have developed a framework for the device integration based on OSGi on a more powerful devices which are called DiY-Gateways. We consider two types of Gateways ("local" and "on-the-cloud") depending of their physical location. Nevertheless both share the same features: Powerful nodes (e.g. laptops, smart-phones, PCs, a specific purpose devices or even software pieces) with several communication interfaces (e.g. Zigbee, BlueTooth, Wi-Fi, Ethernet, USB, etc.). The idea behind these nodes, is to behave as proxies on behalf the heterogeneous local networks and devices, which are connected to them and that do not always have enough resources nor capabilities to be connected directly to Internet (e.g. nodes devoid of IP stack), and therefore to the DiYSE system. These gateways are intended to hide the complexities of the underlying heterogeneous networks/devices by exposing them in a common way – global IDentifiers, device description, I/O interfaces,

etc. and offering the means to monitor and control them from wherever in the DiYSE system.

## 2.2 Presence, Mobility and Remote Object-Access

The ways to connect, expose, and offer the objects and their respective services in the system by easing their integration in the so called Smart Spaces. The objective is to limit the logical/physical context where newcomer objects and services may interact and be addressable or discoverable.

Smart space is a term that not very commonly used in research community. However, some definitions do exist. In most of them the key issue is that smart space is a physical space full of devices and services that function autonomously on behalf of an individual user by providing an ambient and continuous context and reacting and responding to ongoing activities.

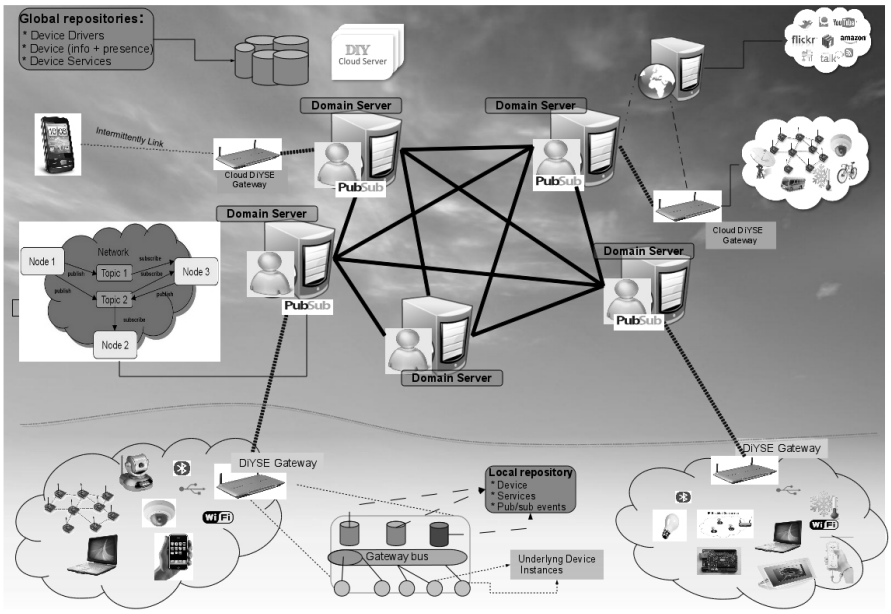


Fig. 1: Proposed architecture for global objects communication.

Normally, a set of devices in such smart spaces, (e.g. home or work environment) are configured in such a way they share common protocols and mechanisms called Service Discovery (SD) to address the heterogeneous device's identification, description, eventing, discovery, etc. Some well known examples are UPnP, DPWS or DLNA. The challenge in these local contexts is that SD mechanisms are very heterogeneous, they do not interoperate together and they are not intended to be extended into a global context such as Internet. This latter constraint is due to SD rely on multicast communication for discovery and eventing within their shared local environments, but such packet delivery technique is not suitable for large scale networks. Our vision and work to improve and solve this issue is to use the XMPP protocol as a substrate for the connec-

tion of distant gateways by extending the Service and Device Discovery and the Eventing to the whole distributed and federated network. The idea behind this approach is that every gataway is aware of the changes in the whole system. The objects could be monitered and addressed by their owners since all of them hava a XMPP prensence not mattering where those are located (e.g. mobile devices).

In order to provide communication between gateways and to interconnect the isolated local domain networks, we have chosen to leverage the eXtensible Messaging and Presence Protocol (XMPP). XMPP is an open XML-inspired Internet protocol traditionally used for online chat communications. Originally based on the Jabber protocol, XMPP has evolved to incorporate features well beyond simple instant messaging such as: event publishing, voice streaming, file transfer and profile information management. The notion of presence, which is central to its operation, refers to the ability for groups of clients to detect other clients connecting and disconnecting from the system. This is critical both to identify when a device or a service becomes available/unavailable and to direct cross-domain client-to-client communication given potentially new locations in the network. Although peer-to-peer implementation of XMPP exists, the typical architecture of XMPP is a pure client-server model, whereby clients connect to a server and servers connect to federated servers for inter-domain communications. Servers Federation guarantees more spreading of resource usage and controlling between services and connected devices. In a federated architecture, each server is responsible for controlling all activities within its own domain and works cooperatively with servers in other domains as equal peers. XMPP therefore, is a decentralized communication network, which means that any XMPP user can message any other XMPP user not mattering where devices are located or connected. XMPP servers can also talk to one another with a specialized server-to-server protocol, providing interesting potential for decentralized social networks and collaboration frameworks. These servers are responsible for authentication, message delivery and maintaining presence information for all users within their domains. If a user needs to get information of one user outside of their own domain, their server contacts the external server that controls the "foreign" XMPP domain and retrieve information to that XMPP server. The foreign XMPP server takes care of delivering information of the intended user within its domain. This same server-to-server model applies to all cross-domain data exchanges, including presence information.