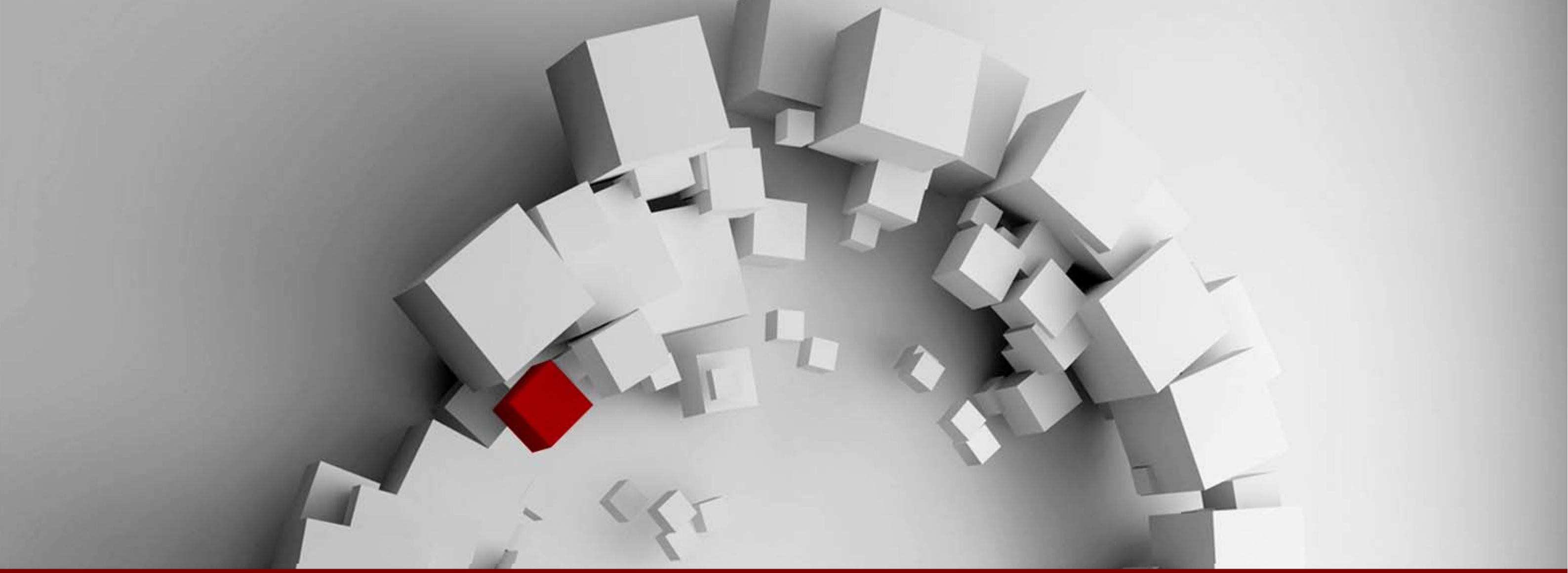

Introduction to the Architectural Reference Model for the Internet of Things

This Introduction to the Architectural Reference Model (ARM) for the Internet of Things which is currently developed by the project partners of the European FP7 Research Project IoT-A shall give the reader a first glimpse into the concepts of the ARM, its origin and its goals. Furthermore, this introduction shall serve as guidance for reading the actual documents issued by the IoT-A project as deliverables. All documents and deliverables referred to in the subsequent pages are publicly available and can be downloaded at http://www.iot-a.eu/arm.

Executive Summary

Today, "Internet of Things" (IoT) is used as a catchphrase by many sources. This expression encompasses a galaxy of solutions somehow related to the world of intercommunicating and smart objects. These solutions show little or no interoperability capabilities as usually they are developed for specific challenges in mind, following specific requirements. Moreover, as the IoT umbrella covers totally different application fields, development cycles and technologies used vary enormously, thus implementing vertical solutions that can be labelled as "INTRAnet of Things", rather than "INTERnet of Things". For instance, in some fields such as manufacturing and logistics, communication and tagging solutions are well established as they provide a clear business benefit in terms of asset tracking and supply-chain management. However, the same solutions do not apply for other fields such as domotics, where business synergies could provide services with clear added-value benefits.

While quite logical at this point, on the long run we believe that this situation is unsustainable. As in the networking field, where several solutions emerged at his infancy to leave place to a common model, the TCP/IP protocol suite, the emergence of a common reference model for the IoT domain and the identification of reference architectures can lead to a faster, more focused development and an exponential increase of IoT-related solutions. These solutions can provide a strategic advantage to mature economies, as new business models can leverage those technological solutions providing room for economic development.

Leaving aside business considerations, and considering only the technical point of view, the existing solutions do not address the scalability requirements of a future IoT, both in terms of communication between and the manageability of devices. Additionally, as the IoT domain comprises several different governance models, which are often incompatible. This leads to a situation where privacy and security are treated on a per-case and per-legislation basis, retro-fitting solutions to existing designs, and this severely hampers portability, interoperability and deployment.

In our vision of the Internet of Things, the interoperability of solutions at the communication level, as well as at the service level, has to be ensured across various platforms.

This motivates, first, the creation of a **Reference Model** for the IoT domain in order to promote a common understanding.

Second, businesses that want to create their own compliant IoT solutions should be supported by a **Reference Architecture** that describes essential building blocks as well as design choices to deal with conflicting requirements regarding functionality, performance, deployment and security. Interfaces should be standardised, best practices in terms of functionality and information usage need to be provided.

The central choice of the IoT-A project was to base its work on the current state of the art, rather than using a clean-slate approach. Due to this choice, common traits are derived to form the base line of the ***Architectural Reference Model*** **(ARM)**. This has the major advantage of ensuring backward-compatibility of the model and also the adoption of established, working solutions to various aspects of the IoT. With the help of end users, organised into a stakeholders group, new requirements for IoT have been collected and introduced in the main model building process. This work was conducted according to established architecture methodology.

Objectives and Outline of the current version of the ARM

The previous version v0.9 of the ARM was published approximately one year ago as project deliverable D1.2, and presented to a large audience during the IoT week 2011 in Barcelona. As a result we received a large number of comments, the majority of them being taken into account already in the new version v1.5 of the ARM.

While the general objective of v1.5 is the same as it was for v0.9, i.e., describing thoroughly an Architectural Reference Model for IoT, this version of the ARM brings to the audience substantial improvements over the previous version, as summarised below:

- All feedback received internally from IoT-A and externally from the stakeholders was taken into account in order to improve the document and in order to make sure that the IoT-A architecture work will eventually meet expectations from the external users;
- Introduction of new views and perspectives beyond the Functional Decomposition view and Security Perspective touched in v0.9. V1.5 comes with the Deployment & Operation and Information views and with the Evolution & Interoperability, Performance & Scalability and Availability & Resilience perspectives. It will be shown in the document how the Design Choices applied at the view levels impact the various quality properties attached to the system architecture materialised by the four perspectives introduced above;
- First version of Best Practices and associated Design Choices which are an initial step towards an aided architecture design for concrete system architects;
- Improvement of the soundness of the whole ARM approach, emphasizing the logical links existing between the various elements and sub-models of the Reference Model and the views and perspectives of the Reference Architecture.

Document structure

This document just gives an introduction to the ARM. It first explains the vision and rationale behind it, as well as the benefits of using the ARM. There-after the process and the methodology used to develop the ARM, as well as how the ARM should be applied when developing concrete systems, is described.

The final section then highlights some of the main business scenarios where the application of the ARM is beneficial, thus also validating the usefulness of creating an Architectural Reference Model for the Internet of Things.

The full ARM is downloadable from http://www.iot-a.eu/arm.

Introduction to the ARM – Vision

Many popular "umbrella" topics like Smart Cities pull a large number of specific application domains like Transportation, Energy, Environment, Assisted Living, most of the time pre-fixed with "Smart" sometimes for obvious marketing reasons but also -more generally- in order to emphasise the fact they embed a certain degree of intelligence and global awareness. This new breed of applications exploits IoT related technologies, however, the resulting applications unfortunately appear as vertical silos only, meaning specific applications with specific architectures, with little place left for inter-system communication and inter-operation. Actually that is where the real issue lies: the smartness of those new applications can only reach its pinnacle if full collaboration between those vertical silos can be achieved.

If we consider also the fact that IoT related technologies come with a high level of heterogeneity, with specific protocols developed with specific applications in mind, it is no surprise that the IoT landscape nowadays appears as highly fragmented. Many IoT-enabled solutions exist with recognised benefits in terms of business and social impact, however they form what we could call a set of **Intra**nets of Things, not an **Inter**net of Things!

