Enabling the business-based
Internet of Things and Services

(FP7 257852)

D2.2.1 Technology Watch Report

Published by the ebbits Consortium

Dissemination Level: Public

Project co-funded by the European Commission within the 7th Framework Programme
Objective ICT-2009.1.3: Internet of Things and Enterprise environments

Document control page

Document file: D2_2_1_Technology_Watch_Report_M6_v10.doc
Document version: 1.0
Document owner: CNet

Work package: WP2 – Requirements engineering and validation
Task: T2.2 – Technology Watch
Deliverable type: R

Document status:
☒ approved by the document owner for internal review
☐ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	Matts Ahlsén (CNet)	2010-10-29	TOC
0.15	Matts Ahlsén, Peeter Kool (CNet)	2010-12-12	New structure
0.2	Ferry Pramudianto (FIT)	2011-01-13	Chapter 5 Energy-aware systems
0.3	Mirko Franceschinis (ISMB), Gonzalo Alcaraz (ISMB), Claudio Pastrone (ISMB)	2011-01-31	Wireless/mobile Sensors and actuator networks. Multiradio technologies. P2P Networks
0.5	Peter Rosengren	2011-02-15	Restructuring of document
0.6	Peter Rosengren	2011-02-17	Added Related projects section
0.7	Peter Rosengren, Matts Ahlsén, Peeter Kool	2011-02-23	Added Operating Systems sub section and extended service discovery
0.8	Ferry Pramudianto (FIT)	2011-02-24	Edited energy section plus added security
0.85	Gonzalo Alcaraz (ISMB), Claudio Pastrone (ISMB)	2011-02-24	Subsection 5.2.1 ZigBee Smart Energy Profiles inserted
0.87	Sten Ahlin, CNet	2011-02-24	Added M2M description to background
0.90	Peter Rosengren, CNet	2011-02-24	Added Executive Summary
0.93	Peter Kostelnik (TUK)	23.2.2011	Section on semantic web languages and recommendations for technology selection.
0.95	Karol Furdik (IS), Marek Paralic (IS)	25.2.2011	Sections on triple stores and publisher subscriber architectures.
0.96	Riccardo Tomasi, ISMB	25.2.2011	Added subsection on TinyOS
0.98	Peter Rosengren, CNet	2011-02-25	Added descriptions of personal health monitoring and energy efficiency to application chapter
0.99	Peter Rosengren, Cnet	2011-02-26	Updated with comments from peer reviewers
1.0		2011-02-28	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
Andreas Zimmerman, FIT	2011-02-26	
Jesper Thestrup, InJet	2011-02-26	

Index:

1. Executive Summary

This deliverable summarises the technology watch activities that have been carried out in subtask 2.3.3 in WP2. After introducing the scope and background for the deliverable it goes into the three important technical cornerstones of ebbits – Internet of Things Architecture, Semantic Data Management and Energy Aware Systems.

The Internet of Things and Services is the current vision for an Internet encompassing any IT artefact, information source or service. The ebbits project aims at developing an interoperability platform for a real world populated Internet of Things domain. The ebbits platform will feature a Service oriented Architecture (SoA) based on open protocols and middleware, effectively transforming every subsystem or device into a web service with semantic resolution.

The document starts by describing some emerging application areas where IoTS is being used today including M2M, personal health monitoring and energy efficiency. Internet of Things and Services is a new and research intensive area which brings together many different disciplines, chapter 4 therefore summarises the most important aspects being tackled in the research world including middleware aspects, service oriented architecture, service composition, radio technologies and operating systems for resource constrained devices.

Semantic Data Management describes aspects related to semantic encoding of events and data, the use of ontologies and of this relates to integration with business systems.

The third important aspect of ebbits is to improve energy efficiency in manufacturing processes. The chapter Energy Aware Systems surveys some important technical developments. Finally, in chapter 6 we survey on-going research project which are related to ebbits and which we intend to follow during the lifetime of ebbits in order to collaborate and exchange results.

2. Introduction

2.1 Purpose, context and scope

The purpose of this deliverable is to provide a first technology watch summary for the ebbits project after 6 months in the project.

The work is performed in the context of Task T2.3 – "Evolutionary requirements refinement". WP2 will manage and undertake the work in carrying out the iterative engineering of requirements, which special focus on the engineering process of initial requirements and re-engineering after the end of each iteration cycle. The aim of this work package is thus to maintain a continuous discovery and analysis of user centric requirements, needs and prospects, to be used in the design, development, implementation and validation of platform and services.

An important subtask in this workpackage is to keep track of development and trends in this area, and document it in a Technology Watch. This deliverable focuses on the general technical aspects relevant for the ebbits project and will not discuss details regarding application specific equipment in the two use cases of ebbits, i.e. manufacturing and food traceability.

2.2 Background

The *Internet of Things and Services* (IoTS) *is* the current vision for an Internet encompassing any IT artefact, information source or service. The ebbits project "Enabling business-based Internet of Things and Services" aims at developing an interoperability platform for a real world populated IoTS domain.

The *ebbits project* will develop the architecture, technologies and processes, which allow businesses to semantically integrate the IoTS into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the IoTS and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information from tags, sensors, and other devices and performing actions on the real-world. The ebbits platform will feature a event-driven Service oriented Architecture (SoA) based on open protocols and middleware, effectively transforming every subsystem or device into a web service with semantic resolution.

.

3. Emerging Application Areas for IoTS

There are many application areas which will benefit from IoTS solutions. Although in ebbits we are focused on manufacturing and food traceability, we will here briefly summarise trends and early adoptions of IoTS-based or similar solutions. Three of the most important application areas at the moment are:

- Machine-to-Machine, M2M
- Personal Health Monitoring and Ambient Assisted Living
- Energy Efficiency

3.1 M2M

Machine-to-Machine (M2M) refers to technologies that allow both wireless and wired systems to communicate with other devices of the same ability. M2M uses a *device* (such as a sensor or meter) to capture an *event* (such as temperature, inventory level, etc.), which is relayed through a *network* (wireless, wired or hybrid) to an *application*, that translates the captured event into *meaningful information* (for example, items need to be restocked). This is accomplished through the use of *telemetry*, the language machines use when in communication with each other. Such communication was originally accomplished by having a remote network of machines relay information back to a central hub for analysis, which would then be rerouted into a system like a personal computer.

However, modern M2M communication has expanded beyond a one-to-one connection and changed into a system of networks that transmits data to personal appliances. The expansion of wireless networks across the world has made it far easier for M2M communication to take place and has lessened the amount of power and time necessary for information to be communicated between machines. These networks also allow an array of new business opportunities and connections between consumers and producers in terms of the products being.

Essentially, it is the exchange of data between a remote machine and a back-end IT infrastructure. The transfer of data can be two-way:

- Uplink to collect product and usage information
- Downlink to send instructions or software updates, or to remotely monitor equipment.

In the past, the high cost of deploying M2M technology made it the exclusive domain of large organizations that could afford to build and maintain their own dedicated data networks. Today, the widespread adoption of mobile technology has made wireless M2M technology available to manufacturers all over the world.

As shown above, wireless M2M applications include connectivity-enabled devices that use a cellular data link to communicate with the computer server. A database to store collected data and a software application that allows the data to be analyzed, reported, and acted upon are also key components of a successful end-to-end solution.

While many M2M deployments will make use of short-range or proprietary radio links, mobile cellular-based M2M solutions will be preferred where mobility is required, or where high data volumes or data transfer rates are involved. Cellular-based M2M can also provide easier installation and provisioning, especially for short-term deployments.

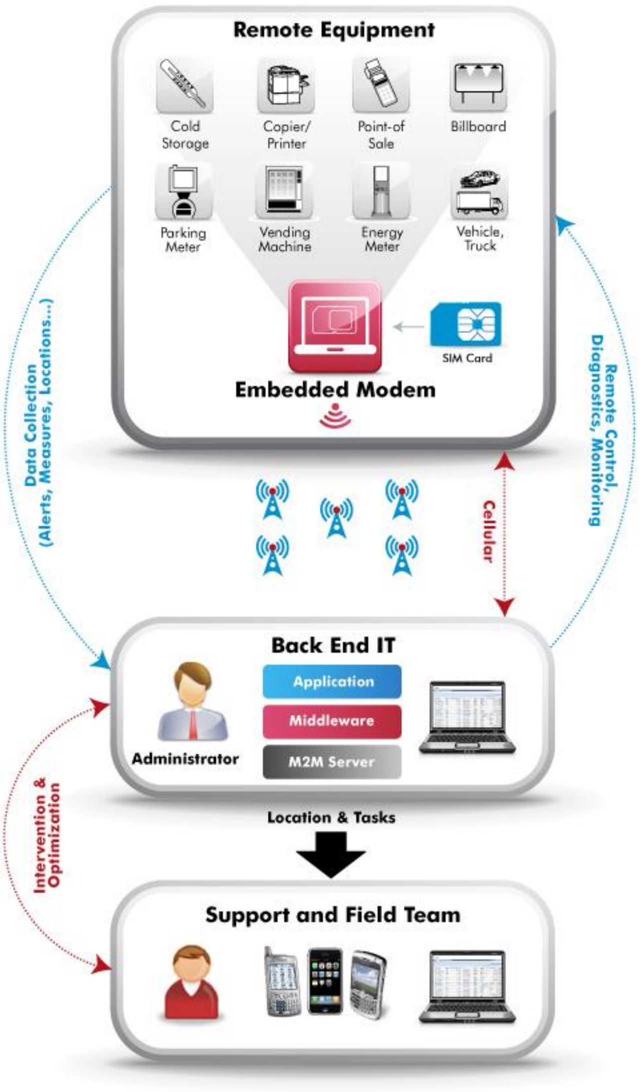

Figure 1: A typical M2M setup (from www.jasperwireless.com)

Telecoms networks will need to be optimised to cater for these new 'subscribers', who may have very different behaviour from current customers. Standardization is required in order to deliver cost-effective M2M solutions, and allow this market to take off.

It's easy to see why machine-to-machine communications have so many applications. With better sensors, wireless networks and increased computing capability, deploying an M2M makes sense for many sectors.

Utility companies, for instance, use M2M communications, both in harvesting energy products, such as oil and gas, and in billing customers. In the field, remote sensors can detect important parameters at an oil drill site. The sensors can send information wirelessly to a computer with specific details about pressure, flow rates and temperatures or even fuel levels in on-site equipment. The computer can automatically adjust on-site equipment to maximize efficiency.

Traffic control is another dynamic environment that can benefit from M2M communications. In a typical system, sensors monitor variables such as traffic volume and speed. The sensors send this information to computers using specialized software that controls traffic-control devices, like lights and variable informational signs. Using the incoming data, the software manipulates the traffic control devices to maximize traffic flow. Researchers are studying ways to create M2M networks that monitor the status of infrastructure, such as bridges and highways.

The Industry in general can increase efficiencies by 10% through smart metering and operations. Traffic accidents can be reduced by 15% trough intelligent traffic systems. The peak load of electricity grids can be lowered by 20% by using smart grids. Intensive care unit time resource can be saved by 20% by the usage of smart M2M sensor systems for remote care of patients. Freight companies can double the usage level of lorries and simultaneously save fuel and reduce negative effects on the environment through smart logistics. The potential is enormous.

Figure 2 Machine-to-machine communications can be used to monitor traffic in real time, like at this Los Angeles traffic center.

Examples of M2M vendors are Jasper[1] which delivers a M2M for telecom operators and device manufacturers to implement M2M services and iMetrik[2]. ETSI is working on standardizing M2M[3].

3.2 Personal Health and Ambient Assisted Living

The healthcare industry is undergoing fundamental changes; one of the more important trends is self-management of chronic diseases such as diabetes.

The remote monitoring area can be divided into telehealth and telecare. In telehealth, numerous medical devices are now available which allows personal monitoring of vital signs like heart rate, blood pressure, glucose level, weight etc. IoTS fits in well in these scenarios, where data is captured remotely by wireless devices and needs to be sent securely to some central service or hospital.

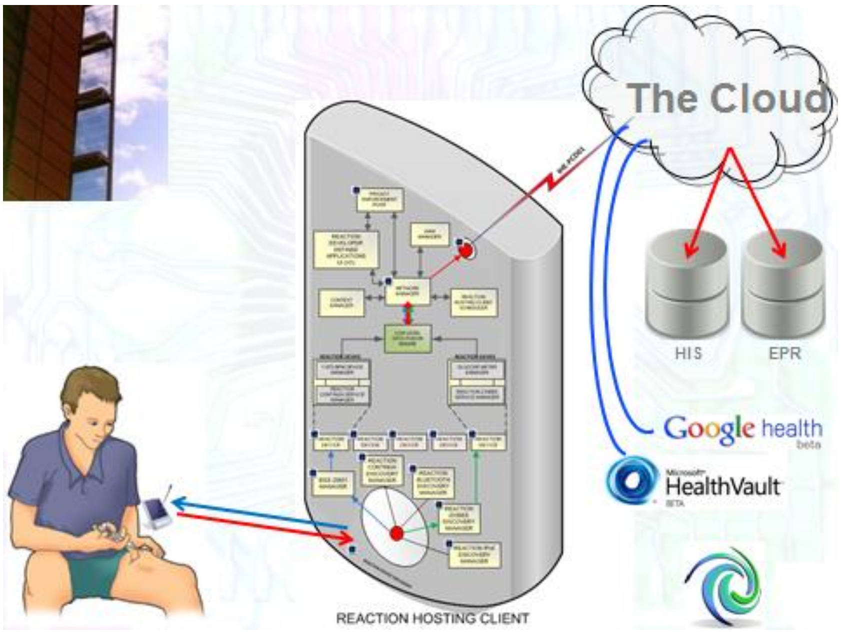

Figure 3 New solutions for personal health monitoring uses IoTS to connect personal medical devices with cloud-based services and hospital systems for efficient self-management of chronic diseases like diabetes.

[1] *www.jasperwireless.com*
[2] *www.imetrikm2m.com*
[3] *www.etsi.org*

Body-worn sensors, and body-area networks can be implemented using IoTS to achieve closed-loop feedback systems for instance for continuous glucose monitoring, see figure.

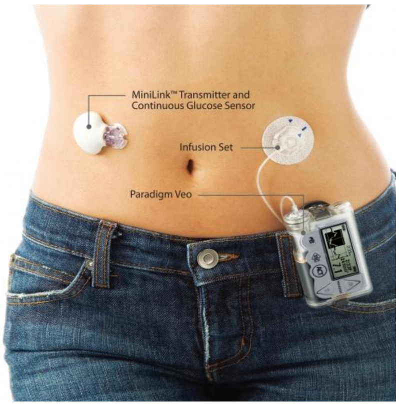

Figure 4 Body-worn sensors and body area networks is examples where IoTS technologies can contribute for more efficient management of chronic dieseases

A difference between M2M applications and personal health monitoring applications is that in personal health applications the devices are either wired or often based on short range radio technology like Bluetooth and ZigBee, as opposed to the GPRS/GSM-based devices which are the focus of M2M. This means that "the last mile" is usually not Internet-based communication.

TeleCare usually refers to support elderly people to live independently at home by using devices and sensors such as fall detectors, movement sensor, and activity hubs. By using IoTS technologies, telecare solutions can be efficiently implemented using open standard technologies, as opposed to previous vendor specific hardware and software solutions, that are incompatible with each other. The Continua[4] Alliance is growing alliance of device manufacturers, software companies, system integrator and other actors which are working towards establishing standards for interoperability.

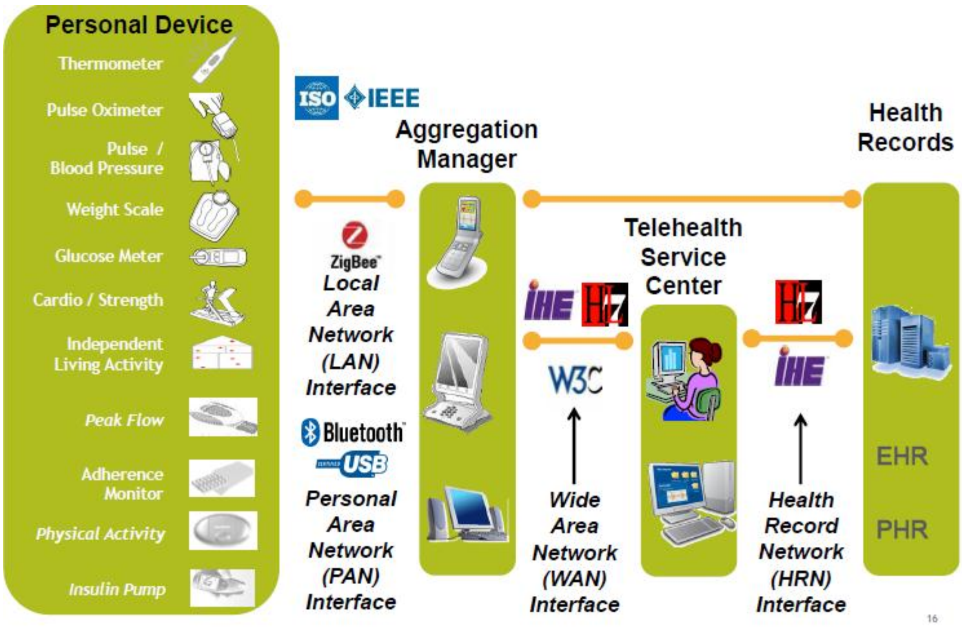

Figure 5: The Continua Alliance strives towards interoperability of medical devices across various networks.

Another important alliance is ANT+ 5 which is focussing on sensors and devices for wellbeing such as heart rate monitors, GPS watches, and training equipment.

3.3 Energy Efficiency

IoTS techniques are often discussed in the context of energy efficiency. Mainly there are two roles IoTS can play. The first one is in the production and distribution of energy, often referred to as

[4] www.continuaalliance.org
[5] www.thisisant.com

Smart Grid. This requires sensors and meters that connect to the Internet and communicates data about consumption back to the producer and energy provider.

The second is on the consumer side, where IoTS is an enabling technology to achieve energy efficiency in smart homes by controlling appliances and other devices at home.

