electronics

MDPI

Article

Domain Specific and Model Based Systems Engineering in the Smart Grid as Prerequesite for Security by Design

Christian Neureiter [1,*], Dominik Engel [1,†] and Mathias Uslar [2,†]

[1] Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, University of Applied Sciences Salzburg, Urstein Süd 1, 5412 Puch, Salzburg, Austria; dominik.engel@fh-salzburg.ac.at

[2] Institute for Information Technology Oldenburg (OFFIS), Escherweg 2, 26121 Oldenburg, Germany; mathias.uslar@offis.de

* Correspondence: christian.neureiter@fh-salzburg.ac.at; Tel.:+43-50-2211-1328; Fax: +43-50-2211-1349

† These authors contributed equally to this work.

Academic Editors: Alfredo Vaccaro and Jin (Wei) Kocsis

Received: 25 March 2016; Accepted: 24 May 2016; Published: 28 May 2016

Abstract: The development of Smart Grid systems has proven to be a challenging task. Besides the inherent technical complexity, the involvement of different stakeholders from different disciplines is a major challenge. In order to maintain the strict security requirements, holistic systems engineering concepts and reference architectures are required that enable the integration, maintenance and evaluation of Smart Grid security. In this paper, a conceptual approach is presented on how to enable the integration of security by design in the development of Smart Grid Systems. A major cornerstone of this approach is the development of a domain-specific and standards-based modelling language on basis of the M/490 Smart Grid Architecture Model (SGAM). Furthermore, this modelling approach is utilized to develop a reference architecture model on basis of the National Institute of Standards and Technology (NIST) Logical Reference Model (LRM) with its integrated security concepts. The availability of a standards-based reference architecture model enables the instantiation of particular solutions with a profound basis for security. Moreover, it is demonstrated how such architecture models can be utilized to gain insights into potential security implications and furthermore can serve as a basis for implementation.

Keywords: Smart Grid; Systems Engineering; Model Driven Engineering; Smart Grid Reference Architecture; SGAM; NISTIR 7628

1. Introduction

Today's electricity system faces major challenges by the ongoing integration of renewable energy resources on distribution level. These resources, often referred to as *Distributed Energy Resources* (DER), are characterized by a volatile injection behavior depending on the availability of their primary energy source such as wind or sun. As a consequence, the maintenance of the overall grid stability requires a sophisticated management of now bidirectional energy flows. Thus, the ICT-based grid automation is going to be extended onto distribution system level which involves the integration of individual participants such as electric vehicles, photovoltaic systems or Smart Meter.

To enable the aforementioned scenario, today's energy system is going to evolve towards a *Smart Grid* which is described by the Electric Power Research Institute (EPRI) as *"modernization of the electricity delivery system so it monitors, protects and automatically optimizes the operation of its interconnected elements - from the central and distributed generator through the high-voltage network and distribution system, to industrial users and building automation systems, to energy storage installations and to end-use consumers*

and their thermostats, electric vehicles, appliances and other household devices" [1]. Some more detailed considerations on possible future directions of the grid can be found for example in [2,3] or [4]. A detailed comparison between the existing grid and the Smart Grid is delivered by Farhangi *et al.* [5].

From a more holistic viewpoint and by following the system classification from [6] it can be argued that the grid evolves from a *massively interconnected, complicated system* towards a *complex system* which is characterized *"by a huge number of elements and connections with a high variety and dynamically established connections"*.

However, the electric energy system is classified as critical infrastructure which implies excessive reliability requirements. To maintain a stable grid operation in the future, the ongoing evolution will have a huge impact on distribution system design as outlined in [7]. Particular considerations on this issue can be found, for example, in a critical discussion published by Moslehi *et al.* [8]. The authors conclude that an architectural approach for transforming the power grid to a "smarter grid" is required. Especially the need for a systematic approach for grid wide integration of the necessary IT technologies is highlighted. Further considerations on present research challenges in this context can be found for example in [9] or [10]. These considerations are complemented by Gungor *et al.* who deliver some more detailed discussions on requirements for the ICT based communication infrastructure [11].

Besides the arising grid related challenges, ICT-related challenges such as privacy or security are lifted to a new level [12,13]. The presence of a huge number of individual interconnected participants increases the potential weak points in terms of security dramatically. A successful exploitation of a potential weak point could limit reliability and lead to a black out with severe consequences. As the availability of electric energy has become a backbone of modern societies, the US Department of Energy states *"The issue of blackouts has far broader implications than simply waiting for the lights to come on. Imagine plant production stopped, perishable food spoiling, traffic lights dark, and credit card transactions rendered inoperable. Such are the effects of even a short regional blackout"* [14].

By now, the awareness on these issues is very high and a broad consensus on the importance of privacy and security for the Smart Grid exists. Thus, these issues are subject of investigations in respect to various different aspects such as encryption and authentication [13], hardware security [15] or, communication infrastructure in general [16]. However, in order to realize and maintain security it is necessary to consider a system from a holistic viewpoint. Moreover, as the goal should be to realize "security by design", appropriate considerations should take place in every stage of the development. Various standardization bodies conducted work in this field that aims at the delivery of suitable concepts. Especially the *NISTIR 7628 Guidelines for Smart Grid Cybersecurity* [17] and the work conducted by the European *Smart Grid Coordination Group* among the European Mandate M/490 [18–21] are well-known and of great importance. However, the proposed considerations are kind of theoretic and take place on a rather high level. Thus, the *implementation* of these concepts during development represents a challenge.

The relation between the two mentioned issues *complexity* and *security* has been brought to the point by Schumacher *et al.*: *"Complexity is the worst enemy of security"* [22] (p. 531). In the foreword of this publication the authors argue that *"[...] security is often afterthought in system design and implementation. The enterprise context and requirements that drive system security are not addressed explicitly, and are not incorporated into system architectures"* [22]. To manage the integration of security in complex systems "by design", the authors relate security with the *Systems Engineering* approach. This approach is well-known in other safety critical domains such as automotive, aeronautics or defense. One of the main goals of this approach is to consider a complex system "as a whole" by involvement of all stakeholders with their particular interests rather than working on isolated aspects.

As outlined by Haberfellner *et al.* [6], *Systems Engineering* goes hand in hand with the concepts of *Model Driven Engineering* (MDE) [23–26]. This approach utilizes formally described and prescriptive (a-priori) models to handle the complexity of a system. By maintaining the fundamental principles of "Separation of Concerns" and "Divide and Conquer" particular concerns of different stakeholders can

be considered individually by still maintaining a complete model of a system. This aspect is of great importance for interdisciplinary development.

One major concern in context of MDE is the selection of an appropriate modeling language. Today, *General Purpose Languages* (GPL) such as UML [27,28] or SysML [29] are widely spread. Much more, extensions of these languages exist that aim at emphasizing particular aspects. For example, in terms of security, UMLsec [30] or the more recently introduced SysML-sec [31] can be mentioned.

A contrasting approach aims at the utilization of *Domain Specific Languages* (DSL) with a dedicated focus on a particular application domain [32–35]. In context of Smart Grids, the concept of DSLs is of special importance. The expected, dynamic characteristic of the Smart Grid will introduce new tasks to utilities. They need to develop appropriate skills for developing, operating and maintaining complex systems. Established concepts such as UML or SysML for sure deliver the necessary capabilities. However, for organizations with no software background, the application of these concepts is rather challenging which limits acceptance dramatically. To successfully introduce MDE to the energy domain it is necessary to develop concepts that rely on the domain experts "natural language". Identifying such a common language for different stakeholders from the application domain can be rather difficult. To develop such a common language the proposed approach supposes work from standardization bodies to be the lowest, common denominator. Thus, one of the key aspects is to utilize domain specific architecture frameworks as basis for the development of a domain specific language.

From a scientific point of view, the incorporation of models for Smart Grid development is subject of different work. Several approaches exist that rely on models for investigating selected aspect such as the integration of certain data models or protocols [36,37]. In contrast, other approaches consider utilization of models on a rather high level. For example, Lopes *et al.* [38] or Kaitovich *et al.* [39] discuss the utilization of models for the purpose of managing complexity. However, both of these two approaches rely on SysML as modeling language. *Domain specific* concepts that allow to (1) develop Smart Grid solutions in respect to reference architecture models from standardization bodies and (2) integrate corresponding security concepts (such as the NIST Guidelines) *by design* are missing.

The paper at hands aims at bridging the gap between *Systems Engineering* and *Smart Grids* in order to tackle *complexity as worst enemy of security*. Thus, a domain specific Systems Engineering concept is proposed. This Systems Engineering concept relies on a dedicated DSL that is developed in reference to Smart Grid standardization work.

The main contribution of this approach is a conceptual framework for Smart Grid specific Systems Engineering that consists of five building blocks. The first building block comprises a development process that aligns engineering of Smart Grid systems with the concepts from *Model Driven Architecture* (MDA) as specified by the *Object Management Group* (OMG) [40]. The second building block states the backbone of this approach. It comprises a DSL that is obtained on basis of the European *Smart Grid Architecture Model* (SGAM) [20] and the American *NIST Logical Reference Model* (NIST LRM) [17]. Building block three is of major importance for the realization of *security by design*. It comprises the development of a *Reference Architecture Model* (RAM) on basis of the NIST LRM and with incorporation of the corresponding security concepts. To complete the picture of MDE, building block four illustrates how to evaluate particular architecture models on basis of certain *Key Performance Indicator* (KPI). Moreover, building block five discusses how to bridge the gap between system architecture and component realization by linking architecture models with particular implementation frameworks. This building block is quite visionary and rather aims at outlining the idea than presenting a concrete implementation.

The remainder of this paper is structured as follows. Section 2 briefly describes the most relevant concepts from standardization bodies that contribute to the presented approach. An overview of the overall concept together with a brief description of the particular building blocks is given in Section 3. Section 4 goes one step beyond as it delivers a more detailed insight in the present state of implementation for all building blocks. To demonstrate the application of the DSL as cornerstone of the approach, Section 5 gives a complete example. Finally, Section 6 concludes this paper with a

short summary, a discussion on the experiences made during application and an outline of future work required.

2. Related Work

As motivated in Section 1, the goal of the presented approach is to provide a *domain specific engineering framework* with respect to *architectural concepts* from standardization bodies. Moreover, contributions from both, American and European *standardization bodies* are incorporated into a *Domain Specific Language* which states the backbone of this approach. Even if already some DSL based approaches exist [41,42], to our best knowledge we are not aware of any other approach that obtains a Smart Grid specific DSL on basis of standardization work. Or, as presented in this approach, on basis of a combination of both, standardization work from Europe (SGAM) and America (NIST LRM). However, according to the focus of this paper, this section briefly introduces various concepts proposed by standardization bodies that contribute to the presented approach.

2.1. IEC 62559-2 Use Case Template

In the field of software engineering the utilization of *Use Cases* is state-of-the-art since the late 90s. Use Cases provide a common methodology for a formal description of functionality, especially when interactions are in focus. A widespread approach is the utilization of the *Unified Modeling Language* (UML) that integrates Use Case modeling. Other domains, however, lack a common concept for such specifications.

As the field of Smart Grid deeply relies on interactions, the IntelliGrid EPRI domain adopted this concept and a formal glossary and categorization for requirements of domain-specific utility communication solutions has been provided [43]. On basis of feedback from utility experiences the *IEC 62559-2 Use Case Template* has been proposed [44,45]. This template gives a framework for a formal description of particular Smart Grid related Use Cases. By now, this template finds broad acceptance within the community. Also, adequate tool support has been developed. A very well-known example is the DKE hosted *Use Case Management Repository* (UCMR) that has been used for common development and information sharing during the work conducted among the European Standardization Mandate M/490 [19].

However, even if guidance for a formal description of Use Cases is given, different drawbacks exist. First, the template is very exhaustive and thus, the description of particular Use Cases is quite time consuming. Second, the template does not provide information about the abstraction level and thus, the granularity of Use Cases developed on this basis differs strongly. For an efficient application of this template additional guidance is necessary. For example, this template could be used to develop Use Cases on a rather high abstraction level whereas classic UML constructs could be applied for developing more low level Use Cases.

2.2. Smart Grid Architecture Model (SGAM)

In the recent past notable efforts have been made in terms of Smart Grids standardization. The *Smart Grid Coordination Group* (SGCG), driven by the European standardization bodies *European Telecommunications Standardization Institute* (ETSI) and *European Committee for Electrotechnical Standardization* (CENELEC) has been issued with the European mandate M/490. Under the umbrella of this mandate, a holistic view on the present standardization landscape should be elaborated in order to identify standardization gaps. In late 2012, the results of four working groups were published [18–21].

A very crucial aspect of this work was the formalization of a certain architecture model, capable of providing a standardized decomposition of Smart Grid systems with a focus on interoperability. As starting point for the specification of such an architecture model the well-known *NIST Conceptual Model* [46] has been chosen. This conceptual model denotes the particular domains of the electric energy system and depicts the interrelations in between with a focus on both, electric connections and

logical interactions. To address the arising changes of the energy system this model has been extended by a *Distributed Energy Resources* (DER) domain as depicted in Figure 1.

For integration of ICT components, responsible for controlling the particular domains of the energy conversion chain, the extended domain model has been combined with the SCADA automation pyramid. This combination yields the Smart Grid plane as depicted in Figure 2. The Smart Grid plane is spanned by the *Domains* axis that decomposes the electric energy conversion chain and the *Zones* axis which provides a hierarchical view on the information management. In [20] the particular domains are described as follows.

- **Bulk Generation:** Representing generation of electrical energy in bulk quantities, such as by fossil, nuclear and hydro power plants, off-shore wind farms, large scale solar power plant (*i.e.*, PV, CSP)—typically connected to the transmission system;
- **Transmission:** Representing the infrastructure and organization which transports electricity over long distances;
- **Distribution:** Representing the infrastructure and organization which distributes electricity to customers;
- **DER:** Representing distributed electrical resources directly connected to the public distribution grid, applying small-scale power generation technologies (typically in the range of 3 kW to 10,000 kW). These distributed electrical resources may be directly controlled by DSO;
- **Customer Premises:** Hosting both-end users of electricity, also producers of electricity. The premises include industrial, commercial and home facilities (e.g., chemical plants, airports, harbors, shopping centers, homes). Also generation in form of e.g., photovoltaic generation, electric vehicles storage, batteries, micro turbines, *etc.*, are hosted.

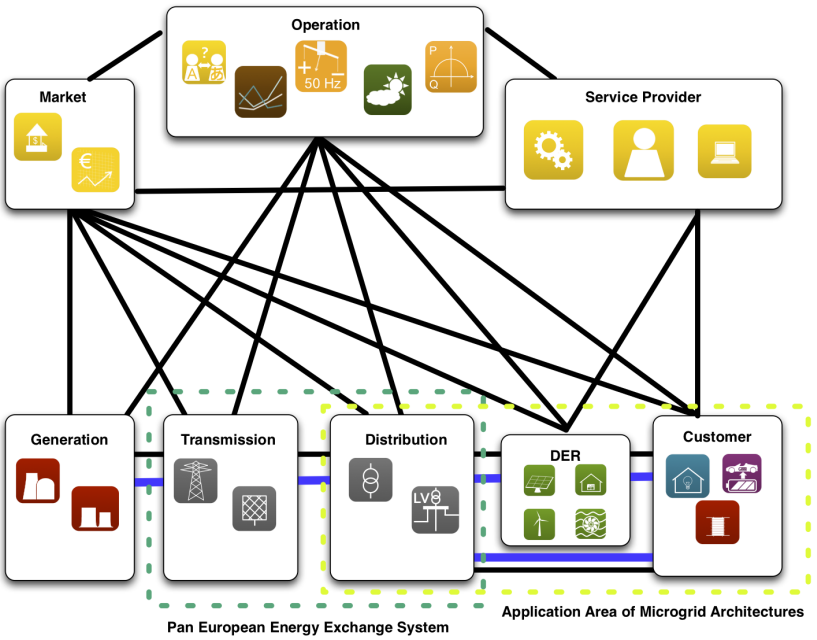

Figure 1. EU extension of the National Institute of Standards and Technology (NIST) Conceptual Model based on [20].

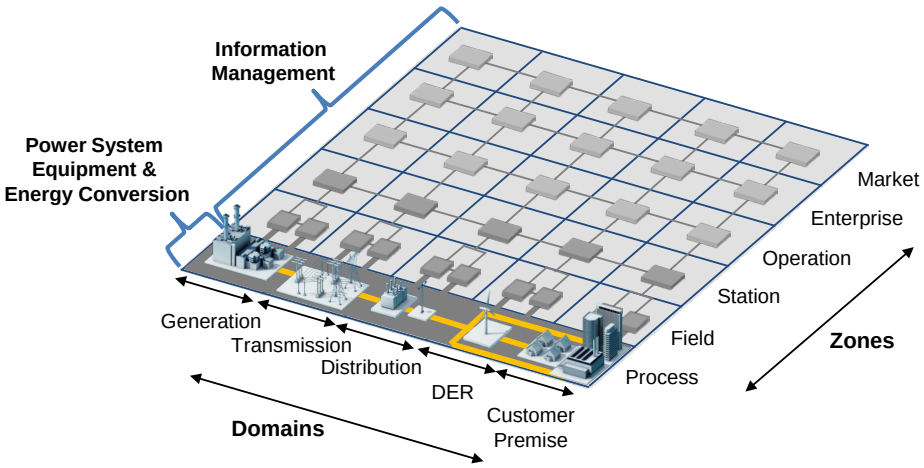

Figure 2. Smart Grid plane spanned by domains and hierarchical zones, reproduced with permission from [20]. Copyright 2012, European Committee for Standardization (CEN).

Also, a detailed description for the integrated zones is given [20].

- **Market:** Reflecting the market operations possible along the energy conversion chain, e.g., energy trading, mass market, retail market, *etc.*
- **Enterprise:** Includes commercial and organizational processes, services and infrastructures for enterprises (utilities, service providers, energy traders, *etc.*), e.g., asset management, logistics, work force management, staff training, customer relation management, billing and procurement, *etc.*
- **Operation:** Hosting power system control operation in the respective domain, e.g., distribution management systems (DMS), energy management systems (EMS) in generation and transmission systems, microgrid management systems, virtual power plant management systems (aggregating several DER), electric vehicle (EV) fleet charging management systems.
- **Station:** Representing the areal aggregation level for field level, e.g., for data concentration, functional aggregation, substation automation, local SCADA systems, plant supervision, *etc.*,
- **Field:** Including equipment to protect, control and monitor the process of the power system, e.g., protection relays, bay controller, any kind of intelligent electronic devices which acquire and use process data from the power system.
- **Process:** Including the physical, chemical or spatial transformations of energy (electricity, solar, heat, water, wind, *etc.*) and the physical equipment directly involved. (e.g., generators, transformers, circuit breakers, overhead lines, cables, electrical loads any kind of sensors and actuators which are part or directly connected to the process, *etc.*).

