Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability

Grant agreement for: Collaborative project

Grant agreement no.: 288445

Start date of project: October 1st, 2011 (36 months duration)

Deliverable D8.6: Final IoT6 Project Report

Contract Due Date	31/07/2014
Submission Date	30/09/2014
Version	final
Responsible Partner	UL
Author List	Latif Ladid, Sebastien Ziegler, Socrates Varakliotis, Peter Kirstein and all WP leaders
Dissemination level	PU
Keywords	Internet of Things, Achievements

Project Coordinator: Mandat International (MI)

Sébastien Ziegler sziegler@mandint.org

Table of Contents

Table of Figures

Abstract

This deliverable documents the entire project's achievements and impacts. It describes these achievements by work package. It then derives a series of recommendations arising from the theoretical considerations and the concrete implementations carried out related to IoT and IPv6. Several conclusions are then presented on how IPv6 and IoT mesh well together, why IPv6 is so important to IoT, what features are particularly relevant and how its impact can be extended further.

1 Executive Summary

The Internet of Things (IoT) is one of the hottest topics in Europe and in the world.

Overall, the Internet of Things has been a confirmed subject for the European Commission since the adoption of the Communication “An IoT Action Plan for Europe” in 2009[1] and is a subject supported by Horizon 2020, the EU Research and Innovation Framework Programme, starting in 2014.

In 2010, Viviane Reading, the EC Commissioner, confirmed the importance of IoT in the EU in her speech in 2010 entitled “Bringing European values to the Internet of Things” http://europa.eu/rapid/press-release_SPEECH-10-279_en.htm.

There is a long history of activities on the Internet of Things in Europe, as also supported by the EC in many research projects. A great deal of information can be found on the Internet of Things European Research Cluster Website, including the latest Cluster book with the Strategic Research and Innovation agenda, which can be downloaded from: http://www.internet-of-things-research.eu.

The IoT6 project has contributed fundamentally and comprehensively to the re-orientation of the IoT concept to use IPv6 as the networking communication of choice. This achievement was only possible through multiple practical, technical and leadership initiatives, to name a few:

- Leading the adoption of IPv6 as key communication protocol;
- Winning ETSI’s support through the contribution to the IPv6 STD book;
- Winning ETSI’s support to lead the ETSI IP6 ISG;
- Leading the IoT Forum and IPv6 Forum;
- Leading the IEEE ComSoc IoT subcommittee;
- Contributing to the IoT Book for the 4th time;
- Leading the IoT Week program;
- Chairing many IERC, IEEE and IPv6 Forum conferences and invited as speakers in standardization;
- Coalition with the IPSO Alliance and ITU-T;
- Extending the IoT6 pilots to other projects.

[1] http://europa.eu/legislation_summaries/information_society/internet/si0009_en.htm

2 IoT6 Presentation

IoT6 stands for “Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture, enabling heterogeneous components interoperability”. IoT6 was a 3 year FP7 European Commission funded research project coordinated by Mandat International from October 2011 until September 2014.

It aimed at exploring the potential of IPv6 and related standards (6LoWPAN, CORE, CoAP, etc.) to overcome the shortcomings and fragmentation of the Internet of Things.

Its main challenges and objectives were to research, design and develop a highly scalable IPv6-based Service-Oriented Architecture to achieve interoperability, mobility, cloud computing integration and intelligence distribution among heterogeneous smart things components, applications and services. Its potential was researched by exploring innovative forms of interactions such as:

- Information and intelligence distribution ;
- Multi-protocol interoperability with and among heterogeneous devices;
- Device mobility and mobile phone networks integration, to provide ubiquitous access and seamless communication;
- Cloud computing integration with Software as a Service (SaaS);
- IPv6 - Smart Things Information Services (STIS) innovative interactions.

Aims and objectives:

The aims and objectives of the project were:

1. To research the **potential of IPv6** and related standards **to support the future Internet of Things** and to **overcome its current fragmentation**.
2. To develop a **highly scalable IPv6-based Service-Oriented Architecture** to achieve interoperability, mobility, cloud computing integration and intelligence distribution among heterogeneous smart things components, applications and services.
3. **To explore innovative forms of interactions with:**
   a) Multi-protocol integration and interoperability with heterogeneous devices;
   b) Mobile & cellular networks;
   c) Cloud computing services (SaaS;
   d) RFID tags and related systems, such as EPCIS;
   e) Information and intelligence distribution.

Main outcomes:

The main outcomes of the IoT6 project are recommendations on the exploitation of IPv6 features for the Internet of Things and an open and well-defined IPv6-based Service Oriented Architecture enabling interoperability, mobility, security, scalability, cloud computing and intelligence distribution among heterogeneous smart things components, applications and services - including with business processes management tools.

3 Main Achievements of the IoT6 Project

3.1 Achievements in WP1

The main objectives of WP1 were the following:

- Identify relevant Use Case scenarios and derive requirements in close cooperation with the IIAB.
- Research, design and define an open IPv6-based service oriented architecture enabling flexible integration interoperability and intelligence distribution among heterogeneous sub-systems of smart things.

The work package was organised in two Tasks:

- Task T1.1 IoT6 Requirements and Scenario Definition: This Task aimed at specifying, analyzing and evaluating the IoT6 requirements, from both technical and conceptual points of view.
- Task T1.2 IoT6 Architecture Design: Based on T1.1, this Task designed and described the IoT6 IPv6-based service-oriented architecture to be developed in order to enable the integration and interaction among various components of the Internet of Things, and their integration with cloud computing applications (Software as a Service) and business processes management tools. It tended towards a unifying (or integrating) framework overcoming the heterogeneity and fragmentation of the Internet of Things. It served as a common reference document for the other WP developments, as well as for the dissemination work. This Task took into account the work developed by other research projects from the IERC Cluster.

During the first 6 months of the project, activities focused on T1.1 “Definition of Use Cases and Derivation of the Requirements”. The identified Use Cases, the high level requirements and a high level architecture were discussed with the IIAB and their feedback used to update relevant outputs.

Based on the outputs of T1.1 (identified Use Cases and the requirements) and taking into account the existing state of the art in terms of IoT architectures, during the second 6 months, the focus of the project was on the initial IoT6 architecture definition. This was done in collaboration with all technical work packages to ensure relevance of the output for all technical items. The work done in Year 1 was documented in deliverables D1.1 (IoT6 Use Case scenario and requirements definition report) and D1.2 (First version of IoT6 architecture & SOA specifications).

The initial architecture defined at the end of Year 1 was used during the second year to drive and guide research in other work packages. At the same time, using the outputs of other work packages, the initial architecture was further detailed and updated. At that time an early specification of the IoT-A architecture reference model [1] was released. As one of the main premises of the project in regard to architecture design was to build on the work of other projects and reuse the outputs where relevant, the initial comparison and mapping of the IoT6 architecture to IoT ARM was done. This enabled us to align the terminology and the functionality of IoT6 architecture with the one recommended by the IoT ARM. The work done in Year 2 is documented in D1.3 (Updated version of IoT6 architecture & SOA specifications).

In the final year of the project, the activities focused on the finalization of the IoT6 architecture using the IoT ARM as the main reference point. In this period, we went

through the process described by IoT ARM methodology in order to produce IoT6 architecture as much compliant with the IoT ARM as possible, given the work already done in the project. We selected one Use Case from WP7 and analyzed it according to the methodology defined by IoT ARM. Based on this analysis, we updated IoT6 architecture and aligned it with the IoT ARM thus giving the opportunity for other projects to easily identify the additional components introduced by the IoT6 project due to specific requirements and focus on IPv6 as well as to reuse it in their projects if applicable. The work done in Year 3 is documented in deliverable D1.4 (Final version of IoT6 architecture & SOA specifications).

3.2 Achievements in WP2

The achievements of WP2 greatly exceeded the *a priori* expectations.

There was, however, a problem in that the DoW was written to explore the potential of IPv6 from the viewpoint of a network stack. The WP was to investigate advanced features and to produce a network IPv6 stack incorporating these features and to incorporate security, scalability, performance and self-healing. The problem was that some of these features depended not so much on the stack as on the interaction between entities, many of which were dependent on the operating system used, and some even on the hardware. Moreover, many of the protocols were standardised in other bodies; this did not impact our using them, but did impact the timing. If the aim was only to provide a stack that could be used by other WPs, it had to be frozen by the end of Year 2; the other WPs could not cope with a stack that changed during the integration phase and changed while preparing for the demonstration.

In order to resolve this dilemma, we developed a stack as required and froze it at the end of Year 2. The stack worked with Linux for gateways and Contiki for small devices, and supported 6LoWPAN, RPL routing and all other IPv6 advanced features, by the third quarter of the second year. This stack was delivered to WP3 and hence the other WPs on schedule. All this work was discussed in detail in deliverable D2.3 (Report on IPv6-based advanced features).

During the third year, detailed experiences were recorded on the performance of this stack, and reported in deliverable D2.4 (Implementation and testing report on IPv6-based IoT6 features). In particular, while Contiki was the agreed choice of OS, several advanced IoT6 features that had to do with security (Datagram Transport Layer Security (DTLS) messages between embedded devices and between devices and gateways) required larger amounts of memory to be implemented.