3.3.1 Smart Grids

In the context of electricity provisioning systems, the currently existing reference scenario, inherited by the past, is rather static. It conceives few large central power stations connected to high voltage transmission systems which supply power to local distribution systems. As a result the power flow is unidirectional and consistent with the paradigm one-to-many: from few power generators to a large number of users.

European Technology Platform Smart Grids6, a platform group invented by the European Commission, envisions that a part of the electricity demand today that are completely satisfied by large central power plants will be provided by distributed generation systems supported by the increase of renewable energy sources. However, the most significant change will concern the philosophical approach to the problem and will be based on the definition of an interactive grid. Each node in the grid will have both an active and a passive role, thus being an electricity consumer and producer at the same time. New regulatory rules will be needed to govern such a more versatile and dynamic scenario. The basis for a Europe-wide market for energy, with more flexible and competitive tariffs, will emerge. New and alternative energy storage methodologies will be considered as well. For instance, BEVs (Battery Electrical Vehicles) could act as mobile energy storage devices and be fully integrated in the electrical distribution grids. This will open new perspectives towards the integration and interoperability of different services, e.g. energy distribution, urban transport, building management and so on, with the final aim to boost competitiveness, job creation, social cohesion and environmental sustainability (i.e. energy efficiency). In such innovative eco-system, the ICT will act as the main enabler.

Power management in Europe has to satisfy a whole new set of requirements in order to leverage the existing potential to reduce energy consumption and greenhouse gas emissions. The idea behind smart metering solutions is to increase the transparency of the power data to identify potential savings. Smart metering identifies consumption in more detail and communicates that information back to the users for monitoring, a technique that is facilitated through smart grids. In contrast to traditional power grids, smart grids support real-time bidirectional communication. Thus, such grids strive to meet the requirements of distributed energy generation and storage systems. They have to be flexible, reliable, accessible and economic to e.g. support real-time optimization of energy flows at local and global level. In such environments domestic and small commercial customers will be able to actively participate in power system markets and provide services to other power system participants. Existing solutions distinguish themselves in the quantity and granularity of the ascertained electrical appliances, reliability, asset cost and data integrity.

3.3.2 Smart Homes and Appliance Control

An important usage of IoTS to achieve energy efficiency is in consumer homes to allow control and monitoring of appliances and their energy consumption. So called smart plugs allows measuring of energy consumption at device level, as opposed to smart meters that measures on the house level.

Using IoTS-based middleware it will make it possible to connect the household with the community and the energy provider to exchange consumption data. A new generation of smart home applications will emerge.

Figure 6 IoT-enabled smart plugs pave the way for energy efficient control in smart homes

[6] http://www.smartgrids.eu

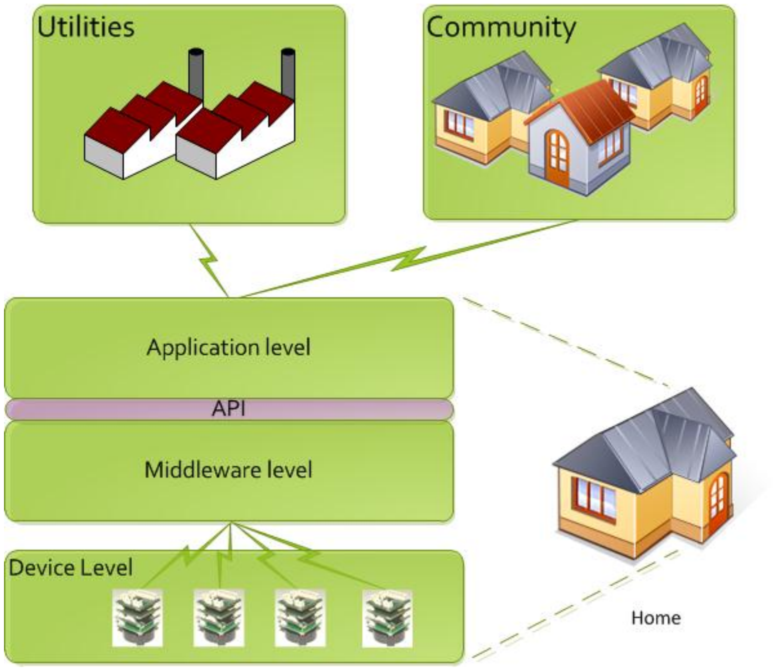

Figure 7 IoT-based middleware is the key to connecting homes in communities and with energy providers, to create the next generation energy applications

4. IoTS Architectural Aspects

The Internet of Things and Services is the current vision for an Internet encompassing any IT artefact, information source or service. The ebbits project is about "Enabling business-based Internet of Things and Services". It aims at developing an interoperability platform for a real world populated Internet of Things domain. The ebbits platform will feature a Service oriented Architecture (SoA) based on open protocols and middleware, effectively transforming every subsystem or device into a web service with semantic resolution.

The ebbits platform will support interoperable business applications with context-aware processing of data separated in time and space, information and real-world events, people and workflows (operator and maintenance crews), optimisation using high level business rules (energy and cost performance criteria), end-to-end business processes (traceability, life-cycle management), or comprehensive consumer demands (product authentication, trustworthy information, and knowledge sharing).

In the following sections we will review the areas of IoTS developments which are of most interest to ebbits.

4.1 RFID/WSN Middleware

IoTS requires flexible configuration and deployment of algorithms for collection, and filtering information streams stemming from the internet - connected objects, while at the same time generating and processing important business/applications events. Such functionalities are generally available within state - of - the art middleware infrastructures. The first generation applications and approaches to Internet of Things explored the use of RFID (Radio Frequency Identification) as a means of identifying and track physical objects using the Internet. Several RFID middleware frameworks are nowadays providing functionality for RFID data collection, filtering, event generation, as well as translation of tag streams into business semantics. These frameworks have been developed as part of both research initiatives [Prabhu06] and vendor products.

Furthermore, several research initiatives have produced open - source RFID frameworks, such as AspireRfid[7], Mobitec[8], and the fosstrak[9] project, which provide royalty–free implementations of RFID middleware stacks.

Several middleware platforms have also been devised in the area of WSN (Wireless Sensor Networks). Specifically, there are platforms addressing only the level of the sensor network, whereas others also deal with devices and networks connected to the WSN. Some middleware platforms are characterised as sensor databases, but there are also publish - subscribe approaches.

Systems like Moteview and ScatterViewer are examples of WSN development and monitoring systems. Other environments like Hourglass, SenseWeb, jWebDust and GSN, provide more complete development and/or programming environments for WSN applications. Between the extremes of high flexibility and tightly coupled approaches, there are several others such as TinyDB, Hood. An in - depth review is available within [Chatzigiannakis2007]. Note that several RFID/WSN middleware platforms have been developed within the framework of EU projects, such as Aspire and Bridge.

4.2 Semantic Sensors Descriptions

Internet of Things requires that devices and sensors can be semantic described in order to exchange metadata about the properties and behaviour of each sensors/device. In ebbits these semantic structures will enable reasoning over multi-sensor and multi-source data.

[7] *http://wiki.aspire.ow2.org*
[8] *http://mobitec.ie.cuhk.edu.hk/rfid/middleware*
[9] *http://www.fosstrak.org*

During the past years we have witnessed the development of a large number of ontologies serving different purposes, applications and domains. For example, [Compton2009] defines a sensor ontology for describing and reasoning about sensors, observations and scientific models, while also facilitating the use of sensors in workflows.

As another example, OntoSensor is a knowledge base of sensors, which can be queried via a Protégé plugin. OntoSensor includes definitions of concepts and properties adopted in part from SensorML, extensions to IEEE SUMO and references to ISO 19115 [Goodwin2006].

The SWAMO ontology has also been developed to enable dynamic, composable interoperability of sensor web products and services, while also providing autonomous agents for system - wide resource sharing, distributed decision making and other autonomic operations. The Sensei[10] project has also created an ontology towards providing a linking between observation models, procedures and complex systems.

In addition to the above ontologies, there have also been significant standardization efforts, mainly through the SWE (Sensor Web Enablement) standards of the OGC (Open Geospatial Consortium). OGC/SWE standards do not provide facilities for abstraction, categorization, and reasoning consistent with standard technologies.

In order to overcome this gap, the W3C SSN-XG (Semantic Sensor Networks Incubator Group) has produced a generic ontology to describe sensors, their environment and the measurements they make. The ontology provides definitions for the structure of sensors and observations, leaving the details of the observed domain unspecified. This allows abstract representations of real world entities, which are not observed directly but through their observable Qualities.

A further and deeper discussion about the use of semantic technologies in ebbits is given in chapter 5.

4.3 Service Oriented Architecture

Wireless Sensor Networks (WSNs) have already been deployed in numerous real-time applications from various domains such as home/building automation, environmental monitoring and utility metering. WSN architectures were initially platform-dependent in order to obtain optimal performance, as well as for some marketing reasons. However, the abundance of WSN applications and the heterogeneity of sensor and actuator technologies, the need for more generic solutions that can fit to several applications at a time has risen.

Service oriented approaches aim to fulfil this gap by decoupling the functionalities of the sensors and actuators from the underlying hardware details or the network infrastructure. The Hydra[11] project pioneered a service-oriented architecture approach based on web service technologies for connecting devices over the Internet. The Socrades middleware architecture enables enterprise-level applications to interact with and consume data from a wide range of networked devices, including sensors. Device abstraction is achieved by device proxies that integrate low-capacity devices to the platform and expose the offered functionalities as services on the middleware. It relies on Web Services for all communication interfaces.

Hourglass proposes a service infrastructure to publish sensor services to be used by different applications. Based on a publish-subscribe mechanism, producers (e.g., presence detection sensor) publish their services and consumers (e.g. parking place finder application) subscribe to interesting services. IrisNet suggest a sensor network at the Internet scale. It provides software components to facilitate the deployment of sensor services.

TinySOA allows programmers to access wireless sensor networks from their applications by using a simple service-oriented API via the language of their choice. Like Hydra the FP7 Sany[12] project base

[10] *http://www.sensei-project.eu*
[11] *http://www.hydramiddleware.eu*
[12] http://www.sany-ip.eu

its SOA implementation on Web services model. It is based on Open Geospatial Consortium (OGC) Sensor Web Enablement for the development of standards for geospatial and location-based services. Sensei uses a REST-based model to represent any physical or virtual entities in the real world. OSAmI uses the dynamic OSGi service platform in order to address a large diversity of co-operating software-intensive systems, including sensor/actuator based systems.

IoTS services can be categorised into two groups: those providing sensor data services (e.g. representing sensor resources as Web services), and high-level services (e.g. services that provide discovery, semantic reasoning, etc. The sensor data services typically gather data from various different resources and provide these as inputs to high-level services and applications (e.g. reasoning, integration, planning, and recommendation services).

An initiative towards standardising the modelling and provisioning of sensor data services is the OGC Sensor Web Enablement[13] (SWE) standards suite that is aimed at web accessible sensor networks and archived sensor data that can be discovered and accessed using standard protocols and APIs.

The standards consist of modeling schemas (Observation and Measurement (O&M) and SensorML) and Web Service interfaces (Sensor Alert Service, Sensor Planning Service and Sensor Observation Service) that facilitate the exchange of information through APIs. [Henson2009] provides a semantically enabled Sensor Observation Service, called SemSOS, which provides the ability to query high-level knowledge of the environment as well as low-level raw sensor data.

52North's SOS implementation is designed to provide a Servlet interface to sensor observation data stored in a database and the sensor descriptions stored in XML files. It proposes an ontology-based model for service oriented sensor data and networks consisting of three main components-ServiceProperty, LocationProperty, and PhysicalProperty. ServiceProperty explains the functionality of a service, while properties in the other two components describe contextual and physical characteristics of the sensor nodes in a wireless sensor network architecture.

High-level services seamlessly integrate the digital world and physical world resources to create context-aware applications and to support various data and event processing tasks. A web service enabled emergency medical response system using sensor resources in demonstrated in [Hashmi2005]. Priyantha *et al.* [Priyantha2008] describe an implementation that allows web service clients to use the sensors and at the same the proposed system minimizes code size and energy at the sensor nodes.

Some application scenarios with reasoning over the semantically annotated sensor data with rules are described in [Sheth2008] and [Henson2009]. The sensor data is annotated with concepts from the OWLtime domain ontology to allow querying for events within a time interval, using temporal concepts such as within, contains and overlaps. The rules allow dynamic assertion of events from the measured sensor values.

4.4 Service creation and orchestration environments

A middleware architecture approach is of importance in the IoT domain due to its role in simplifying the development of new services and integration of legacy technologies into new ones.

As have been explained in previous section middleware architectures proposed for the IoTS domain often follow the Service Oriented Architecture (SOA) approach. The challenges that need to be addressed by a SOA solution in the IoTS domain include: abstracting the devices' functionalities and communication capabilities, provision of a common set of services and a service composition environment.

There are several research projects in the service creation environment area, where the focus is on an easy composition of services and not on their testability. The Unified Service Description Language (USDL) SOA4All[14] is a platform-neutral service description language to support the implementation of Web-based services. USDL covers the business, operational and technical aspects of a service. USDL supports description of both atomic and orchestrated services.

[13] www.opengeospatial.org
[14] *http://www.soa4all.eu*

Different aspects of service description are handled by defined modules within the framework, which can interact with one another. These include service level, legal, pricing, interaction, participants, functional, service and foundation.

The Socrades middleware supports composition of IoT-level services. It implements a service implementation repository that stores all services that are available for composition of new services, orchestration of business process or deployment. The repository stores service metadata and associated content like service implementations.

Service composition is offered through BPEL extensions that that offer support for service (or hosting device) mobility. Composed services are described in BPEL, which communicates with service partners over partner links. Partner links are bound to concrete service types at design time, though the actual endpoints can be unknown. The middleware offers interfaces for deploying/undeploying and for using the composed services.

Service creation and composition is handled by a Task Plan in the Sensei project. The framework's Semantic Query Resolver (SQR) interprets a user's query for a service and translates it into a task or a combination of tasks to be executed by the framework components in order to fulfill the request. Thus, the SQR creates a task plan. A task plan may consist of a single resource operation in the simplest case. In case a single resource cannot fulfill the request, a combination of resources is determined whose interworking can lead to the desired result. In this case, the task plan consists of several atomic resources and associated operations.

The FP7 ICT m:Ciudad[15] project introduces the concept of native mobile User-Generated Service. m:Ciudad allows users to create and provide services, and share the services and contents directly from the mobile device to other mobile users in a community. A service creation tool, the Service Creation Kit [Urdiales2009], is provided to users as a mobile application. Services can be created using templates or by using simple service elements, which can be either content or functional elements. The templates include service logic and interface implementation, but the specific parameters are not included to allow customisation.

4.5 Service and Device Discovery

Service Discovery is the process used by the system when it needs to find a service which solves a particular task or clients' needs (goals). The Service Discovery process returns a list of services that can potentially fulfill these needs. Guinard *et al*. [Guinard2010] differentiate between service discovery which is end-user driven, and network discovery of services which is machine driven and occurs at the network layer. However due to the lack of semantic information described by the service description technologies commonly used today, many service discovery mechanisms rely heavily on keyword matching and are very limited in their ability to provide the users with more complex search tools.

DPWS-based solutions [Abangar2010, Spiess2009, Guinard2010] utilize the WS-Discovery specification of the DPWS stack to find a new resource as it connects to the network and dynamically retrieve metadata about it and the services it hosts. The metadata categories include location or access rights based scopes, device type and message types.

The discovery process works as follows: when a new resource joins the network, it multicasts a 'hello' message via the UDP protocol. By listening to this message, clients/gateway middleware can detect new resources and in a second step, retrieve their metadata. To locate a specific resource or a set of matching resources for a given filter, a client can send a 'resolve' message to the same multicast group and the matching resource sends back a response directly to the client.

The Hydra[11] project pioneered a 3-layered Discovery Architecture in IoTS applications. The middleware platform provides a discovery architecture that builds on UPnP technology. The

[15] *http:// www.mciudad-fp7.org*

approach implements a three layered discovery architecture that includes physical device detection, UPnP network announcement and semantic resolution of devices against a device ontology.

The Hydra model driven architecture (MDA) includes a Discovery Architecture which implements the device discovery process. This architecture is structured in three layers abstracting the discovery functions. The discovery process operates both locally and remotely, so that devices that are discovered in a local network can also be discovered in a peer network over the P2P protocol implemented by the Network Manager.

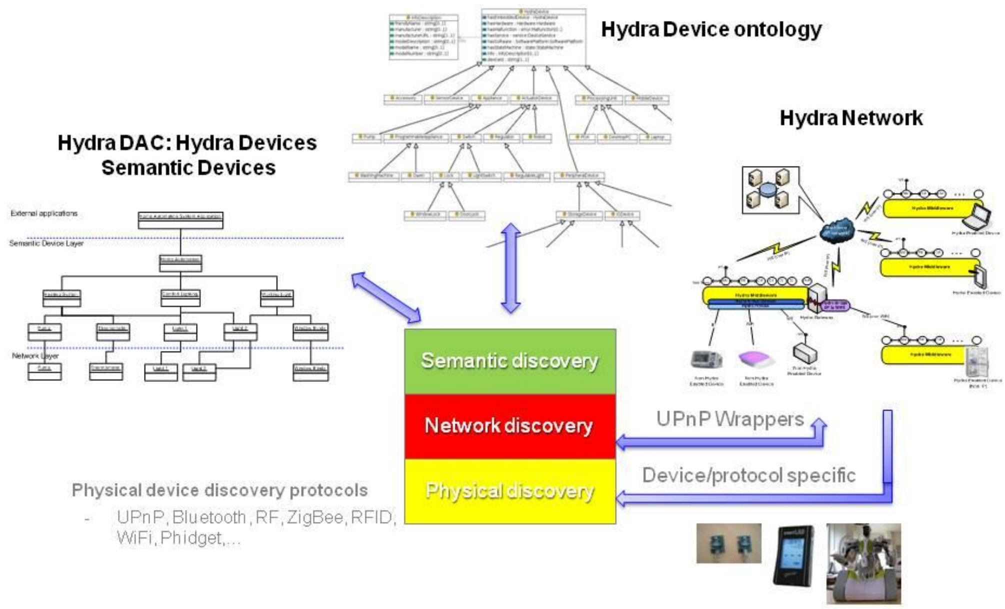

Figure 8: The 3-layered Discovery Architecture is part of the Hydra MDA.