The Smart Grid plane described above provides guidance for decomposing and structuring certain Smart Grid systems but does not yet cover any interoperability aspects. To also address this crucial issue, a layer concept has been introduced on basis of the GWAC Interoperability Stack [47]. Basically, the Smart Grid plane has been projected on five layers that emphasize certain interoperability aspects. The result is the so called *Smart Grid Architecture Model* (SGAM) as depicted in Figure 3. The basic idea of the SGAM is to provide a framework for describing Smart Grid systems on all five layers in order to provide a certain level of completeness. A brief description on the individual *Interoperability Layer* is given in the following on basis of [48]. A more detailed description of the underlying concepts can be found in [20].

- **Business Layer:** Provides a business view on the information exchange related to Smart Grids. Regulatory and economic structures can be mapped on this layer.
- **Function Layer:** Describes functions and services including their relationships from an architectural viewpoint.

- **Information Layer:** Describes information objects being exchanged and the underlying canonical data models.
- **Communication Layer:** Describes protocols and mechanisms for the exchange of information between components.
- **Component Layer:** Physical distribution of all participating components including power system and ICT equipment.

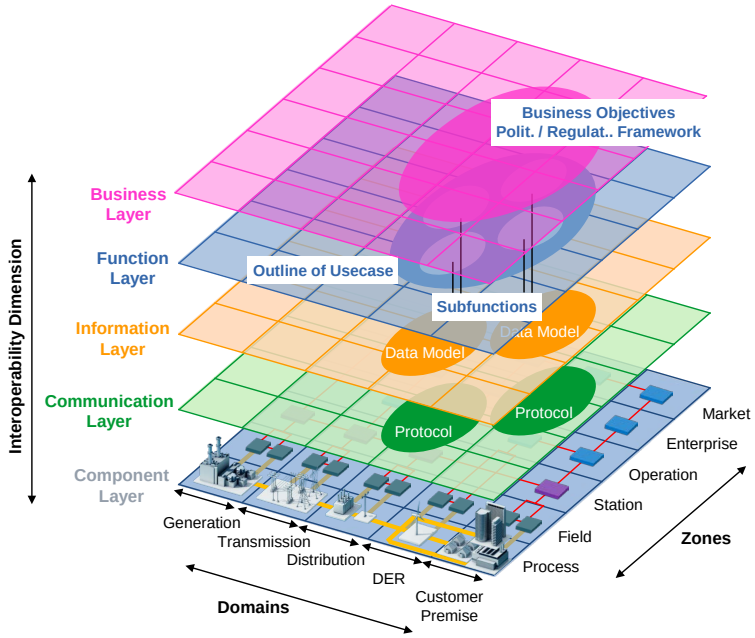

Figure 3. Smart Grid Architecture Model (SGAM), reproduced with permission from [20]. Copyright 2012, European Committee for Standardization (CEN).

Even if the original intention of the SGAM was to identify gaps in standardization, it gained momentum by being used as general architecture framework in several projects. However, even if the clear structure of the SGAM raises acceptance, there is only little guidance on its application. The originally described *Use Case Mapping Process* (UCMP) [20] is intended to identify standardization gaps and thus is only suitable in a limited way for architecture development. In addition, the structure of the SGAM is on a very abstract level and no guidance is given on detailing particular systems.

However, despite the listed drawbacks, the SGAM delivers a very valuable entry point when decomposing and architecting Smart Grid systems. Thus, it finds broad acceptance and is widely used as architecture framework by now.

2.3. NIST Logical Reference Model

The issue of Smart Grid Cybersecurity has been explicitly addressed by the American *National Institute of Standards and Technology* (NIST). In 2010 the first version of the *NISTIR 7628 Guidelines for Smart Grid Cybersecurity* has been published [17]. These guidelines deliver a concept for integration of Security Requirements within Smart Grid solutions together with an initial set of security requirements.

The underlying work is based on a detailed analysis of several existing Use Case collections. On basis of these Use Cases certain Smart Grid actors have been identified and described and were mapped onto the particular domains of the NIST Conceptual Model as discussed earlier (Figure 1). Moreover, interrelations between certain actors have been investigated and communication paths were specified on basis of unique interfaces. The combination of all actors and their associated interfaces yields the *NIST Logical Reference Model* (NIST LRM) as depicted in Figure 4.

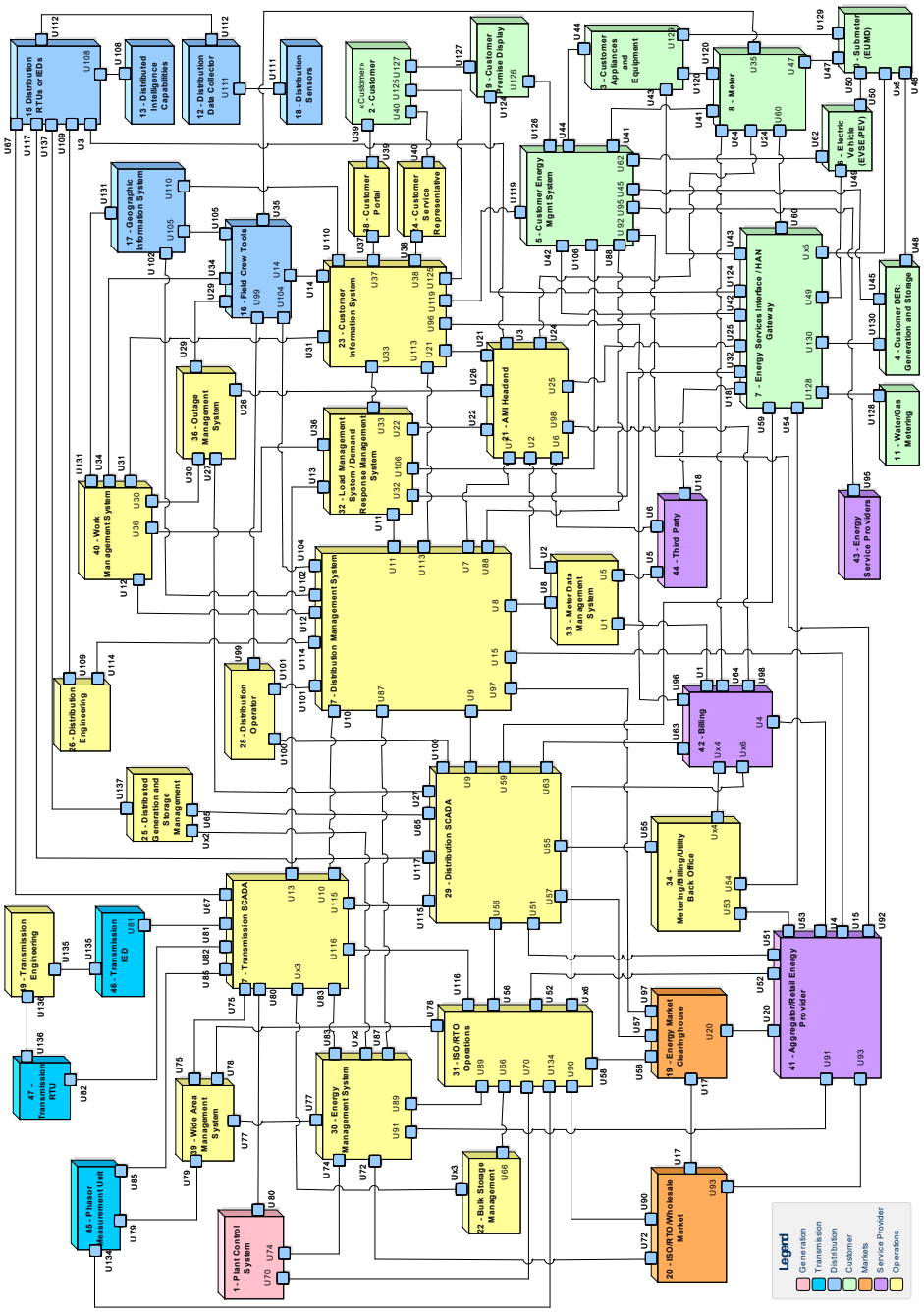

Figure 4. NIST Logical Reference Model based on [17].

The NIST LRM with its actors and interfaces has been used as a basis for further considerations on security. In a first step, different *Interface Categories* were formulated on basis of similar characteristics, specified as *Security Attributes*. Next, for each of the Interface Categories particular security requirements were elicitated. In total, 197 Security Requirements could be identified that were assigned to one of the following groups:

- Common Governance, Risk and Compliance Requirements;
- Common Technical Requirements;
- Unique Technical Requirements.

Finally, all of the interfaces from the NIST LRM were assigned to one or more of the specified Interface Categories. Thus, by following the path *Actor* → *Interface* → *Interface Category* → *Security Requirements* the particular Security Requirements for a certain actor can be obtained. For a better understanding, the conceptual model of the NIST LRM is depicted in Figure 5.

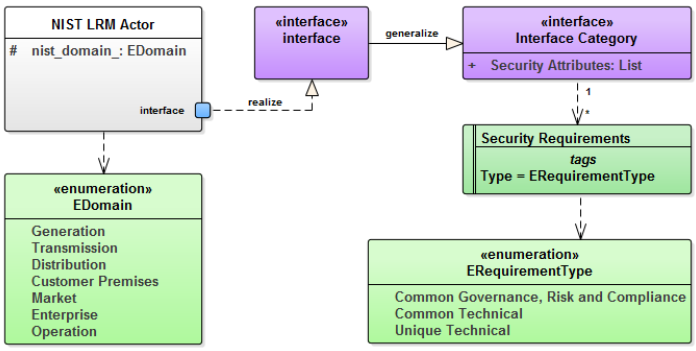

Figure 5. NIST Logical Reference Model (LRM) conceptual model.

Even if the NIST LRM does not claim completeness, it provides a very valuable starting point for security considerations. Considering both, the earlier discussed SGAM and the NIST LRM different aspects can be mentioned. First, the SGAM rather presents an architecture framework than a specific solution. Thus, it provides a *context* for the development of Smart Grid systems. Contrasting to this, the NIST LRM aims at being a *reference architecture* that can serve as a blueprint. As both of these concepts originate from the NIST Conceptual Model an alignment is possible by modeling the NIST LRM in context of the SGAM. However, different drawbacks exist as for example the NIST LRM has a "flat" nature and does not—in contrast to the SGAM—differentiate between business-, logical- or technical aspects. The alignment of these two concepts states a cornerstone of the presented approach and is discussed in more detail in the following sections of this paper.

3. Conceptual Approach

As outlined in Section 1, the integration of *Security by Design* requires a holistic and consistent engineering methodology. Such a methodology needs to be able to address the particular needs of all involved stakeholders over the whole development process. Moreover, a unified process with well-defined artefacts is required that allows for the integration of standardized components and best practice solutions.

In this Section, first the requirements for such an engineering methodology are discussed. Next, a conceptual framework is presented that addresses the particular requirements. The conceptual framework delivers several building blocks which are briefly described. A detailed description of all building blocks together with their realization, application and discussion is given in the subsequent Section.

3.1. Requirements

The requirements for a holistic engineering methodology cover a vast field. First, the methodology needs to be able to address the particular needs of all involved stakeholders such as *Business Analysts*, *Domain Experts* or *Engineers*. Thus, a process is required that gives guidelines along the essential development phases *System Analysis*, *System Architecture* and *Design and Development*.

To enable a consistent development it is essential to be equipped with a common, domain specific language for creation of architectural models. On basis of such a language also best practice solutions such as typical Use Cases or a certain Reference Architecture can be described. Moreover, the availability of a consistent model can serve as a basis for evaluation of architectural solutions.

For example, by integrating individual attributes such as CAPEX or OPEX to particular model elements, sevaral evaluations such as "which costs are related with a certain Use Case?" can be made. By implementing such a domain specific language within a modern modeling tool with some scripting capabilities, some kind of automation can be envisioned as well.

A more precise definition of the requirements for a holistic engineering approach is given in the following:

- Consistent and seamless development process covering *Business Analysis*, *Functional Analysis*, *Architecture Development* and *Implementation*;
- Integration of *Security by Design* over all development phases;
- Domain Specific Modeling Language (DSL) for development of architecture models;
- Utilization of best practice solutions (e.g., standard Use Cases or Reference Architecture) on basis of a common DSL;
- Capability for semi-automatic architecture evaluations;
- Conformity with and guideline for usage of existing standards;
- Seamless integration with implementation framework for particular systems.

3.2. Building Blocks

In order to address the preliminary introduced requirements, a Conceptual Framework built upon the five building blocks has been specified:

- Development Process
- Domain Specific Architecture Framework
- Best Practice Solutions
- Architecture Evaluation
- Implementation Framework

The individual building blocks and their interrelations are depicted in Figure 6, a brief overview is given in the following.

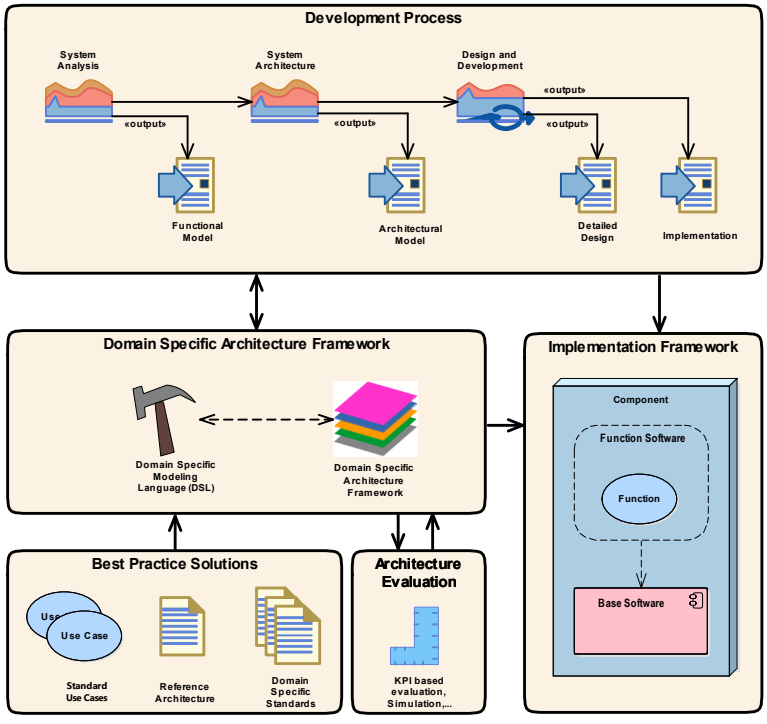

Figure 6. Conceptual Approach.

The **Development Process** building block states the backbone of the Conceptual Approach. It aims at giving guidance through the whole development process. To address the individual needs of all involved Stakeholders, it separates the development process into the three phases

- System Analysis Phase
- System Architecture Phase
- Design and Development Phase

The *System Analysis Phase* aims at defining the needed functionality for a particular system. Thus, it yields a *Functional Model*. To derive the desired functionality, in a first step (*Business Analysis and Risk Assessment*) the underlying business case is investigated which clarifies the motivation for a system. In addition, during this step eventually occurring regulatory constraints can be considered. Another important step during the phase is the execution of a *Risk Assessment*. Detailed considerations on such Risk Assessments for Smart Grids can be found for example in [21]. The way of pursuing such a Risk Assessment strongly relies on the system to be built and thus, no more considerations on this are given in this paper. However, it is very important to perform such an assessment in this very early stage in order to identify particular architectural drivers for further considerations.

The System Analysis Phase typically involves *Business Analysts*, *Domain Experts* and *Requirements Engineers* and focuses rather on required functionality than on technical solutions.

Subsequent to the System Analysis Phase the *System Architecture Phase* aims at the development of a domain specific system architecture. By putting focus on domain specific aspects, the abstraction level of the resulting *Architecture Model* is quite high. Basically, it represents a "black-box" model that aims at describing particular subsystems and their interactions without mentioning their inner composition. The description of the particular components comprises their functionality, interfaces and non-functional requirements such as security. During this stage of development system architects with appropriate domain know-how play a crucial role.

Contrasting to the "black-box" model delivered by the System Architecture Phase, the *Design and Development Phase* aims at the realization of the preliminary modeled system. Thus, the particular components are developed by first creating a detailed design and second, the corresponding implementation. This task typically is executed by the supply chain (developing of particular components) and system integrators (realizing the distributed system as a whole).

However, even if the mentioned development phases are clearly separated, it is necessary to provide a unified and consistent development environment that allows for example a traceability of certain (System level) requirements down to particular components or development artefacts. The necessity for such a unified development environment stresses the need for a holistic and consistent Domain Specific Architecture Framework.

The **Domain Specific Architecture Framework** building block delivers a framework for a consistent development over all development phases as mentioned before. To maintain consistency, the framework is suited to deliver a model that is separated in a *functional model* and an *architecture model* with explicit model transformations in between. Moreover, to raise acceptance of all involved stakeholders, this model should be created on basis of a *Domain Specific Language* (DSL). The presented approach introduces such a DSL that is specified on basis of the SGAM concepts as discussed in Section 2. The availability of such a well-defined modeling language enables the integration of best practice solutions or a model based assessment of particular architectural solutions. For example, different concepts can be evaluated in respect to costs, complexity or security weak points.

To not reinvent the wheel, the building block **Best Practice Solution** introduces the possibility for reusing existing work. In this case, "existing work" does not necessarily mean to integrate complete solutions. Instead, rather the integration of particular aspects is considered. For example, Use Cases created by the community or communication infrastructure designed for a certain purposes can be reused. In the presented approach the reuse of three particular aspects is considered. First of all, during the System Analysis Phase the integration of broadly agreed Use Cases is enabled. Second, during

the System Architecture Phase a possibility is given to rely on Smart Grid Reference Architectures (including security aspects) rather than to start from scratch. Finally, also during System Architecture Phase, guidance is given on how to choose appropriate data model standards or communication protocols. Requiring such a modular concept draws the need for having such reusable elements described on basis of the presented framework. However, in the given approach this has been done vice versa as the framework is built upon such reusable artefacts.

The concepts delivered by the prior described building blocks deliver a complete and consistent model of an overall system architecture. The availability of such a model introduces the possibility for evaluation of particular solutions by utilization of model-based assessments. Thus, the building block **Architecture Evaluation** is introduced. Even if the technology necessary is delivered by the framework, the (semi-)automatic assessment of models is a quite difficult task that requires further considerations. However, to demonstrate the feasibility the presented approach delivers a proof of concept on how to exploit existing models to gain specific insights. To be more precise, information flows are extracted and serve as a basis for evaluation of potential privacy implications. As information items are a critical asset in terms of security, these concepts could also be used to determine the criticality of particular components or interfaces within the architecture. Another approach could be to utilize the extracted data-flow graphs for realization of intrusion detection on basis of process mining.