Another important IoT-specific feature that was developed was GLoWBAL. This was an algorithmic mechanism for assigning IPv6 addresses to devices that might be accessible only through IP-enabled gateways, and had features understood only by a legacy technology. This mechanism was clearly very important as an enabler of large-scale deployment. Its utility was greatly enhanced by features developed in WP3 based on the Digcovery repository. It was included in the stack delivered to the partners.

Another aspect of WP2 was an investigation of how to provide security. While we showed that the popular encryption algorithms could be implemented on small devices, the provision of real security required a complete security infrastructure which was not really part of WP2, and whose provision had not been budgeted. Moreover, we had envisaged smart routing to be a feature that would be implemented at the network level. In practice, this use of the IP header had been deprecated at that level in the Standards Body (IETF), so we agreed to provide it at a service level under WP3. As a result, partly based on the reviewers' comments, we revisited the stack to incorporate datagram security DTLS with the latest versions of both the operating system Contiki, a compatible

version on Linux, and the latest versions of the transport protocol CoAP adopted in the project. The implementations of the security features were found to be platform-dependent in terms of memory requirements, and could not operate on the specific constrained platforms used by the partners in the main demonstrations, which by nature would utilise more application layer resources, even though all other aspects could be identical. For this reason, we decided to validate the implementations only in another set of demonstrations, which simplify the application layer in order to shift available memory towards the secure datagram layer below the application/CoAP layer. It was pursued further with much more advanced features that were incorporated into a complete validation demonstration but not that of one involving the majority of the partners.

Having fulfilled all the other requirements of the other WPs by the end of Year 2, we moved to a very promising set of activities in Year 3. We had come to the conclusion that the use of identifiers was a logical, and much more powerful, extension of the ideas demonstrated in GLoWBAL. Moreover, we found CNRI's Handle system incorporated all the features of real security we required and could be directly combined with all the requisite features. This system was already deployed for other application domains, incorporated a strong security infrastructure and had proven scalability capability. On the request of IoT6, CNRI ensured that their system would both operate with IPv6 features and be IPv6 addressable. Moreover, they were developing a new Release, which would incorporate RESTful programming mechanisms that are central to the IoT6 approach and promised to give us a pre-release version of their new system before the end of the 3rd Year. An important advantage of this approach is the way that IPv6 addresses, Internet services and GLoWBAL interwork. IPv6 addresses are usually stored in the DNS, and this is a key feature of our WP3 approach. Anyone can access the DNS, and hence inspect the IPv6 addresses which under GLoWBAL may reveal features of the target subsystems. With Handle, such access is restricted to authorised users with a sophisticated and fine-scale authorisation. Thus, by using Handle, we have been able to demonstrate it in a Use Case that is a subset of the main one in the project but with secured, deployable and scalable features.

The main additional problem that this approach raised was managerial, not technical. Going from the network service to a complete validation required activity that really belonged in WP1, WP3 and WP7. The reporting in WP1 was straightforward; there were substantial contributions to D1.4 (Final version of IoT6 architecture & SOA specifications) which included portions on scalability, governance and security. However, the duration of WP3 was scheduled to end in January 2014, and the deliverable contents in WP2 and WP7 had been worded slightly differently. We decided, therefore, to describe our approach to the first problem to the reviewers at the end of Year 2 and received their approval. When considering how to report the validation, we determined that the description of the approach would be reported in deliverable D2.4 (Implementation and testing report on IPv6-based IoT6 features), but that deliverable D5.4 (Intelligence distribution tests and evaluation report) was a much more natural home for the detailed treatment of the Use Case even though the work was validation. However, the validation effort was charged to WP7 where it belongs technically. In addition, this work has been widely disseminated in talks, papers and the relevant EC body considering standardization in IoT.

3.3 Achievements in WP3

In this WP, we researched and developed a service layer enabling the interaction with different kind of Internet of Things components. We proposed an architecture and middleware for the scalable integration of actuators and sensors in a network of ubiquitous sensing. The objective was to define mechanisms to support the search for an effective service layer for the sharing of sensor and other smart things information in real time, search and browse, as well as discover resources and information in a distributed

and loosely coupled approach.

- Task T3.1 Overlay Service Layer: Look up and discovery, context-awareness and resource repository. Within this Task, a lightweight multicast DNS (lmDNS) for IPv6-enabled Smart Objects was used in order to overcome the limitations of mDNS which is designed for host-based requirements, where they are not taking into account the design issues and constraints of Smart Objects. As a consequence, this work converged to a global discovery architecture interoperable with DNS called Digcovery and accessible via www.digcovery.net. Individual drivers were designed to interconnect different kinds of objects, things, devices, sensors and tags (RFID, Handle System, legacy technologies, etc.) Finally, a search engine, an access control policies, and a set of management functions were proposed. All these elements contributed towards the key purpose in the IoT6 project, to build an Open Service Layer which makes feasible its full integration into the IPv6 architecture through protocols such as DNS, and other communication interfaces which define the Open Service Layer. Finally, as part of this Task, Local and Global Discovery interactions based on mDNS/DNS-SD and overlay networking solutions were analyzed and how to publish/search globally the resources and devices registered at the local level.
- Task T3.2 Smart Routing Mechanisms: Task T3.2 provided deliverable D3.2 on Smart Routing. Two main solutions were investigated and tested with the gateways, in order to support the traffic differentiation from the IPv6 sensor nodes to a multicast/anycast address on the IPv6 intranet.
- Task T3.3 Service Layer Implementation and Tests: This Task worked on the integration of the initial design and solution under the OSGi framework to provide a common API to be used within WP3-WP6 for validation. Also the Java OSGi bundles were extended to support some of the WP2 functionalities. The IoT6 Stack has been proven in four environments: Contiki-motes, OSGi-Gateway, Digcovery-server and mobile phones. These implementations provide the functionalities of IPv6 connectivity and Open Service Layer defined in WP2 and WP3, respectively. Both implementations support IoT6 interoperability in heterogeneous networks such as wireless sensor devices and legacy technologies (BACnet and KNX). The implementations have been divided into several modules according to their functionalities: IPv6 Addressing, Quality-of-Service, Service Discovery and Web Services, etc. The validation has been done in relation to the Use Case and interaction with the rest of the IoT6 components.

Main outcomes:

- Stable IoT6 stack platform based on:
    - OSGi and Contiki components deployed and tested by other WP;
    - Digcovery platform allows registration and homogeneous access to the information provided by sensors or other devices;
    - Providing context awareness with the aid of MongoDB;
    - DigCovery makes use of enabling IPv6 QoS to control its own traffic.
- IoT6 Open Service Layer enables that Smart objects can be discoverable, accessible, available, usable, and interoperable through IPv6 technologies like:
    - Lightweight multicast DNS (lmDNS) for local discovery in IPv6-enabled Smart Objects;
    - Digcovery for scalable global discovery architecture interoperable with DNS-SD directories;

- Common description based on ontologies (SSN) and profiles (IPSO);
- Elastic Search for look-up and context-awareness queries.

3.4 Achievements in WP4

The main goal of this work package was to bring non-IP based communication systems and mainly the Building Automation Systems (BAS) from their closed domains toward the IPv6 world. All the efforts in this WP have been focused on this problem.

The first step was to understand the system architecture that could support the integration of BAS within the Internet. To this end, we reviewed the main existing building automation protocols, in order to choose the ones to take into account in the design of the architecture and to understand their main features and the constraints that they impose on the architecture itself.

In a second phase, we focused on the high level design of the architecture, choosing among available frameworks and components, with the goal of guaranteeing a seamless integration and management between all the protocols considered. Two different approaches were considered. The first approach was based on enriching the Universal Device Gateway (UDG)[2] with the characteristics needed to satisfy the requirements defined in the work package. The second approach was based on creating a BAS gateway using a generic semantic exploiting a standardised Information Exchange mechanism[3]. The latter approach led to the creation of the IoTSyS integration middleware[4].

When the two systems, UDG and IoTSyS, were built, the partners collaborated together to integrate them. This task was performed in order to demonstrate that the Control and Monitoring System (CMS) developed within UDG could work with different kinds of gateways, despite using different semantics to manage devices from different technologies. The integration of these two systems ensured the satisfaction of the requirements in the WP4 directives on the integration of different legacy technologies.

The WP worked on the implementation of the CMS, improving its previous architecture in order to allow its usage in the scenarios envisaged within IoT6 project. Some key requied adaptations included the capabilities to manage Virtual Variables, Dynamic Targets and Groups of Devices; creation of a homogenized IPv6 mapping to non-IP protocols aligned with IoT6 views[5] (proposed as a standard) that allows the automatic assignment of a unique IPv6 address to each legacy devices residing under the CMS; the capability to distribute the intelligence (logic that is defined within the CMS) among different nodes, thus increasing the scalability; adapting the Graphical User Interface (GUI) that allows to configure easily the devices and to design the IoT6 scenarios desired using all new features introduced.

The next step was to integrate the two solutions (IoTSyS and UDG) into the IoT6 ecosystem. This task was performed by using the IoT6 Stack, which was designed to allow all the IoT6 Components to work together, enabling more complex and fascinating scenarios.

---

[2] http://www.devicegateway.com/

[3] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=obix

[4] https://code.google.com/p/iotsys/

[5] http://datatracker.ietf.org/doc/draft-rizzo-6lo-6legacy/

Lastly, we performed a study about different schemes for Machine-to-Machine (specifically Device-to-Device) interactions, in order to understand their main features and their differences, and to determine which one fits the context of the project. The outcome of this study has been documented in the deliverable D4.4 (Report on heterogeneous device interoperability and multi-protocol integration).