The lowest discovery layer implements the protocol specific discovery of physical devices. This is performed by a set of specialized discovery managers listening for new devices at gateways in a local network. The second layer uses UPnP/DLNA technology to announce discovered physical devices in the local network and to peer networks.

At the top most layer the device type is resolved against the Device Ontology and is mapped to some Device type. It is then placed in the *Device Application Catalogue* (DAC). If an application subscribes to events regarding this type of device, it will be notified that the device is available and has been placed in the Device Application Catalogue.

The middleware provides: 1) Discovery mechanism, 2) Low level protocols, 3) Service execution, 4) Virtualization, and 5) security and trust policies which can directly be used by the developer of Hydra applications. The whole process of the Hydra middleware management of devices and services is reviewed on the following page.

The middleware incorporates support for self-discovery of devices. When an Iota-enabled device is introduced the middleware is able to discover and configure the device automatically. In Figure 8 we see an example of a IoT-device network. Hydra distinguishes between powerful devices are capable of running the middleware natively and smaller devices that are too constrained or closed to run the middleware. For the latter devices, proxies are used and once proxies are in place, all communication is based on the IP protocol.

The Sensei[10] project exposes the sensor services functionalities through the Resource Directory, which accepts resource publications from resource endpoints, and can be queried by users and other components over the Resource Lookup Interface (RLI). Resource directories in different Sensei domains can be peered. These are complemented by the Entity Directory which holds context information about resources. The Sensei resource discovery can be seen from three different perspectives. The first perspective is resource oriented and focuses on the resource discovery by unique Resource IDs or other parts of the resource description. Another discovery mechanism provided by Sensei resource directories is a simple string matching based resolution method using free text tags defined in Resource descriptions.

4.6 Event Management in IoTS

There exist several variants for designing publish/subscribe systems, which offer different degrees of expressiveness and different performance overhead. Topic-based publish/subscribe is rather static and primitive, but can be implemented very efficiently. On the other hand, content-based publish/subscribe is highly expressive, but requires sophisticated protocols that have higher runtime overhead. Additional expressiveness can be achieved by applying content-based filters in the context of statically configured topics, in particular types, to express constraints on properties that are not within discrete ranges (e.g., stock prices) [Eugster2003].

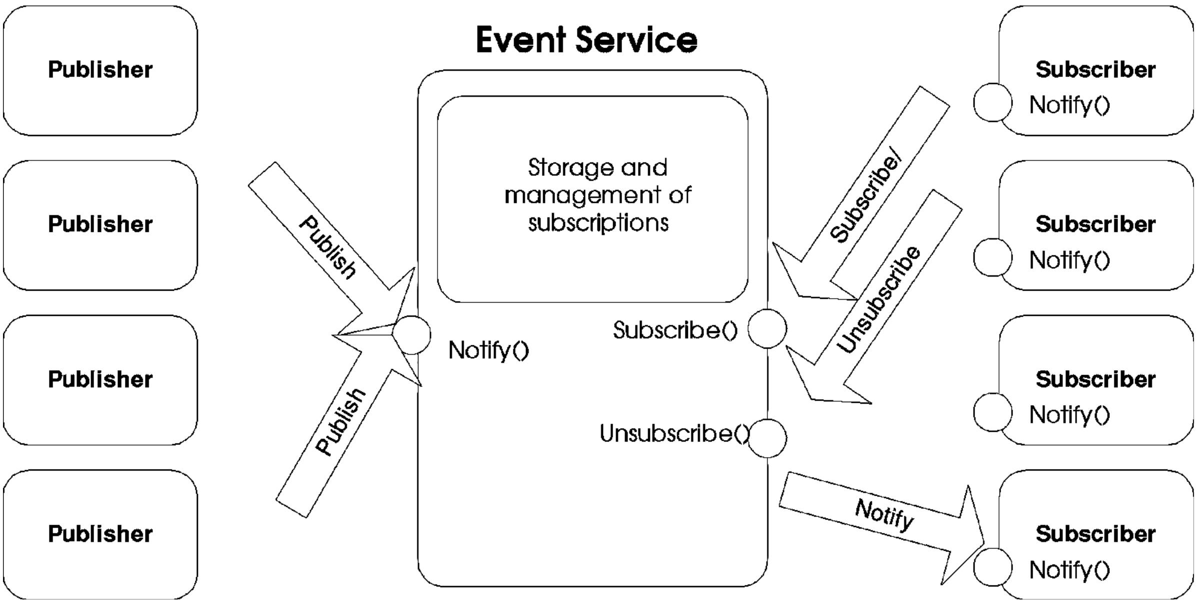

Figure 9 PublishSubscribe A simple object-based publish/subscribe system [Eugster et al, 2003]

Large variety of emerging applications benefit from the expressiveness, filtering, distributed event correlation, and complex event processing capabilities of content-based publish/subscribe systems. These applications include RSS feed filtering, stock market monitoring engines, system and network management and monitoring, algorithmic trading with complex event processing, business process management and execution, business activity monitoring, workflow management and service discovery [Jacobsen2009].

The **DSWare** system [Shuoqi2004] is an event detection middleware for wireless sensor networks. In such scenarios, message delivery with pre-specified time constraints is of paramount importance. A form of hard real-time delivery is provided, but because of the severe limitations of the devices for which DSWare is designed, all the remaining features are implemented using the least resource-consuming approach. For instance, there is no support for particular message orderings.

Padres[16], [Jacobsen2009] is an open source distributed content-based publish/subscribe system developed by the Middleware Systems Research Group at the University of Toronto. The system Padres extends merging-based routing with imperfect merging capabilities. Content-based routing is enabled in cyclic overlays. Cyclic overlays provide redundancy in routes between sources and sinks and thus produce alternative paths between them. Padres also implements other efficient load balancing and recovery algorithms to handle load imbalances and broker failures. The Padres publish/subscribe broker is based on a content-based matching engine that supports the subscription language, including atomic subscriptions, the various forms of historic subscriptions, composite subscriptions with conjunctive and disjunctive operators, the *isPresent* operator, variable

[16] *http://padres.msrg.toronto.edu/Padres*

bindings, and event correlation with different consumption policies. Padres includes a number of tools to help manage and administer a large publish/subscribe network, e.g. a monitor that allows a user to visualize and interact with brokers in real time, and a deployment tool that simplifies the provisioning of large broker networks. Padres is used in several research and development projects, e.g. in the *eQoSystem* project with IBM, it constitutes the enterprise service bus that enables the monitoring and enforcement of SLAs of composite applications and business processes in service oriented architectures, in collaborations with CA and Sun Microsystems, Padres is used to explore the event-based management of business processes and business activity monitoring, in collaborations with the Chinese Academy of Sciences, Padres is used for service selection and for resource and service discovery in computational Grids [Jacobsen2009].

Another interesting framework was created in context the Internet of Things (IoT) research effort – **MAGIC Broker 2** (MB2) developed at the Media and Graphics Interdisciplinary Centre, University of British Columbia [Blackstock2010]. MB2 middleware platform offers a simple, uniform web-based API for building IoT applications and offers developers three built-in programming abstractions: publish-subscribe event channels, persistent content and state storage, and brokerage of services via remote-procedure call. A channel is used as our namespace and conceptual container for other MB2 abstractions. It is used to name the on-line presence of things, and groups of things that comprise IoT applications. MB2 supports a state abstraction that allows clients to request the last n events, as well as read and write name/value pairs in a channel. MB2 can also broker synchronous two-way request-response interactions called services with devices registered with the platform (analogous to a CORBA ORB). MB2 services are similar to those supported by SOAP web services and Java RMI. MB2 supports storage and retrieval of content such as images, videos, text, and HTML documents within a channel in a consistent way. MB2 system was used to create a range of IoT applications involving spontaneous device interaction such as between mobile phones and public displays, and opportunistic or shared sensing and control of devices using a web-based sensor actuator network called Sense Tecnic (STS). The STS platform also includes facilities to process sensor data, effectively creating higher-level sensors. A complex event-processing engine is used to process lower-level sensor events, which are sent back into MB2 for output to higher-level derived sensor feeds that can be used by applications and visuals.

Esper[17] is an Event Stream Processing (ESP) and event correlation engine (CEP, Complex Event Processing) – i.e. it supports requirement to process events (or messages) in real-time or near real-time. Targeted to real-time Event Driven Architectures (EDA), Esper is capable of triggering custom actions written as Plain Old Java Objects (POJO) when event conditions occur among event streams. It is designed for high-volume event correlation where millions of events coming in would make it impossible to store them all to later query them using classical database architecture. Instead of storing the data and running queries against stored data, as the databases do, the Esper engine allows applications to store queries and run the data through. Response from the Esper engine is real-time when conditions occur that match queries. The execution model is thus continuous rather than only when a query is submitted. Esper provides two principal methods or mechanisms to process events: event patterns and event stream queries. Esper offers an event pattern language to specify expression-based event pattern matching. Underlying the pattern matching engine is a state machine implementation. This method of event processing matches expected sequences of presence or absence of events or combinations of events. It includes time-based correlation of events. Esper also offers event stream queries that address the event stream analysis requirements of CEP applications. Event stream queries provide the windows, aggregation, joining and analysis functions for use with streams of events. These queries are following the Event Processing Language (EPL) syntax. EPL has been designed for similarity with the SQL query language but differs from SQL in its use of views rather than tables. Views represent the different operations needed to structure data in an event stream and to derive data from an event stream.

Generic Event Architecture (GEAR) [Casimiro2007] is architecture to provide the possibility of integration of physical and computer information flows in large distributed systems interacting with the physical environment and being composed from a huge number of smart components - systems-of-embedded-systems. GEAR architecture recognises the following layers: *environment* - the physical surroundings, remote and close, solid and ethereal, of sentient objects; *body* - the physical

[17] *http://esper.codehaus.org*

embodiment of a sentient object; *translation layer* - the layer responsible for physical event transformation from/to their native form to event channel dialect - this layer performs observation and actuation operations on the lower side and transactions of event descriptions on the other; *event layer* - the layer responsible for event propagation in the whole system, through several Event Channels – i.e. it provides important event-processing services which are crucial for any realistic event-based system; *communication layer* - the layer responsible for wrapping events into carrier event-messages, to be transported to remote places. GEAR utilises **Cooperating Smart devices** (COSMIC) middleware [Kaiser2005] as an appropriate event model. It allows specifying events with attributes to express spatial and temporal properties. This is complemented by the notion of Event Channels, which are abstractions of the underlying network and enforce the respective quality attributes of event dissemination. Event channels reserve the needed computational and network resources for highly predictable event systems. The COSMIC middleware maps the channel properties to lower level protocols of the regular network and defines an abstract network which provides hard, soft and non-real-time message classes.

TinyCOPS is the implementation of the proposed component framework aligned with the design philosophy of TinyOS 2.0. The flexibility of TinyCOPS to support different sensor node platforms, communication protocols and interaction patterns has been demonstrated experimentally. TinyCOPS makes clear distinction between the metadata and constraint and support attribute-specific operators. Conceptually, however, more important is the difference in the level of decoupling between the middleware service implementation and the communication protocols. It also allows for individual customization of the subscription and the notification delivery protocols and provides infrastructure for address information tunnelling and matching point control. TinyCOPS is concentrated on the class of relatively resource limited sensor network hardware, where compile-time optimization has comparably large impact, and where the run-time modifications are mostly limited to parameter tuning.

4.7 Operating Systems for IoT

For a true Internet Things to be established there is a need to have Internet-based access and control down to the lowest sensors level, i.e. that the sensors and small resource constrained devices are true Internet objects. This requires operating systems and IP-stacks to run natively and embedded in the nodes. The research into this area has been intensive. The two most prominent approaches are Contiki and TinyOS.

4.7.1 The Contiki OS

Contiki[18] is an open source operating system for wireless sensor networks and the Internet of things. Contiki provides low-power networking for resource constrained systems along with a development and simulation environment that makes research, development, and deployment of embedded software easy. Contiki is an event-based OS and is based on the C programming language and it is designed for microcontrollers with small amounts of memory. A typical Contiki configuration is 10 kilobytes of RAM and 48 kilobytes of ROM. Contiki contains the low-power wireless Rime communication stack, the uIP TCP/IPv4 stack, and the IPv6 Ready certified uIPv6 TCP/IPv6 stack complete with 802.15.4 6 lowpan header compression and fragmentation.

Contiki has a complete IPv6 stack including low-power MAC layer and 6lowpan adaptation layer. Contiki's default duty-cycling MAC layer is ContikiMAC which in typical traffic load will keep the radio on less than one percent of the time. The routing mechanism in Contiki is based on IETF-RPL which is a routing protocol for low-power and lossy networks currently under standardization.

To provide a long sensor network lifetime, it is crucial to control and reduce the power consumption of each sensor node. Contiki provides a software-based power profiling mechanism that keeps track of the energy expenditure of each sensor node.

Being software-based, the mechanism allows power profiling at the network scale without any additional hardware. Contiki's power profiling mechanism is used both as a research tool for experimental evaluation of sensor network protocols, and as a way to estimate the lifetime of a network of sensors.

[18] *http://www.sics.se/contiki/*

The Contiki OS is actively developed by a team of around 20 developers associated with many small and large companies such as Cisco, Atmel, SICS, SAP, ST Microelectronics, and Sensinode. The Contiki development community also has an active mailing list with many active Contiki users and developers. For more information about Contiki and the community see the Contiki[19] home page.

4.7.2 TinyOS

TinyOS[20], similarly to Contiki, is an open-source (BSD license) operating system for low-power embedded wireless devices. It is also characterized by a low code and memory footprint, possibility to be used on different hardware platform and optimization of consumption and low-power communication.

TinyOS also provides its own implementation of a 6LoWPAN adaptation layer (called BLIP) plus a number of solutions to cover different communication and application issues e.g. networking, MAC, file-systems, simulators, etc. The main difference with Contiki is that TinyOS uses a slightly different approach concerning the programming language (called nesC). Any TinyOS functionality must in fact be enclosed within an entity called "module", characterized by its "boundaries" i.e. its external interfaces.

A TinyOS interface is a set of plain C functions headers which can be configured to behave either as "commands" or "events", according to the direction of the function call (i.e. "sent" or "received"). A TinyOS module, then, provides or uses a number of interfaces which must be connected to other modules either developed by users or provided by the OS itself, similarly to system APIs.

A TinyOS device, thus, can be programmed as a "network" of modules which exchange commands and events through pre-defined interfaces. Internally, modules are programmed in plain C language, which simplifies the TinyOS learning curve. Currently TinyOS development group is focused on branch 2.x, after a major refactoring occurred after branch 1.x to solve licensing and code organization issues. TinyOS is available to a set of different hardware platforms based on different micro-controllers families including the TI's MSP430 family the Atmel ATMega128 family, Intel's px27ax and others.

4.8 Radio and Network technologies

4.8.1 Multiradio technologies

Nowadays, in our everyday life it is rather usual to observe devices endowed with multiple radio interfaces enabling as many wireless communication technologies. The concept of multiradio is essentially the following: a device can be using, even at the same time, several heterogeneous wireless connections by means of one or more radio front-ends. The most relevant and widespread devices are the so-called Smartphones. The wireless technologies that such devices are more frequently equipped with are 802.11 (Wi-Fi), GPRS/EDGE, UMTS/HSPA and Bluetooth.

As an approach, multiradio will provide smart devices with the possibility to be connected anytime and anywhere not only to the Internet but also to services offered by telecommunication operators such as Wi-Fi calling or Seamless Handovers using technologies like General Access Networks (GAN) or the new standard IEEE 802.21 Media Independent Handover (MIH).

The resulting multiradio smart devices will leverage on the different available radio technologies and will provide seamless connectivity to applications having variable latency and bandwidth requirements. Based on such requirements and on the context (e.g. radio technologies coverage), specific management layers will handle the selection of the most proper communication technologies. In particular, the handover operations present the main challenge since they should be transparent to the application layer and able to cope with diverse wireless technologies.

Some of the relevant technologies considered for a multiradio environment are shown below.

[20] *http://www.tinyos.net*

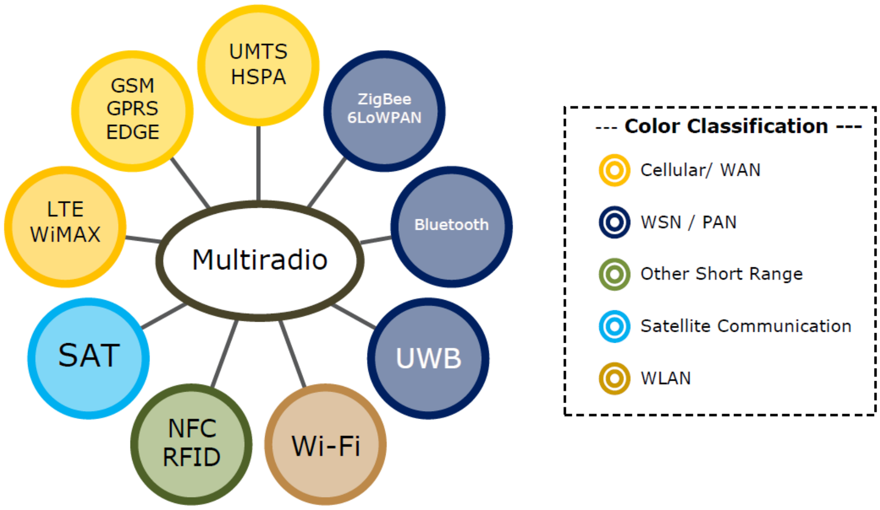

Figure 10: Relevant technologies for multiradio

In recent years, a standardization work has been performed in this research area. More specifically, some working groups have addressed their interest towards the definition of communication standards that natively exploit the availability of different radio technologies (natively multiradio standards). Instead, other standardization and research initiatives have concentrated their efforts on developing handover management solutions enabling seamless communications that exploit the available wireless technologies within a radio range (technologies enabling the multiradio approach).