In the vision of the Internet of Things IoT-A wants to promote, a high level of interoperability needs to be reached at the communication level as well as at the service and the information level, going across different platforms, but established on a common grounding. The IoT-A project reckons that achieving those goals comes in two steps, first of all in establishing a common understanding of the IoT domain (hereafter called Reference Model), and second in providing to IoT system developers a common foundation for building interoperable IoT system architectures (hereafter called Reference Architecture).

A *Reference Architecture* (RA) can be visualised as the "Matrix" that eventually gives birth ideally to all concrete architectures. For establishing such a Matrix, based on a strong and exhaustive analysis of the State of the Art, we need to envisage the superset of all possible functionalities, mechanisms and protocols that can be used for building such concrete architecture and to show how interconnections could take place between selected ones (as no concrete system is likely to use all of the functional possibilities). Giving such a foundation along with a set of design-choices, based on the characterisation of the targeted system w.r.t. various dimensions (like distribution, security, real-time, semantics,…) it becomes possible for a system architect to select the protocols, functional components, architectural options, … needed to build their IoT systems.

The main aim of IoT-A can be explained using the pictorial representation shown below.

As any metaphoric representation, this tree does not claim to be fully consistent in its depiction; it should therefore not be interpreted too strictly. On the one hand, the roots of this tree are spanning across a selected set of communication protocols (6LoWPAN, Zigbee, IPv6,…) and device technologies (sensors, actuators, tags,..) while on the other hand the blossoms / leaves of the tree represent the whole set of IoT applications that can be built from the sap (i.e., data and information) coming from the roots. The trunk of the tree is of utmost importance here, as it represent the Architectural Reference Model (ARM). The ARM is the combination of the Reference Model and the Reference Architecture, the set of models, guidelines, best practices, views and perspectives that can be used for building fully interoperable concrete IoT architectures and systems. In this tree, we aim at selecting a minimal set of interoperable technologies (the roots) and proposing the potentially necessary set of enablers or building blocks (the trunk) that enable the creation of a maximal set of interoperable IoT systems (the leaves).

The ultimate aim of the Reference Architecture

Figure 1: The IOT-A Tree

work is to make sure that concrete system designers will eventually use it. High attention is therefore paid to ensuring the soundness of our work. In particular this version of the ARM aims at making more explicit the various links existing between the various models, views and perspectives, so that it will make the work of systems designers easier.

The ARM Rationale

Figure 2 shows an overview of the process we used for defining the different parts that make the IoT-A ARM. Notice that definitions of terms such as reference architecture, etc. can be found in an external glossary (see www.iot-a.eu/public/terminology). Starting with existing architectures and solutions, generic baseline requirements can be extracted and used as an input to the design. The IoT-A ARM consists of four parts:

- The **vision** summarises the rationale for providing an architectural reference model for the IoT. At the same time it discusses underlying assumptions, such as motivations. It also discusses how the architectural reference model can be used, the methodology applied to the architecture modelling, and the business scenarios and stakeholders addressed.
- **Business scenarios defined as requirements by stakeholders** are the drivers of the architecture work. With the knowledge of businesses aspirations, a holistic view of IoT architectures can be derived. Furthermore, a concrete instance of the reference architecture can be validated against selected business scenarios. A stakeholder analysis contributes to understanding which aspects of the architectural reference model need to be described for the different stakeholders and their concerns. According to common usage, this part constitutes a subset of the vision.

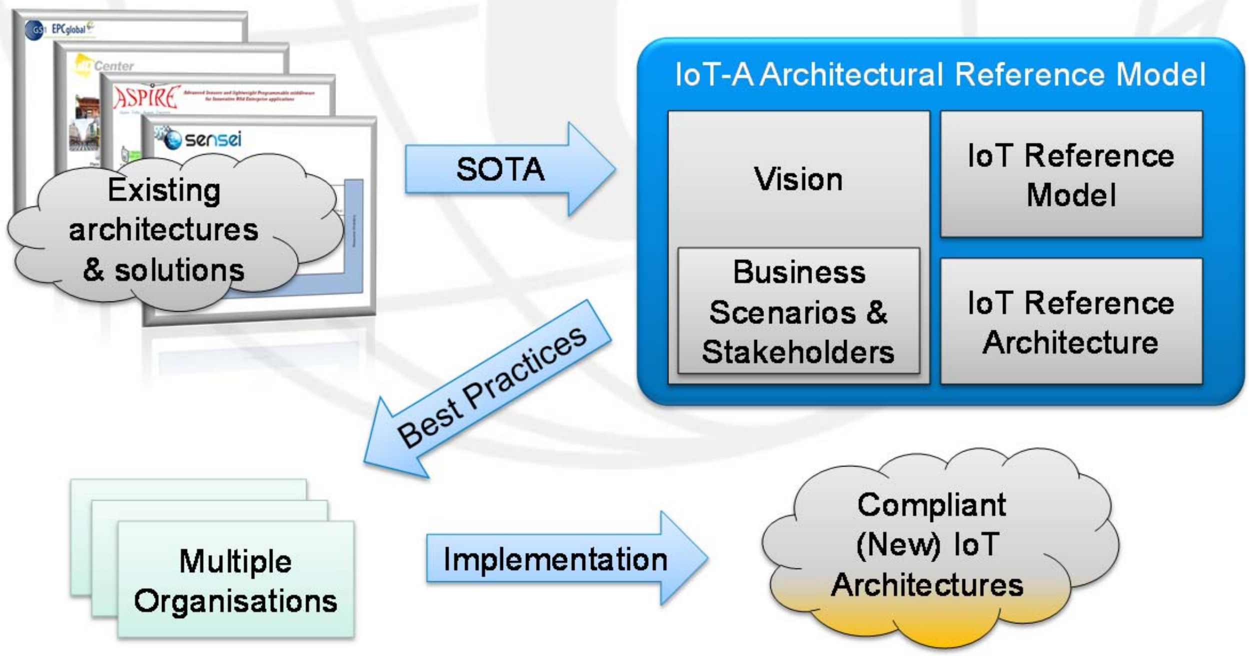

Figure 2: IoT-A architectural reference model building blocks.

- The **IoT Reference Model** provides the highest abstraction level for the definition of the IoT-A Architectural Reference Model. It promotes a common understanding of the IoT domain. The description of the IoT Reference Model includes a general discourse on the IoT domain, an IoT Domain Model as a top-level description, an IoT Information Model explaining how IoT information is going to be modelled, and an IoT Communication Model in order to understand specifics about communication between many heterogeneous IoT devices and the Internet as a whole. The definition of the IoT Reference Model is conforming to the OASIS reference model definition.
- The **IoT Reference Architecture** is the reference for building compliant IoT architectures. As such, it provides views and perspectives on different architectural aspects that are of concern to stakeholders of the IoT. The terms view and perspectives are used according to the general literature and standards The creation of the IoT Reference Architecture focuses on abstract sets of mechanisms rather than concrete application architectures.

To organisations, an important aspect is the compliance of their technologies with standards and best practices, so that interoperability across organisations is ensured. If such compliance is given, an ecosystem forms, in which every stakeholder can create new businesses that "interoperate" with already existing businesses. The IoT-A ARM provides best practices to the organisations so that they can create compliant IoT architectures in different application domains. Those IoT architectures are instances from the Reference Architectures with some architectural choices (called later on *Design Choices*) like considering strong real-time or choosing strong security features, etc. They form thus special "flavours" of the IoT Reference Architecture. Where application domains are overlapping, the compliance to the IoT Reference Architecture ensures the interoperability of solutions and allows the formation of new synergies across those domains.

Benefits of using the ARM

Using the IoT-A ARM can provide many benefits. We list here the most important ones.

Cognitive aid

When it comes to product development and other activities, an architectural reference model is of fourfold use.

First, it aids in guiding discussions, since it provides a language everyone involved can use, and which is intimately linked to the architecture, the system, the usage domain, etc.

Second, the high-level view provided in such a model is of high educational value, since it provides an abstract but also rich view of the domain. Such a view can help people new to the field with understanding the particularities and intricacies of IoT.

Third, the architectural reference model can assist IoT project leaders in planning the work at hand and the teams needed. For instance, the Functionality Groups identified in the functional view of the IoT system can also be understood as a list of independent teams working on an IoT system implementation.

Fourth, the architectural reference model aids in identifying independent building blocks for IoT systems. This constitutes very valuable information when dealing with questions like system modularity, processor architectures, third-vendor options, re-use of already developed components, etc.

