MARSHALL PLAN SCHOLARSHIP REPORT

Generic Data Models and Semantic Retrieval in Smart Grid IT Infrastructures

University of Southern California (USC), Los Angeles, CA USA
Salzburg University of Applied Sciences (SUAS), Salzburg, Austria

Fabian Knirsch, BSc

USC:	Dr. Marc Frincu, MSc
SUAS:	FH-Prof. DI Mag. Dr. Dominik Engel

Salzburg, September 2014

Generic Data Models and Semantic Retrieval in Smart Grid IT Infrastructures

An Approach Towards Model-driven Privacy Assessment

Fabian Knirsch, BSc

Abstract

Future energy systems will heavily rely on the exchange of data and information. In smart grids personal data is collected, transported and processed by many applications and often personal data is the key to effective smart grid systems. The usage of personal data, however, implies privacy issues. To meet legal requirements and to facilitate customer acceptance, full transparency of potential threats needs to be guaranteed for all stakeholders involved. In this report, an approach is presented for the ontology driven classification of use cases based on meta-information. Use cases are modeled in accordance to the Smart Grid Reference Architecture. System engineers benefit from the classification of use cases prior to implementation and customers are provided full transparency of information flows that involve personal data. System design and implementation draws on insights from two model regions in the United States and the European Union. The system is evaluated with respect to the classification of privacy threats and with respect to the system's ability to be extended for more general threats. As a proof-of-concept implementation the classifier is further used as a policy decision point in an existing smart grid middleware.

Acknowledgments

The work described in this report was conducted in cooperation with the Department of Electrical Engineering at the University of Southern California, the Austrian Marshall Plan Foundation and the Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control at the Salzburg University of Applied Sciences. The financial support of the Josef Ressel Center by the Austrian Federal Ministry of Economy, Family and Youth and the Austrian National Foundation for Research, Technology and Development is gratefully acknowledged. Funding by the Austrian Marshall Plan Foundation is gratefully acknowledged.

The acknowledgment of the author, and a word of thanks, is devoted to the people who encouraged this work, to those who have been a source of inspiration, and to those who supported this project through their deep scientific expertise, their ongoing effort and their lasting academic commitment.

A special word of thanks is therefore devoted to Dominik Engel and Christian Neureiter from Salzburg University of Applied Sciences and Marc Frincu from the University of Southern California.

Contents

List of Figures

List of Tables

List of Listings

1

Introduction

Today's power grids rely to a great extent on historically evolved infrastructures. To keep up with current development, such as increasing demand for energy, decentralized power generation from renewable sources and the need to integrate utilities in a world market the concept of *smart grids* is being formed. Renewable energy sources, including wind and solar power, do not necessarily follow customer demands. In future, customer demand should therefore follow energy production as much as energy production is adjusted to fulfill their demands. In a smart grid, all kinds of devices, ranging from power stations to electric cars, (smart) homes and even mobile devices, integrate seamlessly in a network that supports detailed and often real-time information about specific needs and availability. Furthermore, in such an environment a number of actors – individuals and systems – need to collaborate effectively in order to achieve common goals. Thus, efficient information and communication infrastructures are an inevitable component of such an architecture. Enormous amounts of heterogeneous data need to be exchanged, transported and processed.

1.1 Motivation and Objectives

In the United States (US) and the European Union (EU) standardization organization are currently developing roadmaps and models for reference architectures for a smart grid [15, 48]. These models provide an ability to capture all domains, zones and layers involved in and required for a working smart grid. Further, these models are the basis for a systematic capturing and classification of concrete services and use cases [20]. In addition to these models, the need for an assessment of the privacy impact is outlined by [13]. In order to allow a deeply integrated service interaction, such applications can be enhanced with semantic annotations that allow automated communication between services [19, 53].

Smart grids will also rely to a great extent on user interaction and user data. However, whenever user interaction is of high relevance, user acceptance is crucial. To achieve a high degree of user acceptance, applications in the smart grid need to fully align with user requirements. A semantic integration of applications and data sources as well as a semantic access layer give additional value to services and finally to the end user. User acceptance is crucial and can be achieved through user control by providing comprehensive and effective ways to interact with respect to service usage.