The building blocks described so far deliver a concept for a holistic development and evaluation of standards based system architectures. However, there is still a major gap between architecture and implementation that has to be bridged. Thus, a concept for a seamless integration between these two artefacts is introduced by the building block **Implementation Framework**. To allow for such a seamless integration a concept on basis of experiences made in the field of automotive engineering is proposed. In automotive engineering, a wide spread approach for developing control systems is the utilization of the *Automotive Open Software Architecture* (AUTOSAR) framework www.autosar.org. This framework separates a control systems software into two parts. The lower part, denoted as *base software* covers hardware specific details and delivers a standardized interface to particular low level functionalities such as communication stacks. By having such a standardized runtime environment with well-defined interfaces, domain experts (e.g., from mechanical engineering) can focus on the development of functionality (*function software*) without mentioning technical details. This clear separation of concerns provides the basis for sophisticated development concepts such as code generation.

The presented approach proposes a similar concept denoted as *Energy Open System Architecture Framework* (ENOSAR). This implementation framework, similar to AUTOSAR, describes a base software that covers domain specific aspects such as communication protocols or data model standards. In addition, privacy and security related functionalities are provided that can be invoked by the above situated function software.

To bridge the gap between architecture and implementation, the function software is closely related with the functional model as developed during the System Analysis phase. Furthermore, the black-box specification of the architecture model is linked with the configuration of the base software. For example, the chosen communication protocols or data model standards drive the configuration of the base software. Or, on basis of the specified requirements, particular privacy or security mechanisms can be invoked.

The presented implementation framework is rather visionary and in a very early stage. However, a first proof of concept implementation is already available and under evaluation in context of the RASSA Reference Architecture for a Secure Smart Grid in Austria, FFG Project No. 848811 research project. To be more precise, this framework is used to develop different Smart Grid functionalities within the Customer Premises with a special focus on security and privacy.

The outlined Conceptual Framework introduces and describes individual building blocks necessary for a holistic engineering of Smart Grid systems. Such a holistic approach is deemed crucial to enable and maintain *Security by Design*. The given description aims at drawing the big picture of the overall concept and to emphasize the interdependencies between the individual building blocks.

A detailed discussion on the implementation of these building blocks and their degree of maturity is given in the following section.

4. Implementation

4.1. Development Process

As outlined in the previous section, a holistic engineering process is required that allows developing Smart Grid systems in context of a domain specific framework. The presented approach relies on the SGAM as framework and thus, the development process needs to be aligned with this concept. The original SGAM proposal already delivered the *Use Case Mapping Process* (UCMP) [20] that describes how to depict particular systems in context of the SGAM. However, as the UCMP is intended to be used for identification of standardization gaps (e.g., business analysis was done in the end) it was necessary to rethink it for supporting the development of new systems.

In the presented approach, the development is aligned with the *Model Driven Architecture* (MDA) concept as described by *Object Management Group* (OMG) [40]. Basically, the MDA aims at separation of different development aspects and thus defines artifacts over four different layers with specific transformations in between. First, the *Computational Independent Model* (CIM) delivers a functional description of a particular system without mentioning any technological aspect. In the presented process the CIM is yielded by the System Analysis Phase. The second level aims at the decomposition of a particular system by developing a *Platform Independent Model* (PIM) that is not yet linked to any specific technology or platform. The PIM is delivered as outcome of the System Analysis Phase. The technology specific realization is described by the *Platform Specific Model* (PSM) that furthermore serves as basis for the *Platform Specific Implementation* (PSI) as final artifact. These two artifacts are the outcome of the final phase, the Design and Development Phase.

In context of "domain specific" development, the first two artifacts are closely related with the application domain whereas the last two rather require technological based considerations. Thus, the creation of the PSM and the PSI can be done by typical concepts from Systems- or Software engineering whereas the development of the CIM and the PIM should rely on the application domain. In alignment with the SGAM the CIM is related with the upper two layers (Business- and Function-Layer) whereas the PIM corresponds to the Information-, Communication- and Component-Layer. The mapping of the three development phases with the MDA related artifacts can be found in Figure 7, a detailed description of the particular tasks within the development phases is given in the following.

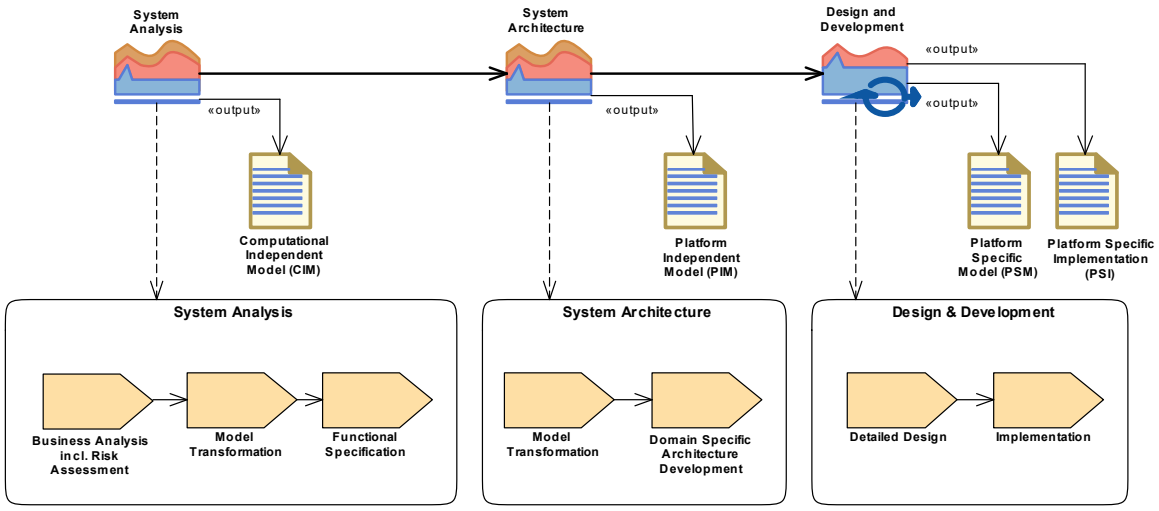

Figure 7. Development Process.

4.1.1. System Analysis Phase

The System Analysis Phase aims at the definition of particular functionalities to be delivered by a certain system. As outlined above, the System Analysis phase yields the *Computational Independent Model* (CIM). To provide a structured approach, the CIM itself is divided into two parts. First, a *Business Model* is used to analyze particular aspects and second, a *Functional Model* describes the necessary functionalities. In terms of the SGAM, the Business Model corresponds to the SGAM Business Layer whereas the Functional Model is represented by the SGAM Function Layer.

The first task of the System Analysis Phase is the **Business Analysis incl. Risk Assessment**. This task aims at identification of involved *Business Actors* (physical or legal persons, also comprising regulatory entities) and their particular *Business Goals*. Moreover, *Business Cases* should be described that are realized in order to balance the needs between certain Business Actors. The description of these Business Cases can be done in any appropriate manner, e.g., by means of *Business Process Modeling Notation* (BPMN).

Another very important task of the Business Analysis task is the execution of a *Risk Assessment*. This Risk Assessment should deal with scenarios for non-fulfillment of the described Business Case. Particular considerations on Risk Assessments for Smart Grids, especially with a focus on security, can be found for example in [21]. However, as the extent and type of Risk Assessments strongly rely on a particular system to be considered, no further specifications are given here. It only has to be mentioned that an adequate risk assessment has to take place in this very early stage to identify particular architectural drivers. Moreover, the results of the Risk Assessment can serve as a basis for further refinement of particular quality requirements such as "security".

Subsequent to the Business Analysis, the **Model Transformation** as second task can be conducted. During this step, particular *High Level Use Cases* (HLUC) can be derived. These HLUC cover functionality necessary to fulfill the originating Business Case. Furthermore, *Logical Actors* are derived from the Business Actors. These Logical Actors can be interpreted as logical entities involved in a particular HLUC. From a technical point of view a *Model Transformation* from the Business Model into the Functional Model is conducted. Thus, the Logical Actors together with the HLUC represent the content of the Functional Model.

Right after the identification of particular HLUC, the above mentioned risk assessment can be used to specify certain *quality requirements*. Due to the differing nature of particular Use Cases no final set of categories for quality requirements can be given and the attributes to be mentioned have to be considered individual. However, in terms of dependability for Smart Grid systems various suggestions exist. For example, in [49] it is proposed to make use of the categories *Privacy, Reliability, Availability, Maintainability, Safety* and *Security* denoted by the acronym *P-RAMSS*.

For a better understanding, the resulting artifacts and their relations are illustrated by a simplified example in Figure 8.

Two *Business Actors* ("Distribution System Operator" and "Customer") with their individual *Business Goals* are involved in the *Business Case* "Flexibility". A risk assessment, dealing with scenarios like "What happens if the Business Case fails?" is conducted as well. All these actors, together with the risk assessment (as attached document) are situated on height of the SGAM Business Layer. During the Model Transformation certain *High Level Use Cases* invoked by the Business Case are identified. In the example, one single HLUC "Condition Monitoring" is depicted. Furthermore, the Business Actors have been *transformed* into the Logical Actors "Distribution System Management" (DSM), "AMI Headend" and "Smart Meter". This model transformation is indicated by *trace* relations between Business- and Logical Actors. In a final step, the quality requirements for the HLUC are elicitated. In the example, only two requirements ("Privacy" and "Security") are used to indicate the integration of quality requirements. However, in real projects requirements elicitation typically delivers a tree of requirements. Thus, it is a common praxis to only depict the root of the individual requirement trees on this level.

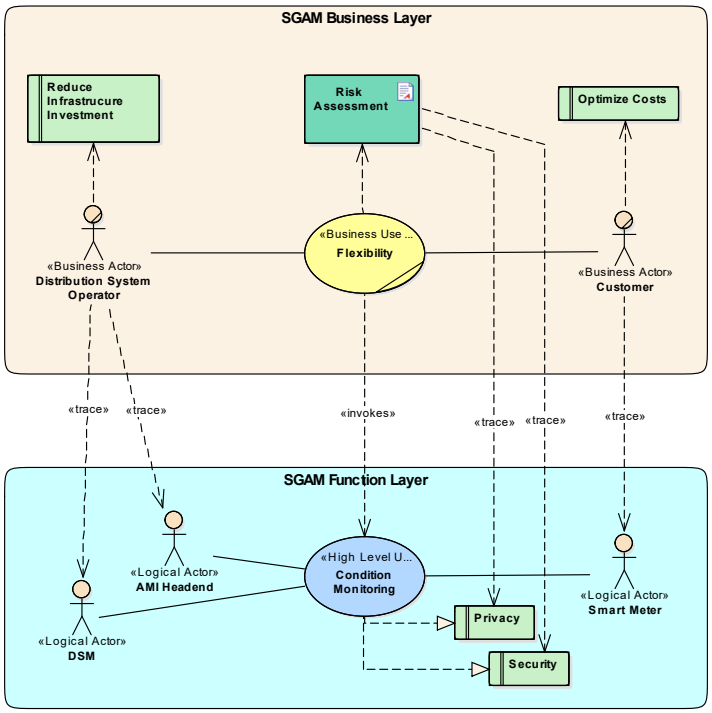

Figure 8. Artifacts of the Computational Independent Model (CIM).

The third task of the System Analysis Phase, the **Functional Specification** deals with the detailed description of the preliminary identified HLUC. The proposed approach suggests utilizing a staged approach. First, the identified HLUC can be described on basis of the IEC 62559-2 Use Case template [45]. Next, by utilization of the concepts from [19], each HLUC is decomposed into several *Primary Use Cases* (PUC). The detailed description of these PUC can be done by means of standard UML mechanisms such as *Sequence Diagrams* or *Activity Diagrams*. To provide a consistent description, a best practice for modeling PUCs is provided. Besides the plain description of functionality, this method aims at identification and description of particular *Information Objects* that are used during interactions. This is important for two reasons: First, the Information Objects transmitted provide a basis for the later development of the SGAM Information Layer and second, in terms of security, Information Objects represent a key asset for further considerations.

Typically, development of Use Cases starts with a natural language description. Transferring such natural language descriptions into more formal models can be a rather time consuming task which limits acceptance dramatically. Thus, an efficient method for obtaining formal descriptions is of great value. Modern UML modeling tools such as *Enterprise Architect* www.sparxsystems.com (which is used later on for implementation of the DSL) are able to increase efficiency by providing features such as *diagram generation*. In the following, a suggested process is described how to efficiently develop formal descriptions of PUCs by utilization of UML *Sequence Diagrams* and *Activity Diagrams*.

In the first step, natural language based Use Case descriptions are rewritten in a semi-formal way by relying on a simple language pattern. This pattern comprises individual sentences, with every sentence being associated with one single Use Case step. Every sentence (or Use Case step) represents either an individual action related to one single actor or an interaction between two actors. The corresponding descriptions for these two possibilities are defined as follows:

- *<Actor 1> <performs action>*
- *<Actor 1> <transmitts information | triggers Action> <to|on> <Actor 2>*

The modeling tool being used allows an import of such a semi-formal, textual description as *structured scenario* (detailed description of Use Cases). Moreover, it semantically parses these sentences

and identifies (on basis of the names) actors involved. Furthermore, it is able to generate *Activity Diagrams* and *Sequence Diagrams* out of this description.

The generated diagrams can be used for further considerations on information being exchanged. First, the Activity-Diagram can be analyzed and *Information Objects* to be transmitted can be modeled. Furhtermore, these Information Objects can be integrated within the Sequence-Diagrams which yields a complete model of all information exchanges for a particular PUC. Again, having the individual Information Objects modeled is an essential step towards consistency during development and also provides a valuable basis for further privacy and security considerations. Another very important aspect to be mentioned is the creation of a complete picture for a certain HLUC as during the description of the information flows the same Logical Actors are used for every PUC. Thus, the other way round, for every Logical Actor it is defined, in which communication it is involved and which information objects are transmitted.

For the sake of clearness, the described concept shall be demonstrated on a simplified example. It is assumed that the HLUC ("Condition Monitoring") from the previous example invokes several PUCs such as "Fetch Transformer Condition" or "Fetch Customer PQ Data" with *PQ* referring to *Power Quality*. The "Fetch Customer PQ Data" PUC comprises a communication between the Logical Actors "Distribution System Management" (DSM), "Customer Energy Management System" (CEMS) and Smart Meter. The semi-formal, textual description of this Use Case looks as follows:

- DSM transmits a PQ-Data Request to CEMS
- CEMS transmits a PQ-Data Request to Smart Meter
- Smart Meter transmits PQ-Data to CEMS
- CEMS transmits PQ-Data to DSM

Next, the textual description can be imported by the modeling tool and both, the *Activity* and the *Sequence Diagram* can be generated automatically. At this time a distinction with UML has to be explained. In UML, actors represent *external* actors interacting with a *system*. Thus, for example, sequence diagrams depicting interaction between actors are not supported by UML2.x. However, in the context of Smart Grids, the definition of "system" and "actor" depends on the stakeholder. For the developer of the Smart Meter, for example, the Smart Meter is the System and the CEMS is an external actor. For the developer of the CEMS the situation is vice verca. However, to describe Smart Grid systems from a holistic point of view, the concept of *Logical Actors* has been introduced and the utilization of Sequence Diagrams appears as appropriate solution to document information exchange between Logical Actors. Another motiviation for integrating the concept of *Logical Actors* is given by the original SGAM description [20] which also utilizes actors as major element on the SGAM Function Layer.

However, on basis of the generated diagrams in a manual analysis step the *Information Objects* can be specified and integrated within the Sequence Diagram. All created artifacts for the description of this PUC are depicted in Figure 9. On the right side of this drawing the generated Activity Diagram can be seen with the identified and modeled *Information Objects* (purple). Furthermore, these Information Objects are attached to the generated Sequence Diagram (lower left of the image) which associates the Information Object flow between the involved Logical Actors. The described process enables a very efficient way of working. Executing these steps for every PUC a particular HLUC comprises, delivers a complete model for all Information Object Flows in between all involved Logical Actors. However, the described steps in total state a rather complex but crucial task for engineering. To provide additional guidance for this process a video tutorial is provided on the *SGAM Toolbox* homepage www.en-trust.at/SGAM-Toolbox that explains the workflow in more detail.

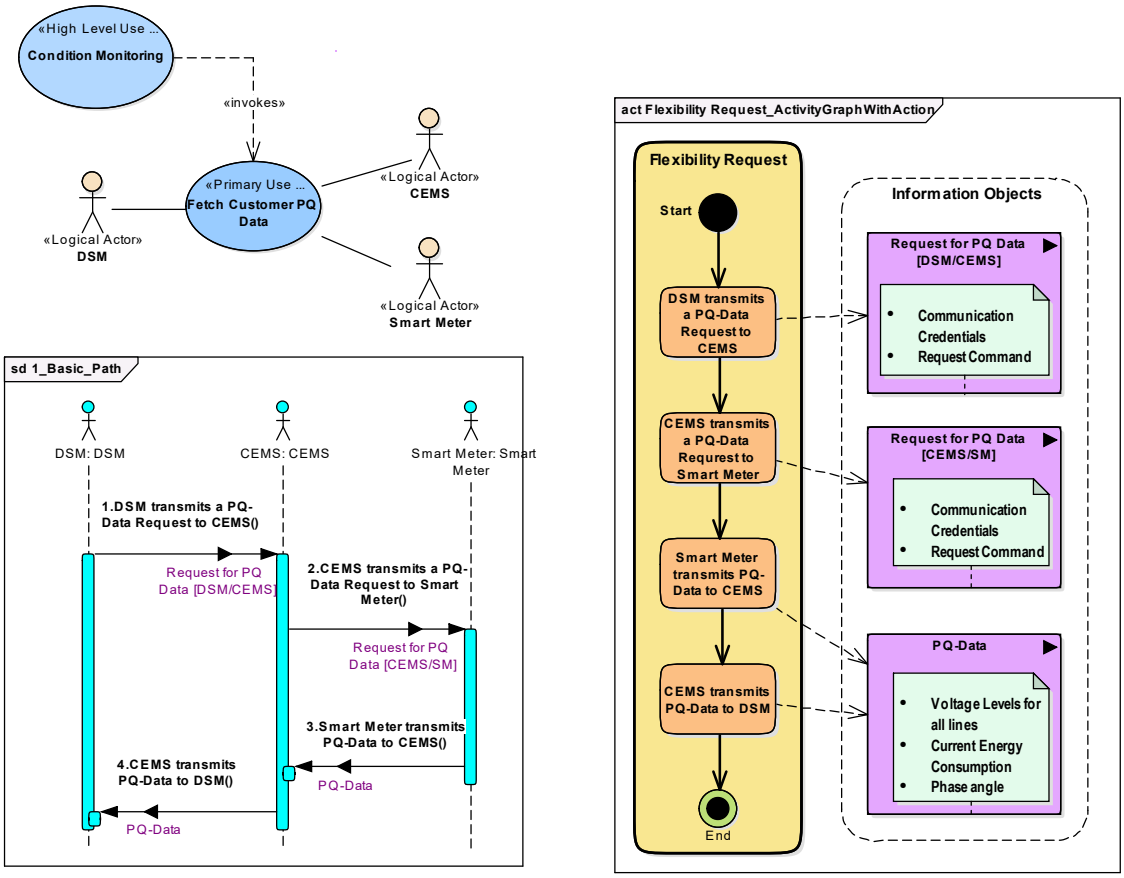

Figure 9. Detailed description of one single Primary Use Case (PUC).