IOT-A Reference Model as a common ground

Establishing a common ground for a field is not an easy task. To be effective, such a ground has to capture as many pertinent vantage points as possible. Establishing the common ground encompasses the definition of IoT entities and describing their basic interactions and relationships with each other. The Architecture Reference Model is providing exactly such a common grounding for the IoT field. Any party envisaging to develop an IoT system that is IoT-A compatible must build on the common concepts provided in the IoT-A Reference Model.

Generation of architectures

Another benefit is the use of the IoT-A ARM for the generation of compliant architectures for specific systems. This could be done by enabling tool support. The benefit of such a generation scheme for IoT architectures is not only the automatism of this process, and thus the saved R&D efforts, but that the generated architecture will provide intrinsic interoperability of the derived IoT systems.

Identifying differences

When using the aforementioned system-generation tools, which are based on the IoT-A ARM, any differences in the derived architectures can be attributed to the particularities of the pertinent use case. When applying the IoT-A ARM, predictions of system complexity, etc. are available for the system parts to be implemented. That makes judging the overall implementation effort for use case implementation easier, and some projects that might not have been realised due to uncertainties in the project plan might become possible. The overall implementation effort is most certainly less than developing an architecture without the help of an architectural reference model.

Benchmarking

Another important use is benchmarking. For example, NASA used a reference architecture of its new exploration vehicle for better benchmarking tenders it was going to receive during a public bidding process. While the reference model prescribes the language to be used in the systems/architectures to be assessed, the reference architecture states the minimum (functional) requirement on the systems/architectures. By standardising the description and also the ordering and delineation of system components and aspects, it also provides a high level of transparency and inherent comparability to the benchmarking process.

Process and Architecture Methodology

This Section provides a meta-perspective of IoT-A process, viz. a look at how the IoT ARM model was derived. It also explains the basic process how concrete systems can be developed by using the ARM First, we need to understand why the reference architecture derived needs to be accompanied by a reference model, before we discuss how the parts of the IoT ARM have been developed.

Introduction

Through the development of an architecture, a solution to a pre-defined goal is found. The development and description of architectures in turn is a modelling exercise. In this respect it is important to point out that the modelling itself does not happen in a vacuum, but rests on a thorough understanding of the domain modelled. In other words, any architecture development is contingent on one's understanding of the domain in question. The same is true for a generalisation of this process, viz. the derivation of reference architectures. Thus, reference architectures also have to be based on a detailed understanding of the domain in question. This understanding is commonly provided in the form of a reference model.

The above discourse motivates why the IoT Reference Architecture is accompanied by a thorough discussion of the IoT domain in the form of an IoT Reference Model. However, this high-level view does not explain how one derives either. What is needed here are both a process and a methodology for deriving the parts of the ARM. The process describes what steps need to be undertaken during the derivation of the architectural reference model, and the methodology describes how these steps are achieved. In other words, the methodology describes how to identify the tasks attached to each development step, and how and in which order to conduct these steps. Both the process and the methodology description are provided in this Section.

The remainder of the text in this Section is organised as follows. To start with, we provide a short discussion of the particularities of reference architectures and how they relate to concrete architectures, and also how they relate to reference models. This information enables us to discuss what high-level actions and input is needed for the derivation of an ARM, and what input is needed in order to guide the transformation of the reference architecture into use-case- and application-specific architectures. With this knowledge at hand we dive into the details of the development process. First, we restate the goals of IoT-A and how we translated them into a step-by-step process. Second, we explain how concrete architectures can be generated from the ARM. Next, we discuss the methodologies available for conducting each step. As it turns out, there is no standardised methodology for the derivation of ARMs. In order to overcome this lack of ARM methodology, we assessed the well-equipped toolboxes for the development of use-case- and application-specific architectures instead. Since these methods intrinsically rely on the specificity of the pertinent use cases and application scenarios, it is found that the methods considered, for instance model-driven engineering cannot always be applied one to one. This Section concludes with a detailed discussion of our requirements process, which is at the heart of our entire architecture process.

Reference model and reference architecture

Reference models and reference architectures provide a description of greater abstraction than what is inherent to actual systems and applications. They are more abstract than system architectures that have been designed for a particular application with particular constraints and choices. From the literature, we can extrapolate the dependencies of reference architecture, architectures, and actual systems (see Figure 3).

Architectures do help in designing, engineering, building, and testing actual systems. At the same time, understanding system constraints better can provide input to the architecture design, and in turn this allows identifying future opportunities. The structure of the architecture can be made explicit through an architecture description, or it is implicit through the system itself.

By extracting essentials of existing architectures, like mechanisms or usage of standards, a reference architecture can be defined. Guidance in form of best practices can be associated to a reference architecture in order to derive use-case-specific architectures from the reference architecture (see Figure 4). Such guidance can, for instance, make new architectures and systems compliant to each other. These general architecture dependencies apply to the modelling of the IoT domain as well.

While the model presented in Figure 3 stops at the reference architecture, the IoT-A architectural reference model goes one step beyond and also defines a reference model. As already discussed earlier, a reference model provides the grounding for a common understanding of the IoT domain by modelling its concepts and their relationships.

Actions and inputs

In the previous Section we discussed how reference architectures relate to architectures and real systems. In order to derive such a reference architecture and the reference model upon which the reference architecture builds, one needs better how they relate to each other and to external input. Such a taxonomy already provides us with a high-level perspective of actions and inputs needed for developing an ARM for IoT.

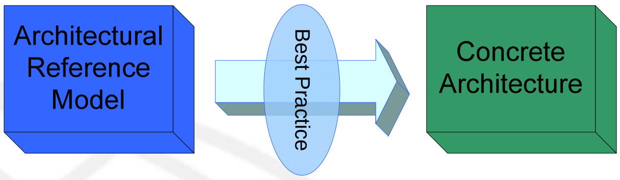

Figure 4: Relation of an architectural reference model, best practice, and concrete architectures.

A high-level taxonomy of how we understand the reference-architecture process is depicted in Figure 5. As discussed earlier, the IoT Reference Model provides guidance for the description of the IoT Reference Architecture. The Best Practice guides the derivation of IoT-A-compliant domain-specific architectures from the reference architecture.

Essential inputs for the definition of the IoT reference model are stakeholder concerns, business scenarios, and existing architectures. It is important to create a common understanding of the IoT domain from the different inputs. This is mainly a modelling exercise, during which experts have to work together and extract the main concepts and their relations of the IoT domain from available

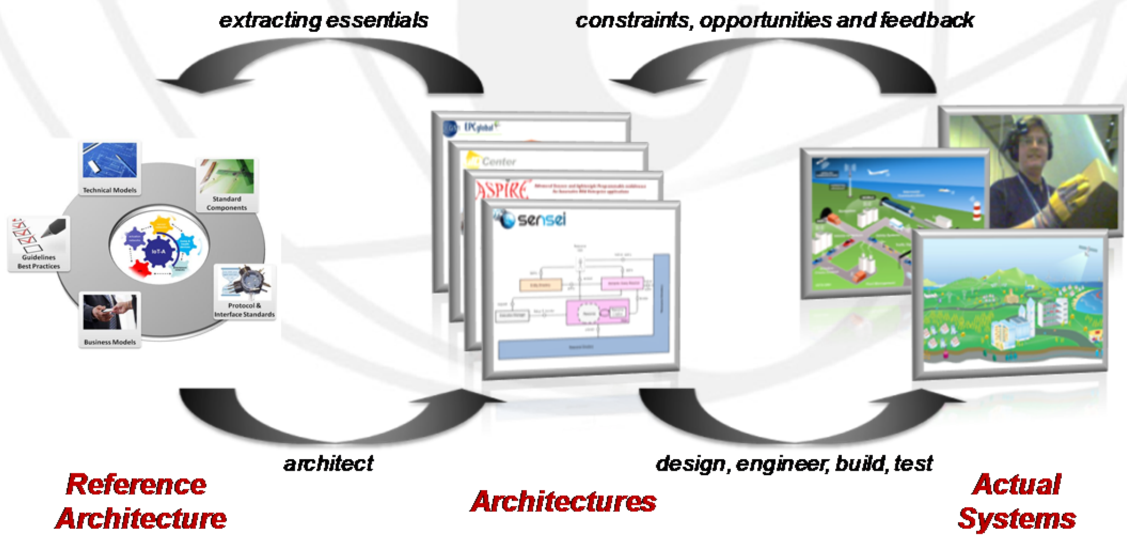

Figure 3: Relationship between a reference architecture, architectures, and actual systems (adapted from literature).

knowledge.