3.5 Achievements in WP5

WP5 “Smart board and intelligence distribution” was aimed at integrating the concepts that were developed within work packages WP2, WP3 and WP4. The main goal was to **implement** and **test** the intelligence distribution tools and some specific routing and security mechanisms dealing with a complex ecosystem of heterogeneous components and heterogeneous applications and services.

The first aim of WP5 was to design an embedded board (“Smart Board”) offering multiple physical interfaces and supporting translation protocols able to integrate legacy devices within IPv6 networks. This embedded board was aimed at providing the different research teams with the same generic compact system able to cope with heterogeneous devices, networks and protocols such as found in Building Automation Systems (BAS).

Part of the challenge in designing that board was to ensure a large spectrum of physical accesses compatible with the numerous Use Case scenarios in deliverable D1.1 and a modularity allowing the rational use of components for different deployment scenarios. The hardware components and part of the firmware were developed in Task T5.1 as documented in deliverables D5.1 (Document on selection of circuits and functionalities) and D5.2 (Smart Board design and realisation report, including board prototype validation tests).

The multi-protocol integration, i.e. the implementation of the IoTSys architecture (designed in WP4), and its deployment on the Smart Board was carried out in Task T5.2 (with the main results described in deliverable D4.3 (Multi-protocol integration report). The multi-protocol interoperability was realised by using the OSGI framework through the development of a set of protocol bundles associated with the main BAS technologies (KNX, Bacnet, M-Bus, En-Ocean, RF-ID and ZigBee). The protocol bundles were deployed on the Smart Board, providing, together with the gateway components, a lightweight access to the heterogeneous technologies through the IoT6 stack.

A dedicated software application hosting the IoTSys as well as the Smart Routing components were developed to handle the configuration of those components at launch (IoT6 Launcher).

The distribution of intelligence (such as studied in WP4, deliverable D4.2) was ensured by integrating the Smart Board software components into the framework of the IoT6-based CMS, (UDG). IoT6-based monitoring functionalities such as resource discovery (based on the Digcovery studied in deliverable D3.1 (Look up and discovery, context-awareness and resource) and the Smart Things Information Service (STIS) have been implemented as well as powerful and flexible mechanisms providing dynamic control rules. See deliverable D4.2 (Multi-protocol architecture and system development report). Those UDG flexible mechanisms can be deployed in a multi-stage, hierarchical configuration and are therefore adaptable to a wide variety of control loops, from the lowest-level, direct control of an actuator set point, to high-level control, such as keeping a room with all its various properties in the comfort zone set by the user.

The content-based `Smart Routing' mechanism described in deliverable D3.2 was implemented within the IoTsys middleware in order to improve the routing capabilities, allowing the more efficient routing of sensor values.

Having implemented such a powerful IoT6 architecture with a plurality of actors/components, it remained to find a coherent way to challenge and test the

implementation. This quite difficult task was the subject of Task T5.3: Intelligence distribution tests and evaluation and was carried out by taking into account the content of deliverable D7.1 (Test process specified), which provided a general framework (abstract architecture) for each Use Case scenario planned in deliverable D1.1 (IoT6 Use Case scenario and requirements definition report). Test cases were performed in order to evaluate the “Smart routing” and QoS features enabled by the Smart Board. The test cases showed how the Smart Board manages reliability the data flows between the heterogeneous devices (IP sensors and legacy actuators) and various control systems such as CMS, Safety Server and SaaS. The evaluation results showed that all test cases were completed properly and the smart routing and QoS mechanisms were implemented successfully in order to achieve intelligence distribution among heterogeneous IoT devices and control systems.

Last but not least, WP5 (in line with deliverable D2.4 (Implementation and testing report on IPv6-based IoT6 features findings) proposed an original way to provide security operations such as authentication, authorisation, confidentiality and integrity using distributed elements based on the Handle System and the DTLS cipher suite. Several Use Cases were proposed for the validation of security activities in the final demonstration. Although the application of a strict security policy through the deployment of a security infrastructure is beyond the scope of the IoT6 project, the Handle System provides most of the technology needed to incorporate a credible measure of security into the sort of Use Cases we are studying in IoT6. The security proposal of the Handle system has shown that this approach could have impact on IoT far beyond the IoT6 project.

3.6 Achievements in WP6

This work package focused on the research challenges related to the integration of IoT6 with mainstream applications, such as business processes applications using the cloud computing platform of Software as a Service (SaaS), mobile networks, and the Smart Thing Information Service (STIS).

In order for IoT6 to be interoperable with STIS, an analysis of unique identifiers was necessary, because STIS has its own identification system and it should be interoperable with that of IoT6, namely IPv6. Deliverable D6.1 (Unique identifier analysis report and STIS, ONS, IoT6 integration) reports on the analysis of unique identifiers and the integration of IoT6 with STIS and ONS. We analysed various unique identifiers such as Uniform Resource Identifier (URI), the Handle System micro ID (uID), Object Identifiers (OIDs), Universally Unique Identifier (UUID), Electronic Product Code (EPC), etc. Among them, we concluded that URI was the best choice for the identifier to enable interoperation among heterogeneous identifiers. In deliverable D6.3 (Interface between IoT6, STIS and ONS validation report), an interoperability test was performed to verify IoT6’s interoperability with STIS, and its result was presented. The tests included unit tests for each interface as well as integration tests between IoT6, STIS, and ONS. We showed the feasibility of design and implementation of integrated system. Finally, we tested and presented the result of the proposed unique identifier implementation in the frame of IoT6 architecture. In deliverable D6.4 (Innovative interactions between STIS and IPv6 through IoT6 report), IEEE 802.15.4-based active RFID tags with STID was introduced as innovative interactions between STIS and IPv6. To enable interaction between these Smart Things and STIS, 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) was employed as a vehicle to integrate non-IP-based Smart Things with STIS. As a result, it was shown that coverage of IEEE 802.15.4-based active RFID networks can be easily extended with the aid of 6LoWPAN networks, which is called 6LoWPAN-based active RFID networks. Also, by integrating a 6LoWPAN gateway with LLRP (Low-level Reader Protocol) readers, Smart Things and STIS can

communicate with each other without any modification, through 6LoWPAN-based active RFID networks.

The requirements for the interoperability of IoT6 with mobile networks were proposed by Task T6.1. This Task had the goal to propose the best option to implement and test the aforementioned forms of interactions between mobile phones and IoT6. Deliverable D6.2 (Ubiquitous access and mobile phone network interactions report) analyzed the registration procedure, addressing of mobile devices, and mobile devices acting as Gateways and half-Gateways in IoT6. It also explored various ways to integrate an IPv6-based Internet of Things into mobile phone networks, enabling mobile devices to provide access to Smart Objects, as well as to use mobile devices as sensors/actuators. Also, for ISPs not supporting IPv6, the tunneling mechanism was used. The ability for Smartphone sensor discovery and data gathering was presented, with possible options for non-IP devices. The feasibility of mDNS implementation on a mobile phone was also studied.

For the interoperability of IoT6 with the business process management tool, the CoAP protocol and JSON format should be supported. Deliverable D6.5 (Business Process Management tools and Cloud Computing applications integration report) demonstrated the feasibility of interaction and a new kind of application. As a result, the RunMyProcess Business Process Management tool was interfaced with the Internet of Things by adding new functionalities to allow CoAP connectors and the development of a CoAP proxy to make the platform visible to CoAP objects. Also, it exposed a vision of interaction between SaaS applications, the Internet of Things and legacy Web services, called `Composite Business Ecosystems for the Web of Everything`.

3.7 Achievements in WP7

The first action in this work package was to describe the Use Cases in deliverable D1.1 (IoT6 Use Case scenario and requirements definition report). These Use Cases were analysed and refined. It was found to be necessary to re-evaluate the mentioned components and how they fit with the final IoT6 architecture. To achieve this task, the Use Case descriptions were subsequently completed and spread among the partners. Thereafter, the Use Case descriptions and sequence diagrams were updated.

Furthermore, an interoperability testing strategy had to be chosen. Different testing strategies were evaluated for testing the interoperability of the IoT6 architecture. Finally, the ETSI EG 202 237 guideline which is also the basis for the PROBE-IT EU FP7 project was selected as suitable.

The first step of the testing guideline was to extract the interface descriptions from the Use Case descriptions and sequence diagrams. Afterwards, the test cases were developed. Deliverable D7.1 (Test process specified) describes the scenarios and interfaces between components of the IoT6 system. The test groups and test purposes were defined, as the basis for the development of formal test cases. As a result, the main outcome of this deliverable was a set of test cases necessary to perform the interoperability tests.

The next task was the development of a concise test plan for the defined test process that could be used to test the interoperability of the IoT6 components. The scenarios of the initial test strategy were adapted to a consolidated “extended Smart Office Use Case” that was agreed upon iby the IoT6 consortium. The test cases originally defined in D7.1 (Test process specified) were adapted to reflect a scenario that involved all elements of the IoT6 architecture and allowed a thorough interoperability test, at the same time economically using available resources. In this context, so-called “abstract architectures” were defined for all steps of the scenario detailing the test setup and allowing the identification of communication dependencies between components and partners. Following, a distributed test plan was worked out, and dependencies between the

partners were identified. The local test setups at the sites of the different involved partners were adapted to support the interoperability test procedure of this work package. Furthermore, the additional equipment needed for the final demonstration was evaluated.