4.1.2. System Architecture Phase

On basis of the identified functionality, the **System Architecture Phase** can be conducted. This phase aims at providing a technical concept for realizing the previously defined functionality. The focus of this phase is put on the identification and description of particular components and their interrelations. Thus, the output of this phase can be seen as black box model for all involved components with the description of the interrelations in between as key issue. Basically, the System Architecture Phase comprises the two tasks *Model Transformation* and *Domain Specific Architecture Development*.

As first task, the **Model Transformation** aims at mapping the prior described Logical Actors onto *Physical Components*. Making this step explicit is quite important as not always a one-to-one relation is given. For example, a particular Logical Actor such as a *Customer Energy Management System* (CEMS) can be realized by a combination of different physical components such as the CEMS itself together with a Communication Gateway. Also, it is possible to deploy different Logical Actors (especially in case of plain software realizations) on a single component such as a physical computer.

The second task of this phase, the **Domain Specific Architecture Development** aims at specifying the interrelations between particular components. These interrelations are described on basis of the lower three layer of the SGAM which comprises the *SGAM Information Layer* (information exchange from a logical perspective), *SGAM Communication Layer* (protocols used for communication) and *SGAM Component Layer* (description of the ICT networks used for communication). For specifying these interrelations, in a first step *Interfaces* are introduced and described in detail for every component. These interfaces form the basis for further considerations on privacy and security. Basically, the risk assessment from the System Analysis Phase is utilized to elicitated detailed privacy and security requirements for every interface of a particular component. The identified requirements

deliver guidance for choosing appropriate data model standards, communication protocols or ICT network topology.

For a better understanding, Figure 10 gives an overview on the described concepts on basis of the previously discussed example. First, the Model Transformation from *Logical Actors* into *Physical Components* (indicated by a *trace* relation) can be seen. In this simplified example all transformations are stated as one-to-one transformations. The interfaces created as basis for interrelations are denoted as *IF A* (between the DSM and the CEMS) and *IF B* (between the CEMS and the Smart Meter). In this example, however, no detailed description of these interfaces is integrated. Second, the relations between certain components are depicted which cover the lower three SGAM layer as described above. To maintain a better overview, only the *Information Object Flow* covering the *PQ Data* has been integrated in this diagram. However, it is important to notice that the *Information Objects* used by the *Information Object Flows* are the same model elements as created earlier in the System Analysis Phase.

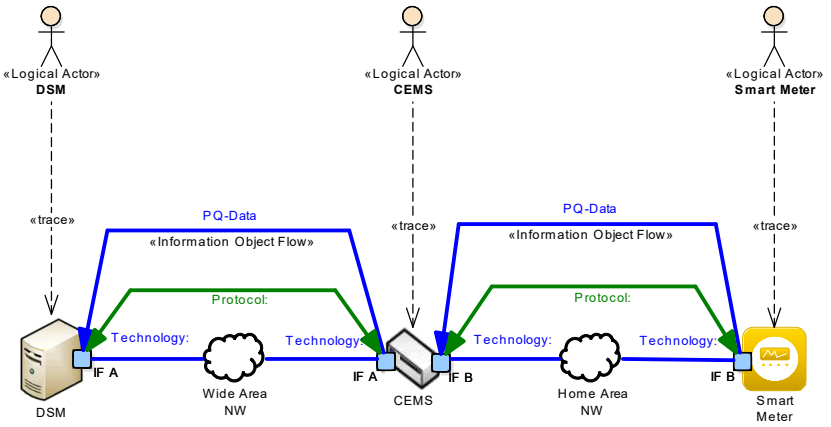

Figure 10. System Architecture.

This very simplified example aims at giving an idea about the necessary steps during this phase. However, conducting these steps introduces different challenges. Thus, similar to the System Analysis Phase, a video tutorial is provided on the *SGAM Toolbox* homepage www.en-trust.at/SGAM-Toolbox that explains all necessary steps in detail.

4.1.3. Design and Development Phase

The final phase of the engineering process is the *Design and Development Phase* which aims at realizing the particular components and integrating them to a complete solution. This phase is typically conducted by the supply chain and system integrators. In terms of the MDA this phase yields a detailed design (*Platform Specific Model*, PSM) and the according implementation (*Platform Specific Implementation*, PSI).

The particular development methodology for realizing certain components should be chosen in respect to their nature. For example, a software solution for user interactions could utilize agile concepts whereas the development of embedded control systems would rather rely on classic approaches. However, to maintain the holistic concept one should strive for a seamless integration of the detailed design with the architectural model. For example, the inner structure of particular components could be designed by means of UML or SysML.

Another aspect in terms of consistency is the utilization of a specific implementation framework that addresses the close integration with the artifacts described so far. For this purpose the proposed approach introduces the *ENOSAR* framework which is discussed in detail in Section 4.5.

4.2. Domain Specific Architecture Framework

The *Domain Specific Architecture Framework* building block represents the core of the presented approach. It delivers a *Domain Specific Language* (DSL) for modeling Smart Grid architectures in context of the SGAM. Thus it fosters a clear separation between the functional aspects (SGAM Business- and Function Layer) and the architectural aspects (the lower three SGAM Layers). As discussed in the previous section, the functional aspects correspond to the CIM whereas the architectural aspects are related to the PIM.

The implementation of this DSL is done as extension to the UML by utilizing the UML inherent "Profile" mechanism. Thus, a seamless integration with existing UML or SysML elements is possible. A simplified representation of the underlying meta-model for this DSL can be found in Figure 11 and is discussed in the following.

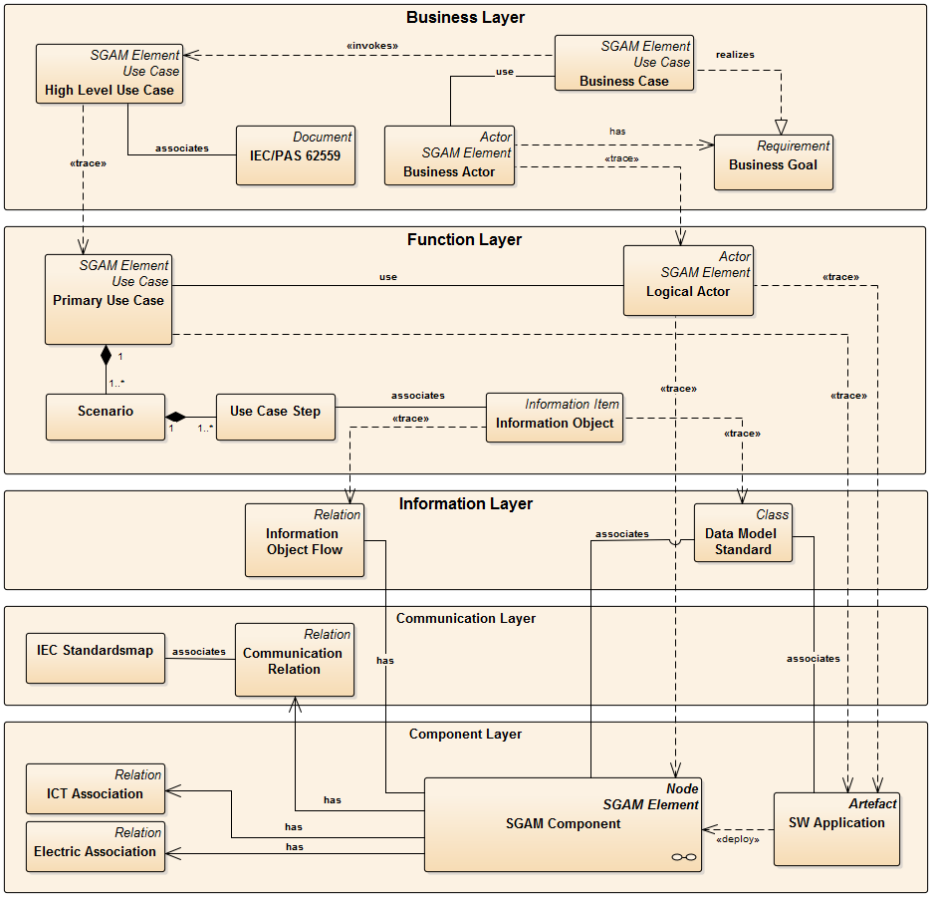

Figure 11. Metamodel of the implemented Domain Specific Language (DSL).

In alignment with the previously described development process, the DSL separates the model into three stages. The relations between elements from different stages are realized by *trace* relations which indicate model transformations as described in Section 4.1.

The first stage of the model provides support for executing the *Business Analysis* task within the *System Analysis Phase*. Basically it delivers model elements of type *Business Actors* (BA), *Business Goal* (BG) and *Business Cases* (BC) to build up the SGAM Business Layer. The way of describing Business Cases in detail may differ between particular users and thus, no formal way for a description is defined. However, as the DSL is an extension of the UML it is possible to make an in-depth description of particular Business Cases by means of any state of the art modeling language such as *Business Process Modeling Notation* (BPMN). Another approach is to only reference or add documents of any type that

hold the detailed description. For modeling particular requirements such as privacy or security it is suggested to make use of existing concepts such as SysML requirements diagrams. Again, as the DSL is an extension of the UML a seamless integration is possible.

The second stage of the model deals with specification of functionality. In reference to the staged approach, the description relies on *High Level Use Cases* (HLUC) and *Primary Use Cases* (PUC). To provide consistency with standardization, the HLUC comprises attributes that reflect the information from the IEC PAS62559-2 Use Case template [45] as introduced in Section 2. The implementation of these attributes is done by means of *tagged values*. By following this concept, a direct import of IEC PAS62559-2 based Use Case descriptions to the model is possible.

For the detailed specification of functionality the model elements *Logical Actor*, *Primary Use Case* and *Information Object* are introduced. To clarify the involvement of *Information Objects* Figure 11 also includes the elements *Scenario* and *Use Case Step* with both being taken from standard UML.

The third stage of the developed DSL delivers elements for modeling architectural aspects (the Platform Independent Model). In terms of the SGAM, these aspects cover the *Information Layer*, *Communication Layer* and *Component Layer*. Thus, the general element *SGAM Component* is introduced. This element serves as super class for the derivation of more specific elements such as *Combined Heat and Power Plant*, *Smart Meter*, *Consumer Energy Management System* and others. These components are obtained by a model transformation out of logical actors which is denoted by a *trace* relation. In addition, several relations between components are defined. These relations comprise *Information Object Flow*, *Communication Relation*, *ICT Association*, *Electric Association* and others. On basis of the described SGAM components and their corresponding relations the lower three layer of the SGAM can be modeled.

The implementation of the described DSL is implemented as part of the publicly available *SGAM Toolbox*. By now, the toolbox could prove its value in various national and international projects. For the authors it was interesting to notice that even if it is centered on the European SGAM it is also used outside Europe. However, even if the basic concept delivers value, some issues still are open. First, as the toolbox is based on standard modeling concepts it is quite applicable for ICT related engineers. For engineers from the electricity domain some training is necessary to become familiar with modeling concepts. This aspect has been tackled by providing additional training material such as video tutorials. In addition, working with the toolbox revealed a certain amount of repeating tasks such as the integration of many PUCs or the mapping of Logical Actors onto Physical Components. Thus, at present the ongoing development of the toolbox focuses on the integration of automation functionality. By now, features are considered for implementation that provide functionality for a wizard-like integration of new HLUCs, PUCs or Components. Moreover, it is considered on how to integrate the NIST LRM as kind of library, Thus, the planned wizard will enable to select particular Physical Components (out of the NIST LRM) as target for the model transformation from Logical Actors onto Physical Components. In addition, features such as the integration of individual model assessment rules on basis of individual attributes or the integration with external tools (e.g., Power System Analysis Tools) have been requested by users of the SGAM Toolbox. For this reason interfaces are implemented that allow for an automatic import of IEC 62559-2 based Use Case descriptions out of the *Use Case Management Repository* [44]. To raise the acceptance within the application domain another interface is being implemented that enables the integration with power system analysis tools such as NEPLAN www.neplan.ch. Beyond the scope of these two interfacing aspects it is necessary to make the toolbox part of a unified toolchain. Particular considerations on how such a unified toolchain could look like already exist. A discussion on the underlying ideas can be found for example in [49].

4.3. Architecture Evaluation

When developing an architecture for complex systems such as Smart Grids, typically different solution concepts need to be considered. Choosing the "best" solution, requires to make architectural decisions which rely on two aspects. First, the *architectural drivers* must be known and second, a method

for assessing two or more solutions in respect to these drivers is necessary. For example, if "costs" is identified as primary architectural driver, a method is required that delivers a comparable indication for the *Key Performance Indicator* (KPI) *Capital Expenditure* (CAPEX) and *Operational Expenditure* (OPEX).

Developing such methods is a major challenge, especially in more complex scenarios such as privacy or security. However, a common baseline for architecture evaluation is the necessity for well-defined architecture models as basis for assessments. The DSL as discussed in the previous section delivers such a formal specification of Smart Grid architectures and thus, provides a suitable basis for a formal architecture evaluation.

To evaluate the feasibility of this approach, first experiments have been made. To be more precise, the architecture specification has been used to develop a method that aims at assessing potential privacy implications. In this approach, data-flow-graphs (DFGs) were extracted out of different architectural solutions. Basically these DFGs map the SGAM Information Layer (Information Objects transported among various physical components) onto Business Actors from the SGAM Business Layer. Thus, it delivers which Information Objects are "visible" for a particular Business Actor. To make a statement on potential privacy implications, these DFGs were exported and served as input for an ontology based privacy assessment. The details of this experiment have already been discussed in [50].

However, even if the workflow for model assessment could be demonstrated from a technical point of view, future research effort is necessary to (1) identify reasonable KPIs and (2) develop appropriate methods for assessment. At present, work is conducted that investigates the possibility for assessing "cost" aspects as described before.

4.4. Best Practice Solutions

As *interoperability* is one of the key architectural drivers in the Smart Grid, concepts are required that enable the reuse of existing solutions or parts of it. However, this is not only limited to solutions for certain functionality but also includes aspects such as typical Use Cases or security requirements for different components. The modeling concept discussed earlier with its formal structure enables the seamless integration of individual elements. However, besides the structure for integration of partial aspects, also the corresponding content is required. By now, three major aspects have been identified that can be reused in particular solutions and will be discussed in the following.

4.4.1. Use Case Management

During the development of Smart Grid systems the specification of functionalities to be delivered by particular solutions is a crucial part. The IEC 62559-2 Use Case template [45] is a cornerstone in a consistent description of Use Cases. As outlined in Section 4.2, the concepts of this template have already been included in the proposed Domain Specific Architecture Framework. However, to maintain interoperability of developed systems and components, it is required to conceive different solutions on basis of the same functionality. Thus, a commonly available set of typical Use Cases that describes particular functionality such as the operation of DERs is necessary.

A major step in this direction is taken by the *Use Case Management Repository* (UCMR) as developed by OFFIS Oldenburg Institute for Information Technology, www.offis.de [44]. The UCMR is a web-based tool that provides a repository for common development and maintenance of Smart Grid Use Cases on basis of the IEC 62559-2 Use Case template. The initial version of this repository has been used among the work of the M/490 mandate for common development of a basic set of Use Cases [19]. Even if the M/490 mandate has finished, the repository with the final set of Use Cases is still available. Today it is hosted by German DKE https://usecases.dke.de/sandbox/ and accessible with the credentials provided in [19].

Since application in M/490, the UCMR tool has continuously been improved and is applied in various projects such as DISCERN www.discern.eu http://ucmr.offis.de/#!loginView. Integration between the UCMR and the introduced SGAM Toolbox is subject of present work as well.

The utilization of a common platform for Use Case development could already prove its value. However, by now, no common instance is publicly available that exceeds the scope of a specific project. For developing a broadly accepted set of Use Cases within the community the availability of such a platform would be of great value. Especially by having the opportunity to directly import Use Cases out of this library into architecture models as described earlier.

4.4.2. Reference Architecture

The Smart Grid as a whole denotes a complex *System of Systems* (SoS), capable of realizing different Use Cases. As a consequence particular components are confronted with extensive interoperability requirements. To maintain these requirements it is necessary to have recourse to well defined components with specified connectivity. To be more precise, a commonly agreed *Reference Architecture* is needed that provides best practice solutions for typical Use Cases, built upon well-defined components and interfaces.

Within the Smart Grid community numerous efforts towards such a Reference Architectures have been made. A good summary on this issue is given for example in [51]. However, in order to gain broad acceptance, approaches made by standardization bodies are of particular interest. In this light, an outstanding piece of work has been delivered in context of the *NISTIR 7628 Guidelines for Smart Grid Cybersecurity* [17]. The proposed *NIST Logical Reference Model* (NIST LRM) as already discussed in Section 2.3 delivers not only a Reference Architecture but also an extensive concept for integration of particular security requirements. Thus, in the proposed approach the NIST LRM has been chosen as basis for providing a Reference Architecture model that can serve as blueprint for architecture development.

The concepts for domain specific architecture development as discussed so far rely on the European Smart Grid Architecture Model (SGAM). Thus, in a first step an alignment between the NIST LRM and the SGAM is necessary. As discussed in Section 2 both, the SGAM and the NIST LRM are derived from the NIST Conceptual Model with its domain concept. Following this concept, the NIST LRM actors can be aligned along the SGAM plane as depicted in Figure 12.

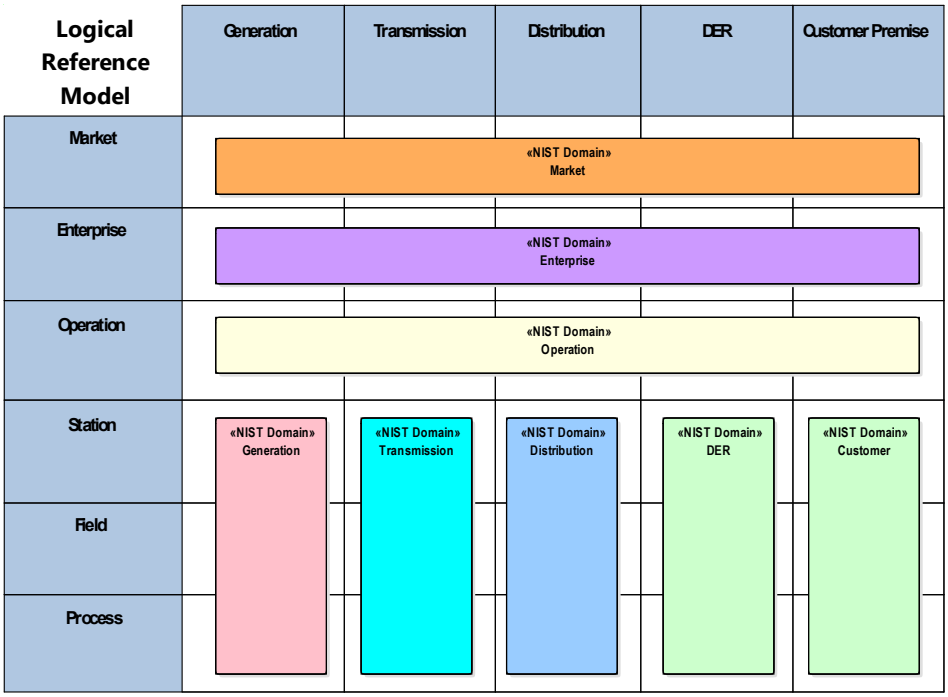

Figure 12. Mapping of the NIST LRM actors onto the Smart Grid Architecture Model (SGAM) plane.