Furthermore, business scenarios, existing architectures, and stakeholder concerns can be transformed into application-specific requirements. When extrapolated, these requirements lead to a set of unified requirements. Unified requirements in turn steer the definition of the IoT Reference Architecture.

Within the ARM, the IoT Reference Model guides the definition of the IoT Reference Architecture, creating dependencies between the Reference Architecture and the Reference Model; once a change is proposed in the Reference Model a clear chain of dependencies can be followed and lead to subsequent changes within the Reference Architecture. By so doing, an overall consistency of the IoT-A ARM is maintained.

As one can see, this high-level representation already identifies high-level actions for the derivation of the ARM and for domain-specific architectures ("understand", "define", etc.). However this view is still too abstract for being of use in the day-to-day development work of the project. What is needed is a detailed architecture process that identifies individual tasks within the development process, that provides insight in the dependencies of said tasks, and that provides a dynamic model of the development process itself (viz. what step follows after the next).

Overall process

ARM development

A process-based view of the ARM derivation is shown in Figure 6. The ARM development process consists of one main process, which is the ARM derivation. Within the ARM derivation two actions are worth mentioning, viz. the domain modelling, which results in the IoT Reference Model, and the functional modelling, which is the main contributor to the IoT Reference Architecture. This process receives input from the requirement-collection process, which in turn receives input from external stakeholders and the state-of-the-art surveys conducted during the early stages of IoT-A.

The work in the ARM-derivation process is presented as an ARM draft. The initial ARM draft v0.9 was presented in deliverable D1.2.

The ARM draft guides the setup- up of the public

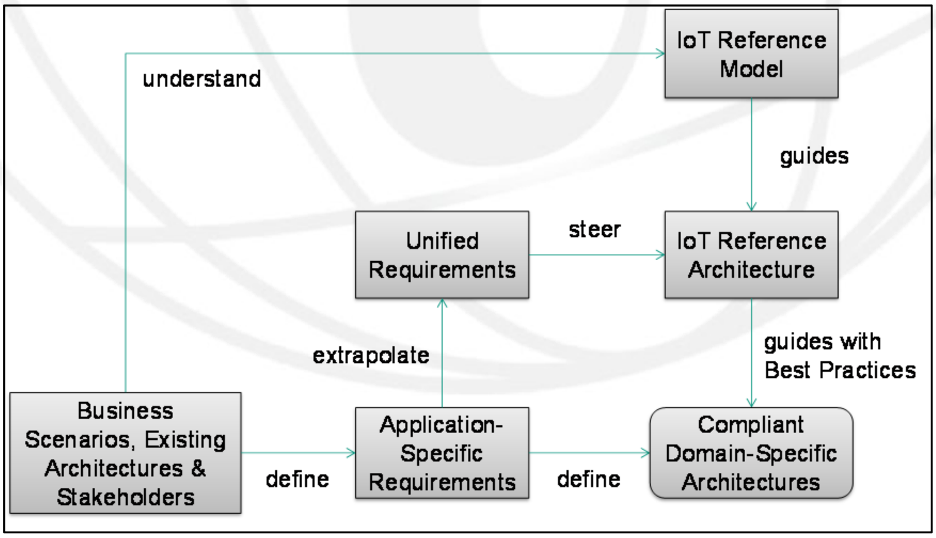

Figure 5: High-level representation of the IoT-Reference-Model and IoT-Reference-Architecture dependencies and model influences.

use-case demonstrations as well as the work of the technical work packages within IoT-A ("technical analysis").

The ARM draft is reviewed by the project's external stakeholders, the demonstration activity, as well as the technical work packages. This review serves as input for a revision of the ARM. In other words, the IoT-A project follows the well-established spiral design and prototyping model. The result from the first iteration of this development cycle is the current document, viz. ARM version v1.5. Before the conclusion of the project two more iterations are planned, resulting in v1.6 and v1.7, respectively.

Besides the architecture and domain analysis we also provide the user of the ARM with best practices for deriving use-case- and application-specific architectures (see Figure 6). Besides being of benefit for the user of the ARM, this process has the side benefit of providing valuable feedback to the ARM derivation itself. When devising guidelines for translating the ARM into a specific architecture, potential gaps and inconsistencies are revealed. Also, the best-practice exercise deepens our understanding of the IoT domain, and provides additional guidance on what aspects of the ARM need further enhancement. Last but not least, studying the translation of ARM into specific architectures and vice versa provides a compelling validation of the usefulness of the ARM.

The spiral-model approach inherent in the ARM development process was chosen for the following reasons. Firstly, each new iteration of the process increases the stability of the ARM. Secondly, due to its multi-step nature, the dissemination of the (embryonic) ARM starts early within the project. Thanks to early publication corrective impulses from peers and external stakeholders are received timely in the development process and can thus positively influence both the applicability of the ARM as well as its acceptance. Third, this approach formalises and coordinates the interaction of the architecture activity within IoT-A with that of the other activities (tech-

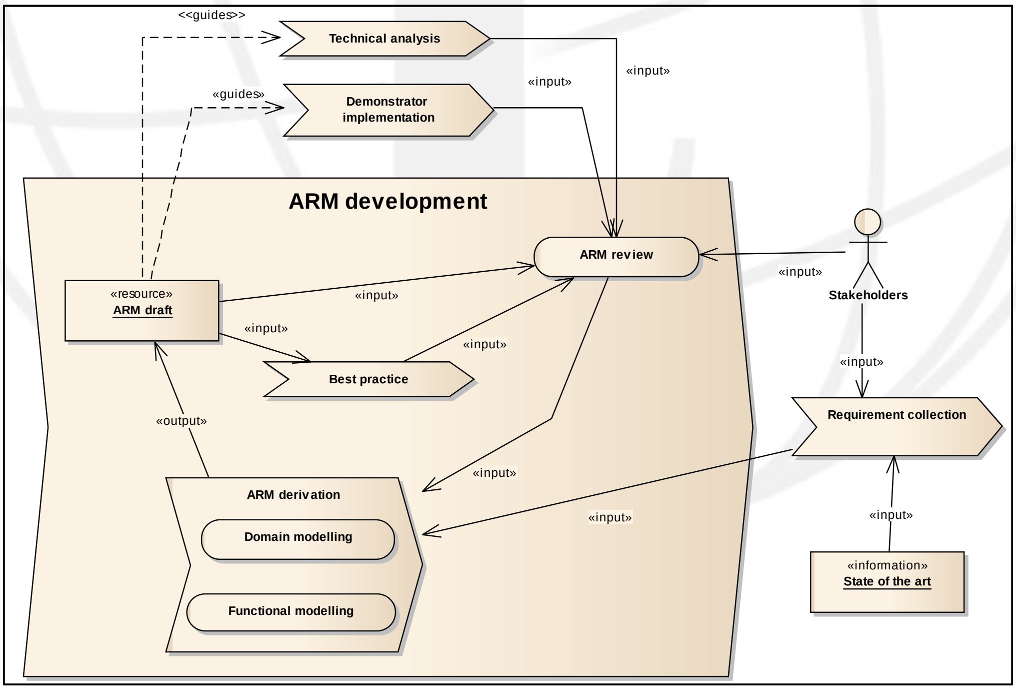

Figure 6: Dynamic view of the IoT-A ARM process.

nical analysis and demonstrator set up), which is expected to enhance the efficacy of this exchange.

Generation of architectures

So far we have only described the genesis of the IoT-A ARM, but not how its use for the generation of specific architectures actually works. While Figure 4 explains that the Best Practice accomplishes the transformation from the IoT ARM to a concrete architecture, the detailed picture is actually more complex.

When applying the ARM in the design of systems, it is likely that different architectures will result subject to the desired properties of the system. So, the best-practice transformation depicted in Figure 4 relies on a use-case description and requirements. This fact is reflected in Figure 8. The role of the ARM is to guide the architect through design choices at hand, and to provide best practices and design patterns for those different choices. The ARM is not operating in a design vacuum but should be applied together with proven design-process practices.

When designing concrete systems, one needs to keep in mind that in practice this will not always be the case. Depending on the engineering strategies used, some of the steps can be done in parallel or even have to be reiterated due to additional understanding gained during the process, or due to changes in the requirements.

Choice of design and development methodology

The choice of a design and development methodology can be understood in two ways: first, a methodology for the ARM development and second, a methodology for generating specific architectures. We have so far only provided high-level views of either case. In reality one needs more guidance, viz. a recipe on how to derive all aspect of the ARM model as well as how to derive the best practices. Simply dissecting them into design steps and processes is not enough; one needs to know how to achieve each step.