Finally, in the test execution phase, several iterations of testing sessions were scheduled according to the test plan. These iterations were performed in a way that all involved partners were monitoring their systems and thus validating the correct execution of the agreed-upon extended Use Cases. The coordination of the testing efforts included regular VoIP meetings (“test days”) on which the agreed upon Use Cases were executed step-by-step by the responsible partners. Each test session was documented, in terms of how many test cases associated with the Use Case could be successfully validated. Furthermore,, in the event of unsuccessful test cases, the reason for their failing was logged and documented in the deliverable D7.2 (Components Instantiations and validation report).

Apart from interoperability testing, the second main focus of the work package was to evaluate the scalability of the IoT6 architecture. Therefore, a proposal for a scalability testing methodology was prepared and presented. Different approaches for scalability assessment have been used to test the various components in the compound. In this case, components of the IoT6 architecture have been analysed based on benchmarks as working prototypes were available. The performed analysis clearly shows the limits of scalability depending on the used hardware resources.

All efforts regarding interoperability and scalability testing were collected and documented in the final test report for this work package in deliverable D7.2. In deliverable D7.3 (Smart IPv6 building deployment, tests and recommendation report), the results of the deployment and tests of the IoT6 architecture and components in a real smart office environment are presented. Further, innovative future applications of the IoT6 architecture have been created and introduced thoroughly in D7.4 (Innovative business processes test and validation report).

3.8 Achievements in WP8

The IoT6 consortium has been actively involved and taking leadership in chairing and organizing peer reviewed, well-known, international conferences, as well as actively participating in standardisation efforts.

A summary of the ***dissemination activities*** includes the following (see more details in deliverable D8.2.3):

- Leadership in IoT Forum and IoT Week
- Leadership in IEEE ComSoC IoT subcommittee
- Leadership in IPv6 Forum to organise IoT6 panels
- Organised directly over one dozen conferences
    - 3 IoT Week events
    - 4 IEEE ComSoc IoT events
    - 2 IPv6 Forum panels
    - 3 esIoT workshops
- Participated in 100 partners and industry conferences
- Authored more than 3 dozen papers, including in:

  - 3 IERC Books
  - IEEE ComSoc papers
- Authored the SME handbook
- Created the IoT6 Website

Standardisation activities

The IoT6 consortium has also approached the industrial community and put effort to spread the knowledge and project achievements through the standardization bodies (IETF, ETSI and ITU-T) as they attract the leading industry players: IETF attracts some 1000 experts and ETSI has some 600 industry members. IoT6 sought to interest the industry with the project solutions and impact the standardization process (mainly in the IETF 6LoWPAN, 6lo and 6TiSCH and ETSI ISG).

- Co-authored the standardisation chapter in the IERC Book led by ETSI
- Formed an ETSI Industry Specification Group for IPv6 and IoT

Exploitation

The IoT6 consortium focused exploitation activities on interaction with the SMEs and IoT industrial sector. The planned activities are organized in 2 strands:

- Organization of dedicated events for presentation and promotion of the benefits and advantages of IPv6 based IoT solutions, focused on events targeting SMEs and IoT industrial sector players. See more details in deliverable D8.4.3.
  - Preparation of promotional material, including professionally designed content, aimed at researchers and industry, with specific focus on SMEs:
  - A series of A1 size posters providing an overview of the project and the main outcomes.
  - SME book, providing a description of the main outcomes of the project.
  - Short version of the SME book: a 10-page leaflet providing a summary of the full SME book.
  - IoT6 comic poster and leaflet.
  - IoT6 gadgets: A set of promotional gadgets such as postcards are created including Augmented Reality (AR) in collaboration with a marketing company.
- Three portable IoT6 demo setups were created to enable easy demonstration of the main project concepts and outcomes.
  - TUV portable suitcase.
  - Demonstration at the FIA conference.
  - Demonstration at the ICT Spring conference.
- Contribution to the creation of the IEEE ComSoc technical working group on Internet of Things (IoT) that was created in November 2012. The group is chaired by Latif Ladid (UL), and Vice-Chaired by: Antonio Jara (HESSO), Antonio Skarmeta (UMU) and Sebastien Ziegler (MI).
- Partners' individual exploitation plans are given. The plans comprise a short

description of each partner and their interests, as well as the opportunities each partner sees for exploiting the results developed within the project. See more details in deliverable D8.4.3.

4 Use Cases Summary

To highlight the benefits of an IPv6-based IoT, four different Use Cases have been implemented in the context of the IoT6 project. They show very well how it is possible to interconnect different devices and create interactions between different services. In detail, first, we illustrate the integration of legacy building automation devices into a homogeneous IoT IPv6-based smart office. Then, an advanced scenario regarding building safety is described. And finally, in the last Use Case, focused on building maintenance, we describe the replacement of a faulty device.

4.1 Use Case 1: Smart Office and Legacy Devices Integration

As there still exist a heterogeneous landscape and large variety of legacy devices and networks of things in the building automation domain, their integration into the Internet through a single interface is still a challenge to face. But, actually it can be addressed by IPv6 and the IoT. In the presented Smart Office scenario, several automation devices of different legacy networks (i.e., BACnet, KNX) are integrated through a gateway which is responsible for translating legacy protocol messages into IPv6 packages and providing a homogeneous view on the underlying heterogeneous networks and associated devices.

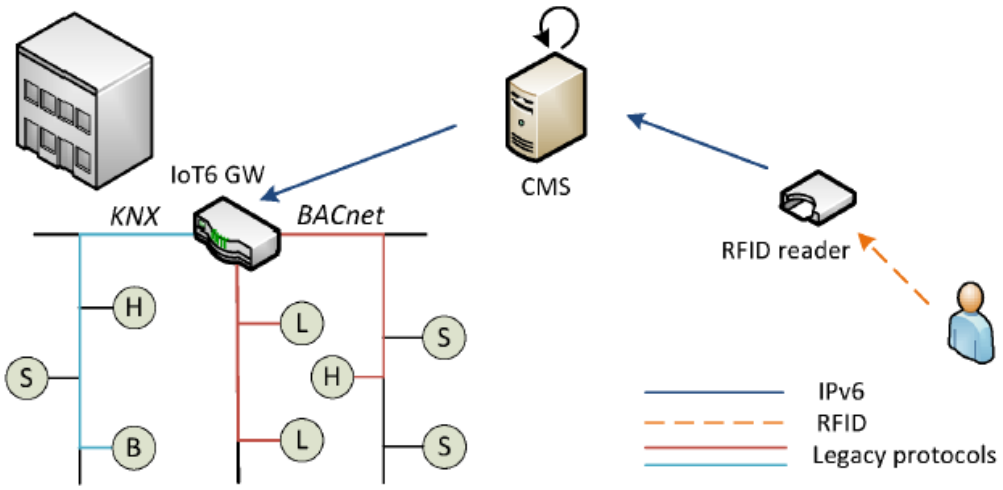

Figure 1: Use Case 1 Smart Office presence

Figure 1 illustrates (i) an IoT6 Gateway (IoT6 GW) integrating several legacy devices, (ii) an IPv6-enabled RFID reader, and (iii) a Control and Monitoring System (CMS) as service in the IoT cloud.

The Smart Office Use Case starts when an employee enters the building and presents his/her RFID badge to the system's RFID reader. As the RFID reader is IPv6-enabled it may directly communicate with the CMS using IPv6. The CMS subsequently chooses the employee's comfort profile for his/her office and sends settings and commands to the IoT6 GW which integrates devices of the particular office into the IoT. The IoT6 GW controls a variety of different devices from heterogeneous building automation networks and masks this heterogeneity by providing a uniform IPv6 interface for all devices. The IoT6 Gateway can in further consequence be used to set user-defined preferences for

the employee in his/her office. In the example case, the heating actuator set point (H) and two brightness actuators (L) integrated through a BACnet network are adjusted. At the same time, also the position of the sunblind (B) which is controlled via a KNX network is adapted according to user preferences. For the CMS, the idiosyncrasies of the different underlying legacy networks make no difference as the IoT6 Gateway transparently integrates these devices into the IoT allowing to adjust them in a uniform way.

A similar situation to the one illustrated in Figure 1 can be observed when the employee leaves the office building. As soon as he/she provides his/her RFID badge to the RFID reader, the CMS is informed that the employee is about to leave the building. In this case, the CMS can execute an energy-saving rule which turns off all devices in the employee's office. Alternatively, a presence sensor in the office combined with a time out could be used to detect absence and initiate the energy-saving scenario.

4.2 Use Case 2: Safety Alert and Dynamic Routing

The second Use Case, slightly more complex than the first one, is focused on an emergency situation and the capabilities of an IoT architecture to deal with this situation. As initial setup for this Use Case, an IPv6-enabled temperature sensor (S) is considered which periodically sends an update of the sensed temperature value to a Control and Management System (CMS) available as service in the IoT. The starting point for this Use Case is a sensor that reads a temperature which is too high (i.e., outside the boundaries of usual operation). The Control and Management System detects this abnormality and flags the received message as an alert message. It sends the value to a smart router which is a component in the IoT that according to the type of the message may take different routing decisions. If a normal temperature message is received, the smart router sends the temperature messages to a Building Energy Management Server (BEMS) that may be responsible for logging and reporting the energy demand of a building.