Unlike the SGAM, the NIST LRM does not comprise a DER domain. Instead, DER related actors are considered as part of the "Customer Premises". To differentiate between these two domains, DER related actors from the NIST "Customer Premises" domain are relocated to the SGAM "DER" domain. To indicate these adoptions, Figure 12 depicts the original NIST domain "Customer Premises" as two separate domains ("DER" and "Customer" in the lower right corner). Further more detailed considerations on the mapping of NIST LRM actors onto the SGAM plane can be found in [52].

The mapping of certain NIST LRM actors on the SGAM plane is a rather straight-forward task. However, in terms of *interoperability* a major drawback of the NIST LRM in comparison to the SGAM comes clear. The SGAM maintains the *Separation of Concerns* principle by clearly differentiating between *Business Actors* (SGAM Business Layer), *Logical Actors* (SGAM Function Layer) and *Physical Components* (SGAM Information-, Communication- and Component Layer). This concept however is not reflected by NIST LRM. Thus, the NIST LRM comprises a mashup of actors of different type.

To fully integrate the NIST LRM with SGAM, further considerations are necessary. First of all, a vertical expansion of the NIST LRM needs to be conducted that differentiates between the three actor types as maintained by the SGAM. By now, the nature of the existing NIST LRM is best fitting on the lower three SGAM layer and thus represents the architectural description comprising components, interfaces and communication paths.

Another important issue to deal with is the specification of functionality on level of the SGAM Function Layer. The functionality described here states the basis for derivation of the NIST LRM. Unfortunately the original specification only provides a rough listing of the mentioned Use Cases with no detailed description. To provide consistency, a more detailed description in alignment with the concepts discussed in Sections 4.1 and 4.2 needs to be conducted.

The last issue to be discussed deals with describing aspects within the SGAM Business Layer. In that case, the NIST LRM does not give an indication on how to mention these aspects. Again, to foster a broad acceptance it is suggested to integrate existing concepts such as *The Harmonized Electricity Market Role Model* proposed by ENTSO-E [53]. For the sake of clearness, the concepts on how to obtain a *Reference Architecture Model* in context of the SGAM are summarized in Figure 13. More detailed considerations on the integration of these concepts can be found in [48].

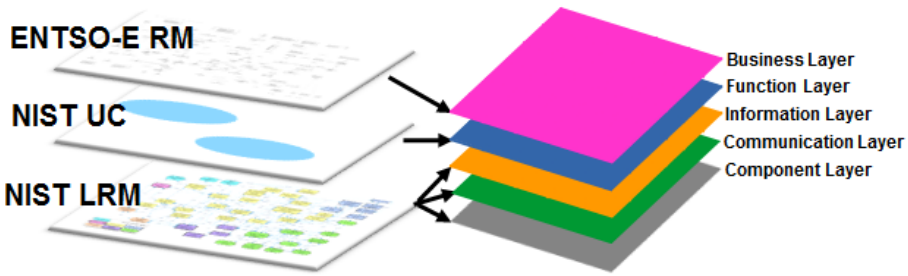

Figure 13. Mapping of the NIST LRM with the SGAM layer.

Despite all mentioned issues, the NIST's detailed specification of particular actors and their interrelations provides a good blueprint for developing Smart Grid architectures. Moreover, the explicit specification of *Interfaces*, their associated *Interface Categories* and the corresponding *Security Requirements* delivers a very applicable way to obtain a basic set of security requirements for every actor at an early stage of development. Thus, this model represents a major contribution to *Security by Design*. Furthermore, the well described concepts behind the NIST LRM support the individual extension of this model by additional actors and interfaces. A demonstration on how to integrate new actors and interfaces is covered by the example in Section 5.

Following the argumentation above, it appears more than justified to select the NIST LRM as basis for development of a Reference Architecture Model. Thus, by utilization of the previously described

DSL the NIST LRM has been modeled in context of the SGAM. The resulting model has been made publicly available www.en-trust.at/sgam-toolbox. Moreover, for a better understanding, an HTML export of this model has been published online as click-through model www.en-trust.at/NISTIR.

By now, this model served as blueprint for architecture development in several projects such as INTEGRA [54] or In2VPP www.in2vpp.de. To be more precise, within these projects particular architectures have been developed by instantiation of the NIST LRM. Again, by deriving concrete actors out of NIST LRM actors particular interfaces to be used are delivered out of the box. As these interfaces are linked directly with their corresponding security requirements this is a major step towards *Security by Design*.

However, the Reference Architecture Model described above only reflects the present state of the art. To tackle the issues discussed before different work is being conducted. To be more precise, under the umbrella of the Austrian RASSA project all discussed issues are investigated in detail to provide a consistent and standards-based Reference Architecture Model.

4.4.3. Standards Mapping

The concepts described so far deliver a possibility on how to develop Smart Grid architectures by invocation of standardized concepts such as SGAM or NIST LRM. On basis of the NIST LRM particular actors and the interfaces in between are described in detail. However, it does not provide guidance on specific data model standards or communication protocols to be used. Such guidance is given by the *IEC Smart Grid Standards Mapping Tool* http://smartgridstandardsmap.com/. This interactive tool delivers support on the selection of appropriate standards in reference to the location within the SGAM plane. By mapping the NIST LRM onto the SGAM plane as discussed before, this tool can be used for selecting adequate protocols or data model standards. However, it is important to notice that the *Standardsmap* extends the original SGAM plane by the two zones *Communication* and *Crosscutting* in order to also provide standards that are not directly referencing a particular location within the SGAM plane.

4.5. Implementation Framework

The last building block of this approach considers how to bridge the gap between (1) the overall system architecture and (2) the implementation of particular components. Thus, an implementation framework is envisioned, that can serve as platform for the realization of functionality as specified during the architecture phase.

The content of this building block is subject of the author's current work and thus, the presented considerations rather present conceptual considerations than a ready-to-use implementation. However, by now a first prototype implementation is under development. In the following paragraphs, (1) the goals of this implementation framework, (2) the underlying concepts and (3) the implementation concept for the first prototype are briefly discussed.

The goals behind this implementation framework address various issues. First of all, a close relation with functional descriptions from the architectural model is targeted. Thus, a dedicated runtime environment for particular devices is conceived that provides certain lower level functionalities (e.g., communication stacks) to be used by the functional parts of the software. The availability of such a well-defined runtime environment enables engineers to concentrate on functionalities rather than to deal with low level aspects such as communication stacks or hardware specific issues. Moreover, it enables a certain level of portability as functional implementations can be deployed on various hardware platforms that dispose of an implementation of this runtime environment.

For the envisioned concept, portability and modularization have been identified as main architectural drivers. Thus, a *layered architecture* has been selected as architectural pattern. In the field of automotive engineering a similar approach is well established. Here, the ***AUT****omotive* ***O****pen* ***S****oftware* ***AR****chitecture* (AUTOSAR) www.autosar.org separates the software on embedded devices into a *base software* and a *function software*. The base software provides certain domain specific functionalities

such as communication stacks for domain specific protocols (e.g., CAN, Lin, FlexRay and others). Moreover, well-defined interfaces are provided that can be used by the on-top function software. The mentioned architecture has proven to be of great success for several reasons. First of all, AUTOSAR as common and well-known development platform provides a structure that enables separation of work among the value chain. Typically, platform development is conducted by the supply chain whereas function development can be done by *Original Equipment Manufacturers* (OEM). This clear partitioning enables OEM's to switch between different hardware platforms from different suppliers. In the field of automotive, this is crucial for wide spread second or third supplier strategies.

In terms of Smart Grids strategic considerations are less in focus. However, the AUTOSAR concepts illustrate how cooperative development of control systems in a safety critical environment can look like. Moreover, it demonstrates how development can be enabled by a clear separation of concerns between platform aspects (Supply chain develops hardware and base software) and functionalities (OEMs integrate functionality)

The implementation framework envisioned in this building block follows a similar concept. Thus, it has been denoted as ***EN***ergy ***O***pen ***S***ystem ***AR***chitecture (ENOSAR). Similar to AUTOSAR, ENOSAR separates the device software into an *ENOSAR Base Software* and an *ENOSAR Function Software*. The base software aims at a close link with the underlying hardware. A particular configuration (e.g., which protocols to be used) should be obtained on basis of the architectural model. Moreover, it is intended to provide well-defined interfaces to be used by the function software. The function software is closely linked with functionality as described within the architectural model. Investigations on how to obtain functional software out of the architecture model (source code generation) are currently made by the authors. An overview about the ENOSAR concepts can be seen in Figure 14, further considerations on selected building blocks are given in the following.

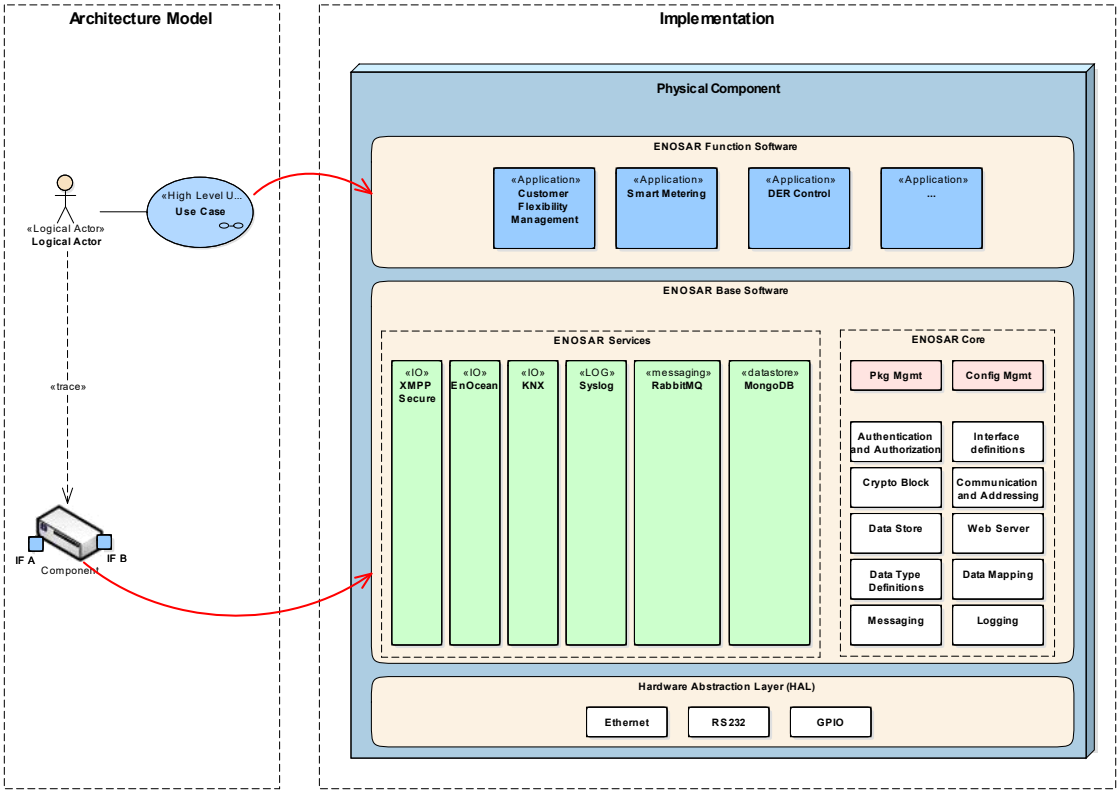

Figure 14. Concept of the ***EN****ergy* ***O****pen* ***S****ystem* ***AR****chitecture* (ENOSAR) Framework.

Figure 14 depicts a rough overview on the implementation framework. The underlying ideas, together with the current work on implementation are described in the following.

The **ENOSAR Base Software** is the key element of the implementation framework. It is located upon a *Hardware Abstraction Layer* (HAL) that is supposed to handle hardware specific aspects such as drivers. The interfaces provided for the function software are denoted as *ENOSAR Services*, depicted on the left side of the base software. These services comprise communication stacks for domain specific protocols such as *XMPP Secure, OPC-UA, EnOcean (Home Automation protocol) or IEC 61850*. In addition, a messaging and a datastore concept are specified. Figure 14 lists *RabbitMQ* respectively *MongoDB* as concerning technologies as they are used in the ongoing implementation of the prototype.

A key element of the ENOSAR Base Software is the **ENOSAR Core**. It comprises basic functionality such as *package* or *configuration management*, the internal *Messaging* concept, *Logging* or a *Webserver*. A very crucial element of the core is the *Crypto Block*. Besides the implementation of state-of-the-art cryptographic routines such as AES block cipher, this block is used for implementation and evaluation of innovative concepts especially in the field of *Privacy Enhancing Technologies* (PETs). To be more precise, the two concepts *Homomorphic Encryption* [55] and *Wavelet Encryption* [56,57] are in focus. Both of these technologies can be used to address privacy issues in terms of Smart Metering. The first concept aims at privacy aware aggregation of data. This technique, for example, aggregates already encrypted real-time Smart Metering values from individual households and relays them to a Distribution System Operator (DSO). By decrypting the received data, the DSO obtains an aggregated value (e.g., the sum of n households). This aggregated value can be good enough for sensing of network conditions but does not enable to conclude on an individual's behavior. The second concept, *Wavelet Encryption* addresses a conditional access on data on basis of a certain key. For example, a DSO could be equipped with a key that enables access to Smart Metering data with low resolution for billing purposes. In contrast, a third party service could be granted access to high resolution data for the purpose of delivering a particular service.

The **ENOSAR Function Software** should be derived on basis of the functional description from the architectural model. Thus, it focuses on algorithmic aspects and is intended to make use of functionality provided by the base software. Again, the maturity of this concept is on a conceptual level and concerning aspects are part of present work. A main focus of this work is put on the following aspects:

- Stronger formalization of functionality
- Model-to-Code transformation
- Partitioning of the ENOSAR Base Software (e.g., where to locate which functionalities.)
- Integration of PETs
- Interfacing between base software and function software

The described research work is conducted at the Josef Ressel Centerwww.en-trust.at in Austria. To be more precise, exploratory case studies considering selected Use Cases such as "Smart Metering" or "Electric Vehicle Charging" are realized in order to learn about conceptual aspects. The prototype implementation is conducted on Raspberry Pi hardware platforms. As Hardware Abstraction Layer (HAL) the Java Virtual Machine has been selected and thus, the implementation of the prototypes is done in Java. To provide a dynamic setup of the software modules being used, OSGI www.osgi.org has been utilized as versatile and modular service framework. As basis for communication between different Smart Grid participants the XMPP protocol has been selected for implementation. Furthermore, it has been extended by several security mechanisms to support End-to-End security. In addition, *Home Area Network* (HAN) protocols such as enOcean, ZigBee or KNX are integrated. For interactions with customers, communications with various customer appliances on Android basis are realized.

Again, the specification of such an implementation framework is a challenging task and research—especially on the integration of PETs—is still under progress. Thus, the concepts presented within this building block rather aim at giving a conceptual outlook than presenting finished work.

5. Architecture Modeling Example

The core of the presented approach addresses the domain specific architecture development in reference to existing standards. For a better understanding of the underlying concepts, they are demonstrated on basis of a simplified example within this section. The example used is based on the one introduced in the original SGAM proposal [20]. It is developed in alignment with the development process described in Section 4.1 and modeled by utilization of the *SGAM Toolbox* (Section 4.2). Focus of the presented example is to demonstrate the methodology and thus the example does not reflect the complexity of a real project.

However, for the given example let's assume the following scenario: The distribution system infrastructure of a certain *Distribution System Operator* (DSO) is running against its limits. Instead of making an expensive investment on infrastructure, the DSO considers to rather utilize flexibility ("Active Grid Operation") instead. To be more precise, the DSO considers two scenarios. The first scenario, *Generation Flexibility*, adresses control functionality on *reactive power* of certain *Distribution Energy Resources* (DER) such as wind power plants or photovoltaics. The second scenario, *Load Lexibility* adresses limitations in the network by reducing or shifting loads. In the given example, one particular Use Case ("*Control Reactive Power of DER*" corresponding to the first scenario is used to demonstrate the proposed modeling approach.

To ease the reading of the modeling example, Table 1 lists the abbreviations being used.

Table 1. Abbreviations used in the following example.

Abbreviation	Meaning
BA	Business Actor
BC	Business Case
BG	Business Goal
CIM	Common Information Model
DDC	Distributed Data Collector
DER	Distributed Energy Resource
DSM	Distribution System Management
DSO	Distribution System Operator
HES	Head End System
HLUC	High Level Use Case
IF	Interface
IO	Information Object
LA	Logical Actor
PUC	Primary Use Case
WAN	Wide Area Network

5.1. System Analysis—Business Layer

The very first task when architecting a system is to conduct the *Business Analysis* task as part of the *System Analysis Phase*. This task aims at identification of the motivation behind a certain system to be built, the stakeholders involved and their particular interests. Figure 15 demonstrates the utilization of the developed DSL in combination with existing modeling languages (e.g., BPMN) to describe a particular business case.

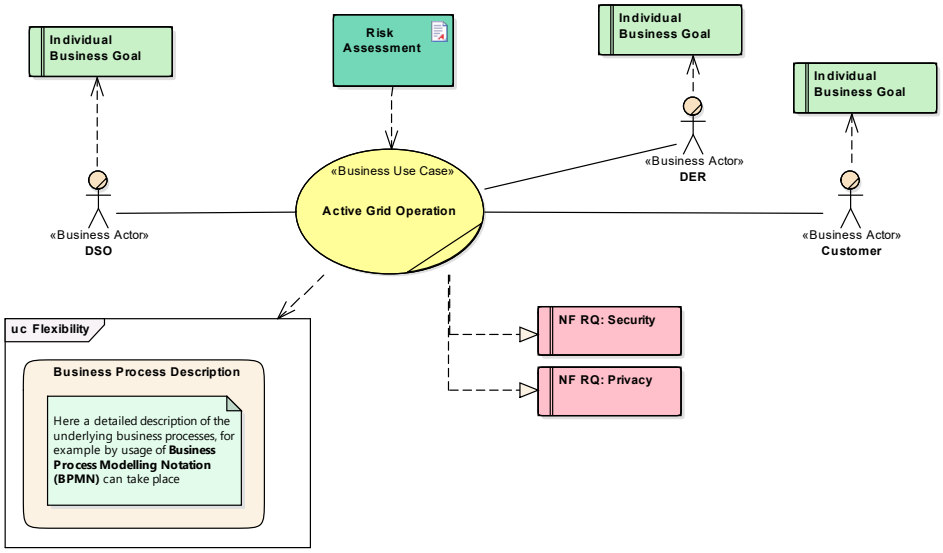

Figure 15. Business Case Analysis.