In the case of the ARM there are, to our knowledge, no standardised approaches for developing such a model. Furthermore, the IoT usage domain is, compared to typical reference-architecture domains, extremely wide and varied, and common denominators are thus rather few and abstract. This high level of abstraction in terms of the domain to be modelled stands in contrast to input needed for established and standardised methodologies such as, for instance, Aspect-Oriented Programming, Model-Driven Engineering, Pattern-Based Design, and SysML. All these methodologies were designed for very concrete use cases and application scenarios. Unfortunately, this high degree of specificity is even defining their inner workings. In other words, if one applies them to generalised use cases, one does often not get generalised models on the abstract level of an ARM as desired, since the processes of which said methodologies are constituted do not work for generalised use cases.

We illustrate the above issue with two examples, Model-Driven Engineering (MDE) and Pattern-Based Design (PBD). In the first case, the methodology is not directly applicable, while, in the second case, the methodology can potentially be generalised for deriving the best-practice transformation in Figure 9.

Model-Driven Engineering for the generation of Model-Driven Architectures is standardised by the Object Management Group. Its application area is the development of software systems. It provides an approach for, first, specifying a system independently from the platform; second, specifying platforms; third, choosing a particular platform for

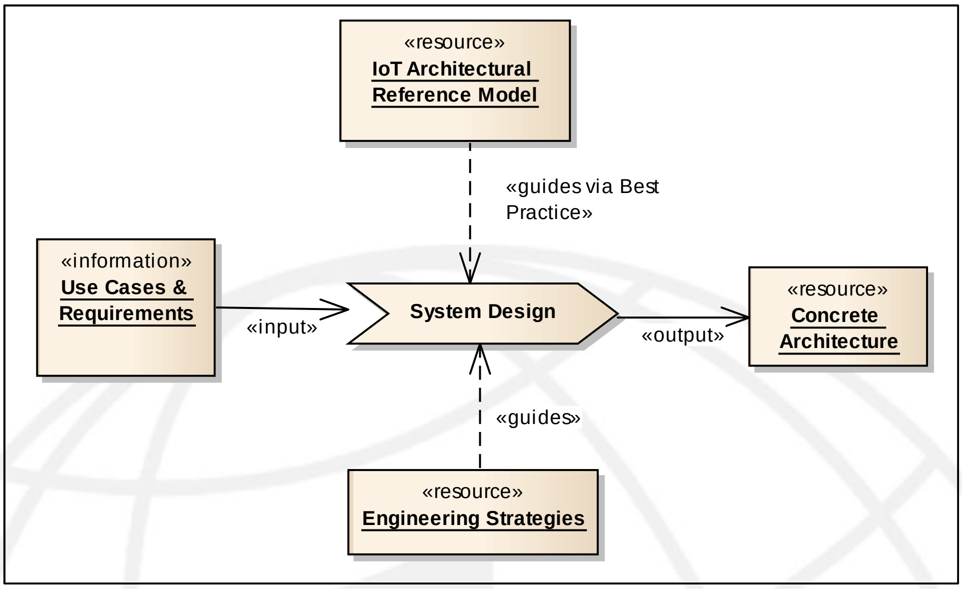

Figure 8: Process for the generation of concrete architectures.

the system; and fourth, transforming the system specification into that of a particular platform. The goals behind this approach are portability, interoperability, and reusability through the architectural separation of concerns. So, on the face of it, all this sounds very similar to the goals of our ARM development process.

In Figure 7, the main idea of model-driven architecture is shown. A platform-independent model, viz. an architecture, is to be transformed into a platform-specific model, viz. an implementation. An example for the former is a GUI user interface described in UML, and the latter

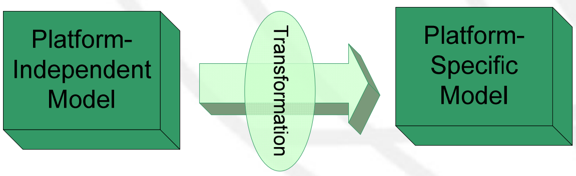

Figure 7: Generalised architecture approach according to the Model-Driven-Architecture methodology, a.k.a. Model-Driven Engineering.

is an implementation of said interface in a cellphone model featuring a particular operation system.

While this sounds very much like the Best-Practice transformation depicted in Figure 4, it is not the same. This becomes clearer in Figure 9.

Figure 9 is pieced together from Figure 4 (ARM) and Figure 7 (Model-Driven Engineering). As one can see, both the ARM and the Model-Driven-Engineering approach are linked to each other through platform-independent models, but they reside on different levels of abstraction. While the general idea of a model transformation, as promoted by MDE, resonates with our ARM approach (see Figure 4), the methodology developed for the derivation of transformations between platform-independent and platform-specific models can, upon a thorough analysis, not be transferred and adapted for the derivation of Best-Practice transformations.

Pattern-Based Design is a technique that reuses repeatable solutions to solve commonly occurring problems. In this design method one records how object-oriented designers identify recurring design problems. The corresponding solutions are then documented, and a reuse of the solutions is strived for. Consequently, the design process becomes increasingly flexible, elegant, and, most important, reusable. The solutions are divided into Sub-solutions, where "A design pattern identifies the participating classes and instances, their roles and collaborations, and the distribution of responsibilities. Each design pattern focuses on a particular object-oriented design problem or issue". From this short discussion it becomes clear that (a) Pattern-Based Design was developed for implementation processes, viz. the transformation to the right in Figure 9, and that (b) the only way this method can be applied for the derivation of the best-practice transformation in the same Figure would be by trying to translate the ARM into a particular architecture and to see whether the "book-keeping" approach prescribed by Pattern-Based Design yields valuable insight. At the current stage we do not know whether this is possible and aim at finding out during on-going best-practice development, which, among others, encompasses the derivation of a concrete architecture.

In Table 1 we summarise how we use ideas borrowed from standardised architecture methodologies for our work on the higher abstract level of an ARM.

Methodology	Aspect adopted in our work
Aspect-Oriented Programming	Delineation of functionalities by aspects. This is embodied in the concept of Functionality Groups
Model-Driven Engineering	General concept of transformation from a generic to a more specific model. We use this concept for describing and developing our Best Practice.
Pattern-Based Design	We will test the efficacy of this method upon deriving a concrete architecture as a best-practice test case.
Views and Perspectives	We adopt the concept of views and perspectives for the derivation of the IoT Reference Architecture, viz. we arrange all aspects of our reference architecture according to views and perspectives). The same is done for the unified requirements.

Table 1: Usage of standardised architecture methodologies for the development of the IoT ARM.

Requirements process

The IoT Reference Model by itself does not specify the technical particularities of an IoT system. For example, how are things identified and addressed in an IoT context? Or: how are these things associated with services? Such particularities are addressed in the IoT Reference Architecture. In order to build such a reference architecture, we not only need the IoT Reference Model and the methodology to do so, but also technical requirements that can be used for inferring particularities of the architecture. This is reflected in Figure 5.

In this Section we explain how the requirements for the IoT ARM have been inferred. The collection of

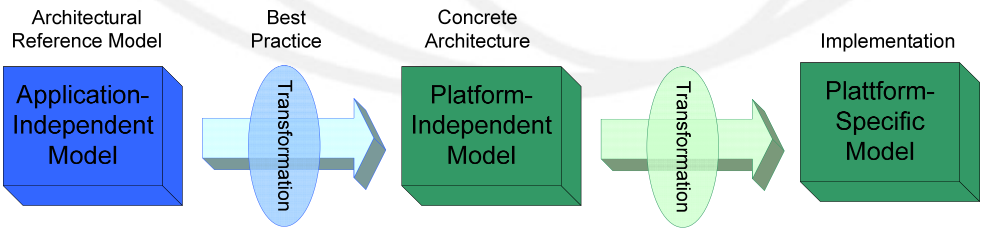

Figure 9: Relation of the Best-Practice-driven derivation of concrete architectures form an architectural reference model and the derivation of implementations from said concrete architecture.

requirements was done in a three-pronged process:

1. The rich experience and knowledge of the project partners guided the derivation of a minimum-requirement list, which also had a major influence in drafting the Reference Modelmodel. The state of the art concerning thing-centric communication and Internet technologies was considered, and a list of internal requirements was deduced.