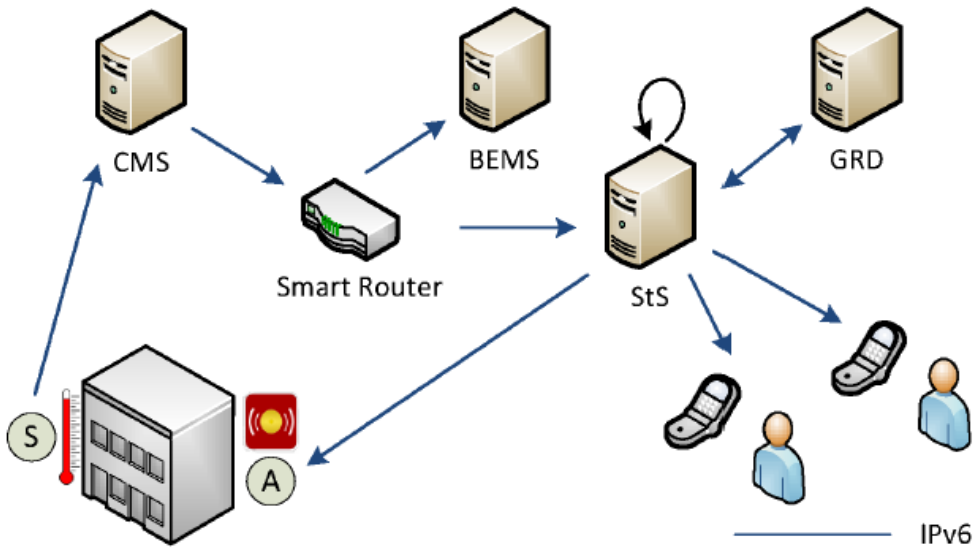

Figure 2: Use Case 2 Safety Alert

In the present case (excessive temperature), however, the smart router detects the priority of the message (alert) and according to the tagging carried out by the CMS, forwards the value to a specific Safety Server (StS) which is responsible for handling alert situations. As the StS receives the abnormal value, it firstly contacts the Global

Resource Directory (GRD) to gather information about the location of the sensor. If the StS already has a list of alarm devices with their location, it can compare the location of stored devices with the location of the temperature sensor to directly turn on an IPv6 enabled alarm device (A) in the vicinity of the alert situation. If the StS has no pre-stored alarm devices for the area for which the alarm was reported, it is possible to issue another query to the GRD service requesting alarm devices that are in the vicinity (e.g., found in 15-meter radius) of the alert. Any device capable of signaling an alarm which is found can subsequently be switched on. Furthermore, the StS may have a list of mobile phones of persons in charge for alert situations (e.g., fire wardens or system engineers). In this case, the CMS has to gather information about the current location of the IPv6 enabled mobile phones from the Global Resource Directory through an additional query. After this information is received, the CMS can inform responsible persons near the area of interest about the alert situation via their mobile phones. As Figure 2 shows, all communication is handled via IPv6 which emphasises the diversity of devices and components that may be integrated in the IoT. If legacy devices are involved, either on the sensor or on the actuator side, again an IoT6 Gateway can be used for integration as described before.

4.3 Use Case 3: Building Maintenance

The third Use Case is related to building service maintenance. It involves a variety of IoT components and demonstrates how these components in combination with IPv6 communication can be used to detect device failures in a building and investigate as well as fix the cause. In the shown case (cf. Figure 3), a number of sensors are connected to the IoT through an IoT6 gateway (IoT6 GW) which assures that all legacy sensors can be accessed in a uniform way through IPv6 communication (as in the Use Case 1). This Use Case starts with the failure of a legacy component in the subnet controlled by a specific IoT6 gateway, e.g., a temperature sensor. Usually, a Control and Management System (CMS) observes the value of a temperature sensor to, for example, detect safety situations or perform energy reporting (as in Use Case 2).

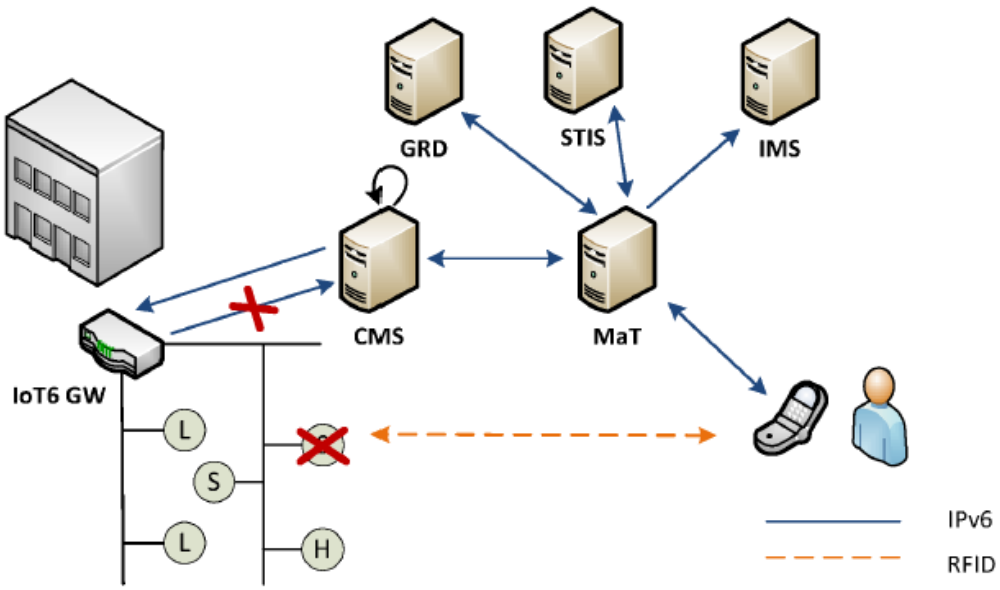

Figure 3: Use Case 3 Building Maintenance

In the case an observed sensor silently fails, a time-out occurs at the CMS, indicating that something is wrong with the device. A message is generated at the Control and

Management System and sent to the Maintenance Tool (MaT) for further examination. In its simplest form, the Maintenance Tool could also run on a local CMS but presently is pictured as global service in the IoT cloud. As soon as the MaT gets the message about the failure of a device, it creates an alert ticket and sends out failure notifications to a variety of mobile phones of responsible persons (e.g., system engineers). The group of recipients may again be based on the current location of the mobile phones for which a look up call to the Global Resource Directory would be necessary (as in Use Case 2). A person associated with one of the contacted mobile phones seeks out the faulty device and uses a maintenance app on the mobile phone to scan its RFID tag. The information is relayed to the MaT which needs to find out the device which is associated to the respective RFID tag. Therefore, it queries the Global Resource Directory (GRD) for the location of the Smart Things Information Service (STIS), a database-like service that keeps associations between RFID tags and devices. The MaT further sends back information to the maintenance app running at the mobile device providing the system engineer with more information about the device. With the help of this information, the engineer has the possibility to run diagnostics on the device. In this case, the CMS further acts as an intermediary between Maintenance Tool and the IoT6 Gateway, accepting and relaying messages from the Maintenance Tool to the IoT6 Gateway. In case the device's defectiveness is confirmed, a replacement order needs to be made. This order can again be performed using the MaT. The information previously retrieved from the STIS may in this case further be used to directly order the spare part from an Inventory Management System (IMS), another service in the IoT. If the address of the IMS is not yet known by the MaT, it first again has to issue a request to the GRD. Alternatively, the IMS may be part of the maintenance tool in which case the separation of the two services can be omitted.

4.4 Use Case 4: Secure Personalised Management of Office Resources

The fourth Use Case is different. The environment is the Smart Office, and the scenario is a realistic portion of its routine operation. However, the purpose is to show the full gamut of operations that are required to set up and run such a scenario securely, in a way that can be generalised and scaled. The integration with the main demo's real sensors and actuators equipping such an office is not attempted; this has been done in the other Use Cases. The main reason for the difference is that the large-scale, real-life, identifier resolution, repository and security was made available (from outside on a pre-release basis) far too late to be incorporated into the activities of the partners deploying such devices. Another reason was the increased embedded system resources required for the extended security operations featured in Use Case 4. Nevertheless, Use Case 4 does use an infrastructure of real sensors (proximity, RFID reader and temperature) and actuators (multiple LEDs, embedded or attached to small IoT6 controllers) to demonstrate the operation of the system. Furthermore, this Use Case, adopts all the lower level protocols and functionality used in the others: 6LoWPAN, IPv6 address auto-configuration, RPL, CoAP, deployed on both the Contiki and the Linux IPv6 stacks where required, as well as RESTful interfaces and programming. It even uses an improved form of GLoWBAL that obscures the process of IPv6 address mapping to identifiers for increased privacy. However, it uses also DTLS for inter-device communication, and interaction with the Handle system to store and obtain securely IPv6 addresses and other attributes, fine-grained authorisation of services on devices, and security tokens. The Handle system used is part of an IPv6-enabled infrastructure with world-wide deployment and proven scalability, flexibility and governance.

The basic scenario is that an office contains a locked door, lights, a heating and air-conditioning unit (HVAC). A person tries to enter the office with an RFID card. If so authorised, the door can be opened, the lights are switched on and the HVAC set at a certain temperature. The HVAC temperature is regularly monitored while the person is in the room, and is adjusted if the room conditions change too much. When the person

leaves the room, this is detected by a presence sensor; the lights and HVAC are then turned off.