The intention of the given scenario is to cope with energy peaks on basis of an active grid operation. Basically, the *Distribution System Operator* (DSO) wants to control the injection from *Distributed Energy Resources* (DER) and the energy consumption by *Customers*. Thus, the parties involved are introduced as the Business Actors *DSO*, *DER* and *Customer*. Each of these actors has its own interests which are denoted as *Business Goals*. The concerning model elements cover a detailed description of the individual goals.

In order to balance the interests between the individual stakeholders, the Business Case (BC) *Active Grid Operation* is introduced. This model element can cover more detailed considerations of underlying aspects. For example, a *Strength-Weakness-Opportunity-Threat* (SWOT) analyses or individual business processes, such as the workflow for integration of new participants can be covered by this description. The underlying considerations can be noted in arbitrary ways, for example simply by using plain text, attaching documents or in a more formal way by utilization of *Business Process Modeling Notation* (BPMN). Figure 15 indicates the integration of such information as in the lower left corner of the picture a preview on the diagram "behind" the BC is included.

Besides the identification and description of the BC it is of special interest to also consider the associated risk. Considerations for a domain specific risk assessment can be found for example in [21] or [58]. However, in general the introduced risk is based on *potential impact* and *likelihood* (for a successful attack). In this early stage when no architectural solution yet exists, the *likelihood* is rather hard to determine whereas first considerations in terms of *potential impact* already can be made. In the given example, the risk assessment has been done by utilization of an external document template that is associated with the Business Case.

On basis of the risk assessment a first set of dependability requirements can be derived. In Figure 15 this is indicated by one particular security and one privacy requirement. These requirements typically will span a tree within the model that continuously is refined with every development phase. For decomposition of these requirements the utilization of SysML *Requirement Diagrams* is a well-known approach. The *SGAM Toolbox* used to create this example is an extension of an existing modeling tool and thus supports the integration with standardized modeling concepts such as SysML. Furthermore, also integration with external tools such as *Requirements Management Tools* (RMT) is possible. However, as this example focuses on the overall methodology, it does not provide a more granular description here.

As can be seen, specific considerations on security and privacy already take place in the very beginning. This is of great importance as privacy and security requirements can be identified as architectural drivers and thus have impact on architectural decisions to be made.

To bring the considerations made in a domain specific context the created model can be aligned within the SGAM by placing it within the *SGAM Business Layer* as depicted in Figure 16. This can be of importance to identify eventually existing regulatory constraints for particular domains. If so, the regulatory instance together with its requirements can be added as additional Business Actor. However, as the given example focuses only on the methodology, regulatory constraints are not considered here in more detail.

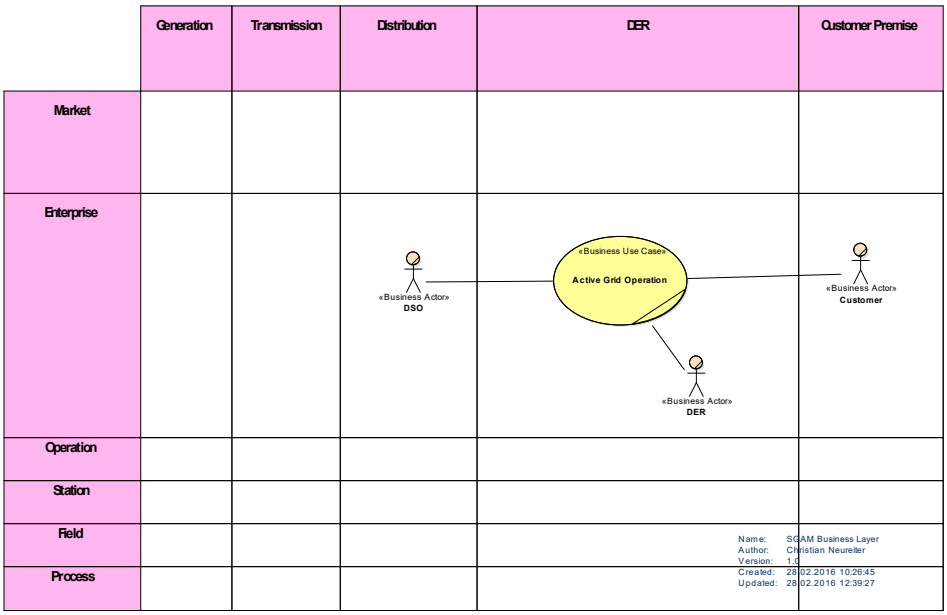

Figure 16. SGAM Business Layer.

5.2. System Analysis—Function Layer

The initial step in defining functionality is to derive particular *High Level Use Cases* (HLUC) on basis of the preliminary described *Business Case*. For a detailed description of the individual HLUCs the IEC 62559 Use Case template is used which is reflected by the DSL. In the given example three HLUC (Generation Flexibility, Control Reactive Power of DER and Load Flexibility) are specified. The relations between a business case and HLUCs are of type "invoke". Moreover, relations between particular HLUCs can be described by standard UML Use Cases relations. In the example, this is demonstrated as the HLUC "Generation Flexibility" invokes another HLUC "Control Reactive Power of DER". Figure 17 depicts the derivation of HLUCs on basis of the given Business Case. In addition, this illustration depicts the association of previously defined requirements with the derived HLUCs. However, typically the derivation of HLUCs goes hand in hand with further refinement of these requirements.

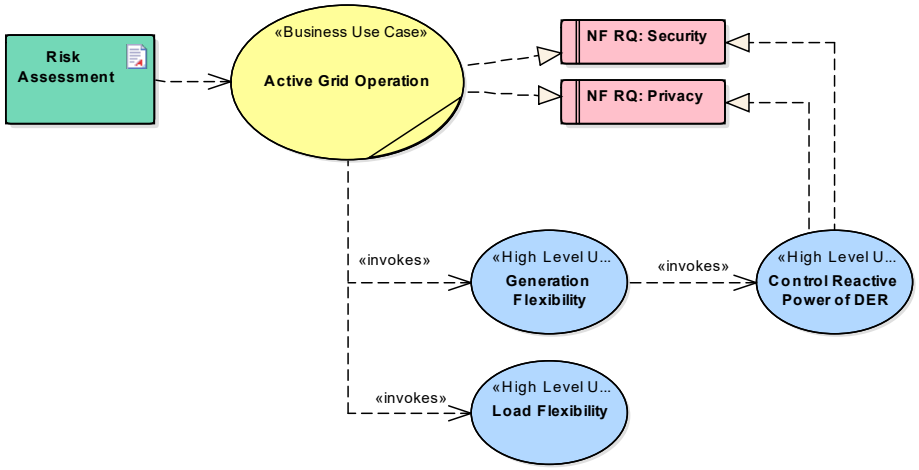

Figure 17. Derivation of High Level Use Cases.

The elaboration of the detailed functionality of one HLUC is typically done iteratively. In the beginning, an initial model transformation is conducted that derives particular Logical Actors (LA) out of the Business Actors. Figure 18 exemplarily depicts the model transformation for the two business actors *DSO* and *DER*. The initial transformation typically is not complete as during the detailed analysis additional actors arise. However, to provide consistency throughout the whole model it is important to maintain the model transformation and assign every newly introduced LA to a certain BA.

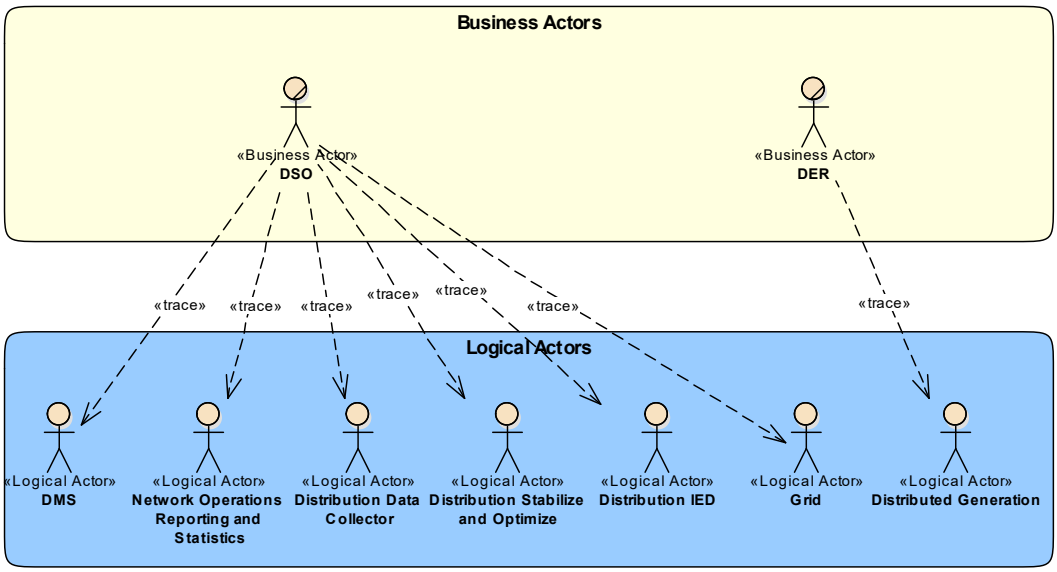

Figure 18. Model Transformation: Business Actors into Logical Actors.

The detailed analysis of one particular High Level Use Cases starts with a decomposition into more granular Primary Use Cases (PUC). In the example given, the HLUC "Control reactive power of DER" is decomposed into the five PUCs *Data Acquisition*, *SCADA*, *Volt/Var Control*, *DER Control* and *Audit* (Figure 19).

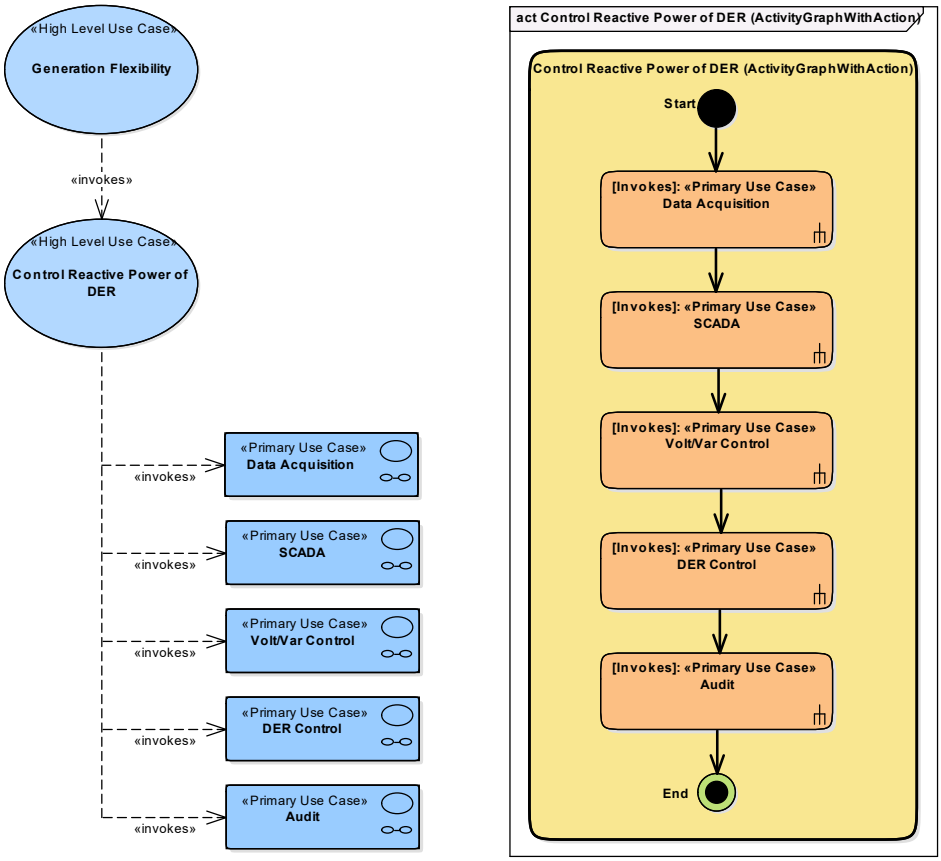

Figure 19. Decomposition of a single HLUC.

Moreover, the interrelations between the particular PUCs are depicted as UML Activity diagram. In that case, the investigated HLUC is described as *Activity* and all of the invoked PUCs are integrated as corresponding *Actions*. The described concept is very well established within the UML and thus, many tools typically support some auto-generation of the activity diagrams out of the box. The availability of such features like *diagram generation* is not only a "nice to have" issue as they raise efficiency for modeling dramatically which is a key factor for acceptance.

Subsequent to the identification of the individual PUCs, they can be scrutinized and described in detail. The main interest lies in a complete description of a single PUC which again can be done by all available concepts of UML. However, in terms of privacy and security the information exchanged between individual actors is a major asset. Thus, another goal of this step is to identify and describe all involved *Information Objects* (IO). Besides the focus of privacy and security the introduction of explicitly described IOs is a valuable concept for providing consistency within the developed architecture. Figure 20 exemplarily depicts the detailed description for the PUC "DER Control" with all mentioned aspects. The according development is described in the following in detail.

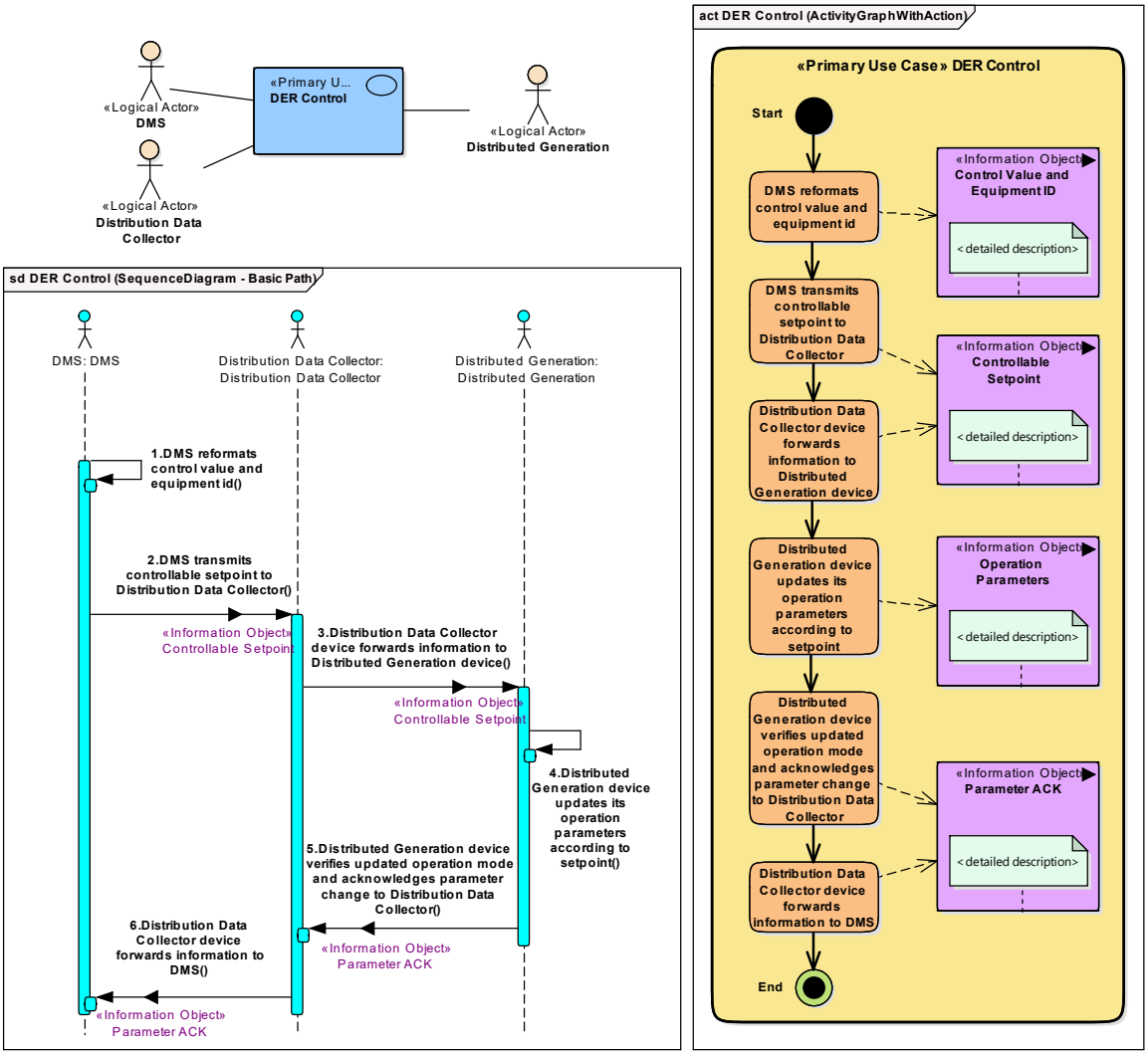

Figure 20. Detailed description of a single Primary Use Case (PUC).

Typically, description of Use Cases starts with a textual description. The SGAM Toolbox as extension for the *Enterprise Architect* www.sparxsystems.com modeling tool supports diagram generation out of a textual description. Thus, in a first step the textual description can be added to the PUC as *Structured Scenario*. If *Logical Actors* (LA) are already assigned to the PUC they will automatically be recognized. Missing actors can be introduced to the model and need to be assigned to an appropriate Business Actor as described above. After introducing all missing actors, the UML Sequence and Activity Diagram can be auto-generated.

In the example used for demonstration the PUC "Control Reactive Power of DER" comprises six individual steps. Two of them (1 and 4) describe local actions for only one LA, the others comprise communication activity as can be seen in the Sequence Diagram (lower left corner of the image). The Activity Diagram (right side) has been created similar to the one used for describing the HLUC before. Here, the PUC is represented as "Activity" that includes the individual steps as corresponding "Actions". As can be seen, the given PUC is executed as straight sequence of actions. However, for more complex Use Cases all concepts from UML Activity Diagrams such as branches, conditions or simultaneous execution can be used.

Provided with the Activity Diagram, the individual steps can be analyzed and certain *Information Objects* (IO) can be defined. The example given comprises four IOs (on the very right side). At this stage, the IOs can be described in arbitrary detail. It is important to notice that these model elements will

be used not only for privacy and security considerations but also for further development. Thus, an early clarification of the information to be exchanged helps to maintain consistency and is useful in multiple ways.