The following sub-sections present the current state of the art of the two classes of multiradio solutions just mentioned.

Natively multiradio standards

At the moment, there is only one communication standard adopting multiradio approach by design, that is, Bluetooth 3.0 + HS.

This standard was endorsed by the Bluetooth SIG on April 2009. In fact, the version 3.0 + HS of the Bluetooth Core Specification[21], are the only multiradio technology available in the market. It provides data transfer speeds up to 24Mbit/s by using two types of radio links. The Bluetooth link is required for negotiation and establishment of a high data rate IEEE 802.11 connection.

The main enhancement that the specification provides with respect to previous versions is the Alternate MAC/PHY (AMP) which is an alternative controller used to transfer large quantities of data. The specification defines two controllers: Basic Rate/Enhanced Data Rate (BR/EDR) and AMP. Functionalities such as device discovery, connection establishment and connection maintenance are offered by the BR/EDR controller. In case higher data rate is needed and the devices involved can support AMP capabilities, the core system will provide the mechanism to move data traffic from the BR/EDR controller to the AMP controller (IEEE 802.11) in order to get higher data transfer speeds.

Several commercial solutions supporting Bluetooth 3.0 + HS are already available. A list of compliant chipsets is reported in the following.

- CSR9000 Chipset from CSR[22]
- BCM2070, BCM2075, BCM43225, BCM94312 and BCM4325 chips from Broadcom[23]
- WiLink™ 7.0 chip from Texas Instrument[24]

[21] www.bluetooth.com
[22] http://www.csr.com/products/32/csr9000
[23] http://www.broadcom.com/press/release.php?id=s442042
[24] http://ti.com/wilink7pr

Technologies enabling the multiradio approach

As previously mentioned, specific technology solutions have been designed to exploit the availability of different radio resources. Some of them base on the concept of opportunistic communication that will be described later in this section. Other solutions include the already introduced GAN and the standard IEEE 802.21 MIH.

4.8.2 Opportunistic networking

Opportunistic networking is an emerging communication paradigm related to an intermittent communication where nodes can alternate periods of activity to periods in which they are not available, either because they are switched off or because they are out of the wireless access point range. During off periods, nodes store the information they need to transfer and wait for the next on period. Once the nodes are able to communicate, they send their data using the relevant available interface. In light of this communication behaviour and data delivery strategy, opportunistic networks are commonly referred as supporting the "store-carry-and-forward" communication paradigm, inherited by the well-known "store-and-forward" typical of traditional packet switching networks.

The concept of opportunistic networking is useful within a multiradio environment because it could provide an alternative solution to technologies allowing continuative communication. In fact, opportunistic networking could work jointly with handover management systems to enhance the communication in areas characterized by poor wireless coverage.

IEEE 802.21 MIH

The standard IEEE 802.21 MIH was created in 2008 in order to support interworking between IEEE 802.X and 3rd Generation Partnership Project (3GPP) networks. More specifically, it provides a framework which could support a complex exchange of signaling information aiming to enable seamless handover in a multi-connectivity environment.

Its main contribution is the creation of a new layer called MIHF (MIH Function) between L3 and L2 of the OSI model which acts as an intermediate, coordinating the exchange of information between the devices involved and helping to make the handover(HO) decision and execution.

The service sets provided by the standard are the following:

- *Event* (e.g. link up/down, link events in general)
- *Command*
  - *Link* – L2 control and configuration
  - *MIH* – Help on HO procedure (e.g. network selection)
- *Information* (e.g. neighbor networks information)

Actually, there are not commercial solutions based on the considered standard. However, there are two open source projects whose aim is to provide an implementation of the overall IEEE 802.21 MIH framework.

- ***ODTONE***[25]

  Supported features: Basic IEEE 802.21 MIH framework with IEEE 802.11/IEEE 802.3 Interfaces, Android compilation support, MIH Sensor SAP (v0.3).

  Releases: ODTONE 0.1(2010); ODTONE 0.2 (2010); OPTONE 0.3 (2011, current).

- ***Open MIH***[26]

  Supported features: Basic IEEE 802.21 MIH framework

  Releases: Open MIH 0.1 (2009, current)

[25] http://helios.av.it.pt/projects/odtone
[26] http://sourceforge.net/projects/openmih

4.8.3 General Access Network (GAN)

GAN is defined in the 3GPP specifications TS 43.318**Error! Reference source not found.** and TS 44.318. Its main goal is to provide interworking between IP-based networks (e.g. Wi-Fi) and 3GPP. The target is not only limited to Wi-Fi networks but as the standard is mobile phone oriented and the majority of the smartphones in the market are Wi-Fi enabled, the commercial solutions adopting GAN use the technology mainly to interconnect Wi-Fi and 3GPP networks.

The architecture proposed in the standard includes the following components:

- IP Network (Wi-Fi)
- 3GPP Network with a GANC (GAN Controller)
- User Equipment (UE)

The GANC acts like a Radio Network Controller (RNC). The GANC is connected to the IP Network through an interface called Up, allowing the user to utilize his cellular services and credentials with just being connected to the IP network.

The following service providers are implementing GAN in their networks:

- T-Mobile US → Wi-Fi Calling[27]
- Rogers and FIDO → Wi-Fi Calling[28]
- FIDO → Wi-Fi Calling[29]
- Cincinnati Bell → Fusion Wi-Fi[30]
- Orange UK → UMA[31]

The services basically consist in allowing the user to make phone calls through a Wi-Fi connection when he/she is at home, reducing the cost of the calls and improving coverage.

4.8.4 P2P networks

As described in the deliverable D5.1, P2P networks are distributed systems characterized by distinguishing peculiarities, which can be shortly summarized as follows:

- absence of a centralized controlling entity, anything (operations, storage, addresses, etc.) is distributed all over the system
- lack of hierarchy among nodes, which are equally privileged, equipotent participants in the application (in one word, just peers)
- cooperation among peers in performing tasks is a distinctive aspect
- peers share their resources, which can be processing power, disk storage, network bandwidth or other

P2P concept became very popular at the beginning of this century for a specific Internet application, that is, the file sharing. It has quickly favoured the exchange of music and video contents among Internet surfers and the evolution of ever more efficient systems. In fact, most of the successful commercial P2P systems fall in this category, where the shared resource is primarily the contents (i.e., files).

However, referring to the last item, other systems can be labelled as P2P, sharing different types of resources. For instance, sharing processing power is an efficient solution to perform very complex tasks.

[27] http://mytouch.t-mobile.com/mytouch-4g-wifi-calling
[28] http://www.rogers.com/web/content/add-ons/callingservices?tab1_content&submenu5
[29] http://www.fido.ca/web/content/monthly/fidouno
[30] http://www.cincinnatibell.com/consumer/wireless/fusion_wifi

[31] http://shop.orange.co.uk/shop/show/offer/uma

In Section 0 we list some of the most known file sharing systems, introducing them in chronological order of appearance and, thus, of evolution. In Section 0, instead, we report other systems in which the main shared resource is not merely a file but it can consist of processing power or disk storage.

File sharing systems

The evolution of file sharing systems, originally appeared at the end of the last century, has been quite dynamic, many different systems followed one another. Here is a not exhaustive list:

Napster

http://www.napster.com

Napster was the first file sharing system to appear, actually based on a centralized server for IP address resolution while contents were distributed among peers. By allowing to freely share contents, it was the originator of possible copyright violations followed by consequent compensation requests by lawyers of majors and content producers. After the shutdown in July 2001, today Napster is active again, but under payment, and differently organized.

Gnutella

http://rfc-gnutella.sourceforge.net/

Gnutella overcame Napster drawback of having a single tracker by distributing among the peers not only the shared files but also availability and location information. Gnutella distributes uniformly the control information in the network and adopts flooding-based strategies to localize resources (unstructured resource location), thus suffering from scalability problems. Last updates on the website date to 2003 when Gnutella 0.6 was the last stable version, although yet in late 2007 Gnutella was one of the most popular file sharing networks on the Internet.

Freenet

http://freenetproject.org

From the architectural perspective, Freenet is rather similar to the approach taken by Gnutella. Differently from Gnutella, Freenet is still active (the current version 0.7.5 is available for Windows systems free of charge) but has partially extended its offering. Indeed, besides sharing files, the system lets subscribers browse and publish "freesites" (web sites accessible only through Freenet) and chat on forums.

Kazaa/FastTrack

http://www.kazaa.com

http://developer.berlios.de/projects/gift-fasttrack

Like other file sharing systems, Kazaa Media Desktop, the correct name for Kazaa, was based on the FastTrack P2P protocol. Its life has passed through a large number of copyright-related lawsuits, mainly filed by RIAA. Kazaa is now running under license as a legal music subscription service by Atrinsic, Inc.

BitTorrent

http://www.bittorrent.com

BitTorrent is another second generation P2P file sharing systems, whose network name coincides with the P2P protocol at its basis. BitTorrent is now maintained by company BitTorrent, Inc. There are numerous BitTorrent clients available for a variety of computing platforms (namely, Windows, MAC and Linux). For Windows machines, v7.2 is the last available version.

eDonkey

http://en.wikipedia.org/wiki/EDonkey_network

Best suited to share big files among users, the eDonkey P2P file sharing network is thus generally used to share video files, full music albums and computer programs. In the past, the eDonkey network was supported by the MetaMachine Corporation, now out of business, while currently no organisation does that and eDonkey works by being fully supported by its users alone. The related client software can be downloaded free of charge from several different sources.

eMule

http://www.emule-project.net/home/perl/general.cgi?l=1

http://sourceforge.net/projects/emule/files/

It is one of the most recent P2P file sharing systems, available for Windows machines, and it was thought as an alternative to eDonkey, a network which is indeed now supported by latest eMule versions (v0.40+).

Other examples of P2P systems

File sharing systems are not the only ones which follow the P2P basic concept of resources sharing. Other distributed systems have been designed for different purposes preserving the sharing approach. By sharing processing power, for instance, seemingly impossible number crunching tasks are now possible and this takes less time than it would by just a single individual. Here are some heterogeneous examples:

Chord project

http://pdos.csail.mit.edu/chord/

Chord is a MIT general framework to build scalable and robust distributed systems based on P2P ideas. Several systems are founded on Chord and on its distributed hash lookup mechanism related to Distributed Hash Tables (DHT): e.g., Cooperative File System (CFS), UsenetDHT and OverCite.

Cooperative File System

http://pdos.csail.mit.edu/papers/cfs:sosp01/

The Cooperative File System (CFS) is a P2P read-only storage system: it allows anyone to publish and update their own file system and provides read-only access to others. The resource shared in this system is therefore disk storage. Based on Chord location protocol, CFS is scalable in terms of number of servers. It runs on Linux, OpenBSD, and FreeBSD. Performance is such that CFS delivers data to clients as fast as FTP.

4.8.5 Grid computing systems

http://en.wikipedia.org/wiki/Grid_computing

Grid computing systems represent an infrastructure for the distributed computing. They are considered when the need for processing large amounts of data is satisfied by using a large amount of distributed resources. Such systems are intrinsically related to a coordinated sharing of resources within a virtual organization. The term 'grid computing' refers to the integration of computer resources from multiple administrative domains in order to pursue a common aim. Cluster computing is pretty similar to grid computing, the differences being that grids tend to be more loosely coupled, heterogeneous, and geographically dispersed, and that, commonly, a single grid is used for a variety of different purposes. Technically, grid software libraries known as middleware are the key elements on which the implementation of such systems is based. Two specific examples of distributed systems associated with sharing of processing power follow.

SETI@Home

http://setiathome.ssl.berkeley.edu/

SETI@home, where SETI stands for Search for Extra-Terrestrial Intelligence, is one of the most known distributed computation projects in the world. It is managed by the Space Sciences

Laboratory at the University of California at Berkeley in the United States and it is based on public volunteering. It exploits the processing power jointly provided by hundreds of thousands Internet-connected computers with the aim of searching possible extraterrestrial intelligence life by analyzing radio signals.

Great Internet Mersenne Prime Search

http://www.mersenne.org/

http://en.wikipedia.org/wiki/Great_Internet_Mersenne_Prime_Search

The Great Internet Mersenne Prime Search (GIMPS) is a project that relates to the use of distributed and shared resources available on the Internet (in substance, disk storage and, mainly, processing power). By means of an optimized program able to check the primality of a number, the final goal is to compute Mersenne prime numbers, i.e., those ones that can be written in the exponential form 2^p-1 where p is a prime number.

4.9 Security in IoTS applications

In the ebbits platform security has to satisfy business requirement containing high expectations for confidentiality, integrity, availability and robustness. Industry regulations have to be taken into account while also handling issues rising due the use of distributed systems. For this state-of-the-art solutions are examined and further developed.

Trust

The term trust is used in the meaning *Alice trusts Bob's public key if she is sure that Bob is the legitimate owner of the associated private key*. There are two reference implementations for trust available in the middleware. One based on Public Key Infrastructure (PKI) and another using Web-of-Trust (WoT).

Virtualization

The virtualization aim is twofold. On one hand it should abstract from hardware and on the other hand it should protect against attacks aiming in tracing individuals. This is achieved with the use of non-persistent HIDs.

Semantics

The usage of semantic web technology (SWT) avoids the hard coding of security assumptions but instead represent them in an external ontology. This way it is possible to make security-relevant decisions based on the security ontology.

Security is provided by many components in the middleware each responsible for different aspects of the system. It is not the aim of this document to provide a detailed overview of realization of these components so we will only give a high level overview of services provided.

4.9.1 Linksmart Security

Linksmart is an Open Source software that is commercialisation of the middleware from the Hydra project. It contains several security modules:

Crypto Manager

Cryptographic operations are required for protecting the middleware communication from eavesdropping and modification, for authentication of devices and users. These components are merged in the so called Crypto Manager which serves various services like the creation and verification of digital signatures, encryption and decryption and generation and confidential storage of keys.

Trust Manager

The Trust Manager can be used to verify if a token offered by an entity is trustworthy. The decision if an entity is the legitimate owner of a key is based on trust models. The Trust Manager can implement any kind of trust model taking a token as input and returning a trust value as response. Two implementations are already available in the LinkSmart, one for PKI and one for WoT.

Security Library

To protect the message exchange between entities security protocols and cryptographic operations have to be applied. Two kinds of communication are distinguished. One is between different managers inside the middleware which use LinkSmart Security based on symmetric cryptography based on keys established in development time. The other case is between different devices which may be unknown to each other prior to communication. For this case a protocol is used which uses asymmetric cryptography for authentication and to establish shared key.

Access Control Policy Framework

The Access Control Policy Framework provides policy-driven, access protection for IoT-devices and applications, building on the XACML (eXtensible Access Control Markup Language) standard. The Policy Decision Point (PDP) is responsible for making the access decisions based on the XACML request it receives from a Policy Enforcement Point (PEP) and the set of policies that have been published to it.

Obligation Policy Framework

Obligation policies are a realization of the event-condition-action (ECA) policies, using some advanced techniques like semantic reasoning, complex event processing and enforcement monitors to increase the benefits of the policy framework. Obligation policies shall help developers in setting up a distributed system that automatically adapts its settings and implementations upon context changes.

5. Semantic Data Management

The aim of this section is to provide a state of the art in the areas of semantic web languages, standards and triplestore implementations. The survey is focused on the mostly investigated areas in the area of semantic web, relevant to the semantic data management in ebbits.

The commonly used standards for formal ontology definition are outlined. As there is quite lot of effort in the area of semantic description of web services, the most relevant standards are listed. As the ebbits also aims to formalize the various kinds of rules, including business rules or service composition/orchestration rules, the overview of existing semantic description of rules standards are investigated.

One of the most relevant technologies is the frameworks enabling the storage, processing, reasoning and querying the semantic information. As in ebbits there is the quite strong assumption, that there will be required to handle bigger volumes of semantic data, the overview is focused on the triple stores – the native semantic storages, instead of providing the survey on the existing reasoners, which does not seem to be too much relevant for the assumed architecture design.

5.1 Formalisms for modelling web services

OWL-S (Web Ontology Language for Services)

OWL-S [OWL-S] is OWL ontology for semantic description of the web services. The structure of the OWL-S consists of a service profile for service discovering, a process model which supports composition of services, and a service grounding, which associates profile and process concepts with the underlying service interfaces.

Service profile has functional and non-functional properties. Functional properties describe the inputs, outputs, preconditions and effects of the service (IOPEs). The non-functional properties describe the semi-structured information intended for human users for service discovery, e.g. service name, description and parameters which incorporate further requirements on the service capabilities (e.g. security, quality of service, geographical scope, etc.).

Service model specifies how to interoperate with the service. The service is viewed as a process which defines the functional properties of the service (IOPEs), together with details of its constituent processes (if the service is a composite service). The service model functional properties can be shared with the service profile. OWL-S distinguishes between atomic, simple, and composite processes. OWL-S atomic processes can be invoked, have no sub-processes, and are executed in a single step from the requester's point of view. The simple processes are used as elements of abstraction, they are viewed as executed in a single step, but they are not invocable. Composite processes consist of simple processes and define their workflows using control constructs, such as sequence, split, if-then-else or iterate.

Service grounding enables the execution of the web service by binding the abstract concepts of the OWL-S profile and process model to concrete messages and protocols. Although different message specifications are supported by OWL-S, the widely accepted WSDL is preferred.

WSMO (Web Service Modelling Ontology)

WSMO [WSMO] is a conceptual model for describing semantic web services. It consists of four major components: ontologies, goals, web services and mediators.

Ontologies provide the formal semantics to the information used by all other components. WSMO specifies the following constituents as part of the description of ontology: concepts, relations, functions, axioms, and instances of concepts and relations, as well as non-functional properties, imported ontologies, and used mediators. The latter allows the interconnection of different ontologies by using mediators that solve terminology mismatches.