The approach presented here builds on insights from such a semantic service integration and goes a step further, by providing a semantic access layer to classify services and use cases based on their meta-data and therefore in the early design phase of the development cycle. In this report, a thorough investigation of requirements for a semantic access layer in the smart grid is conducted. State of the art approaches are evaluated with respect to their ability to incorporate meta-data and additional information such as privacy requirements, as well as their ability to allow service classification and retrieval with respect to such meta-data. Such existing approaches are also investigated regarding their capability to provide a source of expert knowledge.

An approach is presented that defines an ontology to classify smart grid services and use cases according to their meta-data. Classification is based on a generic and extensible data model to incorporate expert knowledge from various domains and draws on insights from smart grid model regions from the US and the EU. The main objectives of this work are to (i) provide a semantic access layer that allows use case classification with respect to user privacy requirements; and (ii) provide the basis to evaluate use cases in terms of service classification for more general impacts. In a concrete implementation this approach is evaluated in a productive smart grid middleware. For this purpose, the ontology based classification system is used as a Policy Decision Point for privacy-aware data retrieval where users are provided with a detailed privacy impact analysis for specific smart grid use cases.

The following requirements and objectives have been identified:

R_1 **Identify meta-information for classification.** A thorough analysis of state-of-the-art approaches in the domains of semantic service integration, use case analysis in the smart grid and privacy requirements engineering will reveal a set of information items that are potentially applicable to classify use cases and services.

R_2 **Design of an ontology for classification of use cases.** Classification will be done using an extensible and generic data model, an ontology. Ontologies allow the incorporation of expert knowledge, the usage of implicit knowledge as well as easy extensibility. The design of the ontology as well as the expert knowledge will draw on insights from model regions in the US and the EU.

R_3 **Use ontology based classification in a policy decision point.** A concrete implementation will serve as a policy decision point in a productive smart grid middleware. Typical use cases will be applied to test the classification system.

R_4 **Evaluate the system by applying new, previously unseen use cases.** For evaluating the classification system, new, previously unseen use cases will be classified. For these use cases the risk is assessed qualitatively and quantitatively.

R_5 **Evaluate the system's ability to generalize and to allow service classification.** Evaluation will show the ability of the system to be generalized to classify services as a whole. Furthermore it will be shown to what extent it is possible to go beyond privacy by being able to use arbitrary expert knowledge for classification.

R_6 **Implementation is open source.** The implementation will be available as open source. The package includes the source code as well as depending files and an appropriate documentation.

1.2 Structure

This report is structured as follows: Chapter 1 outlines the motivations and objectives of this work and defines a list of requirements to be met by this work. Chapter 2 introduces terms and definitions that are of relevance for this work. To this end, the concept of smart grid is explained, the term and meaning of privacy is presented and finally the aim and the principles of classification are shown. Chapter 3 focuses on work related to the ontology driven classification of use cases. Especially software architectures, semantic retrieval systems and existing standards in the field of this work are investigated. Chapter 4 outlines ontology design principles and ontology modeling issues. Further, the concept of pre-classification is discussed and the information that is explicitly and implicitly contained in the ontology is presented. Finally, the qualitative assessment based on attack vectors is introduced. Chapter 5 describes the model of threat patterns in detail and how these patterns are formed based on expert knowledge.

The chapter concludes with a thorough description of the quantitative assessment which yields the final risk value. Chapter 6 describes the implementation of the system in detail. Methods for representing graphs and threat patterns in XML, the classification implementation and corresponding issues are discussed. The policy decision point illustrates a prototypical application for the system in a working smart grid middleware. Chapter 7 outlines the evaluation of the system, including the methodology, sample use cases and the system's ability to generalize. Chapter 8 summarizes this report and gives a prospective for future work.

2

Terms and Definitions

This chapter introduces terms and definitions of relevance for the following work. First, the concept of smart grids is described. Secondly, privacy is defined and the key concepts and characteristics of privacy are identified. Further, the importance of privacy-aware data retrieval in the smart grid is emphasized. Ontolgies and state of the art formal languages to model knowledge are presented as well as the concept of reasoning to discover implicit knowledge. The principles of the classification process are outlined in order to introduce the following chapters. Finally, the concept of a Policy Decision Point is briefly introduced.