In a final step of the PUC description, the preliminary specified IOs can be added to the concerning steps within the sequence diagram. Doing this for every single PUC draws a complete picture of all PUCs and LAs involved within this HLUC. This complete picture can be visualized in context of the SGAM Function Layer as depicted in Figure 21. Again, as a modeling language is used for development, this image does not have to be drawn manually. Instead, it is a result of the detailed descriptions of all individual PUC considerations.

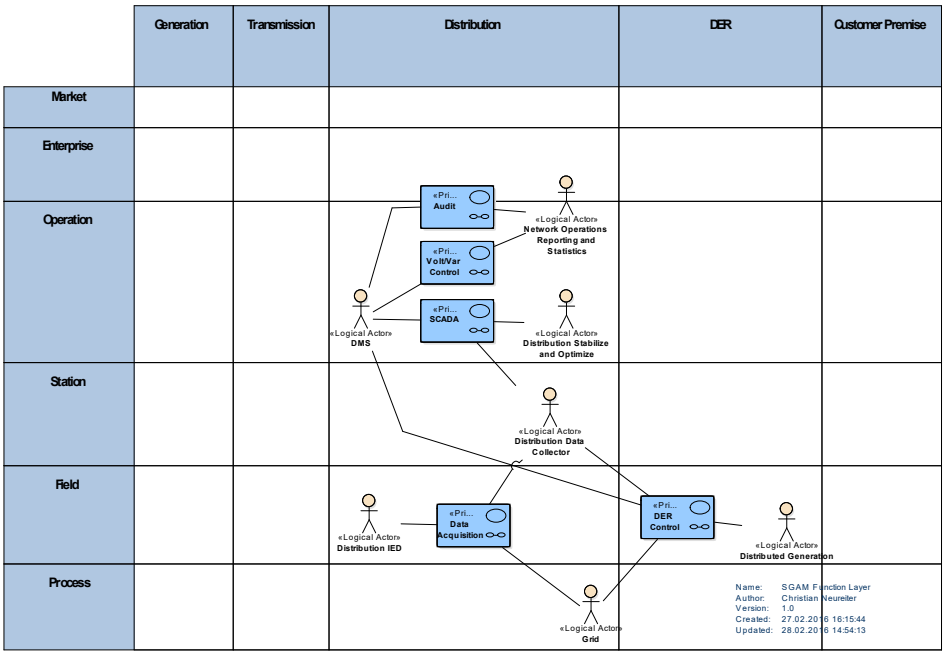

Figure 21. SGAM Function Layer.

5.3. Model Transformation

Right after the functional specification within the upper two layers of the SGAM, the architectural development can take place. The first step is to map the *Logical Actors* from the *SGAM Function Layer* onto *Physical Components* ("Model Transformation"). The identified components can be related with actors from the NIST LRM. Thus, appropriate interfaces together with their associated security requirements can be obtained. Figure 22 depicts this mapping for the given example.

As one can see in the figure, the mapping is not necessarily a one-to-one mapping. Rather, one single LA (as for example "Distribution Data Collector") can be realized as combination of multiple physical devices ("Distribution Data Collector" and "Head End System", HES). Vice versa, it is also possible for one physical component to realize the functionality of different Logical Actors (e.g., the "DMS Computer" realizes the LAs "DMS" and "Distribution Stabilize and Optimize").

Another very important aspect that can be seen is that not for every physical component a suitable reference can be found within the NIST LRM. In the given example, this pertains for the Head End System. In that case, two aspects can be considered. First, the planned solution is not conform to the NIST LRM and maybe could be reconsidered. Or, the questioned component needs to be added individually. As the development of the Smart Grid typically is no green field approach, the second scenario is more common. In the example given the HES should be integrated manually to demonstrate how to individually extend the NIST LRM. A special focus here is put on maintaining the security concepts introduced by the NIST LRM.

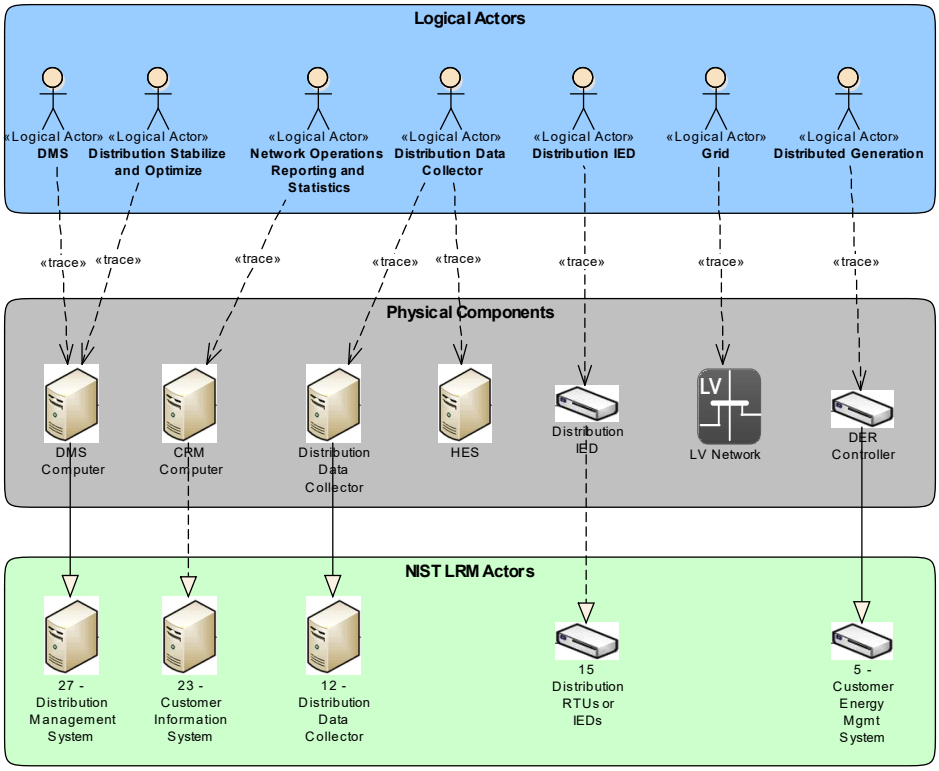

Figure 22. Mapping of Logical Actors onto physical components.

When introducing a new component it is necessary to identify and describe the interfaces with surrounding components. In the given example, the HES will be interfacing with three components (DDC, DMS and DER). Thus, the three interfaces *U201, U202* and *U203* are created and instantiated for the HES. Furthermore, each of these interfaces is considered in respect to the available interface categories discussed in Section 2.3. After the appropriate interface categories have been identified, the interfaces can be related with them. Doing so provides a direct link between the newly created component and the associated security requirements delivered by the NIST LRM. Figure 23 depicts the creation of *Interfaces* and their relation with particular *Interface Categories* for the newly created component *HES*.

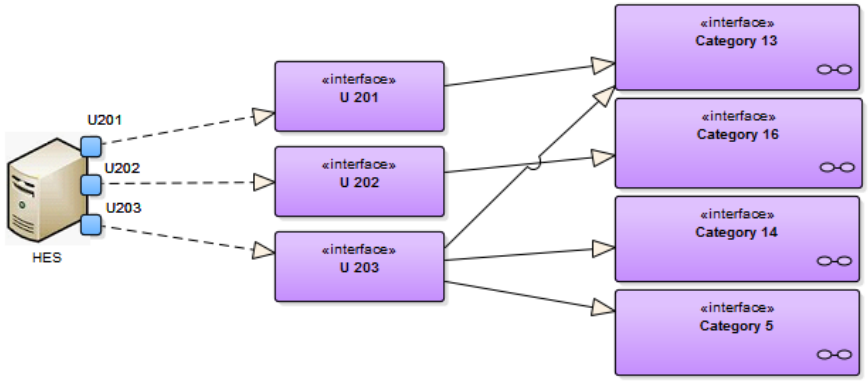

Figure 23. Integration of interfaces for a new component.

Due to their huge number the security requirements associated with every *Interface Category* are not included in the image. To brows through the corresponding requirements, the online available click-through model www.en-trust.at/NISTIR can be consulted.

However, the associated security requirements provided by the NIST LRM are developed *"bottom up"* and intended to be *High Level Security Requirements*. Thus, they rather serve as starting point and need further particularization. The necessary refinement can be done on basis of the *"top down"* requirements developed so far in the preliminary steps. This concept fosters the combination of both, a *top down* and a *bottom up* approach which is a very common way for developing security requirements.

5.4. System Architecture—Information Layer

The instantiation of the physical components together with their instantiated security requirements states the basis for the architectural development. In a first step the *Business Context View* as part of the *SGAM Information Layer* can be developed. This view aims at depicting the *Information Object Flows* among the participating physical components and can be derived by considering (1) the detailed description of all involved PUC's and (2) the mapping from *Logical Actors* onto *Physical Components*. Figure 24 depicts the *Business Context View* for the "Control Reactive Power of DER" HLUC.

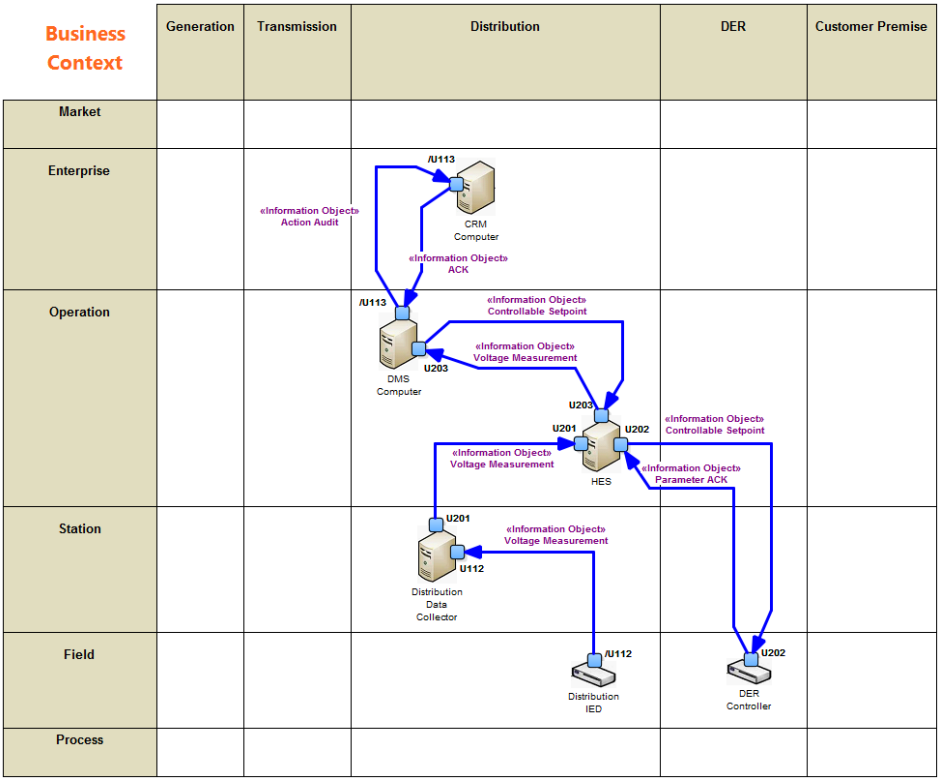

Figure 24. SGAM Information Layer: Business Context View.

The second specified view on the *SGAM Information Layer* is the *Canonical Data Model*. This view deals with the assignment of appropriate data model standards to be used. Selecting the best fitting standard requires detailed considerations. As discussed in Section 4.4.3, the *IEC Smart Grid Standards Mapping Tool* can deliver guidance for choosing appropriate standards. However, for the given example mainly CIM and IEC 61850 related standards have been selected as can be seen in Figure 25.

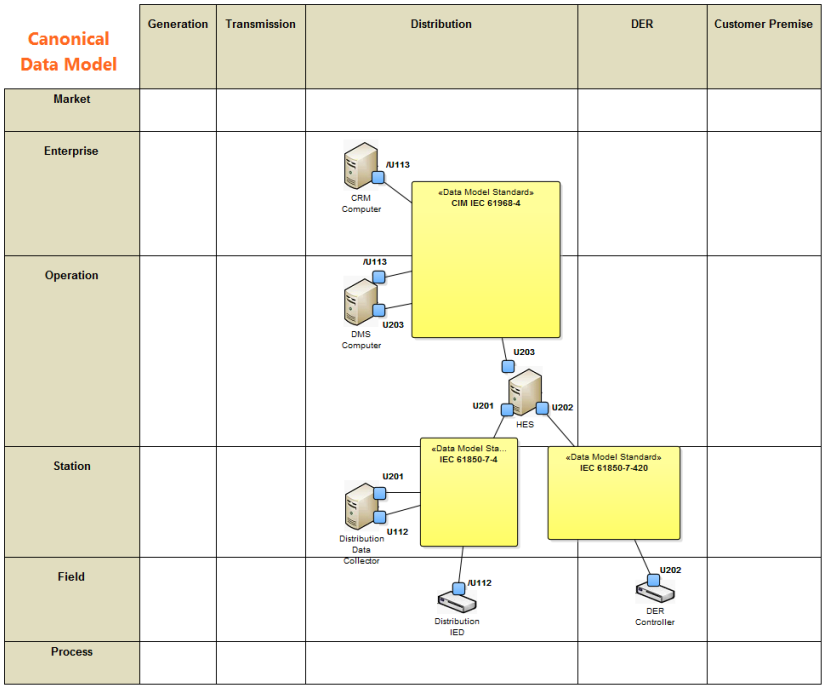

Figure 25. SGAM Information Layer: Canonical Data Model.

5.5. System Architecture—Communication Layer

Developing the *SGAM Communication Layer* is quite similar to the development of the *SGAM Information Layer*. The already existing components with their realized interfaces can be placed within the concerning diagram. Again, the *IEC Smart Grids Standards Mapping Tool* provides guidance on the selection of appropriate communication protocols which are integrated in the model as *Communication Relation*. Figure 26 depicts the *SGAM Communication Layer* for the example with the selected communication protocols.

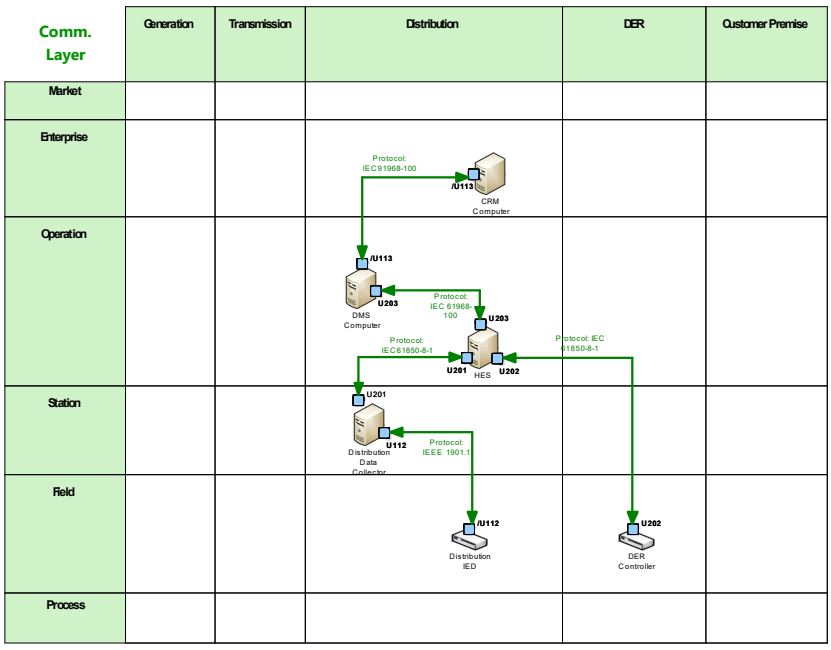

Figure 26. SGAM Communication Layer.

5.6. System Architecture—Component Layer

The previously described SGAM Information respectively SGAM Communication Layer is focusing on the Point-to-Point relation between two components. Contrasting to this, the SGAM Component Layer delivers the possibility to completely describe the ICT architecture to be realized. Thus, additional elements such as gateways or firewalls can be integrated. Another very important aspect is the representation of particular network segments. Similar as before, the elements within the diagram represent the nodes of a particular ICT architecture. The edges (relations) are used to define the communication technology such as Ethernet, Coax or GSM.

The example depicted in Figure 27 depicts how the existing components utilize different gateways to communicate over certain network segments. Moreover, the relations in between are intended to specify the communication technology used. For example, the *Distribution IED* fetches the sensor value over a two-wire connection whereas it communicates over RS485 with the (local) *Distribution Data Collector* (DDC). The DDC itself is connected to the WAN via a GPRS modem with a direct Ethernet link between the DDC and the gateway.

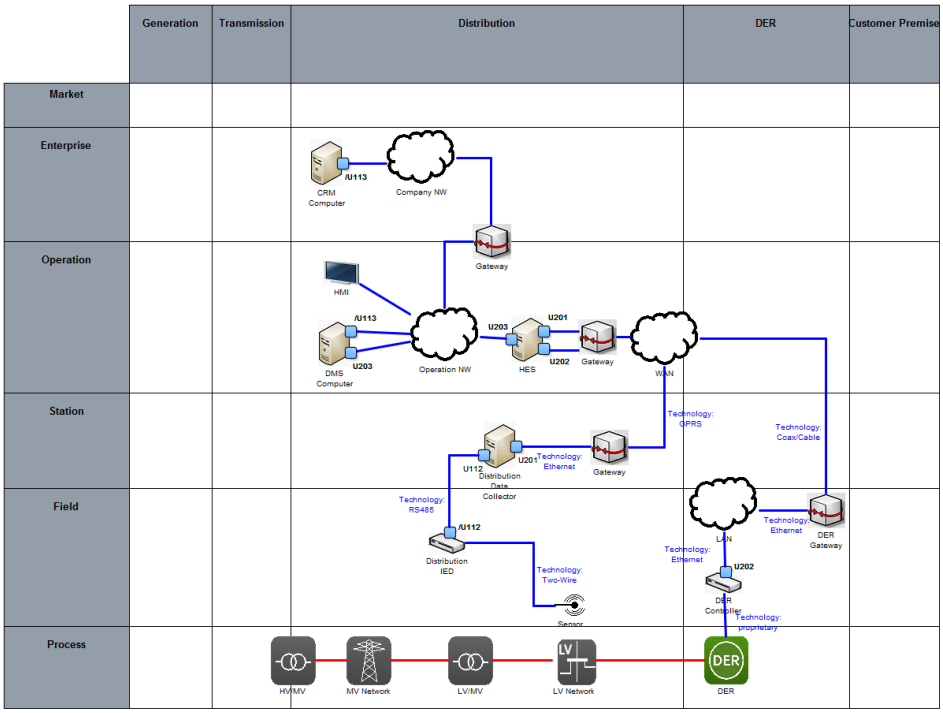

Figure 27. SGAM Component Layer.