Goal specifies objectives that a client might have when consulting a web service, i.e. functionalities that a web service should provide from the user perspective. In WSMO a goal is characterized by a set of non-functional properties, imported ontologies, used mediators, the requested capability and the requested interface.

A *web service* description in WSMO consists of five sub-components: non-functional properties, imported ontologies, used mediators, a capability and interfaces. The capability of a web service defines its functionality in terms of preconditions, post-conditions, assumptions and effects. A capability may be linked to certain goals that are solved by the web service via mediators. Preconditions, assumptions, post-conditions and effects are expressed through a set of axioms and a set of shared all-quantified variables. The interface of a web service provides further information on how the functionality of the web service is achieved. It describes the behaviour of the service from the client's point of view (service choreography) and how the overall functionality of the service is achieved in terms of cooperation with other services (service orchestration). A choreography description consists of the states represented by ontology, and the if-then rules that specify (guarded) transitions between states. The ontology that represents the states provides the vocabulary of the transition rules and contains the set of instances that change their values from one state to the other. Like for the choreography, an orchestration description consists of the states and guarded transitions. In extension to the choreography, in an orchestration transition rules, that have as a post-condition the invocation of a mediator that links the orchestration with the choreography of a required web service, can also appear.

Mediators describe elements that aim to overcome structural, semantic or conceptual mismatches that appear between the different components that build up a WSMO description.

WSMO is formalized using the Web Service Modelling Language (WSML) which is based on description logic, first-order logic and logic programming formalisms.

WSDL-S (Web Service Semantics)

WSDL-S [WSDL-S] is a small set of proposed extensions to Web Service Description Language (WSDL) by which semantic annotations may be associated with WSDL elements.

WSDL-S defines URI reference mechanisms to the interface, operation and message WSDL constructs to point to the semantic annotations defined in the externalized domain models. WSDL-S defines the following extensibility elements and attributes:

- *modelReference* element - allows for one-to-one associations of WSDL input and output type schema elements to the concepts in a semantic model;
- *schemaMapping* attribute - allows for many-to-many associations of WSDL input and output complex type schema elements to the concepts in a semantic model. It can point to a transformation (for example XSLT) from XML data to the external ontological data in RDF/OWL or in WSML;
- *precondition* and *effect* elements - are used on WSDL interface operations to specify conditions that must hold before and after the operation is invoked. The conditions can be specified directly as an expression with format defined by the semantic language or by reference to the semantic model;
- *category* element - provides a pointer to some taxonomy category. It can be used on a WSDL interface and is intended to be used for taxonomy-based discovery.

BPEL4WS (Business Process Execution Language for Web Services)

BPEL4WS[32] is a specification that models the behaviour of web services in a business process interaction. It is based on the XML grammar which describes the control logic required to coordinate web services participating in a process flow. An orchestration engine can interpret this grammar, thus it can coordinate activities in the process. BPEL4WS is a layer on the top of WSDL (Web Services Description Language). WSDL defines the specific operations and BPEL4WS defines how the operations can be sequenced. Every BPEL4WS process can be considered as a web service using WSDL describing the public entry and exit points for the process. WSDL data types are used within a

[32] http://www.ibm.com/developerworks/library/specification/ws-bpel

BPEL4WS process to describe the information that passes between requests. WSDL might be used to reference external services required by the BPEL4WS process. BPEL4WS provides support for both executable and abstract business processes. The executable process models a private workflow. The abstract process specifies the public message exchanges between parties. The executable processes provide orchestration support while the business protocols (abstract processes) focus more on the choreography of the services.

Support for basic and structured activities is included. The basic activities might be receiving or replying to message requests as well as invoking external services. The structured activities specify what activities should run in what order – the whole process flow. These activities also provide support for conditional looping and dynamic branching. The structured activities might specify that certain activities should run sequentially or in parallel. *Containers* and *partners* are two important elements within BPEL4WS. A container is a variable for exchange in the message flow. A partner could be any service that the process invokes or any service that invokes the process. Each partner is mapped to a specific role that it fills within the business process. This is managed by containers.

In BPEL4WS, a set of activities can be grouped into a single transaction – it means that the steps enclosed in the scope should either all complete or all fail. Within this scope, the developer can then specify compensation handlers that should be invoked if an error occurs. BPEL4WS provides a robust exception handling mechanism through the use of throw and catch clauses, similar to the Java programming language.

5.2 Semantic storage

5.2.1 Semantic triple stores

A wide variety of triple stores is available nowadays for storage of semantic information in RDF and/or OWL. The semantic triple stores are often integrated into a framework that provides querying interfaces and data maintenance capabilities. In the following paragraphs, we provide a survey of some of the most known triple stores together with a short description of technology used, functionality provided, advances, and licensing policies.

JENA

Jena[33] is a popular and frequently used Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine. Jena includes an API for both RDF and OWL, together with capabilities of reading and writing RDF in RDF/XML, N3 and N-Triples. The built-in repository enables in-memory and persistent storage of semantic data, together with the reasoning and inference by means of SPARQL query engine. The SDB and TDB are two subsystems for persisting RDF and OWL data in Jena. TDB is focused on the data access by means of Jena APIs, while SDB is for the RDF storage and query specifically to support SPARQL. Java framework is available under open source license.

Sesame

Sesame[34] is a popular open source Java framework for storage, inference and querying of RDF data. It can be used as a database for RDF and RDF Schema, or as a Java library for applications that need to work with RDF internally. Sesame is internally organised into a modular and layered architecture, where the semantic data repository and the respective Storage And Inference Layer (SAIL) interacts with functional modules such as the SeRQL, RQL and RDQL query engines, the admin module, and RDF export. Access to these functional modules is available through Sesame's Access APIs, consisting of two separate parts: the Repository API and the Graph API. The Repository API provides high-level access to Sesame repositories, such as querying, storing of RDF files, extracting RDF, etc. The Graph API provides more fine-grained support for RDF manipulation, such as adding and removing individual statements, and creation of small RDF models directly from code.

[33] jena.sourceforge.net
[34] http://sourceforge.net/projects/**sesame**

AllegroGraph

AllegroGraph[35] RDFStore is a high-performance persistent RDF graph database. It is capable to handle billions of triples in a good performance. AllegroGraph supports SPARQL, RDFS++, and Prolog reasoning from numerous client applications. The AllegroGraph features include effective database replication mechanisms, pre-indexing of triples, full text and free text indexing, powerful query analyzer, transaction processing, and many others. AllegroGraph is written in Common Lisp; the clients are available for all main platforms (Java, Python, Perl, etc.). Licensing of AllegroGraph depends on the allowed capacity of the store and ranges from free version (< 50 mil. triplets) to commercial enterprise version.

BigOWLIM / SwiftOWLIM

SwiftOWLIM and BigOWLIM [BigOWLIM2010] are variants of the OWLIM family of semantic repositories, called also as RDF database management systems. This framework provides native RDF engines, implemented in Java and compliant with Sesame and Jena, robust support for the semantics of RDFS, OWL Horst, OWL 2 QL and OWL 2 RL, high scalability, loading and query evaluation performance. It is declared by vendors that the SwiftOWLIM is the fastest semantic repository in the World: it supports non-trivial inference with tens of millions of statements on contemporary desktop hardware. The advantage of BigOWLIM is its scalability and multi-user query performance. OWLIM framework is based on the Triple Reasoning and Rule Entailment Engine (TRREE) of Ontotext. It is implemented in Java and packaged as a Storage and Inference Layer (SAIL) for the Sesame RDF database. SwiftOWLIM is available for free for any purpose; BigOWLIM is provided free of charge for research, evaluation and development purposes. For commercial use, licenses of BigOWLIM are offered at a flat pricing model, where the price is proportional to the capacity of the servers on which the engine will be installed.

BigData

Bigdata[36] is a horizontally-scaled, general purpose storage mechanism for ordered data (B+Trees), designed to operate on either a single server or a cluster of commodity hardware. Bigdata uses dynamically partitioned key-range shards and thus it may be deployed on 10s, 100s, or even thousands of machines and new capacity may be added incrementally without requiring the full reload of all data. The Bigdata RDF database supports RDFS and OWL Lite reasoning, high-level query (SPARQL), and datum level provenance. Bigdata is written in Java and is freely available under an open-source license (GPL v2).

Mulgara

The Mulgara Semantic Store[37] is an open source, massively scalable, transaction-safe, purpose-built database for the storage and retrieval of RDF, written in Java. It provides the RMI or embedded data access, JRDF and REST programming interfaces to the semantic repository. Data access and querying is allowed by means of TQL or SPARQL. Mulgara is licensed under the Open Software License v3.0.

OntoBroker / Ontoprise

Ontobroker[38] is a deductive, object-oriented database system that has originally been developed as a research prototype at the AIFB Karlsruhe as part of the Semantic Web initiative. As Ontobroker had matured, it went commercial and is now available through Ontoprise [Ontobroker]. The OntoBroker is an implementation of highly scalable Semantic Web middleware. It supports all W3C Semantic Web recommendations such as RDF(S), OWL, SPARQL, RIF and ObjectLogic. The new ObjectLogic is best of breed of RDF, OWL and F-Logic concerning the expressive power and evaluation performance. The Ontobroker semantic framework includes the RDF triple store, query and inference engine. It is well integrated into the general OntoBroker suite, allowing close interaction between the other supported knowledge representation formats of OWL and F-logic.

[35] *http://www.franz.com/agraph/allegrograph*
[36] *http://www.systap.com/bigdata.htm*
[37] *http://www.mulgara.org*
[38] *http://ontobroker.semanticweb.org*

Virtuoso

OpenLink Virtuoso [Virtuoso] is a SQL-ORDBMS and Web Application Server hybrid that provides SQL, XML, and RDF data management in a single multithreaded server process. Triple store access is available via SPARQL, SIMILE Semantic Bank API, ODBC, GRDDL, JDBC, ADO.NET, XMLA, WebDAV, and Virtuoso/PL (SQL Stored Procedure Language). Virtouso [Virtouso2007] is also an OWL Reasoner, which supports a subset of OWL subclass or sub property relations. It also includes a Live SPARQL Query Service Endpoint in all installations. The product is developed in C language and is available in Open Source and Commercial editions.

Redland

Redland[39] is a set of free software C libraries that provide support for RDF. It includes object based libraries and APIs for manipulating the RDF graph, triples, URIs and Literals, together with querying mechanisms of SPARQL and RDQL. Redland provides a storage for graphs in memory and persistently with Sleepycat/Berkeley DB, MySQL 3-5, PostgreSQL, AKT Triplestore, SQLite, files or URIs. The triple stores are accompanied with data aggregation and recording provenance support. All Redland packages are free software / open source software, which are released under the LGPL 2.1, GPL 2 or Apache 2 licenses as alternatives.

5.3 Rules support in Semantic Technologies

The usage of the rules on the top of the ontologies is one of the goals of W3C plan for the Semantic Web Architecture development. At present there exist plenty of languages for ontology description, related to the Semantic Web, with different expressional power. These languages are useful for quite precise declarative knowledge description of the objects, their properties and relations. The expressional capabilities of these languages have various disadvantages and problems, which need to be solved. The rules integrated with ontologies may solve the problems related to ontology languages. Moreover, the rules can also extend the expressional and reasoning capabilities and functionality of ontology languages.

For example, many of the limitations of e.g. OWL stem from the fact that, while the language includes a relatively rich set of class constructors, the language provided for talking about properties is much weaker. In particular, there is no composition constructor, so it is impossible to capture relationships between a composite property and another (possibly composite) property. The standard example here is the obvious relationship between the composition of the “parent” and “brother” properties and the “uncle” property. The complex relationships between composed properties cannot be captured (even the relatively simple “uncle” example cannot not be captured (because “uncle” is not one of “parent” or “brother”). More generally, the most of the ontology languages are based on so-called Open World Assumption, which means, that knowledge is assumed to be incomplete. In such circumstances, it may happen that inference does not have to lead always to reasonable conclusions. The problems related to Open World Assumptions may be partially solved by using the rules, which usually support the Closed World Assumption. It is assumed that everything that cannot be inferred from the knowledge is false. The rules support the solution of this issue by using the procedural elements adopting the several types of negation (e.g. negation as failure (naf) or strong negation) influencing the reasoning.

In general, the rules can be used for various purposes, for example:

- deductive rules used for inferences based on dependencies between some ontology properties, such as the transfer of properties from parts to wholes,
- meta-reasoning rules used for facilitating meta-reasoning on ontology in control or knowledge engineering tasks acquisition, validation or maintenance of an ontology
- connecting rules between ontologies required for reasoning across several domains

[39] *http://librdf.org*

- mapping rules for mapping ontologies in data integration, and querying heterogeneous sources

Considering the amount of existing standards and approaches for ontology description, the integration of rules is a complicated problem. In the last few years, there has been a significant progress in the area of design and development of more or less specific formalisms for rule definition and their integration with various ontology languages. The aim of this section is to provide an overview of existing standards and approaches in the three areas:

- Rule languages: containing the overview of existing standards and approaches to rule formalization
- Rule markup languages: containing the overview of standards and approaches aiming to define a uniform format of rules used mainly for rule exchange between the rule systems
- Specific systems integrating rules: containing the list of selected systems implementing specific kinds of rules and inference mechanisms.

The rule inference engines are mostly based on the several approaches using various kinds of logics. The theory behind the rule semantics and inference and the overview of rule engines description is out of the scope of this section.

5.4 Rule languages

OWL2 Rule Language (OWL2 RL)

The OWL2 RL profile is aimed at applications that require scalable reasoning without sacrificing too much expressive power. It is designed to accommodate both OWL 2 applications that can trade the full expressivity of the language for efficiency, and RDF(S) applications that need some added expressivity from OWL 2. This is achieved by defining a syntactic subset of OWL 2 which is amenable to implementation using rule-based technologies in the form of first-order logic. The design of OWL 2 RL was inspired by Description Logic Programs DLP and pD*.

A suitable rule-based implementation is assumed to have the desirable computational properties; for example it can return all and only correct answers to the certain kinds of query. Such an implementation can also be used with arbitrary RDF graphs.

Restricting the way in which constructs are used makes it possible to implement reasoning systems using rule-based reasoning engines, while still providing desirable computational guarantees. These restrictions are designed so as to avoid the need to infer the existence of individuals not explicitly present in the knowledge base, and to avoid the need for nondeterministic reasoning. This is achieved by restricting the use of constructs to certain syntactic positions.

The OWL2 RL is implemented in the OWLIM family triple stores [BigOWLIM2010], [SwiftOWLIM2010].

Semantic Web Rule Language (SWRL)

SWRL is a W3C submission for a rule language combining the sublanguages of OWL (OWL-DL and OWL-Lite) with the Unary/Binary Datalog RuleML sublanguages.

One of the goals of SWRL is to overcome the known limitations of ontology languages by adding the rules on the top of the ontologies. Anyway, the OWL-DL extension with the rules is, in general, undecidable, but decidable fragments are known (e.g. DL-safe rules).

The main strengths of SWRL are its simplicity and its tight integration with OWL. SWRL extends the OWL axioms with Horn-like rules combined with the knowledge-base. Rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. Usually, SWRL rules are part of an OWL ontology encoded in XML or in abstract syntax, which is quite hardly human readable. The XML concrete syntax is a combination of the OWL presentation syntax with the RuleML syntax. Translation from the XML syntax to

RDF/XML could be easily accomplished by extending the XSLT transformation for the OWL XML presentation syntax. The SWRL abstract syntax enables the definition of rules in a more human readable way.

SWRL does not support non-monotonic inference, but suitably-restricted SWRL rules can be straightforwardly extended to enable procedural attachments and/or non-monotonic reasoning (negation-as-failure and/or prioritised conflict handling).

SWRL is increasingly supported by DL reasoners, e.g. KAON2, Pellet, Racer-Pro, Hoolet or Boosam.

Web Service Modelling Language Rules (WSML-Rule)

WSML-Rule is the rule based sublanguage of Web Service Modelling Language[40] (WSML)specified as the formalization of Web Service Modelling Ontology (WSMO) [WSMO]. The rules are defined as the logical expressions in WSML. Basically, the logical expression syntax has its foundations in F-Logic, but uses slightly different language keywords.

WSML-Rule allows the unrestricted use of function symbols and no longer requires safety condition, i.e., variables which occur in the head are not required to occur in the body of the rule. WSML has an XML syntax for exchange between machines and an RDF syntax for exchange over the Semantic Web. In difference to SWRL, WSML-Rule allows for nonmonotonic negation. The WSML-Rule dialect is supported by the IRIS and MINS reasoners.

Web Rule Language (WRL)

WRL [WRL2005] is a rule-based ontology language for the Semantic Web [WRL2005]. The language is located in the Semantic Web stack next to the Description Logic based ontology language OWL. WRL defines three variants, namely Core, Flight and Full. The Core variant marks the common fragment between WRL and OWL. WRL-Flight is a Datalog-based rule language. WRL-Full is a full-fledged rule language with function symbols and negation under the Well-Founded Semantics.

WRL adheres to a conceptual model for ontologies, developed in the WSMO effort, which is independent of any logical language paradigm. The basic ontology meta-model of WRL consists of concepts, relations, instances, and axioms. The ontology vocabulary can be specified using WRL or OWL, or using their common semantic subset, denoted by the WRL-Core subset of WRL and the OWL-DLP subset of OWL. The common superset of WRL and OWL, here called "FOL++", might be a First-Order Logic with particular extensions to incorporate the non-monotonic features.

WRL extracts the rule-based variants of WSML, namely WSML-Flight and WSML-Rule, as well as the basic inter-operation layer with Description Logics, namely WSML-Core, leaving out the Web Service-specific elements, such as Goals, Web Services and Mediators (thus the rule example would look the same as in the case of WSML-Rules). WRL inherits from WSML the conceptual syntax for the specification of ontologies and the logical expression syntax for the specification of rules as part of an ontology. WRL has an XML exchange syntax which is based on RuleML.