2. A group of external IoT stakeholders was established and queried for their use cases and their expectations toward IoT. They were also asked for their objectives, concerns, and business goals. As far as feasible, these overarching aspirations were broken down into requirements.

Usually, such stakeholder aspirations are not made as system requirements, rather as use-case specific goals. Therefore, each stakeholder aspiration was thoroughly analysed, and suitable translations into requirements were sought. Stakeholder aspirations can be rather general (strategic objectives, concerns, or business goals) or they can be very specific, i.e., a stakeholder spells out what kind of functionality or performance she/he needs. An example for the former is the functionality of the IoT systems. For instance, European Telecommunications Standards Institute ETSI raised the following concern: *"Today, due to sub-optimal processes, a lot of time and money is wasted. This situation could be improved a lot by tracking all the items/things, providing context data on them at any time and location, allowing for automated evaluation of the collected data and reacting immediately on a dangerous situation to protect against the breakdown-break-down of items."* This addresses the functional view, but it does not clearly address what functionalities are needed in order to meet this aspiration. In our requirement-engineering process, we broke this concern down into two distinct functional requirements.

- "The system shall enable centralized or decentralized automated activities (control loops)."

- "The system shall enable the planning of automated tasks."

The above example was provided in order to briefly illustrate our requirement process. The functional view is a recurring item in the list of unified requirements. This view is represented as a block-diagram in the IoT Reference Architecture, which in itself constitutes a central result of the IoT-A project and an indispensable input for the development of a compliant IoT system.

Business Scenarios validating the ARM

Business scenarios play an important role in the external validation of the Architectural Reference Model (ARM). Business scenarios help defining application-specific requirements, i.e., they are one source of input regarding what potential systems and applications need to implement and deliver, if they are to realise certain business scenarios. At the same time, business scenarios help understanding the IoT Reference Model as such, as the domain components described in the reference model are reflected in the respective business scenarios, i.e. the reference model provides a formalised and abstracted model of the entities and their relationships that are brought to life within the different business scenarios.

Rationale and Introduction

The primary aim of business scenarios is to provide an external validation of the ARM in economic terms, i.e., business scenarios should demonstrate that concrete systems built utilising concrete ARM compliant architectures are economically viable and beneficial, so that it makes sense for business stakeholders to develop business scenarios based on IoT-A models and best practices. Ideally, business scenarios should cover a diverse set of relevant application fields in order to demonstrate the broad applicability of IoT-A, especially since one of the primary goals of IoT-A is to develop an IoT Reference Architecture that transforms the isolated island solutions of the “intranets of things” as we know them today into a domain-spanning interoperable infrastructure of IoT platforms that is viable from an economic point of view and facilitates novel business opportunities.

Due to limited space, we will here only briefly discuss the main application fields for which viable business scenarios compliant to IoT-A can be developed, and then focus only on one central application field in the focus of the project, namely the retail domain.

The narrowed focus of the use cases comes from the fact the stakeholder group of the IoT-A project focuses mostly on selected application fields. Within these fields, stakeholder aspirations can of course be diverse, because of differences in their background and differences in their business views. Nevertheless, there are some common themes in stakeholder aspirations that make us confident that there is some potential for generalizing business scenarios:

- Many stakeholders see IoT as a means of improving their current business. IoT will thus serve various business goals and strategic objectives, such as future-proof, lowered costs, etc.

- Other stakeholders see IoT as a disruptive technology, which will aid them in creating new applications and thus new business opportunities (selling access to sensor data, etc.).

- In order to achieve a maximum of flexibility of IoT technology and its use, short product-development cycles, and a maximum leverage of existing and new solutions to common problems is needed. For that reason, many stakeholders advocate open IoT platforms and frameworks. The underlying business goal for this advocacy is to lower costs in product development. Strategic objectives are to enhance product interoperability and to shorten the development cycles. The latter is important for responding to customers’ emerging needs in an agile manner.

- Since active supervision of IoT interactions is even more elusive than monitoring today’s Internet traffic, security and privacy have, as expected, been identified as a core topic. Privacy is strongly related to the overall acceptance of IoT. If individuals and other users cannot experience a sufficient level of privacy when utilising IoT technology, this will critically challenge the acceptance of this novel technology. Security equates of course not only to privacy, but also to the protection of the IoT against interferences, such as service attacks, trojans, viruses, etc.

Fields of Application

In order to maximise the impact of our architectural reference model, we have to identify those scenarios where IoT technologies have a special relevance, taking into account that these scenarios frequently share the same applications, sensors, stakeholders and, of course, users. We will base this identification on scenarios that have been kindly provided by the IoT-i[1] project.

Transportation/ Logistics

In transport logistics, IoT improves not only material flow systems, but also global positioning and auto-identification of freights. Additionally, it increases energy efficiency and decreases thus energy consumption. In conclusion, IoT is expected to bring profound changes to the global supply chain via intelligent cargo movement. This will be achieved by means of continuous process synchronisation of supply-chain information, and seamless real-time tracking and tracing of objects. It will provide the supply chain a transparent, visible and controllable nature, enabling intelligent communication between people and cargo.

Smart home

Future smart homes will be conscious about what happens inside a building, mainly impacting three aspects: resource usage (water conservation and energy consumption), security, and comfort. The goal with all this is to achieve better levels of comfort while cutting overall expenditure. Moreover, smart homes also address security issues by means of complex security systems to detect theft, fire or unauthorized entries. The stakeholders involved in this scenario constitute a very heterogeneous group. There are different actors that will cooperate in the user's home, such as Internet companies, device manufacturers, telecommunications operators, media-service providers, security companies, electric-utility companies, etc.

Smart city

While the term smart city is still a fuzzy concept, there is a general agreement that it is an urban area which creates sustainable development and high quality of life. Important areas of a smart city are , encompassing economy, people, governance, mobility, environment and living. Outperforming in these key areas can be done through strong human or social capital and/or ICT infrastructure. For the latter, a first business analysis concludes that several sectors/industries will benefit from more digitalised and intelligent cities (examples for a city of 1 million people[2]:

- Smart metering, 600.000 meters, US $ 120 million opportunity
- Infrastructure for charging electric vehicles, 45.000 electric vehicles, US $ 225 million opportunity
- Remote patient monitoring (diabetes), 70.000 people, US $ 14 million opportunity
- Smart retail, 4.000 stores, US $ 200 million opportunity
- Smart-bank branches, 3.200 PTMs, US $ 160 million opportunity

Smart factory

Companies will be able to track all their products by means of RFID tags in a global supply chain; as a consequence, companies will reduce their opera-

[1] See http://www.iot-i.eu/

[2] Cf. R. Nicholson, "*Smart Cities: Proving Ground for the Intelligent Economy*", 2010

tional expenditure and improve their productivity due to a tighter integration with enterprise resource planning and other systems. Generally, IoT will provide automatic procedures that imply a drastic reduction in the number of employees needed. Workers will be replaced by bar-code scanners, readers, sensors and actuators, and in the end by complex robots that are as efficient as a human. Without any doubt, these technologies will bring opportunities for white-collar workers and a big number of technicians will be necessary to program and repair these machines. This is synonymous to a transfer to maintenance jobs, but it also constitutes a new challenge for providing all blue-collar workers with an opportunity to move toward these types of jobs and to avoid unemployment.

Retail

IoT realises both customer needs and business needs. Price comparison of a product; or looking for other products of the same quality at lower prices, or with shop promotions gives not only information to customers but also to shops and business. Having this information in real time helps enterprises to improve their business and to satisfy customer needs.

Obviously, big retail chains will take advantage of their dominant position in order to enforce the future IoT retail market, as it happened with RFID adoption, which was enforced by WalMart in 2004. Particularly, companies with controlling positions, such as WalMart, Carrefour, Metro AG, etc. are able to push the adoption of IoT technology due to their sizeable market power.

e-Health

Controlling and preventing is one of the main goals of future health care. Already today, people can have the possibility of being tracked and monitored by specialists even if both are not at the same place. Tracing peoples' health history is another aspect that makes IoT-assisted eHealth very versatile. Business applications could offer the possibility of medical service not only to patients but also to specialists, who need information to proceed in their medical evaluation. In this domain, IoT makes human interaction much more efficient because it not only permits localization, but also tracking and monitoring of patients. Providing information about the state of a patient makes the whole process more efficient, and also makes people much more satisfied.

The most important stakeholders in this scenario will be public and private hospitals and institutes such as, e.g., the Institute of Applied eHealth at Edinburgh Napier University, which partook in the first stakeholder session of IoT-A. It is worth mentioning that telecommunications operators are quite active in e-health (for instance, O2 UK).