The representation of the SGAM Component Layer as given in Figure 27 is suitable to only provide an overview on the overall ICT architecture and thus, it helps to maintain the "big picture". The given abstraction level is clearly not detailed enough for planning and maintaining the particular ICT architecture. However, to provide consistency over the whole model, the detailed documentation for the ICT architecture also should be included here. A very feasible way is to utilize the *network segment* elements to provide an in-depth description.

5.7. Further Processing

The individual steps described so far demonstrate step-by-step modeling of a Smart Grid solution. In terms of the SGAM, the model has been developed layer-by-layer. By maintaining vertical relations ("traces", reflecting model transformations) between the individual layer, a complete model can be

obtained. Figure 28 depicts selected aspects of the example model on basis of a "front view" (e.g., looking from the front on the SGAM cube).

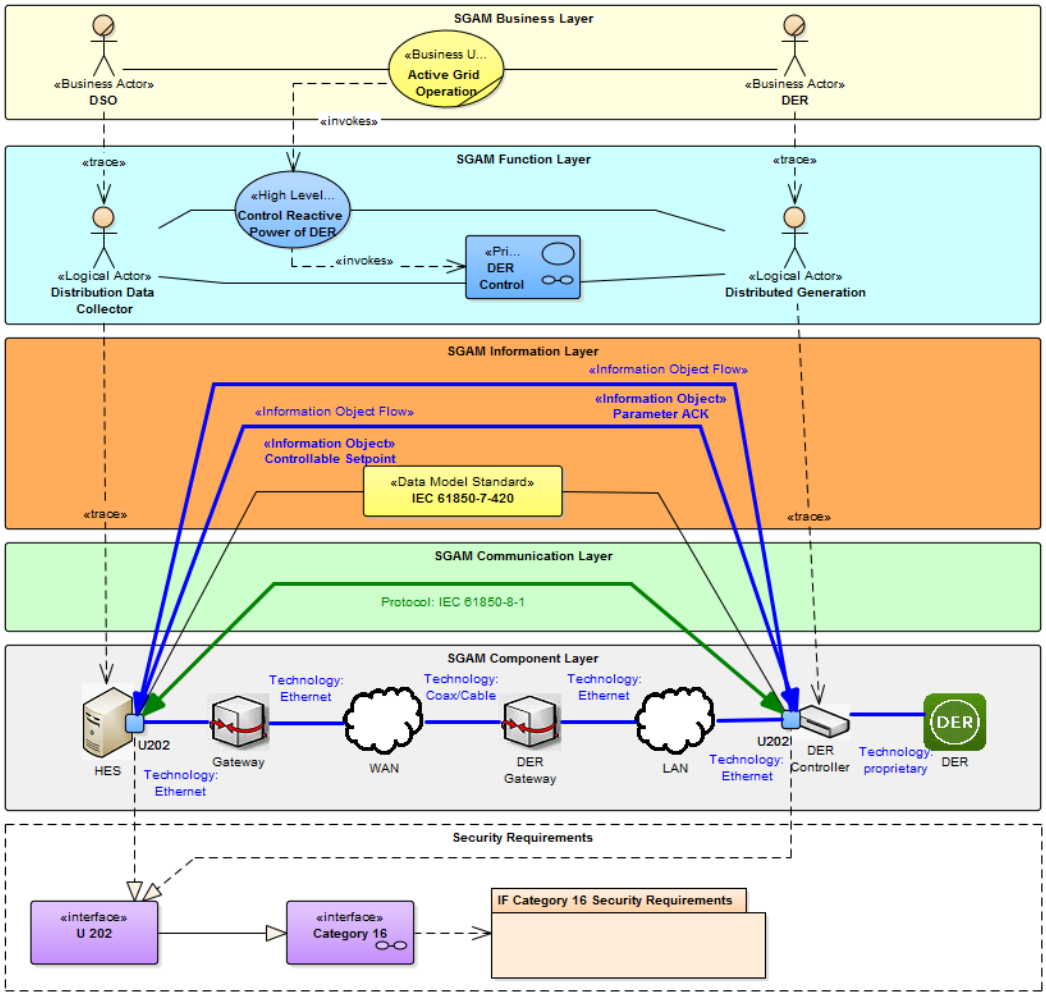

Figure 28. Front view on the example model.

As it can be seen, a complete picture can be derived that describes all individual aspects mentioned by the SGAM. In addition, as the physical elements are derived from the NIST LRM they include the related interfaces which can be traced to the concerning security requirements via their interface categories.

Besides the presentation of a complete picture, the availability of such an architectural model can serve various purposes. For example, as a certain Business Case can be traced through the model to all involved components and their relations, it is possible to assess various attributes of the related components. This could be, for example, attributes like CAPEX or OPEX to determine costs related with a particular architectural solution. Or, by delivering a description of particular components (functionality on basis of Use Cases, data model standards used, information objects exchanged or security requirements associated) the model could serve as a basis for their realization. Of course such a model as the one from the example only delivers a basis and further refinement is necessary. However, as the DSL is an extension to UML a seamless integration of additional aspects, e.g., non functional requirements is possible.

6. Conclusions

Security by Design is a crucial issue when developing Smart Grid systems. Not only due to the criticality of the electric energy system but also for acceptance among end users which are expected to play a major role in the grid. An important prerequisite in this context is the availability of a holistic and domain-specific development process. The approach proposed in this paper aims at providing such a development concept. It is built upon existing work with a special focus on output from standardization bodies from both, USA and Europe. Thus, no fundamentally new concepts are introduced, moreover different existing bits and pieces are combined to envision such a holistic concept.

The presented approach identifies five interrelated building blocks that contribute to Smart Grid engineering. For each of these building blocks the present state of implementation is discussed. In this discussion, different degrees of maturity are revealed. Work on domain specific architecture description, for example, already delivered quite applicable results. Moreover, necessary integration work for standardization concepts, especially between USA and Europe, could be identified. Other concepts, such as the integration between architecture development and implementation appear rather visionary at the time. In this context, the *ENOSAR* framework has been introduced that sketches a possible way for seamless integration. However, in order to enable such a holistic engineering methodology, a unified process together with an appropriate tool-chain is required. At present, different efforts are made towards this direction as outlined in [49].

The authors' current work is concentrating on a tighter integration between different standardization concepts in order to obtain a consistent reference architecture as envisioned in Section 4.4.2. Besides, also the advancement of the *ENOSAR* framework and especially its integration with the *SGAM Toolbox* is put in focus. All of this work is done under the currently running RASSA research project the *Josef Ressel Research Center* www.en-trust.at is involved in.

Acknowledgments: The financial support of the Josef Ressel Center by the Austrian Federal Ministry of Science, Research and Economy and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Funding by the Austrian Federal Ministry for Transport, Innovation and Technology and the Austrian Research Promotion Agency (FFG) under Project 848811, "RASSA Architektur", is gratefully acknowledged. Financial support and participation in the development of the *SGAM Toolbox* by *Siemens Corporate Technology, Munich* is gratefully acknowledged.

Author Contributions: Mathias Uslar contributed to the NISTIR 7628 and SGAM mapping concept presented in the paper, the alignment with the EU M/490 work as well as the fine-tuning of the textual and graphical presentation. Dominik Engel contributed to the overall concept presented in the paper, as well as the fine-tuning of the textual presentation. Christian Neureiter developed the domain specific language and the SGAM-Toolbox. Furthermore he wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Electronic Power Research Institute. *Vision for a Holistic Power Supply and Delivery Chain*; Technical Report 1018587; EPRI: Palo Alto, CA, USA, 2009.
2. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart Grid—The New and Improved Power Grid: A Survey. *IEEE Commun. Surv. Tutor.* **2011**, *99*, 1–37.
3. Lo, C.; Ansari, N. The Progressive Smart Grid System from Both Power and Communications Aspects. *IEEE Commun. Surv. Tutor.* **2011**, *14*, 799–821.
4. Gungor, V.C.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G.P. Smart Grid Technologies: Communication Technologies and Standards. *IEEE Trans. Ind. Inform.* **2011**, *7*, 529–539.
5. Farhangi, H. The path of the smart grid. *IEEE Power Energy Mag.* **2010**, *8*, 18–28.
6. Haberfellner, R.; de Weck, O.L.; Fricke, E.; Vössner, S. *Systems Engineering. Grundlagen und Anwendung*; Orell Füssli: Zürich, Switzerland, 2012.
7. Brown, R.E. Impact of Smart Grid on distribution system design. In Proceedings of the 2008 IEEE Power and Energy Society General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, PA, USA, 20–24 July 2008; pp. 1–4.

8. Moslehi, K.; Kumar, R. A Reliability Perspective of the Smart Grid. *IEEE Trans. Smart Grid* **2010**, *1*, 57–64.
9. Fan, Z.; Kulkarni, P.; Gormus, S.; Efthymiou, C.; Kalogridis, G.; Sooriyabandara, M.; Zhu, Z.; Lambotharan, S.; Chin, W. Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 21–38.
10. Yan, Y.; Qian, Y.; Sharif, H.; Tipper, D. A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 5–20.
11. Gungor, V.C.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G.P. A Survey on Smart Grid Potential Applications and Communication Requirements. *IEEE Trans. Ind. Inform.* **2013**, *9*, 28–42.
12. McDaniel, P.; McLaughlin, S. Security and Privacy Challenges in the Smart Grid. *IEEE Secur. Priv. Mag.* **2009**, *7*, 75–77.
13. Khurana, H.; Hadley, M.; Lu, N.; Frincke, D.A. Smart-Grid Security Issues. *IEEE Secur. Priv.* **2010**, *8*, 81–85.
14. US Department of Energy. *The Smart Grid: An Introduction*; US Department of Energy: Washington, DC, USA, 2008.
15. Paverd, A.J.; Martin, A.P. Hardware Security for Device Authentication in the Smart Grid. In *Smart Grid Security*; Springer: Berlin, Germany, 2013; pp. 72–84.
16. Mo, Y.; Kim, T.H.J.; Brancik, K.; Dickinson, D.; Lee, H.; Perrig, A.; Sinopoli, B. Cyber Physical Security of a Smart Grid Infrastructure. *Proc. IEEE* **2012**, *100*, 195–209.
17. The Smart Grid Interoperability Panel—Cyber Security Working Group. *Guidelines for Smart Grid Cyber Security*; Technical Report NISTIR 7628; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2010; Volume 1–3.
18. Smart Grid Coordination Group. *First Set of Standards*; CEN-CENELEC-ETSI Smart Grid Coordination Group: Brussels, Belgium, 2012.
19. Smart Grid Coordination Group. *Sustainable Processes*; CEN-CENELEC-ETSI Smart Grid Coordination Group: Brussels, Belgium, 2012.
20. Smart Grid Coordination Group. *Smart Grid Reference Architecture*; CEN-CENELEC-ETSI Smart Grid Coordination Group: Brussels, Belgium, 2012.
21. Smart Grid Coordination Group. *Smart Grid Information Security*; CEN-CENELEC-ETSI Smart Grid Coordination Group: Brussels, Belgium, 2012.
22. Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F.; Sommerlad, P. *Security Patterns: Integrating Security and Systems Engineering*; John Wiley: Hoboken, NJ, USA, 2006.
23. Schmidt, D.C. Model-driven engineering. *IEEE Comput. Soc.* **2006**, *39*, 25.
24. Kent, S. Model Driven Engineering. In *Integrated Formal Methods*; Springer: Berlin, Germany, 2002; pp. 286–298.
25. Bézivin, J. In search of a basic principle for model driven engineering. *Novatica J. Spec. Issue* **2004**, *5*, 21–24.
26. Favre, J.M. Towards a basic theory to model model driven engineering. In Proceedings of the 3rd Workshop in Software Model Engineering, Lisbon, Portugal, 11th–15th October, 2004; pp. 262–271.
27. Object Management Group. *OMG Unified Modeling Language (OMG UML), Superstructure*; Object Management Group: Needham, MA, USA, 2009.
28. Object Management Group. *OMG Unified Modeling Language (OMG UML), Infrastructure*; Object Management Group: Needham, MA, USA, 2009.
29. Object Management Group. *OMG Systems Modeling Language (OMG SysML) Version 1.2*; Object Management Group: Needham, MA, USA, 2010.
30. Jürjens, J. UMLsec: Extending UML for secure systems development. In *"UML" 2002—The Unified Modeling Language*; Springer: Berlin, Germany, 2002; pp. 412–425.
31. Roudier, Y.; Apvrille, L. SysML-Sec: A model driven approach for designing safe and secure systems. In Proceedings of the 2015 3rd International Conference onModel-Driven Engineering and Software Development (MODELSWARD), Angers, France, 9–11 February 2015; pp. 655–664.
32. Van Deursen, A.; Klint, P.; Visser, J. Domain-Specific Languages: An Annotated Bibliography. *Sigplan Not.* **2000**, *35*, 26–36.
33. Hudak, P. Modular domain specific languages and tools. In Proceedings of the 5th International Conference on Software Reuse, Pittsburgh, PA, USA, 20–24 July 1998; pp. 134–142.
34. Mernik, M.; Heering, J.; Sloane, A.M. When and how to develop domain-specific languages. *ACM Comput. Surv. CSUR* **2005**, *37*, 316–344.

35. Fowler, M. *Domain-Specific Languages*; Pearson Education: Boston, MA, USA, 2010.
36. Andrén, F.; Stifter, M.; Strasser, T. Towards a Semantic Driven Framework for Smart Grid Applications: Model-Driven Development Using CIM, IEC 61850 and IEC 61499. *Inform. Spektrum* **2013**, *36*, 58–68.
37. Brédillet, P.; Lambert, E.; Schultz, E. CIM, 61850, COSEM standards used in a Model Driven Integration approach to build the Smart Grid Service Oriented Architecture. In Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, MD, USA, 4–6 Ocotober 2010; pp. 467–471.
38. Lopes, A.J.; Lezama, R.; Pineda, R. Model Based Systems Engineering for Smart Grids as Systems of Systems. *Procedia Comput. Sci.* **2011**, *6*, 441–450.
39. Kaitovic, I.; Lukovic, S. Adoption of Model-Driven methodology to aggregations design in Smart Grid. In Proceedings of the 9th IEEE International Conference on Industrial Informatics, Lisbon, Portugal, 26–29 July 2011; pp. 533–538.
40. Object Management Group. Model Driven Architecture (MDA) MDA Guide rev 2.0. Whitepaper, http://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf (accessed on 27 May 2016).
41. Adolf, D.; Ferranti, E.; Koch, S. SmartScript—A domain-specific language for appliance control in Smart Grids. In Proceedings of the IEEE 3rd International Conference on Smart Grid Communications (SmartGridComm), Tainan, Taiwan, 5–8 November 2012; pp. 465–470.
42. Schütte, S.; Scherfke, S.; Sonnenschein, M. Mosaik-smart grid simulation API. In Proceedings of SMARTGREENS 2012—International Conference on Smart Grids and Green IT Systems 2012, Porto, Portugal, 19–20 April 2012; pp. 14–24.
43. Rohjans, S.; D'nekas, C.; Uslar, M. Requirements for Smart Grid ICT-Architectures. In Proceedings of the 3rd International Conference on Innovative Smart Grid Technologies Europe (ISGT Europe), Berlin, Germany, 14–17 October 2012; pp. 1–8.
44. Trefke, J.; Rohjans, S.; Uslar, M.; Lehnhoff, S.; Nordström, L.; Saleem, A. Smart Grid Architecture Model use case management in a large European Smart Grid project. In Proceedings of the 4th Innovative Smart Grid Technologies Europe (ISGT Europe), Lyngby, Denmark, 6–9 October 2013; pp. 1–5.
45. International Electrotechnical Commission. *IEC 62559-2 Use Case Methodology—Part 2: Definition of the Templates for Use Cases, Actor List and Requirements List*; IEC: Geneva, Switzerland, 2015.
46. Office of the National Coordinator for Smart Grid Interoperability. *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0* ; Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2010.
47. The GridWise Architecture Council. *GridWise Interoperability Context-Setting Framework*; The GridWise Architecture Council: Richland, WA, USA, 2008.
48. Neureiter, C.; Uslar, M.; Engel, D.; Lastro, G. A Standards-based Approach for Domain Specific Modelling of Smart Grid System Architectures. In Proceedings of the 11th International Conference on System of Systems Engineering (SoSE), Kongsberg, Norway, 12–16 June 2016.
49. Neureiter, C.; Engel, D.; Trefke, J.; Santodomingo, R.; Rohjans, S.; Uslar, M. Towards Consistent Smart Grid Architecture Tool Support: From Use Cases to Visualization. In Proceedings of the 5th International Conference on Innovative Smart Grid Technologies Europe (ISGT Europe), Istanbul, Turkey, 12–15 October 2014; pp. 1–6.
50. Knirsch, F.; Engel, D.; Neureiter, C.; Frincu, M.; Prasanna, V. Model-driven Privacy Assessment in the Smart Grid. In Proceedings of the 1st International Conference on Information Systems Security and Privacy (ICISSP), Angers, France, 9–11 February 2015; pp. 173–181.
51. Appelrath, H.J.; Beenken, P.; Bischofs, L.; Uslar, M. *IT-Architekturentwicklung im Smart Grid. Perspektiven für eine Sichere markt-und Standardstandard Integration Erneuerbarer Energien*; Springer: Berlin, Germany, 2012.
52. Uslar, M.; Rosinger, C.; Schlegel, S. Security by Design for the Smart Grid: Combining the SGAM and NISTIR 7628. In Proceedings of the 38th International Computer Software and Applications Conference Workshops (COMPSACW), Vasteras, Sweden, 21–25 July 2014; pp. 110–115.
53. European Network of Transmission System Operators for Electricity. *The harmonised Electricity Market Role Model*; European Network of Transmission System Operators for Electricity: Brussels, Belgium, 2014.
54. Mattle, P.; Neureiter, C.; Kupzog, F. Projekt SGMS—INTEGRA Übergang zu netz- und marktgeführtem Betrieb im Smart Grid. In Proceedings of the 4th Workshop on Communications for Energy Systems; Austrian Electrotechnical Association: Vienna, Austria, 26 September 2013; pp. 44–52.

55. Engel, D.; Eibl, G. Multi-Resolution Load Curve Representation with Privacy-preserving Aggregation. In Proceedings of IEEE Innovative Smart Grid Technologies (ISGT), Lyngby, Denmark, 6–9 October 2013; pp. 1–5.
56. Engel, D. Wavelet-based Load Profile Representation for Smart Meter Privacy. In Proceedings IEEE PES Innovative Smart Grid Technologies (ISGT'13), Washington, DC, USA, 24–27 February 2013; pp. 1–6.
57. Engel, D.; Eibl, G. Wavelet-Based Multiresolution Smart Meter Privacy. *IEEE Trans. Smart Grid* **2016**, *99*, 1–12.
58. Neureiter, C.; Eibl, G.; Engel, D.; Schlegel, S.; Uslar, M. A concept for engineering smart grid security requirements based on SGAM models. In *Computer Science—Research and Development*; Springer: Berlin, Germany, 2014; pp. 1–7.