Semantic Web Service Language Rules (SWSL-Rules)

SWSL-Rules are a logic-based language for specifying formal characterizations of Web service concepts and descriptions of individual services. It includes two sublanguages: SWSL-FOL - a full first-order logic language, which is used to specify the Semantic Web Service Ontology (SWSO), and SWSL-Rules - a rule-based sublanguage, which can be used both as a specification and an implementation language.

Generally, *the SWSL-Rules language* is designed to provide support for a variety of tasks that range from service profile specification to service discovery, contracting, policy specification, and so on. The language is layered to make it easier to learn and to simplify the use of its various parts for specialized tasks that do not require the full expressive power of SWSL-Rules. SWSL-Rules supports the negation as failure, non-monotonic reasoning and uses F-Logic syntax.

SWSL [SWSL2005] has not been directly implemented extensively yet. This is because the SWSL effort has to date focused mainly on requirements, specification, and use case scenario

[40] *http://www.wsmo.org/wsml/*

development. SWSL plans to embark upon more ambitious implementation efforts, largely using tools for RuleML and F-Logic, including notably SweetRules and Flora-2. SweetRules is the set of tools for semantic web rules and ontologies revolving around the RuleML, SWRL and OWL.

Extended RDF (ERDF) Rules

ERDF is an extension of RDF adding the support of negation and the rules [Wagner, 2008]. RDF does not support any negation concept, thus ERDF comes with the two types of negation:

- Negation as failure - also called weak negation, it is intended to decide if it is possible to prove a specific ground fact.
- Strong negation - is used to explicitly provide negative information in the knowledge base.

For more, ERDF adds the rule support to RDF. The rules can be defined in two ways: (1) an XML syntax expressed with the help of R2ML and (2) a non-XML syntax, based on Jena Rules syntax, which is extended with support to express the negation. The non-XML syntax for ERDF rules is an extension of Jena Rules syntax, adding support for expressing strong and weak negation. One of the limitations for this syntax is that it does not allow expressing disjunction. It is still possible to express disjunction if the respective formula is in disjunctive normal form by splitting that rule in many rules with the same head. If the formula is not in the disjunctive normal form it has to be normalized before splitting it.

The ERDF implementation is realized as the extension of Jena API[33].

5.5 Rule markup languages

RuleML

RuleML is a markup language developed to express both forward (bottom-up) and backward (top-down) rules in XML for deduction, rewriting, and further inferential-transformational tasks. It is defined by the Rule Markup Initiative, an open network to develop a canonical Web language for rules using XML markup and transformations from and to other rule standards/systems.

RuleML has defined several sublanguages, such as:

- Object-Oriented RuleML: the frame-like knowledge representation supporting the facts (instances) and rules (methods).
- ASP RuleML: support for answer-set programs in XML Schema. This variant facilitates the specification of a number of ASP-related constructs in a general manner. Moreover, it constitutes a base language for specific ASP extensions, such as HEX-programs.
- RDF: An experimental RDF translator for a subset of RuleML, available in XSLT
- RuleML Lite: developed basically as a RuleML subset compatible with RDF and OWL-DL that covers unary and binary Datalog facts, rules, and queries. The RuleML Lite design has interacted with the SWRL design via the joint committee.
- FOL (First Order Logic) RuleML: shares/reuses most of the earlier RuleML LP syntax, incorporating First-Order-Logic quantifiers and disjunctions as well as equivalence and negation. FOL RuleML strives for a strict separation of declarative content from procedural performatives.

R2ML

R2ML is a comprehensive and user-friendly XML rule format that allows interchanging rules between different systems and tools, enriching ontologies by rules, connecting the rule systems with R2ML-based tools for visualization, verbalization, verification and validation [R2ML].

R2ML is comprehensive in the sense that it integrates :

- the Object Constraint Language (OCL): an OMG standard used in information systems engineering and software engineering [OCL],

- the Semantic Web Rule Language (SWRL): a proposal to extend the Semantic Web ontology language OWL by adding implication axioms,
- the Rule Markup Language (RuleML)

R2ML is a *usable* language in the sense that it allows structure-preserving markup and does not force users to translate their rule expressions into a different language paradigm such as having to transform a derivation rule into a FOL axiom, an ECA rule into a production rule, a function into a predicate, or a typed atom into an untyped atom.

Notice that R2ML, like OCL and OWL/SWRL, provides a *rich syntax* for expressing rules supporting conceptual distinctions, e.g. between different types of terms and different types of atoms, which are not present in standard predicate logic. However, the user does not have to be familiar with all of R2ML's language elements in order to use it productively.

5.6 Systems integrating custom rules

Jena2

The Jena Semantic Web Framework implemented in Java contains a general purpose rule-based reasoner (in addition to RDFS, OWL, and transitive reasoner).

Custom rules can be defined above the RDFS or OWL ontologies. The rules can be extended with the built-ins, which are the predefined functions – procedural primitives supporting inference e.g. with comparison procedures, string manipulation, regular expression support, mathematical operations, or variable binding.

The rule reasoner can be also further extended by registering the new procedural primitives. The default set of procedural primitives support RDFS and OWL implementations and can be easily extended.

The rule-based reasoner works in the three modes:

- forward-chaining mode (using the RETE algorithm);
- backward-chaining mode (using the logic programming engine with the execution strategy similar to a Prolog engine);
- hybrid mode combining the forward and backward rules

The difference between the forward and backward chaining rules is just in using the direction of **->** operator.

AllegroGraph prolog rules

Prolog is an alternative query mechanism for AllegroGraph triplestore [Allegro], where queries can be specified in a declarative way. The AllegroGraph is implemented in Lisp language and the Prolog is an internal part of Lisp. The queries written in prolog can be directly integrated with the triplestore, by loading them into internal memory, or can be sent to the server via http client using special dedicated service.

Using of prolog rules in AllegroGraph is used as an alternative way of querying of the triplestore. The rules enable to write the queries in terms of higher level concepts specified by the rules. When some query refers to the new concept specified by the rule, the results of rule application are automatically retrieved. That means, the rules enable to specify concepts or relations additional to the meta-model, which can be used to extend the basic querying.

Rule engine uses the Prolog backward chaining engine.

6. Energy aware systems

6.1 Smart metering of devices

Smart metering usually refers to the adoption of various types of intelligent consumption meters in order to increase the transparency of energy consumption data, identify potential savings and enable automatic energy management solutions. For instance, smart meters can provide detailed information concerning the actual consumption and make it available for monitoring and billing purposes. Then, the same information can be made available to users in order to increase their awareness. In such a context, an Advanced Metering Infrastructure (AMI) represents the networking technology solutions enabling the bi-directional communication with smart meters. An example of a smart grid project is the Italian system installed by Enel S.p.A. The Telegestore project (National Energy Technology Laboratory 2007) is the first commercial scale use of smart grid technology to the home, and delivers annual savings of 500 million euro at a project cost of 2.1 billion euro. Another example, in Germany, the city of Mannheim installs small distributed generators that feed electricity to the grid. They utilize decentralized cooperation that fosters swarm behavior (Buchholz, Nestle et al. 2009). Moreover, In Germany utility providers such as YelloStrom and eOn have started to do test field with smart meter infrastructure for households. However, these meters only provide users with feedback of their energy consumption.

The Address Project[41] conducts research in the area of smart grids. One goal is the development of technologies for distributed control and real-time network management. The Address Project has a strong focus on enabling smart grid technology. This includes installing new equipment on the customer's side as well promoting distributed power generation and storage.

E-Energy[42] is a funding program from the German Ministry of Economics and Technology aiming at building an infrastructure for smart grid and smart metering technologies. Customers should be able to actively monitor and analyse their energy consumption and additionally receive information about real-time pricing information. This is achieved by enabling the aforementioned bidirectional communication technology. Furthermore, the need for managing distributed power generation is addressed by the consortium. Currently different technologies are tested in the field and being installed in real households.

Another project testing smart metering technologies in the field right now is SAVE@Work4Homes[43] a European project supported by the Intelligent Energy Europe program. Its goal is to reduce energy consumption for social housing in Europe by enabling smart metering technologies allowing tenants to monitor, analyse and control their energy consumption. Via Energy Awareness Services, users can access respective information e.g. over the Internet.

Other project targeting environmental sustainability, energy efficiency and new power distribution/generation business models is the BeyWatch[44] project funded by European Commission Seventh Framework Programme (FP7). BeyWatch aims to design, develop and evaluate an innovative, energy-aware and user-centric solution, able to provide intelligent energy monitoring/control and power demand balancing at residential area and office buildings.

6.2 Smart Meter Communication Technology

Commonly used technologies are POWERLINE and EIB/KNX (EN 50090,ISO/IEC 14543). Whereby POWERLINE is used to transport information across long distances in power grids, EIB/KNX tends on

[41] *http://www.addressfp7.org*
[42] *http://www.e-energy.de*
[43] *http://save.atwork4homes.eu*
[44] http://www.beywatch.eu

building automation as a basis of intelligent device control. EIB/KNX exists for more than 15 years and has a market share less than 5%. The reason for this is the usage of high-priced components and the additional installation of a required bus system wire.

LonWorks45 is a networking platform for building automation. The relevant communication protocol, twisted pair and power line signalling technologies, and IP tunnelling method have been recently included into the standard ISO/IEC 14908. A further standard communication solution for home and building automation is BACnet46 (ISO 16484-5).

Moreover, different advanced local wireless communication technologies are being increasingly adopted to enable the interconnection of objects like smart meters and actuators, involved in the definition of an energy management system.

While the use of a wired infrastructure is suitable for specific scenarios where direct access to the system is available (e.g. monitoring/metering of electrical loading at a power distribution panel), in other scenarios it could be not feasible (e.g. when retro-fitting existing old buildings). In this last case, the adoption of wireless technologies could represent a simpler and/or more cost effective solution. Both high data-rate solutions as e.g. Wi-Fi (IEEE 802.11) and low data-rate Wireless Sensor and Actuator Network (WSAN) solutions as e.g. IEEE 802.15.4(IEEE 2006), ZigBee(ZigBee Alliance and HomePlug Powerline Alliance liaison April 2010), Z-Wave47 and EnOcean48.

In particular, WSAN technology characteristics in terms of low-power operation, self-configuration and self-organization attitude specifically provide the means for the definition of energy-efficient and flexible network infrastructures for smart meters and actuators. In addition, ZigBee Alliance is particularly active in the field of smart metering as well as building automation: a specific ZigBee Application Profile is focused on smart energy and presents several advanced features for smart meters and smart appliances.

6.2.1 ZigBee Smart Energy Profiles

Within the ZigBee Alliance, two different application profiles have been specified to support energy-related features in WSANs.

The first one is called Smart Energy Profile 1.0 and is based on the updated version of ZigBee protocol released in October 2007 and the ZigBee Cluster Library (ZCL). The second application profile, called Smart Energy Profile 2.0 [ZigBee2010], is the novel smart energy specification which adapts the profile to run over IPv6-based networks. A brief description of both the considered profiles is provided in the following.

Smart Energy 1.0 adopts the ZigBee 2007 protocol version and extends the ZCL, a collection of pre-defined application messages (i.e. clusters), to include attributes and commands useful for applications aiming to provide energy consumption monitoring and to enable a more efficient energy usage. More specifically, the key relevant applications are related to metering, pricing, demand/response and load control. The profile is intended to be implemented in scenarios such as familiar homes or complex apartments, but can be adapted to any location. It is worth mentioning that security aspects have been carefully addressed within the profile: new advanced features based on elliptic curve cryptography have been introduced.

Smart Energy 2.0 is a standard being jointly designed by the ZigBee Alliance and the HomePlug Powerline Alliance49. Its main objective is to provide a networking and application layer platform supporting the interaction between customer devices and energy services providers, thus promoting the adoption of monitoring and control features to reduce global energy consumption. The profile is intended to run on any network relying on IPv6 protocol. More specifically, IEEE 802.15.4 has been considered together with the adoption of the IETF 6LoWPAN standard as adaptation layer.

[45] http://www.echelon.com/products/lonworks_platform.htm
[46] http://www.bacnetinternational.org/
[47] http://www.z-wave.com
[48] http://www.enocean.com
[49] http://www.homeplug.org/

At the moment, a first draft of a HTTP-based RESTful application protocol has been defined. Basically, a Smart Energy device can be considered as a server hosting application level capabilities exposed as Uniform Resource Identifiers (URIs). The actions (i.e. get, update, extend or delete) on such resources can be performed through the well-known HTTP methods GET, PUT, POST and DELETE. The data models used are defined in the International Electro technical Commission's (IEC) 61970-301 (IEC 2009) and 61968-11 (IEC 2009).

A more detailed introduction to Zigbee Smart Energy Profiles can be found in the following deliverables from the ebbits project:

- D5.1.1 Concepts and technologies in intelligent service structures 1
- D7.2 Event and data structures, taxonomies and ontologies

6.3 Energy-aware middleware

eDiana is an existing Artemis project targeting to increase energy efficiency of embedded devices. The eDiana project aims at achieving a reference model-based architecture based on the concept of cells (households) and macro cells (residential and non-residential buildings). Such cells can then be interconnected to form more complex networks of whole districts. Technically eDiana[50] aims at realizing these goals by developing an open middleware helping to integrate cells into existing power grids.

Sofia[51] is another Artemis project which is of relevance to energy efficiency. Sofia project is targeting to make "information" in the physical world available for smart services - connecting physical world with information world. Although the Sofia project is not targeting energy efficiency one of their applications is dealing with this. Sofia is advocating an ontology-based approach for automatic generation of device code. However, their main focus is mainly on powerful mobile devices like Nokia phones

Another project targeting at similar goals is the AIM project[52], funded by European Commission Seventh Framework Programme (FP7). The main objective is to develop technologies for managing energy consumption in domestic environments in real-time. Target groups are either power distribution network operators who monitor power consumption of larger residential areas or residential users who monitor and manage their home network. AIM distinguishes between the home and the outside network. Residential users administer their home networks while functionalities are exposed as services to the outside network via a gateway offering functions for policy management, device discovery, and proactive configuration.

ME3Gas[53] is an Artemis JU funded Project developing a middleware and applications for a smart metering infrastructure. In addition to providing energy saving technology, a main goal of the project is to raise awareness about the electricity and gas consumptions for both residential and commercial buildings, with a full savings potential in these areas estimated to be around 27 % and 30 % of energy use respectively. Furthermore the aim is to put the consumer in control to effortlessly optimize energy efficiency.

6.4 Energy Efficiency in Future Factories

[Karnouskos2009] envision energy efficient future factories. They claim the granularity of energy consumption information will go into embedded devices such as PLCs, mobile devices, and sensors so that each production process can be continuously monitored and dynamically improved. Continuous energy monitoring and controlling is enabled by web services for devices (DPWS) as addressed in Socrades Project. Web service in manufacturing environment allows service oriented architecture to be implemented in the shop floor and thus ease the integration of enterprise planning level into manufacturing operational. Energy efficiency will go beyond simple monitoring of

[50] http://www.artemis-ediana.eu/
[51] http://www.sofia-project.eu/
[52] http://www.aim-project.net/
[53] http://www.sics.se/projects/me3gas

machines. Correlation of context information to consumption data is needed to be able to identify the effectiveness of manufacturing processes.

Energy efficiency should also be implemented in the whole product lifecycle so that the consumers can also know the energy spent for producing the product. Being able to know energy needed at each step of a product lifecycle, a better strategy can be developed to optimize the corresponding processes (e.g.: layout of the machines, material processing order). Energy labelling such as found in electronic devices can be extended not only to rate their energy consumptions but also the energy needed to produce them in different processes such as manufacturing, logistic, etc.

A procedure for the energy/emission analysis of production processes of discrete manufacturing systems has been investigated [Devoldere2007], [Cannata2009]. They identify six main steps: objective definition, identification, evaluation, energy/emissions measurement, analysis, reaction. These steps are aligning with lean manufacturing concept that was developed by Toyota. In their case study, they examine energy consumption in different state of the manufacturing machines. And through the 6 steps defined before, they identified “worthless” energy consumption during idle, thus the reaction to this is to shut off the machines when they are not needed and when the idle consumption over period of time is bigger than the power needed to start up the machines.

7. Related on-going projects and clusters

ebbits exploits the knowledge gained from related research projects conducted previously by the ebbits partners. We also plan to collaborate and exchange ideas and knowledge with ongoing projects and participate in concertation activities and workshops organised by the commission. Some examples of ongoing research activities/projects that we have identified as relevant for ebbits are:

7.1 IERC-IoT – Cluster of European Research Projects on the Internet of Things[54]

The IERC cluster is bringing together a number of EU-funded projects and promotes a common vision of the Internet of Things. ebbits is a member of the cluster. Most of the projects in the IERC IoT cluster are obviously of interest to ebbits. Here we mention a few running FP6 and FP7 projects in the cluster that is relevant:

ASPIRE

ASPIRE, Advanced Sensors and lightweight Programmable middleware for Innovative RFID Enterprise applications, is researching an RFID middleware which can integrate with low-cost hardware as well as legacy IT-systems and network infrastructure. Their research on resource-constrained devices is of interest to ebbits, while our work on distributed intelligence and semantic knowledge infrastructures could help them extend their architecture.

IoT-A

IoT-A, the European Lighthouse Integrated Project addressing the Internet-of-Things Architecture, proposes the creation of an architectural reference model together with the definition of an initial set of key building blocks. Together they are envisioned as crucial foundations for fostering a future Internet of Things. Using an experimental paradigm, IoT-A will combine top-down reasoning about architectural principles and design guidelines with simulation and prototyping to explore the technical consequences of architectural design choices.

IMS2020

IMS2020 is a coordination action running until 2011 with a focus on Intelligent Manufacturing Systems. Especially one of their key areas – "Energy Efficient Manufacturing" is of interest to ebbits. We hope to be able to give input from ebbits´ user requirements and architecture design to this coordination action.

RACE networkRFID

RACE networkRFID is 3 year CIP-PSP project that started in 2009. The network considers its mission is to create opportunities and increase the competitiveness of European Member States in the area of RFID thought leadership, development and implementation. At the same time it will position RFID technology within the mainstream of information and communications technology (ICT).