Environment

Applications in the environmental domain have many overlaps with other scenarios, such as smart home and smart city. The key issue in these scenarios is to detect means that help to save energy. One prominent example is Smart Grid. Concerning this application area one needs to highlight initiatives that imply a more distributed energy production, since many houses have a solar panel today.

As a vital part, smart metering is considered as a pre-condition for enabling intelligent monitoring, control, and communication in grid applications. The use of IoT platforms in Smart Metering will provide the following benefits:

- An efficient network of smart meters allows for faster outage detection and restoration of service. Such capabilities redound to the benefit of customers

- Provides customers with greater control over their energy or water consumption, providing them more choices for managing their bills.

- IoT deployment of smart meters is expected to reduce the need to build power plants. Building power plants that are necessary only for occasional peak demand is very expensive. A more economical approach is to shape the demand by either to incentivize customers to reduce their demand through time-based rates or other programs, or by service-level agreements that allow temporarily turning off devices which are not needed (e.g., the freezer for 20 minutes).

In order to describe a well-defined business model it is necessary to define what needs to be done in the business, which are the metrics for success, which are the problems that must be solved and the plans that solve these problems. Knowing which part of the problem is possible to solve and how much time is needed and which part cannot be solved is an important step that we must take into account when we develop concrete business cases for some of the application fields discussed above.

As we can only go into the details of one business case in the context of this document, we will pick a use case from the application field of retail, as this is a central application field for the project, and apply an appropriate business case methodology to it. This methodology is outlined in the next section. The use of a methodology instead of merely calculating "some kind of business case" enables us to perform comparisons between different application fields, for instance when we consider health under an economic perspective within the context of the forthcoming deliverables.

Business Case Methodology

As demonstrated above, there is a huge potential for realising IoT applications in different application fields that are based on architectural concepts of IoT-A and potentially bring novel business opportunities, for instance when sensor technology contributes to changing distribution models for perishable goods, so that e.g. fruits or vegetables can still be sold to the consumer before their quality deteriorates and the goods are wasted. However, in order to make such scenarios possible large investments are needed in, e.g., hardware, software, installation, configuration, maintenance, business process reengineering and training of personnel. To justify such investments, a 'business case' (BC) is usually developed, describing the benefits, costs and risks of each investment alternative.

BCs commonly appear as spread-sheets, often accompanied by presentations or explanatory documents. They may be presented by the project leader (BC 'owner' or 'champion') to senior management, which is responsible for prioritizing BCs and making investment decisions. This way, the BC can be used to decide about investment before project execution ('ex-ante'), to evaluate progress during project execution and to determine to what extent the proposed value of the investment has been realized after project execution ('ex-post'). Naturally, the development of BCs is a complex task. First, collecting, transforming and aggregating the required information demands interdisciplinary teamwork and expertise in a wide range of fields such as business strategy, business operations ('work practice'), information technology, accounting and project management. Second, BCs are based on assumptions concerning the future development of certain variables. Predicting those variables requires accurate data and reliable analysis methods. Third, BCs are subject to a constantly changing business environment, requiring an agile BC development process to adapt to these changes.

Within the context of IoT-A, BCs should be based on a generic BC process to allow for their development, use and improvement across different application fields. We therefore base the BC process on a framework that is being developed in the IoT project SemProM[3] that proposes a BC framework which is based on a generic BC process, consisting of six steps: Scope, Processes, Criteria, Methods, Results, Conclusion. During this process, domain-specific components consisting of criteria and methods may be reused.

The BC framework provides a set of spread-sheets in Microsoft Excel that accompany the process proposed. In the following section, we apply this framework to two of the primary retail use cases of IoT-A, namely the NFC Based Shopping Assistant in combination with the sensor- based quality control.

Retail Business Case

The use case shows how IoT technologies like sensor technologies built into consumer electronic devices and NFC tags coupled with the Internet of Things Architecture can provide useful meta-information to the customer to enhance the overall shopping experience and at the same time significantly reduce the costs for consulting that sales personnel in the retail stores need today, as there are no such systems in widespread use. The use case demonstrates a direct human-to-machine interaction.

From a business perspective, the use case is primarily interesting because the NFC-based product information has the potential to reduce the consultation time of the sales personnel in the store.

In order to make the business case somewhat more complex, we also integrate the sensor based quality control use case and assume that both scenes are interconnected, because they are based on a common architecture, namely a concrete architecture based on the IoT-A Reference Architecture.

The sensor based quality control scene shows how sensors monitor perishable goods in a store. Depending on the luminance, humidity, and temperature of the environment, the estimated future quality of the perishable products is determined and prices are reduced even before a perceivable degradation of quality occurs. By applying this sensor based quality control and combining it with dynamic pricing, it is ensured that the goods are sold before quality degradation is likely to occur. From a business and industry perspective, the scene demonstrates two important retail-related concepts: Dynamic pricing and quality control of perishable goods. Dynamic pricing as a real-time tool for price optimization strategies has always been crucial. In contrast to the state of the art, dynamic pricing in the featured use case is not performed on static information such as best-before dates in the backend ERP system, but it is based on real time IoT data gathered from a sensor infrastructure. As about 20% of perishable goods never reach the consumer, but are disposed of before, the utilization of IoT sensors is also an interesting concept to implement quality control of perishables and thus reduce waste and increase profits at the same time.

As we have stated before, we assume that both the self-contained NFC-based product information and the sensor based quality control are based on the same technical system realised in accordance with the IoT-A ARM. Therefore, we calculate their antici-

[3] http://www.semprom.de/

pated effects in a combined business case. The actual Excel sheets are available on the IoT-A web-site at www.iot-a.eu, but in the following tables we already provide some of the respective criteria, on which the calculations are based, as well as instantiations of these criteria calculated for cases, when the IoT-A -based use cases are realised and when they are not realised (= the baseline).

In our calculations we base our BC on an example case for German Retailers trading fast moving consumer goods (FMCG) in a higher market segment. The following two tables outline the respective parameters used.

Benefits > Sales per hour > Consulting time > Self-contained product information via NFC based Shopping assistant

Description: Sales per Hour is a traditional KPI to measure salesperson or store performance

Reference: Lipschitz(2006): Retail Key Performance Indicators (KPI), http://www.hillsorient.com/articles/2006/07/121.html

Example case for German Retailers trading FMCGS: Consulting time for organic products purchases

Instantiation with Traditional Consulting by Sales Personnel

Parameters	2010	2011	2012	2013	Metric	Reliability	Source
RR: Relevant retailers	31174	30550	29939	29340	abs. number	fact	Statistisches Bundesamt, consolidation is likely c
HS: Households per Store	300	305	310	315	abs. number	accurate estimate	Estimated from http://salzburg.orf.at/stories/439
HO: Households	40'200'000	40'119'000	40'039'000	39'959'000	abs. number	fact	Statistisches Bundesamt, decrease in sync with
CO: Consumers	70'000'000	69'860'000	69'720'000	69'580'000	abs. number	rough estimate	Active population is decreasing 0.2% each year in
IOC: 'Intensive' Organic Consumers	0.02	0.02	0.02	0.02	%	rough estimate	Bruhn, Maike (2002) Warum kaufen Verbraucher
CR: Consumers requiring Consulting	0.1	0.1	0.1	0.1	%	guess	Estimation
VW: Retail Store Visits per week	1.5	1.5	1.5	1.5	abs. number	accurate estimate	HJ Becker (2005) Einkaufszentren in Berlin http://
MWT: Mean Waiting Time for Consulting	110	110	110	110	seconds	accurate estimate	Jungbauer et al (2005) Does Clothing Make the Ma
MCT: Mean Consulting time per customer	240	240	240	240	seconds	accurate estimate	Estimate derived from pharmacy domain http://w
Derived Measures							
RC: Relevant Consumer store visits per week (CO*IOC*VW)	2'100'000	2'095'800	2'091'600	2'087'400	abs. number		
CW: Consulting time per week (RC * CR * (MWT + MCT))	20'417	20'376	20'335	20'294	hours		
RT: Weekly Consulting time per retailer (CW / HO / HS / RR)	**4'750**	**4'732**	**4'714**	**4'694**	hours		