There are always two stages in IoT activities: *Set-up* and *Operational*. Of course one sometimes switches frequently between the two.

The individual steps in the operational phase of this Use Case are the following:

Initial Conditions

- The door is locked, the lights are off and the HVAC is switched off.

Door Entry Request:

- A person requests entry by placing an RFID card near a card reader.

Authorisation:

- A Smart Office application checks that he or she is authorised to enter.

Door Opening and Actuating/Logging:

- If so, a process is invoked that actuates the door opener, notes the entry on the logging database, switches on the lights, and sets the air-conditioning to an pre-agreed level.

Sensing/Logging:

- When somebody enters the room, the presence sensor becomes aware of this, and logs an entry in logging database. This may include knowledge of the badge owner, and of anyone not wearing a badge – or an authorised individual.
- *When the* presence sensor indicates that there is no longer anyone in the room, this is noted in the log, and the services revert to the unoccupied condition.

Temperature Control

- While there are people in the room, the temperature is monitored at regular intervals. If it goes outside pre-specified thresholds, the HVAC setting is changed accordingly.

Clearly this scenario is seemingly simple, but the underlying operations are sophisticated. Moreover all inter-entity communication is sent securely via secure DTLS messages. Examples of secure operations are described below for key stages.

Door Entry Request:

- The RFID card is read by a door sensor and the stored RFID value is read.
- The door sensor controller sends the RFID value, a device identifier, and the authentication token of the door sensor to the application.

Authorisation:

- The application uses the RFID value and device identifier to obtain the Handle ID.
- The application sends its Authorisation ID and the Handle ID to the Handle Server.
- The Handle Server authorises the application to check credentials.
- The application sends the RFID value and sensor authentication token to the Handle Server.
- The Handle Server checks the authenticity of the door sensor, the correctness of the RFID value, and the authorisation of the person to enter the room.

- If all are correct, it sends the IP address, device ID and security token of the door opener to the application.

Door Opening and Actuating/Logging Requests:

- If the person is authorised, the application authorisation is repeated to obtain the IP and device addresses of the lights, the same parameters of the HVAC.
- Assuming they are so authorised, these parameters are returned to the application.
- If all are authorised, a process is invoked that actuates the door opener, using its security token, and notes the entry on the logging database.
- The process switches on the lights, and sets the air-conditioning to a pre-agreed level. The relevant network and security tokens returned are used.

Door Opening and Actuating/Logging:

- The different remote devices verify the security tokens they have received, and that they authorise the operations requested. If so, they carry out the operation requested.

Presence Sensing/Logging:

- When somebody enters the room, the presence sensor becomes aware of this, and notifies the application with the sensor identification and authorisation and the RFID of the person and any others with or without badges. It sends an information-logging request to the application with the usual identification and authentication.
- The process of the relevant Handle identification is repeated; the authentication of the presence sensor and its authorisation for the logging operation is confirmed from the Handle Server. The secure presence logging entry is then sent to the logging process.
- When the presence sensor indicates that there is no longer anyone in the room, this is noted in the presence log, and the services revert to the unoccupied condition.

Temperature Control:

- We will not go through all the detailed transactions for this case. It is only different because the application initiates the operation, and hence sends periodically authorised requests to the temperature sensor. Based on the authorisation token, the sensor controller can gauge whether that operation is authorised. If it is, the temperature sensor sends back the relevant reading, duly authenticated.
- If the temperature is within range, nothing is done. If not, the relevant authorised new temperature is sent to the HVAC. As usual, the address of the HVAC can be obtained from the Handle Server.

Both the Handle client and the application can, of course, cache many of the values indicated above. Hence the number of Handle accesses can be greatly reduced. Moreover, it is an application decision which of the above operations really need authentication and/or authorisation. When they are required, it is clearly a straight forward operation – given this infrastructure.

There must be, of course, a set-up phase, which is associated with the Building Manager/Supervisor and Inventory of a building structure, i.e. the *model* of the building. One could use this to derive a form of structured identifiers that describe the model.

In this Use Case, the set-up phase consists of associating the auto-configured IPv6

addresses of each device controller to device identifiers, assigning the appropriately structured device identifier and authentication token to each device, and setting the authorisation privileges associated with each security token. The Handle system has fine-grained control of the permission for *create, delete, read* and *modify* both of Handles and of their attributes. Thus in this Use Case, the IPv6 addresses may come from Buildings Manager's model of the building or a completely different stakeholder's model like a Fire Department. This will probably result in different *logical views* of the physical devices (IPv6 addresses, Handles and privileges for the same device) but both Handle and IPv6 can cope with the different views described in software. The Authentication tokens may be intrinsic to the device. The security tokens may be set by one stakeholder. The privileges associated with that token, and with the authentication of a requestor, may be the function of another stakeholder. Thus in this Use Case, the credentials for creating and modifying Handles, the provision of security tokens, the allocation of IPv6 addresses, and privileges associated, will all be part of the set-up phase.

The benefits of the above scheme become more evident if one takes a higher-level view of the building infrastructure. For example, the Fire Department would have a different model from the Buildings Supervisor, of course. Thus their identifier would be constructed with a different 'algorithm'. The IP address associated with the identifier, may then be a less explicit attribute, which obeys the rules needed for an end-point following the network rules corresponding to the identifier's view of the network addresses. One could rely on the security of the Handle system to allow explicit algorithmic derivation of the IPv6 addresses, or even a random one, obeying the rules of the endpoint from a network perspective. We value this feature as a significant improvement over the Y2 GLoWBAL address mapping.

Thus while the actual steps in this Use Case are seemingly simple, and the devices controlled are standard IPv6-enabled IoT embedded devices with additional non-IP circuitry, the Use Case illustrates a very large range of software defined applications that can be set up securely and operated securely within a ServiceNet.

4.5 IPv6 Business Case: Mobile Phone as a sensing Tool

In this section, one of the many possible business cases that could be deployed using proposed architecture for IPv6 end point sourcing is given. Specifically, this case demonstrates how data from the phones sensors could be accessed from the Internet and used for forming the bigger picture about the environment. Smartphone has a number of embedded sensors, like GPS, microphone, speaker, camera, and light, etc. that could be used for environmental monitoring. For example, data gathered from many different sound sensors on phones could provide information about the noise level in the different parts of the city to form the noise level map. In observed cases, a mobile device can have its own sensors (embedded) or different sensors can be connected wirelessly, for example via Bluetooth, when mobile phone acts as half-gateway for sensors from devices that do not support IPv6 protocol, allowing them to be accessible via IPv6 network. The phone, while on the IPv4 mobile network, does not have a static IP address and every time when the phone is switched off and on, it obtains new IP address from the network. On the other hand, if the phone is on the WiFi, through the IPv4 network, port forwarding on the local router must be provided. Here we demonstrate the usage of IPv6 addressing system that enables every IoT device to have a unique IP address which facilitates implementation by avoiding port forwarding. Communication with mobile phones is done over CoAP protocol, while Digcovery system is used for the service discovery. Two set-ups are presented. In the first one, smartphone is used as end point that could be accessed directly through the IPv6 address.

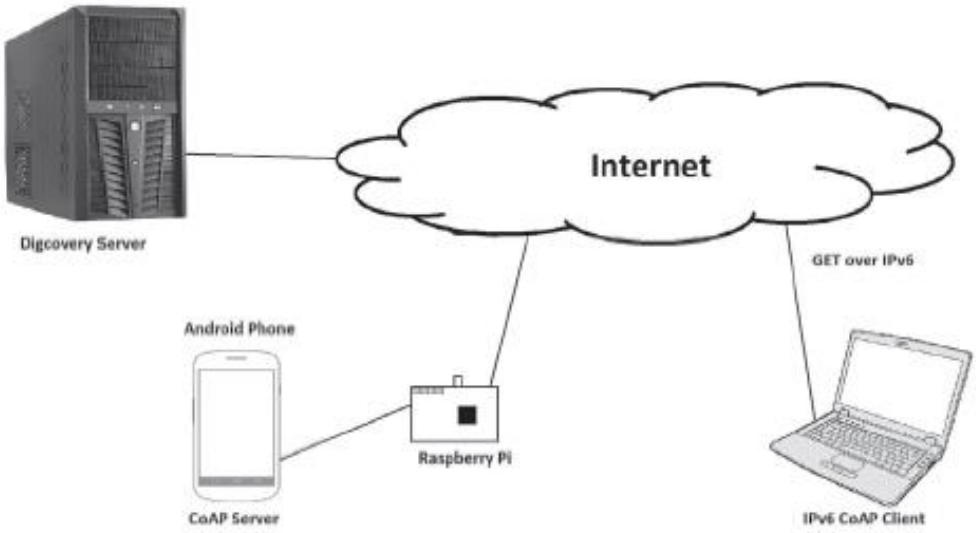

Figure 4: IPv6 communication between Laptop and CoAP Server

The REST CoAP server is used so every sensor could be accessed independently, through its own interface. CoAP is an application layer protocol designed to lower the complexity for the constrained networks but, also, to enable communication over the existing Internet infrastructure. A second set-up shows how smartphone could be used as a half-gateway for non-IP devices. In that way, access to the devices that use Bluetooth or infrared, is provided. A smartphone application is responsible for registering the phone's sensors into a Digcovery directory. Digcovery is introduced as a service discovery system on the IoT6 project. It has a CoAP interface build-in in order to enable communication with the constrained devices on the network edge. On low power devices it is too complicated or impossible to implement DNS protocol, and usage of a CoAP for discovery enables development of more distributed systems. It allows end devices to discover services that it needs.