2.1 Smart Grid

Electrical power grid structures and concepts mostly date back to a time when energy was produced centrally and initially designed to fulfill a given demand. Today, society and technology are steadily moving towards a more sustainable usage of resources and shifting away from the exploitation that has driven early techniques of electricity generation. Renewable energy resources, such as wind, water and solar power are becoming the dominant sources for the sustainable production of energy. These sources, however, do not follow customer demands; by contrast customer demand will have to follow energy production. Subsequently, new intelligent and adaptive approaches for transmission and distribution need to be established. The grid, the linking – to date still missing – component between production and usage, is therefore referred to as being a smart grid [4, 10, 35].

The term smart grid, however, comprises much more than a pure technical point of view on distribution networks. In a smart grid all kinds of stakeholders, ranging from businesses to network operators and customers interact and integrate to achieve common goals. In [4], the following main reasons and objectives for a smart grid are identified: (i) foster usage of renewable resources; (ii) increase of efficiency and reliability in energy production and delivery; (iii) integrate customers allowing them to make informed decisions; and (iv) increased support for electric vehicles. In the past, the increasing demand for power has been addressed by extending distribution networks and by building new power plants. This was a cost-efficient and reasonably effective way to cope with aging infrastructure [10]. The focus is now shifting to the increase of the efficiency of the system as a whole. Today, cost-intensive capacities have to be provided to cope with infrequent peak-loads. In future, load should be curtailed which requires the integration of customers. The usage of electrical vehicles is seen as one of the key concepts for a future distributed and temporary storage of electrical energy [4].

While some of the challenges are in the field of electrical engineering, others are in computer science and business administration. To master these challenges of a smart grid, stakeholders from historically different fields have to cooperate efficiently, including electrical engineers and software engineers as well as an all new approach to future business models. This establishes the need to find a common terminology and taxonomy to describe use cases, services and the entire smart grid. In the US a mandate to develop a framework for such a standard has been issued to the National Institute of Standards and Technology (NIST) in 2007 through the Energy Independence and Security Act [4]. Their efforts resulted in the NIST Framework and Roadmap for Smart Grid Interoperability Standards [47, 48].

In Europe a similar approach has been carried out in 2011 under European Union Mandate 490 [23]. The European standardization organizations CEN, CENELEC and ETSI have been requested to establish a framework to coordinate, develop and integrate future standardization. This finally resulted in the publication of the Smart Grid Reference Architecture [15]. This reference architecture proposes the following main components:

(i) **European Conceptual Model.** The European Conceptual Model is heavily based on the model introduced by the NIST Framework and Roadmap for Smart Grid Interoperability Standards. Both models define seven domains and provide a basis for mapping the appearing communication flows. The NIST model, however, was initially designed for the US and therefore had to be adapted to fully meet European requirements for the smart grid: an additional domain for distributed energy resources has been added to take into account historically evolved infrastructures characterizing the European grid.

(ii) **Architecture Viewpoints.** The Architecture Viewpoints are abstractions of different stakeholders in the smart grid and represent interoperability layers. The set of layers consists of business layer, function layer, information layer, communication layer and component layer.

(iii) **Smart Grids Architecture Model.** The Smart Grids Architecture Model (SGAM) allows the design of use cases in the smart grid independent of underlying technologies. The focus is on the capturing of interoperability issues. The model as a whole has three dimensions, domains (generation, transmission, distribution, customer premises), zones (market, enterprise, operation, station, field, process) and interoperability layers. This model will be introduced in more detail in the following sections and it is the basis for further evaluation and classification of use cases.

(iv) **SGAM Methodology.** The SGAM Methodology allows the analysis of standards and a methodology for assessing smart grid use cases. This methodology is discussed in detail below.

Merging zones, domains and interoperability layers result in the SGAM Framework. This three dimensional model covers the interoperability layers introduced above and involves the aspects of the following list [15]. Furthermore, this list gives an overview of the components and their linkage for use case mapping as proposed in the SGAM Meta-model which integrates seamlessly into the architecture [20, 49]:

(i) **Business Layer.** This layer is the information exchange from a business point of view. It includes market parties, business models and entire products and services and describes the common business goal as well as the individual requirements for each actor. This layer therefore defines a business case, as well as business actors and their business requirements.