SENSEI

Sensei develops holistic framework for the large scale deployment of interoperable wireless sensor and actuator networks (WS&AN), along with universal interfaces to access them.

7.2 Ongoing FP7 framework projects

Pobicos

Pobicos is an ongoing STREP project in FP7 with the goal to actively research and offer support for opportunistic pervasive computing applications by building a platform that enables the easy programming of partially unknown, heterogeneous object communities. Established links with the Hydra partners already exist so we will have good opportunities to liaise with Pobicos. From them we can obtain information about application development tools for opportunistic systems developed in

[54] http://www.internet-of-things-research.eu

POBICOS. We can share information about the distributed intelligence and event management being researched in ebbits.

CONET

CONET, Cooperating Objects NETwork of Excellence, is an FP7 network of excellence which aims at building a strong community in the area of Cooperating Objects. The network is working on raising awareness and creates a community of researchers to enable cooperation towards a sustainable architecture that is able to cope with the vision of Cooperating Objects. Obviously this is highly important to ebbits and we will seek to participate in the network.

IPAC

IPAC, Integrated Platform for Autonomic Computing, is an FP7 project aiming at delivering a middleware and service creation environment for developing embedded, intelligent, collaborative, context-aware services in mobile nodes. Of special interest to ebbits is their knowledge and ontology engineering techniques.

PECES

PECEC, Pervasive Computing in Embedded Systems, enables seamless cooperation of embedded devices across various smart spaces in a context-dependent, secure and trustworthy manner. Their work on context management is highly relevant for the ebbits project, and our partner Fraunhofer is also partner in the PECES project.

SM4ALL

SM4All aims at creating an embedded middleware platform for pervasive and immersive environments for all. The P2P architecture in combination with their work on context awareness makes it interesting for us to collaborate with them. Well-established links between ebbits partners and SM4All already exists through their use of the Hydra middleware.

7.3 Artemis Joint Undertaking

Artemis JU aims to tackle the research and structural challenges faced by the industrial sector. The objective is to define and implement a Research Agenda for Embedded Computing Systems. There are several ongoing projects within the Artemis Joint undertaking which are relevant for ebbits to collaborate with.

Sofia

Sofia is an Artemis project which is of relevance to ebbits. The Sofia project is targeting to make "information" in the physical world available for smart objects and services - connecting physical world with information world. Sofia is advocating an ontology-based approach for automatic generation of device code. With Sofia we can share knowledge about cross-industry interoperability and ebbits could contribute to open source software developed in SOFIA. SOFIA could utilize smart objects developed in ebbits.

Genesys

The objective of the Genesys project is to develop a cross-domain reference architecture for embedded systems that can be instantiated for different domains. Among other things the project is addressing power/energy efficiency in embedded system which is highly relevant for ebbits.

eDiana

eDiana is another ongoing Artemis project targeting the building sector with a focus on energy efficiency. The eDiana project aims at achieving a reference model-based middleware architecture based on the concept of cells (households) and microcells (residential and non-residential buildings). Their work on middleware implementation for energy efficiency makes them a good candidate to share knowledge with.

8. Conclusions

The developments in M2M should be closely followed. Given the involved of many large and important actors with an interest in this field we can expect several new applications, services and start-up companies to appear.

SoA-based Middleware still relevant

The SOA-based architecture selected in ebbits is still relevant. The middleware solution based on the results from Hydra, i.e. LinkSmart middleware, still represents the best and most efficient choice for the ebbits architecture. The support in LinkSmart middleware for using semantic technologies for sensor, device and service descriptions is in line with state-of-the-art, as well as the 3-layered discovery architecture pioneered in the Hydra project. In this way, we can manage "legacy" device both wired and wireless, but are flexible open to the new GSM-based mobile M2M devices.

Embedded OS to support IPv6 at sensor level

As for the lowest operating system level, we recommend Contiki and the use of IPv6. It is a very efficient embedded OS with strong backing in both industry and academia. IPv6 is also very most industrial developments are heading, for instance next generation ZigBee is based on IPv6.

Be practical about Semantics

In the last 10 years, there was a lot of effort in designing the several semantic standards, formalisms in various areas including the ontology design, web services semantic extensions, semantic support for rule-based reasoning, but also in the area of implementations of semantic web frameworks or the reasoners. There is a lot of existing standards, some in the phase of proposal, some having stable implementations, some still lacking in implementation.

On the one side, it is good to have the overview of existing technologies, as it can be very helpful when taking into account the requirements on the architecture design. On the other side, as many of semantic technologies are still evolving and lot of them are still in the phase of development, it is more reasonable to be practical and take into account the status of particular semantic technologies, having the features matching the ebbits requirements.

Look for efficient semantic storage

From the experiences from previous projects, the best point, where to start, when selecting the proper formalisms and the implementing technologies, is the current state of the art in the area of triple stores – the native semantic storages with efficient querying and reasoning capabilities. The reason is, that each triplestore has its own implementation of storage, the scope of language expressiveness, reasoning capabilities, querying and possibly the rule support. For example, the OWLIM [BigOWLIM, 2010] family of triple stores can best handle the complexity OWL-Lite dialect of OWL and supports the OWL2 RL rule language, where both of language formalisms used is the W3C standards. On the other side, AllegroGraph or Jena family triple stores can handle the full OWL language, but implement the custom rule languages, which are not standardized. Thus, the selection of the underlying triplestore will have the direct impact on the possibilities, what can be represented, queried and how.

The overview of selected perspective triple stores and their features in the relevant areas are more precisely summarized in deliverable D4.1 Analysis of Semantic Stores and Specific ebbits Use Cases.

Investigate federation techniques

There is also the possibility to combine the different technologies for different purposes and use so-called federation techniques, which are native to semantic web, when combining several technologies, but on the other side, this approach may increase the complexity of the whole architecture.

9. List of Figures and Tables

10. References

[Abangar2010] Abangar, H., Barnaghi, P., Moessner, K., Tafazolli, R., Nnaemego, A. & Balaskandan, K.(2010) A Service Oriented Middleware Architecture for Wireless Sensor Networks. Proceedings of Future Network & Mobile Summit 2010. Florence, Italy.

[BigOWLIM2010] OWLIM – Semantic Repository for RDF(S) and OWL, Ontotext AD, OWLIM Primer, 2010

[Blackstock2010] Michael Blackstock, Nima Kaviani, Rodger Lea, Adrian Friday: MAGIC Broker 2: An Open and Extensible Platform for the Internet of Things. Internet of Things (IOT), 2010, Tokyo, Japan, Pages: 1-8, 2010

[Cannata2009] Cannata, A., S. Karnouskos, et al. (2009). Energy efficiency driven process analysis and optimization in discrete manufacturing. Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE.

[Casimiro2007] Antonio Casimiro, Jörg Kaiser, Paulo Verissimo: Generic-Events Architecture: Integrating Real-World Aspects in Event-Based Systems. Architecting Dependable Systems Lecture Notes in Computer Science, 2007, Volume 4615/2007, Pages: 287-315

[Chatzigiannakis2007] Chatzigiannakis, I. Mylonas, G. Nikoletseas, S., "50 ways to build your application: A survey of middleware and systems for Wireless Sensor Networks" in the Proc. of IEEE Conference on Emerging Technologies and Factory Automation, 2007. ETFA.

[Compton2009] Compton, M., Neuhaus, H., Taylor, K., Tran, K., Reasoning about Sensors and Compositions. In Proceedings of the 2nd International Workshop on Semantic Sensor Networks (SSN09) at ISWC 2009, 2009.

[Devoldere2007] Devoldere, T., W. Dewulf, et al. (2007). "Improvement potential for energy consumption in discrete part production machines." Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses: 311-316.

[Eugster2003] Eugster, Felber, Guerraoui, Kermarrec, The Many faces of Publish/Subscribe, ACM Computing Surveys, Volume 35 , Issue 2 table of contents, Pages: 114 - 131, 2003

[Goodwin2006] Goodwin C., Russomanno, D.J., An Ontology-Based Sensor Network Prototype Environment, Fifth International Conference on Information Processing in Sensor Networks (Poster), IEEE, Nashville , 2006

[Guinard2010] Guinard, D., Trifa, V., Karnouskos, S., Spiess, P. & Savio, D. (2010) Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. IEEE Transactions on Services Computing, 3,3 223-235.

[Hashmi2005] Hashmi, N., Myung, D., Gaynor, M. & Moulton, S. (2005) A sensor-based, web serviceenabled, emergency medical response system. Proceedings of the 2005 Workshop on End-To-End, Sense-and-Respond Systems, Applications and Services.

[Henson2009] Henson, C., Pschorr, J. K., Sheth, A. P. & Thirunarayan, K. (2009) SemSOS: Semantic Sensor Observation Service. Proc. of the 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009). Baltimore, MD.

[Jacobsen2009] Hans-Arno Jacobsen, Vinod Muthusamy, and Guoli Li: The PADRES event processing network: Uniform querying of past and future events. it - Information Technology, 51(5)250-260, 5 2009.

[Kaiser2005] Kaiser, Jörg, Brudna Cristiano, Mitidieri Carlos, COSMIC: A real-time event-based middleware for the CAN-bus, Journal of Systems and Software - Special issue: Parallel and distributed real-time systems archive Volume 77 Issue 1, July 2005 Elsevier Science Inc. New York, NY, USA

[Karnouskus2009] Karnouskos, S., A. W. Colombo, et al. (2009). Towards the energy efficient future factory. Industrial Informatics, 2009. INDIN 2009. 7th IEEE International Conference on.

[OWL-S2004] Martin, D., et.al., OWL-S: Semantic Markup for Web Services, W3C Member Submission, 2004, available at: http://www.w3.org/Submission/OWL-S/

[Priyantha2008] Priyantha, B., Kansal, A., Goraczko, M. & Zhao, F. (2008) Tiny Web Services for Sensor Device Interoperability. Proceedings of the 7th international Conference on information Processing in Sensor Networks.

[Sheth2008] Sheth, A. P., Henson, C. & Sahoo, S. S. (2008) Semantic sensor web. IEEE Internet Computing,12,4 78-83.

[Shuoqi2004] Shuoqi Li, Sang H. Son, Stankovic, John A. Event detection services using data service middleware in distributed sensor networks, Proceeding IPSN'03 Proceedings of the 2nd international conference on Information processing in sensor networks, 2004.

[Spiess2009] Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., Souza, L. & Trifa, V. (2009) SOA-based integration of the internet of things in enterprise services. Proceedings of IEEE ICWS. Los Angeles, CA, USA.

[SWSL2005] Semantic Web Service Language - SWSL Battle et al, Member submission, 2005, available at http://www.w3.org/Submission/SWSF-SWSL.

[Urdiales2009] Urdiales, D., Heras, R. d. l., Narganes, M., Davies, M., Zhdanova, A. V., Christophe, B.,Maknavicius, L., Immonen, M., Heinil, J. & Barnaghi, P. (2009) m:Ciudad- Unleashing mobile userprovided services. Proceedings World Wide Web Conference (WWW2009). Madrid, Spain.

[Virtuoso2007] Blakeley, C., RDF Views of SQL Data (Declarative SQL Schema to RDF Mapping), OpenLink Software, 2007

[WRL2005] Web Rule Language - WRL Bruijn et al, Member submission, 2005, available at http://www.w3.org/Submission/WRL.

[WSDL-S] Web Service Semantics - WSDL-S Akkiraju et al, Member submission, 2005, available at http://www.w3.org/Submission/WSDL-S.

[WSMO2005] Lausen, H., et.al.,. Web Service Modeling Ontology (WSMO). W3C Member Submission, 2005, available at: http://www.w3.org/Submission/WSMO

[ZigBee2010] ZigBee Alliance and HomePlug Powerline Alliance liaison (April 2010) "Smart Energy Profile 2.0 Application Protocol Specification, ZigBee Smart Energy Profile 2.0 Technical Requirements Document."

Appendix: WSAN Products and Vendors

In this appendix we provide pieces of information concerning worldwide companies involved in Wireless Sensor and Actuator Networks (WSAN) and the commercial solutions they have proposed. A short discussion of products and their peculiarities is followed by the company presentation.

The list of the below mentioned companies is certainly not exhaustive mainly due to the exponential market growth we have been observing. The rapidly changeable market situation is responsible for the news included in this section becoming partially stale after short time.

Most of the pieces of information reported in this section have been collected on the Internet. The sequential order in which the companies are presented is not necessarily related to the relevance and the importance.

We structure the contribution into three main sections. Sections **Error! Reference source not found.** and 10.2 are intended to distinguish among companies selling uniquely transceivers and companies that commercialize complete solutions which integrate the many modules constituting a WSAN node. On the other hand, Section 10.3 is devoted to other companies with important expertise in the field of WSAN.

10.1 Transceiver Vendors

This section presents companies, along with related products, whose business is based on, or includes, the sale of wireless transceivers for WSAN technology.

10.1.1 Ember

Website: http://www.ember.com/

Founded in 2001, Ember is a Promoter of the ZigBee Alliance, with a seat on its board of directors. Ember's solutions are ZigBee compliant and the physical layer also conforms to the IEEE 802.15.4 standard.

Ember products are System-on-Chip (SoC) or Network Co-Processor Chips, based on ZigBee protocol stack, designed for vendors that want to integrate low-power, low-cost, mesh solutions in a single node.

10.1.2 Texas Instruments

Website: http://www.ti.com/

Texas Instruments (TI), another Promoter of the ZigBee Alliance, covers many different sectors in the ICT world market. A leading position in the production of semiconductors and MCUs is held as testified by two highly popular products: the TI MSP430 microcontroller and the CCxxxx family of wireless transceivers, integrated in many WSAN commercial platforms.

TI offers cost-effective low-power RF solutions for the ISM band, ZigBee/RF4CE and a variety of short-range applications in the sub-1 and 2.4 GHz frequency bands. The industry-leading CC2420 is well known since it was the first 2.4 GHz single-chip RF transceiver compliant with IEEE 802.15.4/ZigBee. The CC2520 represents TI's second-generation 2.4 GHz IEEE 802.15.4/ZigBee RF transceiver.

10.1.3 Freescale

Website: http://www.freescale.com/

Freescale, a very large company involved as Promoter in the ZigBee Alliance, is specialized in embedded semiconductor solutions and commercializes both RF transceivers and microcontrollers. In addition, comprehensive platforms combining proprietary hardware and software components are in Freescale portfolio. One of them is called ZRP-1 and includes the MC1319x family of transceivers and the HCS08 8-bit family of microcontrollers. The transceivers, MC13191, MC13192 and MC13193, work in the 2.4GHz frequency band and differ for their potentiality. The MC13191 can be considered for point-to-point and star topology applications and can be interfaced to a variety of Freescale microprocessors. Only SMAC (Simple Media Access Controller), the proprietary MAC protocol, supports this platform. The MC13192, in combination with an appropriate microcontroller and software tools, also supports a complete IEEE 802.15.4 standard for star and mesh networks. Finally, the MC13193 is the only one able to load the ZigBee protocol stack implementation.

10.1.4 Digi

Website: http://www.digi.com/

Digi International was founded in 1985 as DigiBoard and subsequently, in 1989, it went public as Digi International. It is leader in commercial grade device networking and its innovation mainly regards wireless machine-to-machine (M2M) device networking products.

Digi Drop-In Networking solutions provide end-to-end wireless access to electronic devices in places where wires will not work or cannot be used, making it easy for customers to effectively “drop-in” a wireless M2M solution.

10.1.5 STMicroelectronics

Website: http://www.st.com/internet/com/home/home.jsp

STMicroelectronics (STM) is a world leader in semiconductors. Its product portfolio includes solutions for ZigBee large-scale mesh networks and wireless tracking networks. STM proposes a complete family of products with ZigBee networking technology and its IEEE 802.15.4 radio allow large wireless network deployments and interoperability thanks to its open standard specifications. The offer consists of hardware ICs, software libraries, and development tools.

10.1.6 Atmel – MeshNetics - LuxLabs

Website: http://www.atmel.com/

In 2009 Atmel acquired all MeshNetics ZigBee Intellectual Property Rights from LuxLabs (Meshnetics). Thanks to the acquisition, which included BitCloud ZigBee PRO software and ZigBit wireless modules, Atmel can now offer a complete wireless solution to electronics equipment manufacturers. It covers 700/800/900Mhz frequencies through to 2.4 GHz, boasting a link budget of over 120dB.

The BitCloud ZigBee PRO software stack is a full-featured, second generation embedded ZigBee certified stack for reliable, scalable, and secure wireless applications running on Atmel wireless platforms. The stack offers support for large wireless ZigBee networks consisting of hundreds of devices that is optimized for ultra-low power consumption and is provided with an application programming interface (API) for easy customization by OEMs and system integrators.

ZigBit, a ZigBee low-power and high-sensitivity module based on both MCU (ATmega 1281v) and transceiver (AT86RF230) from Atmel, has a strongly reduced size, less than a square inch of space, in order to build networks with mesh topologies based on MeshNetics stack profile.

10.1.7 Jennic

Website: http://www.jennic.com/

Founded in 1996, Jennic has joined the ZigBee Alliance with the role of Participant. It supplies integrated silicon chips and software to designers and manufacturers of equipment for home automation, commercial building automation and industrial process monitoring. Jennic platform combines a 2.4GHz IEEE 802.15.4 radio transceiver, the low-power IEEE 802.15.4 compliant JN5121 microcontroller, 64kb of ROM and 96kb of RAM. The current drainage of the radio chip is not particularly attractive from the energy consumption point of view: 50mA in receive state, 40mA in transmission mode and a sleep current of 5µA.

10.1.8 Microchip Technology

Website: http://www.microchip.com

Microchip Technology Inc. is a provider of microcontroller and analogue semiconductors.

Microchip radio frequency products target the unlicensed ISM frequency bands. They include the 2.4 GHz IEEE 802.15.4 radio transceiver (MiWi and MiWi P2P solutions) and the sub-GHz transmitter, receiver, and transceiver solutions in the 300-900 MHz frequency range. Microchip also provides the MiWi Development Environment, a proprietary wireless solution to develop wireless applications. The MiWi Development Environment (MiWi DE) package includes support for Microchip proprietary protocols – MiWi Mesh and MiWi P2P – and is optimized for low-power, low-data rate, cost sensitive application.