Another device (a laptop in this case) searches the directory for the required service. After receiving the required description, a client application on the laptop communicates with the phone and collects measurements from the sensors on the phone. Android-based smartphone was used for implementation of the IPv6 CoAP server, a Raspberry Pi was acting as an IPv6 border router and finally a laptop as an IPv6 client (Fig. 4). Raspberry Pi is basically a Linux machine and therefore, it could be set to be a router for the local network enabling Internet access to the local devices. Raspberry Pi is converted to be a WiFi hot spot for the IPv6 network. In this way, IPv6 enabled devices could get the IPv6 address through the Raspberry. A full /56 prefix is assigned to a Raspberry, enabling the distribution of IPv6 connectivity to an entire network. A DHCP server is built on the Raspberry which assigns unique IPv6 address to every device that tries to connect with it. A static IPv6 address, accessible from the web is assigned to the Raspberry Pi.

In the second set-up case, shown in Fig. 5, access and communication to the external device connected via Bluetooth with the phone is presented. In this set-up, mobile phone acts as a half-gateway for sensors from devices that do not support IPv6 protocol or, like in this case, do not have IP stack at all. These devices are connected to the phone via Bluetooth, Infrared, etc. Since IoT means connected devices via the Internet, it is crucial to show how these devices could have an Internet access over the IPv6 network. The role of half-gateway is to communicate through IPv6 but still to be able to connect to a device via Bluetooth or Infrared. The mobile phone performs registration of these devices in Digcovery thus allowing their discovery and obtaining measurements. In the full

gateway implementation additionally protocol adaptations, security and privacy aspects should be supported.

In this setup, an Android phone, with CoAP Server implemented, is used as IPv6 half-gateway for the Bluetooth enabled device MindWave.

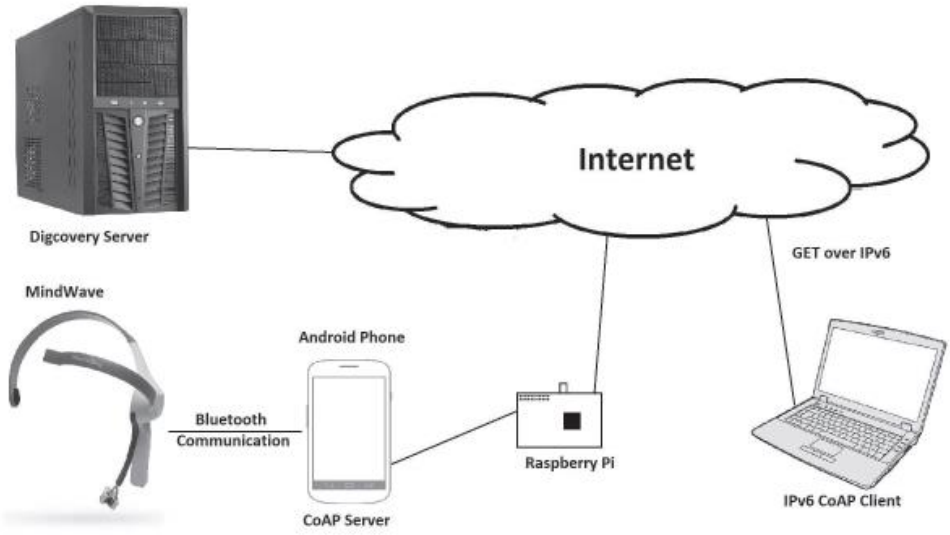

Figure 5: IPv6 communication between Laptop and MindWave device.

The MindWave device is able to read brain wave activity and to send raw measurements to the smartphone. As in the first test case, Raspberry Pi is set as the Border Router for IPv6. A connection between the MindWave and the smartphone is established using Bluetooth. An application installed on the phone communicates over the IPv6 network, reads and process the EEG (Electro Encephalograph) data from the MindWave and interpret it as the level of attention and meditation. The same application has a CoAP server that waits for the request from the Internet.

With the presented business case, we have demonstrated how sensors on mobile phones could be used for environmental monitoring, and how non-IP devices could be connected in the IPv6 network. Since existing protocols on application layer that operates in request-response model are not a good match for low-power, resource-constrained devices, CoAP protocol is used.

CoAP is a lightweight application protocol based on UDP that supports multicast requests, caching and REST Web services between the end-points, and is seen as a future protocol for IoT. Digcovery is a global discovery platform and is used for service discovery. This platform is used to locate the different domains and the wide deployed directories with the different resources. Raspberry Pi is acting as an IPv6 border router for the local network enabling Internet access to the local devices. It is converted to be a WiFi hot spot for the IPv6 network. In this way, IPv6 enabled devices could get the IPv6 address through the Raspberry. As mentioned above, Raspberry Pi is operating under the Linux OS but it is also an embedded device with digital GPIO's (General Purpose Input/Output), that provides many opportunities. Raspberry Pi could have any kind of server build-in (CoAP or HTTP) and access to GPIO's that gives the chance to control any device connected to the Raspberry. In that way, many of home and office devices could be controlled from the browser, desktop application or even smartphone application.

5 Standardisation Efforts

Following the recommendations of the reviewers from the second review to take a strong step in getting the findings and results out to industry and major stakeholders, this section describes the new initiatives undertaken in Y3 by the IoT6 project partners for the standardisation and awareness creation during and beyond the life-time of the project.

The IoT6 project partners did not spare any effort to make coalitions and partnerships and took convincing stances and positive energy and attitude to win STD influencers and key industry advocates to endorse the mature results of the project and spread them for adoption.

The IoT6 project has been very fortunate in participating directly and leading some initiatives in the strategic standardisation bodies such ETSI, the IETF, ITU and IEEE ComSoc and standards influencing projects such as the IoT Forum and IERC Cluster. The embedding of ETSI in the IoT6 in the Industry Advisory board with one of its leaders proved to be of strategic value which led to credible and influencing recommendations and acceptance of the IoT6 results.

The creation of the ETSI IP6 ISG and the IEEE ComSoc IoT/5G/SDN-NFV as well as the IoT Forum and IERC will contribute to the dissemination of best practices beyond the project lifetime with a strategic sustainability vision also beyond IoT including pre-standardisation efforts of IoT or MTC for 5G networks.

Some achievements with lasting impacts in this area are:

- Leading the adoption of IPv6 as key communication protocol;
- Winning ETSI's support to lead the ETSI IP6 ISG;
- Leading the IoT Forum;
- Leading the IEEE ComSoc IoT subcommittee;
- Contributing to the IoT Book for the 4th time;
- Leading the IoT Week program;
- Chairing many conferences and invited as speakers in standardization events.

6 Overall Impact

"By the year 2020 there will be one billion computers, 5 billion users of mobile communication systems, ten billion appliances, one hundred billion sensors, and one billion billion electronic tags, most of them Internet-enabled. Getting it right means a huge economic potential. Getting it wrong would be catastrophic."

Viviane Reding, European Commissioner

This is to summarise how IoT6 addressed the expected impacts listed in the call:

Impact 1: IoT6 has opened a new range of Internet enabled services based on truly Inter-connected physical and virtual objects, person to object and object to object communication as well as their integration with enterprise business processes.

IoT6 has truly paved the way to a new range of Internet enabled services and their integration with enterprise business processes by enabling integration of cloud computing, heterogeneous devices, mobile phones networks and STIS. IoT6 has defined **open architecture** leveraging the capacity of IPv6 to provide ubiquitous access and seamless communication among a large population of mobile and networked smart objects located in diverse geographical locations thus enabling a cost effective **integration and interoperability of heterogeneous smart things** and systems. I**ntegration of the Internet of Things with cloud computing** will be enabled through software as a service (SaaS), such as enterprise **business process management tools**. A **multi-protocol translation Web service** providing interoperability among heterogeneous smart things and systems using different communication protocols as well as **IPv6 proxy services for legacy devices t**o ease their integration into the future Internet will be developed. IoT6 will provide **STIS-IPv6 integration**, thus enabling a global addressing scheme for STID and IPv6 as well as mechanisms for address registration and update, and sensor information exchange between IoT6 and STIS. I**ntegration with mobile networks will be achieved** through interfaces such as IMS.

The extension to the use of an IPv6-enabled Digital Object Architecture will allow secure description of the heterogeneous objects and processes that transcends the network address space. The secure linkage of identifiers with IPv6 addresses and security tokens will greatly facilitate multiple stakeholders to provide independent services to common IoT sensors and actuators. It will also ease the problems of IoT governance. It divorces the management of the Internet IPv6 address space from the Identifier space. This will allow new Stakeholders, including complete industries, to manage their identifier space without impacting the management of the Internet.

Impact 2: Novel business models based on object connectivity and supporting innovative Internet services.