Instantiation with IoT-A assisted Self-Contained Product Information

Parameters	2010	2011	2012	2013	Metric	Reliability	Source
RR: Relevant retailers	31174	30550	29939	29340	abs. number	fact	Statistisches Bundesamt, consolidation is likely c
HS: Households per Store	300	305	310	315	abs. number	accurate estimate	Estimated from http://salzburg.orf.at/stories/439
HO: Households	40'200'000	40'119'000	40'039'000	39'959'000	abs. number	fact	Statistisches Bundesamt, decrease in sync with
CO: Consumers	70'000'000	69'860'000	69'720'000	69'580'000	abs. number	rough estimate	Active population is decreasing 0.2% each year in
IOC: 'Intensive' Organic Consumers	0.02	0.02	0.02	0.02	%	rough estimate	Bruhn, Maike (2002) Warum kaufen Verbraucher
CR: Consumers requiring Consulting	0.1	0.1	0.1	0.1	%	guess	Estimation

Scope | Processes | Criteria | Methods & Values | Results

Table 2: Sample Instantiations for the Retail

Criteria	Method	Unit	Goal	Impact	Business Process	Reference
Benefits			max.			
Sales	MS = sum(AS) * reference time frame	€	max.		06. Sales	Lipschitz(2006): Retail Key Performance Indicators (KPI), http
Sales per hour	HS = sum(all items sold) / sum(hours within fiscal		max.		06.04 Sales Processing	Lipschitz(2006): Retail Key Performance Indicators (KPI), http
Consulting time	CT = customers * store visits * time per custor	time	min.		06.04 Sales Processing	
Self-contained product information			max.		06.04 Sales Processing	
Average sale	AS = total price of items sold per sale / receiq	€	max.		06.04 Sales Processing	Lipschitz(2006): Retail Key Performance Indicators (KPI), http
Price optimization	price = f(optimization)	€	max.		06.03 Offering Design	Dynamic Pricing in the Presence of Inventory Considerations
Dynamic quality based price optimizatic	price = f(quality)	€	max.	Sensor-Based Quality Control a	06.03 Offering Design	Dynamic Pricing in the Presence of Inventory Considerations
Transparency of origin	price = sum(prices of products with transpare	€	max.		06.03 Offering Design	
Items per sale	IS = sum(all items sold per reference time fr	int	max.		06.04 Sales Processing	Lipschitz(2006): Retail Key Performance Indicators (KPI), http
Conversion rate	CR = (Number of visitors / number of receipts	%	max.		06.01 Customer Management	Lipschitz(2006): Retail Key Performance Indicators (KPI), http
Attraction through product information	AT = (Number of visitors accessing additional	%	max.	The additional information tha	12.03 Store Design	Senecal, S. & Nantel, J., The Influence of Online Product Reco
Supply chain optimization	SCO= CO = sum(other wasted products) / sum	%	max.		05.04. Transport Planning	Dada, A. and Thiesse, F. 2008. Sensor applications in the sup
Cool chain integrity	CCI = QB / sum(all perishable products)	%	max.		05.04. Transport Planning	
Quality based issuing of perishables	QB = sum(wasted perishable products)	int	min.	The continuous monitoring of	05.06 Warehouse Control	
Assortment selection	AS = sum(CCP) + sum(other products)	int	max.		06.02 Assortment Selection	Shelby H. McIntyre, Christopher M. Miller, The selection and
Conscious consumer products	CCP = TO + sum(other conscious consumer pro	int	max.		06.02 Assortment Selection	
Transparency of origin	TO = sum(products with transpareny of origin	int	max.		06.02 Assortment Selection	
Costs						
Store operation costs	SOC = IPM + EC					
Instore peripheral management	IPM = sum(software infrastructure costs in-st	€	min.	A Real World Integration Platfo	12.03 Store Design	
Employee costs	EC = sum(salary of each person)	€	min.	For example, the Shopping Assi	11.05 Organisational Managem	Resatsch, F.; Sandner, U.; Leimeister, J. M. & Krcmar, H. Do Po
Acquisition of trading goods	ATG = sum(all below)				03. Acquisition of Trading Goods	
Production	PRO = production costs	€	max.		04. Production	
Raw materials and supplies	RMC = Costs of Material	€	max.		04. Production	
Procurement	PRC = Total costs for procurement	€	max.		03. Acquisition of Trading Goods	
Goods receipt	GR = Costs per Good received	€	min.		05.01 Goods Received	
Waste costs	WC = Costs of Waste	€	max.		05.02 Inventory Management	
Inventory costs	INV = Current Inventory * Item Costs	€	max.		05.02 Inventory Management	
Risks						
Dependability on technical infrastructure	DEP = dependability	%	min.	IoT infrastructures introduce a significant amount of technical components, both on the products and on the backend. Core		
Sales personnel competencies	SPC = necessity of being a domain expert	%	max.	As NFC based shopping assista	11.05 Organisational Management	
Digital divide causing loss of consumer base	DD = digital divide (DDH + DDL)	%	min.	With consumer-facing compone	06.02 Assortment Selection	Coroama et al (2003): Living in a smart environment. The soc
high education population	DDH = digital divide high education	%			06.02 Assortment Selection	Coroama et al (2003): Living in a smart environment. The soc
low education population	DDL = digital divide low education	%			06.02 Assortment Selection	Coroama et al (2003): Living in a smart environment. The soc
Technical faults and errors	FE = faults and errors	%	min.	Any technical infrastructure in	12.03 Store Design	
Process re-engineering threats						
Handling and storage procedures	HS = handling and storage	%	min.	Any RFID based system is heav	05.04. Transport Planning	Finkenzeller (1999): The RFID handbook
Sales, return and recall related information governe	IG = information governance	%	min.	Legal restrictions impose signi	06.01 Customer Management	Coroama et al (2003): Living in a smart environment. The soc

Scope | Processes | Criteria | Methods & Values | Results

Table 3: Criteria for the Retail Business Case

The core result of the business case calculation for the retail domain is that, apart from the reduced waste of perishables due to the sensor based quality control, the consulting time of sales personnel being reduced significantly, in our case about 30%, so that IoT-based scenarios indeed appear to have a significant business impact. For the detailed results, please consult the respective spread-sheet.

The business case as we have calculated it for a retailer from the FMCG field does not yet take into account the savings and benefits of software systems that are compliant with the IoT-A Reference Architecture and that follow the best practices and design choices laid out by the IoT-A project. While we envision the best practices to be a strong and central contribution of the project, it is currently hard to calculate its economic impact.

Conclusions & Outlook

Apart from the best practices, we also believe that substantial economic benefits can emerge from the modular and component-based approach to building IoT systems that are compliant with the IoT-A Reference Architecture. These additional business effects will also be taken into account for future business cases.

While we can already state that from an economic perspective the IoT-A approach simply makes sense, it is also important to note that solid business scenarios are only a precondition to the application of the ARM. In order to implement Internet of Things use cases based on the IoT-A Reference Architecture, the project aims at providing much more than just the Reference Architecture itself and an economic validation: The business cases are just one building block towards a fully featured **"Cook book"** with information on various aspects concerning the implementation of IoT systems. It will provide best practices for the various modules that the ARM comprises and will discuss design choices that academics and practitioners alike will be faced with when implementing concrete systems based on IoT-A.

The latest project deliverable D1.3 already includes lavish sections on design choices and best practices that will be further augmented and refined in order to develop an easily accessible guide for implementation that will certainly have an impact beyond the IoT-A partners and the projects in the IERC cluster. Our leitmotif was and remains to provide the industry standard for implementing future IoT systems based on a modular, common approach and an accessible, yet sophisticated, reference architecture.

Please follow our activities towards the concretization of the ARM and the best practices on working with the ARM by visiting our website at www.iot-a.eu and getting the latest version of the ARM. In order to give feedback and help shaping future versions of the ARM and the best practices for implementation, please make sure to get in touch with us and our stakeholder group about which you can also find information on our website.

Imprint

Editors:

Sebastian Lange (VDI/VDE-IT), Andreas Nettsträter (Fraunhofer IML), Stephan Haller (SAP), Francois Carrez (University of Surrey), Alessandro Bassi (Hitachi)

Authors:

Martin Bauer (NEC), Nicola Bui (Consorzio Ferrara Ricerche), Francois Carrez (University of Surrey), Pierpaolo Giacomin (Hitachi), Stephan Haller (SAP), Edward Ho (University of St.Gallen), Christine Jardak (Siemens), Jourik De Loof (Alcatel-Lucent), Carsten Magerkurth (SAP), Andreas Nettsträter (Fraunhofer IML), Alexandru Serbanati (Sapienza University of Rome), Matthias Thoma (SAP), Joachim W. Walewski (Siemens), Stefan Meissner (University of Surrey)

IoT-A
Internet of Things - Architecture

SEVENTH FRAMEWORK PROGRAMME