10.2 Modules Vendors

This section focuses on companies that sell complete wireless modules falling in the WSAN technology.

10.2.1 Zolertia

Website: http://www.zolertia.com/

The main site of Zolertia, a company focused on the most advanced WSAN hardware platforms, is located in Spain.

The last WSAN module developed by Zolertia, called the Z1 module, is a general purpose development platform (very similar to CrossBow Telos, see Section 10.2.8: both integrate CC2420 with MSP430) presenting a low-power 802.15.4/6LoWPAN/ZigBee compatible radio and a wide range of easily pluggable sensors, these being two very attractive features. Indeed, there are not yet a lot of 802.15.4/6LoWPAN/ZigBee platforms available on the market.

10.2.2 Telit Wireless Solutions

Website: http://www.telit.com/

Telit Wireless Solutions deals with Machine-to-Machine (M2M) communications. The company develops, manufactures and markets M2M modules which enable machines, devices and vehicles to communicate via wireless networks, as well as a wide range of communication modules in the sector of cellular technologies, but also in short-range technologies such as Wi-Fi modules and ZigBee. In 2008, Telit moved from a "cellular M2M" solutions provider to a "wireless M2M" solutions provider, including Short Range RF technologies in its product offering.

10.2.3 Libelium

Website: http://www.libelium.com/

Like Zolertia, Libelium is a Spanish company, founded in 2006. It designs and manufactures hardware for the implementation of wireless sensor networks, mesh networks and communication protocols for all sorts of distributed wireless networks.

Libelium's main products are Waspmote, a low-power consumption sensor device originated by the EU FP6 project WASP and compliant with ZigBee standard; Meshlium, a router integrating WiFi mesh (2.4GHz - 5GHz), ZigBee, GPRS, GPS and Bluetooth technologies in a single unit; and, finally, N-vio, a proximity marketing and message sending platform via Bluetooth. The distinguishing features of Libelium platform are the multi-radio communication, the availability of multiple sensors (for instance, the ones suitable for weather, pollution and agricultural purposes) and accessories that can be plugged (including RFID readers).

10.2.4 Sensinode

Website: http://www.sensinode.com/

In the WSAN domain Sensinode commercializes the K320 platform. Beside ordinary sensor nodes the solution includes one NanoRouter that basically consists of an embedded PC with ARMLinux and Ethernet. The device includes a MSP430 family microcontroller (specifically, F54xx), a sensor of light, two buttons for external commands, two leds, external input/output ports and a USB interface. Referring to the IoT paradigm, it is worth noting that Sensinodes nodes are equipped with a 6LowPAN stack called NanoStack.

10.2.5 Arduino

Website: http://www.arduino.cc/

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language and the Arduino development environment. Arduino projects can be stand-alone or they can communicate with software on running on a computer.

10.2.6 STMicroelectronics

Website: http://www.st.com/internet/com/home/home.jsp

Already presented as one of the biggest companies in semiconductor and transceivers production, STM is also active in the development of WSAN modules. Based on ZigBee stack, SPZBxxx family they is designed for low-power consumption purposes and optimized for embedded applications. In addition, it is worth mentioning the STM32W platforms, a System on Chip (SoC) solution based on CORTEX-m3 and integrating accelerometer and temperature sensors. A Contink demo running 6LowPAN is included but not officially released and supported.

10.2.7 Dust Networks

Website: http://www.dustnetworks.com/

Dust Networks was founded in 2002 in California. Development of industry standards for the full and broad adoption of WSN technology is a company mission. In this light, Dust Networks is a member of several industry standards groups such as ISA, the HART Foundation, the ZigBee Alliance and WINA.

10.2.8 CrossBow – Memsic

Website: http://www.xbow.com/ http://www.memsic.com/

This company, founded in 1995 and headquartered in San Jose, California, is very well known because it was the first to start a profitable cooperation with the University of California at Berkeley for the commercialization of Smart Dust sensor nodes. The characterizing feature and, at the same time, the main added value of Crossbow's products (now shared with Zolertia) is the platform open architecture, based on TinyOS operating system. In 2009 Crossbow sells its WSAN product lines to Memsic.

Memsic wireless sensors provide advanced monitoring, automation and control solutions for a range of industries. The applications for wireless sensor networks are almost limitless with many industries and applications having specific technology requirements such as reliability, battery-life, range, frequencies, and topologies, size of the network, sampling rate and sensor use.

10.2.9 Coronis

Website: http://www.coronis.com/

French company founded in 2000, Coronis Systems became a subsidiary of Elster Group GmbH in 2007.

Coronis created Wavenis, which includes Wavenis RF transceiver and wireless communication protocol. Its Wavenis devices are integrated into systems with fixed network and mobile usage models, Ethernet, and Internet services. The company offers wireless products and development platforms for creating custom devices for system integrators in various markets. Its products are used in sensor monitoring and data collection process, smart water and energy meter management, home comfort, building and industrial automation, home healthcare and monitoring, security and control, and environment and temperature monitoring applications. The company has served remote water and energy meter monitoring, home comfort, home and building automation, healthcare, industrial automation, building management, access control, cold-chain management, and UHF RFID markets for identification, tracking, and locating people and objects.

10.2.10 EKA Systems

Website: http://www.ekasystems.com/

EKA Systems was founded in 2000 to provide reliable, Internet-enabled, wireless device networking technology for monitoring, control, and automation applications.

EKA Systems technology allows connecting electric, water, and gas meters and control devices into a single, efficient data network in a reliable way, in order to improve utility operations while protecting the environment and conserving valuable resources.

10.2.11 Texas Instruments

Website: http://www.ti.com/

The presence of Texas Instruments, one of the Promoters of the ZigBee Alliance, in the ICT world market covers many different sectors. Concerning the embedded system field, TI holds a leading position also in modules production.

The CC2430 is the first true System-on-Chip ZigBee solution while CC2431 is the first system-on-chip (SoC) with a hardware location engine. The CC2480 is the first product from TI's new Z-Accel family of ZigBee-certified network processors that simplifies design and reduces time-to-market.

Finally the latest addition to ZigBee products family is the already mentioned second-generation 2.4 GHz CC2530 System-on-Chip suitable for RF4CE and Smart Energy applications with up to 256 KB flash memory.

10.2.12 Freescale

Website: http://www.freescale.com/

From 2009, Freescale has put on the market the Xtrinsic family product. This new brand of sensors is designed with a combination of integration, logic and customizable software on the platform to deliver smarter, more differentiated applications.

Freescale sensors that are classified under the Xtrinsic brand exhibit integrated algorithms or are integrated platforms with multiple sensors and a processor that provides a high degree of contextual awareness and decision making. Xtrinsic sensing solutions represent products across the automotive, consumer, medical and industrial markets.

10.2.13 Helicomm

Website: http://www.helicomm.com/

It is a privately held company founded in 2002. Participant to the ZigBee Alliance, it provides wireless networking platforms for monitoring and control applications and carries on partnerships with microcontroller and radio transceiver manufacturers who license Helicomm's design and software.

Helicomm provides both embedded modules to develop generic applications, as well as tailored "application specific" modules, such as smart metering, environment sensors, etc.

10.2.14 Watteco

Website: http://www.watteco.com/

Watteco is a French company providing smart energy hardware and software solutions for smart grid deployment. After being a vendor of ultra-low Power Line Communication (PLC) modem solutions dedicated to the smart home control and energy efficiency emerging markets, now Watteco family of PLC chips have been introduced in the smart grid market by facilitating the integration of low rate PLC solution into an IEEE 802.15.4 and IPv6 Internet Protocol network. The use of Watteco technology facilitates the interoperability of wireless and wire-line equipment.

Watteco proposes both OEMs to create networks of smart energy devices and smart energy software to create monitoring solutions.

10.3 Other Company Profiles

In this section companies that have raised important expertise in the field of WSN and provide solutions at different stage of the WSN-based service development are listed. For instance, companies involved into developing commercial software solutions for WSN, as well as developing turnkey system solutions for selected fields, can be found.

10.3.1 E-Senza

Website: http://www.e-senza.com

E-Senza is a developer and manufacturer of wireless device networking products and solutions for connectivity among sensors, instruments, actuators, meters and systems. E-Senza wireless data infrastructure products address process industry, industrial automation, building automation, energy management, environmental monitoring, as well as logistics applications.

E-Senza proposes wireless device networking adapters with interfaces to sensors and field devices as well as gateways to connect to fieldbus systems. E-Senza also provides protocol stacks, hardware modules, and engineering services for integration. E-Senza developed a framework called SenzaNET, that is standard-based (on IEEE 802.15.4, WirelessHART and 6LoWPAN), low-power and with low latency (due to the use of a precise time synchronization algorithm), robust and secure.

10.3.2 Tridium

Website: http://www.tridium.com/

Tridium is involved in open platforms, application software frameworks, automation infrastructure technology, energy management and device-to-enterprise integration solutions. Tridium software frameworks extend connectivity, integration and interoperability to the millions of devices deployed today and empowers manufacturers to develop intelligent equipment systems and smart devices that enable collaboration and communication between the enterprise and edge assets.

10.3.3 Software Technologies Group

Website: http://www.stg.com

Software Technology Group (STG) develops software protocol for manufacturers of industrial wireless product developers, compliant with IEEE 802.15.4, ZigBee and WirelessHART.

More recently STG has developed expertise and partnerships to provide also hardware and turnkey systems solutions for end customers. Some examples of what has been developed are a ZigBee wireless fire extinguisher and an IEEE 802.15.4 wireless networked storage system.

10.3.4 Millennial Net

Website: http://www.millennialnet.com

Founded in 2000 by two MIT researchers, Millennial Net develops wireless sensor networking software, systems, and services that enable OEMs and systems integrators to implement wireless sensor networks. Millennial Net has developed a WSAN platform, called MeshScape 5, with proprietary protocols to manage routing, delivery reliability, responsiveness. MeshScape 5 networking utilizes the IEEE 802.15.4 unlicensed radio band or other types of radios including 433MHz and 900MHz ISM band.

Some of MeshScape 5 innovations are: mesh nodes and coordinator enabled to run at low power for battery operations over several years; active frequency hopping to avoid potential interferences for robust operation in highly noisy environments; low-latency capability to reduce the end–to-end packet delivery time to below seconds within a multi-hop environment, even with battery operated mesh node. The platform is independent in order to adapt to customer-preferred hardware platforms. Finally, Millennial Net products are appropriate for remote monitoring of industrial plants.

The network scalability is indicated as the main performance feature, making it possible to deal with applications and topologies involving about a hundred nodes. The network works at both 900MHz and 2.4GHz frequency bands.

10.3.5 Semtech

Website: http://www.semtech.com/

Founded in the '60s, the core activity of Semtech regards sensor interfacing/data acquisition, 8-bit microcontrollers for embedded systems, radio transceivers and audio codecs. In 2005, Semtech acquired Xemics, a company involved in ultra-low power analogue, radio frequency and digital integrated circuits. This has favoured the growth of Semtech in the wireless and sensing fields, with ultra-low power devices as short range highly integrated RF transceivers. This background allowed the realization of applications as home and building automation (wireless access control, wireless alarm, wireless light control, etc.), remote metering and control, GPS positioning for asset tracking, hearing aids and wireless headsets.

10.3.6 Cisco - Arch Rock Corporation

Website: http://www.cisco.com/

Cisco recently acquired Arch Rock Corp (September 2010) and its wireless sensor networking technologies. Arch Rock was founded in 2005 inheriting the ten-year research experience carried out at the University of California at Berkeley and at Intel Research. Indeed, David Culler and other researchers who developed TinyOS (the well-known open-source operating system for embedded systems) and realized the Berkeley Mote are among its founders.

10.3.7 MolToSenso

Website: http://www.moltosenso.com/

MolToSenso is a hi-tech Italian firm with a specific focus on wireless networks of cooperating objects, consisting of heterogeneous wireless sensors and actuators nodes.

MolToSenso has developed turnkey systems as well as solutions to be integrated with third parties equipment. Systems are originally designed by MolToSenso (at hardware, firmware and software level) in order to provide, in a modular fashion, customized solutions for customers with heterogeneous needs. The application areas are structural monitoring, environmental and building automation (either home or hotel), energy monitoring, industrial processes, logistics, distributed security, e-health, and so on.

Olkas is the basic module that integrates the 32 bits ARM Cortex-M3 microcontroller, an SD card slot, a real time calendar and a low-power IEEE 802.15.4 based (at 868 MHz) transceiver with MolToSenso proprietary mesh routing algorithm designed on top, or, alternatively, a low-power transceiver with 2.4GHz ZigBee Pro stack protocol. Different power sources can be plugged to Olkas (solar panel, USB, mains, etc.). A multiplatform software (running on Microsoft Windows, Linux and Mac OS X) allows the user to configure and manage the network remotely, thus logging on a PC the data monitored from the environment.

10.3.8 C-Labs

Website: http://www.c-labs.it/

C-Labs started its activities as a consulting company, then became a company specialized in hardware and software design for WSANs. In particular, they developed WINE, a proprietary protocol stack, able to connect, with a star topology, different low-power nodes. This protocol can be used in different applications, as telemetry, security, and home automation.

10.3.9 Zensys - Sigma Designs

Website: http://www.sigmadesigns.com/

Sigma Designs, through its acquisition of Zensys, provides one of the key enabling technologies of the intelligent home, the Z-Wave wireless mesh network ecosystem, a proprietary protocol for WSANs, alternative to ZigBee and 6LoWPAN. Sigma Designs offers a family of low-cost, low-power integrated MCU/transceiver chips, embedded with Z-Wave, along with a suite of development tools and services that companies can use to create wireless products and solutions for residential and light commercial applications.

Sigma Designs has developed a fully integrated RF communication module that uses the unlicensed Short-Range Device (SRD) frequency band of 902MHz-928MHz in the US and 868.0-868.6MHz in Europe and a proprietary patented protocol stack.

This solution enables also wireless control in residential and light commercial environments, as the new IR (Infra Red) replacement.

10.3.10 UniBand Electronic Corporation

Website: http://www.ubec.com.tw/

As many other new companies involved in electronic components and embedded systems, UniBand Electronic Corporation (UBEC), from Taiwan, joined the ZigBee Alliance and developed ZigBee compliant products, belonging to the U-series and ZG-series. UBEC develops single-chip solutions, integrating wireless radio transceiver operating at 2.4 GHz, PHY layer baseband and MAC layer architecture. They are controllable by various MCUs such as 8051 to apply in low-rate wireless applications that include home automation, consumer electronics, toys, industrial automation.

10.3.11 Dash7 Companies

Dash7 (http://www.dash7.org) is both a wireless sensor networking and a localization technology using the ISO/IEC 18000-7 open standard for low-power radio frequency devices operating in the 433 MHz unlicensed spectrum, available for use worldwide.

The original ISO 18000-7 standard was ratified in 2004 for localization and goods tracking. In 2009, the Dash7 Alliance, a non-profit industry consortium interested in promoting interoperability among Dash7-compliant devices, was formed to advance the use of DASH7 wireless data technology by developing extensions to the ISO 18000-7 standard, ensuring interoperability among devices and educating the market about Dash7 technology beyond the localization technology, which is the native area of interest of the ISO 18000-7 standard.

Many companies, members of the Dash7 Alliance, have started producing Dash7-compliant hardware products.

10.3.12 Confidex

Website: http://www.confidex.fi/

Confidex is an important supplier of high-performing RFID tag solutions and services in order to make supply chains, transactions and authentication of goods and people more efficient and secure. It serves customers representing a broad range of industries in Europe, North America and Asia.

10.3.13 Evigia Systems

Website: http://www.evigia.com/

Evigia is one of the industries involved in exploiting integrated sensor and ASIC (application specific integrated circuit) technologies to improve the functionality and cost of wireless and sensing products. These advances allow better performances and lower costs of asset-management supply chains. The network functionality, visibility, and security control are increased too, while the underlying hardware products themselves benefit from smaller size, lower power consumption, and lower cost.

10.3.14 Identec Solutions

Website: http://www.identecsolutions.com/

Identec Solutions technology offers tracking solutions that manage critical process and optimize supply chain flow. Privately held since 1999, Identec Solutions has provided asset-management solutions and support to many organizations.

10.3.15 RFind Systems

Website: http://www.rfind.com/

RFind Systems' Tag To Tag Communication technology deals with industrial manufacturers' needs in terms of logistics and warehouse asset management. Deriving location data from multiple sources, it provides point-to-point, x-y coordinates, or choke point locating. It can be used in industrial environments where metal, open spaces, bandwidth conflicts and interfering radio frequency signals are typical issues.

10.3.16 Savi Technology

Website: http://www.savi.com/

Savi Technology is a provider of smart asset management solutions and services for public sector and commercial supply chains worldwide, leveraging low-power wireless sensor networks for real-time information to optimize management, effectiveness, security, and profitable return of assets.

10.3.17 Sirma Technology

Website: http://syrmatech.trustpass.alibaba.com/

Syrma Technology is part of the Tandon Group of Companies, dealing with electronic manufacturing and outsourced design services. Syrma has built a reputation of providing electronic manufacturing services to original equipment manufacturers (OEMs) across several industry segments.

10.3.18 Udea Wireless Technologies

Website: http://www.udea.com.tr/

Udea is a developer of RF receiver, transmitter and transceiver modules, as well as complete wireless RF modules for ISM band wireless applications.

10.4 Other Companies

Just to give an intuition of the large number of companies currently involved in the embedded systems market, here is a brief list of others that we have not mentioned yet. **Nordic Semiconductor** is a semiconductor company specializing in design solutions at the microchip level in the areas of shot-range wireless radio communication. The Spanish **Atalum** (networking software), the British **Telegesis** (ZigBee manufacturer), the American **Tendril Networks** (smart grid solutions) and the South-Korean **Radiopulse** are several among many others that joined the ZigBee Alliance and started to develop ZigBee WSAN platforms.