IoT6 developed an **open service oriented architecture** easing the integration of different products and services through the Internet. It interconnect**s the Internet of Things with the Internet of Services** through IPv6. It paves the way to **innovative ecosystems of companies**, enabling the aggregation of complementary products and services from different companies in order to provide ad hoc solutions to the customers. It has instigated the creation of new business opportunities and revenue generating business models, stemming from the ability to structure ad-hoc platforms of heterogeneous products. These business models involve **new roles and stakeholders, such as "solution brokers"**, who will provide ad hoc combination of resources and services. In order to evaluate this approach, several business scenarios

will be evaluated within the project that will be clearly linked with Future Internet priorities areas like intelligent buildings. For instance, an **on-line maintenance management tool** will enable new maintenance services for building automation components particularly relevant for building and construction industry, including some members of the IAB.

Impact 3: Emergence and growth of new companies, in particular SMEs, offering innovative technical solutions for making everyday objects readable, recognizable, locatable, addressable and/or controllable via the Internet.

IoT6 provided a handbook targeting SMEs to support their exploitation of IoT6 outputs as well as transition to IPv6. IoT6 has directly supported two emerging SMEs:

RunMyProcess and a spin-off company of the UDG project. More generally, it facilitates in a sense the entry for new SMEs, by enabling them to integrate specific solutions with other solutions through an open framework. The focus on smart buildings paved the way to innovative technical solutions with huge business opportunities for companies who can offer flexible and secure solution to the users. The project has instead of initiating an alliance to support the development and dissemination of IoT6 architecture; it has taken leadership of IoT Forum and is the leader of the IPv6 Forum.

These channels enable direct support to the dissemination and adoption of IoT6 results by industries and SMEs.

Impact 4: Consensus by industry on the need (or not) for particular standards. More widely accepted benchmarks. Consensus by all stakeholders on the governance of the "Internet of Things" including key management aspects.

IoT6 is closely linked with major industries, international forums, standardization bodies and other research projects with a European and international perspective. IoT6 is in very good position to align and contribute to the consensus by industry and other stakeholders on the need and critical use of IPv6 for the Internet of Things, with a proposed open and decentralized service-oriented architecture.

7 Main Recommendations on IoT and IPv6

IPv6 is good for IoT and IoT is good for IPv6. There are several arguments and features that demonstrate that IPv6 is actually a key communication enabler for the future Internet of Things:

- Adoption is just a matter of time

  The Internet Protocol is a must and a requirement for any Internet connectivity. It is the addressing scheme for any data transfer on the web. The limited address capacity of its predecessor, IPv4, has made the transition to IPv6 unavoidable. Google's figures are revealing an IPv6 adoption rate following an exponential curve, doubling every 6 months.

- Scalability

  IPv6 offers a highly scalable address scheme. The present scheme of Internet Governance provides at most 2×10^{19} unique, globally routable, addresses. This is many orders of magnitude more than the 2×10^{9} that is possible with IPv4 and the 10^{13} that is the largest estimate of IoT devices that will be used this century. It is quite sufficient to address the needs of any present and future communicating device still allowing it to have many addresses.

- Solving the NAT barrier

  Due to the limits of the IPv4 address space, the current Internet had to adopt a stopgap solution to face its unplanned expansion: the Network Address Translation (NAT). It enables several users and devices to share the same public IP address. This solution is working but with two main trades-off:

  The NAT users are borrowing and sharing IP addresses with others. While this technique allows single stakeholders to mount large applications, it becomes completely unmanageable if the same end-points are to be used by many different stakeholders; this would occur in an IoT deployment where the same sensors are to be used by multiple, independent, stakeholders. Secondly the mechanism cannot be used to access specific end-points from the Internet.

- Multi-Stakeholder Support

  IPv6 provides for end devices to have multiple addresses and an even more distributed routing mechanism than the IPv4 Internet. This allows different stakeholders to assign IoT end-device addresses that are consistent with their own application and network practices. Thus multiple stakeholders can deploy their own applications, sharing a common sensor/actuation infrastructure, without impacting the technical operation or governance of the Internet.

- IPv6 Features

  Many features have been built into the basic IPv6 specifications that are very useful both for the operation and the deployment of IoT. Besides the features already mentioned, these include multicast, anycast, mobility support, auto-configuration and address scope.

- Over the last decade, many new higher level protocols have been developed that are both useful for IoT and are well-suited to devices with constrained resources. Examples are 6LowPAN (wireless nets), COAP (transport with web services) and DTLS (secured datagrams). Indeed there is a whole REST environment targeted at constrained devices.

- Tiny operating systems and network stacks

  IPv6 application to the Internet of Things has been researched for many years.

The research community has developed several operating systems like TinyOS and Contiki that are relatively small and support the above protocol suites and environments. While the main IPv6 is very rich in possible features, these reduced environments have often restricted carefully the features available in order to meet IoT needs while reducing the size of the underlying system and leaving more space for applications. For example, a basic Contiki system takes less than 20KByte, and even one supporting a full IPv6 stack and the other high-level protocols including DTLS can probably fit into 70 Kbytes.

- Increased hardware support

  The operating system and network stack (with security) could be made much more compact by providing more hardware support in the chipset (or a co-processor). However, such initiatives would detract from the efficient porting of the system to other chipsets. It would be desirable to make such upgrades for large deployments in commercial environments.

- Mapping of physical systems onto IPv6 address and Privacy extension

  We have shown it is possible to map many features of the physical IoT devices onto IPv6 addresses. This can ease large-scale deployments though at the cost of revealing to anyone interested architectural features of the IoT devices because of the transparency of the Domain Name Service entries.

  In contrast, IPv6 provides for privacy by automatically randomising the suffix of the IPv6 address to hide the MAC address or any serial number used as identifier when connecting to the Internet. This feature is made available on all operating systems automatically.

  Of course, these two techniques have contradictory aims and effects; their relevance are determined by the needs of the IoT application.

- Use of Identifiers and improved functionality

  We have shown that by the use of Identifiers in conjunction with IPv6, one can take advantages of IPv6 features without their drawbacks. For example, with systems like Handle the structure can mirror the topology of a deployment, while the security features of the identifier system precludes unauthorized access to this information. At the same time, IPv6 addresses can be attributes of the Handle Identifiers, but use the privacy enhancements at the same time.

- Enabling the extension of the Internet to the Web of Things

  Thanks to its large address space, IPv6 enables the extension of the Internet to any device and service. Experiments have demonstrated the successful use of IPv6 addresses to large-scale deployments of sensors in smart buildings, smart cities and even with cattle. Moreover, the CoAP protocol enables the constrained devices to behave as Web services easily accessible and fully compliant with REST architecture.

- Mobility

  IPv6 provides strong features and solutions to support mobility of end-nodes, as well as mobility of the routing nodes of the network. The project has also achieved some interesting results on including Mobile IP in the Contiki stack.

- Address auto-configuration

  IPv6 provides an address self-configuration mechanism (Stateless mechanism). The nodes can define their addresses in very autonomous manner. This enables drastic reduction of IoT configuration effort and deployment cost. With an Identifier-based system like Handle, this technique can be combined with

automated procedures to derive authentication tokens from the device, and have access control features added.

- Fully Internet compliant Gateways

  IPv6 Gateways can be fully Internet compliant. In other words, it is possible to build a proprietary network of smart things or to interconnect one's own smart things with the rest of the World via a gateway that is fully compliant with IP requirements towards the Internet.

- Standardisation

  Some of the IoT6 developments like GLowBALIP and the Identifier system would benefit hugely if their attributes were standardized in this context much more rigidly for IoT. EC initiatives should support directly such standardization possibly in a Support Action.

- Dissemination

  Much detailed dissemination has been achieved in IoT6. However, applicability to new applications by new entities would require even more dissemination. Again this activity could be included in a further Support Action.

8 Conclusions

The IoT6 project is very pleased to state that it has achieved its very well outlined objectives, work plan and overall impact right from the outset to the end with very impactful sustainability initiatives. The project has justified all the statements below in its Deliverables, and demonstrated specific subsets in its applications and demonstrations.

The IoT6 project has:

- Researched and exploited the rich features of IPv6 and related standards (6LoWPAN, CORE, CoAP, DTLS etc.) to support the future Internet of Things by developing a layer enabling IPv6 features (such as discovery, self-configuration, security, scalability, multiple addresses for an object, and ubiquitous access, etc.) to be exploited by the service layer.
- Researched, designed and developed an open and distributed IPv6-based Service-Oriented Architecture enabling interoperability, mobility, cloud computing and intelligence distribution among heterogeneous smart things components, applications and services.
- Used this IPv6-based architecture to develop innovative forms of interactions for the Internet of Things with:
    a) Heterogeneous devices using different communication protocols (including legacy devices), by exploring innovative schemes for achieving multi-protocol integration and interoperability.
    b) Integration with Mobile telephone networks to provide ubiquitous access and seamless communication among a large population of mobile and networked smart objects located in diverse geographical locations, with solutions such as IP Multimedia Subsystem (IMS) and the Mobile Internet Protocols.
    c) Cloud computing applications and services (Software as a Service), including business process management tools.
    d) Smart Thing Information Service (STIS), exploring STID-IPv6 interactions and possible adaptation and extension of STIS to any IPv6 device.
    e) Information and intelligence distribution with “Distributed resource repositories” (for look-up and discovery services) and “Smart routing”.

The above features can be extended for increased IoT governance, deployability, security, flexibility and scalability by suitable integration with a system like Handle. The detailed justification is given in deliverable D1.4 (Final version of IoT6 architecture & SOA specifications).

References

[1] Internet of Things Architecture, IoT-A: Final Architectural Reference Model for the IoT, http://www.iot-a.eu/public/public-documents/d1.5/view