(ii) **Function Layer.** This layer represents applications, systems and services. The functions are derived from the business cases and supposed to be independent from actors and physical implementations. This view is a high level use case that is associated with a number of primary use cases that are invoked. Both are used by SGAM actors. Each primary use case composes a scenario and consists of a number of steps necessary to perform the scenario behind.

(iii) **Information Layer.** This layer shows how information objects are exchanged, involving functions, services and components. Each information object is provided by a data model standard and conveyed through an information object flow.

(iv) **Communication Layer.** This layer describes concrete protocols and mechanics for the interoperable exchange of information. The communication relation of components is modeled in this view.

(v) **Component Layer.** This layer contains the physical location of systems, applications and network infrastructure. The components are realized SGAM actors (from the function layer) and have both, an information and communication technology association and an electric association.

Research in the field of smart grid is carried out throughout the US and the EU. A number of model regions provide valuable contributions to research in the field of smart grids. In this report, the focus is on two recently established model regions in the US in Los Angeles, California[1] and in the EU in Salzburg, Austria[2].

The Los Angeles project is primarily focusing on in-field evaluation of demand response approaches, electric vehicle integration, customer behavior and security features. The project is scheduled to start an operational demo in 2014 [63]. The campus of the University of Southern California is an active part of the model region and the location for a prototypical implementation of a smart micro grid. Fields of research in the Salzburg project cover the implementation of a pilot for relevant technologies and the analysis of the impact of smart grids for customers. One of the main objectives of affiliated research is user-centric control and privacy in the smart grid.

Based on insights from both model regions the road maps for the following years are roughly as follows: as a first step customers are equipped with smart meters to gain real-time (or almost real-time) information on current loads. Following that, demand prediction and demand response for load curtailment are scheduled to be applied in demonstration micro grids. Furthermore, electric vehicles and their effective integration are subject of both model regions. Cross-cutting issues that are addressed in both, the US and the EU include security and privacy.

In summary, the smart grid greatly relies on the exchange of information between participating actors. The smart grid also poses the need for scalability. In future,

[1] `https://www.smartgrid.gov/project/los_angeles_department_water_and_power_smart_grid_regional_demonstration`

[2] `http://www.smartgridssalzburg.at/modellregion-salzburg/`

new, at the present day unknown, use cases will emerge and they will have to integrate seamlessly into then existing infrastructures at that time.

Whenever information is exchanged, data storage, protocols and security are topics of interest. Therefore, modeling IT infrastructures for the smart grid is a challenging task. Furthermore, when personal information is passed through a system and is subject to the processing of multiple actors, considering privacy aspects is inevitable, especially the excessive application of metering infrastructures and data analysis facilities makes the smart grid prone to privacy impacts.

This project again draws on the insights gained from experts in both model regions and use cases are modeled and evaluated with respect to individual characteristics.

2.2 Privacy

In Merriam-Webster [71] the term *privacy* is defined as "freedom from unauthorized intrusion". Even though this definition is meant to be related to an individual, this statement is immediately valid for data directly or indirectly related to an individual. The following section introduces state of the art definitions for privacy, as well as characteristics considered to be within the scope of privacy-aware data processing. Ethical, political or historic aspects of privacy are not within the primary focus of this section. In practice, especially in related standards [14, 46], the terms security and privacy are often used simultaneously and without explicit differentiation.

For this work it is assumed that security violations involves the *unlawful* acquisition of data (e.g., by certain attacks or by breaking system access points). Privacy violations, by contrast, are the unintended use of *legally* acquired data; unintended use is meant to eventuate from storage, processing or analysis of data apart from the original purpose (e.g., smart meter data originally collected for billing, is used to track the customers presence at home).

The definition for privacy and privacy violation that is used in this work is based on a threat tree. Figure 2.1 depicts a slightly modified version of the threat tree proposed by Neureiter et al. [50]. Privacy violation is set as "someone has personal information" *and* "misuse of personal information". Misuse happens either by purpose or unintended. Further, the possessing of personal information can either be by intention or unintended. In the latter case someone gains information by unauthorized combination of data sets or someone gains unintended information from one data set only.

Furthermore, throughout this report a terminology based on [5] is employed. The term *provider* refers to the party (individual or organization) that originally owns the data,

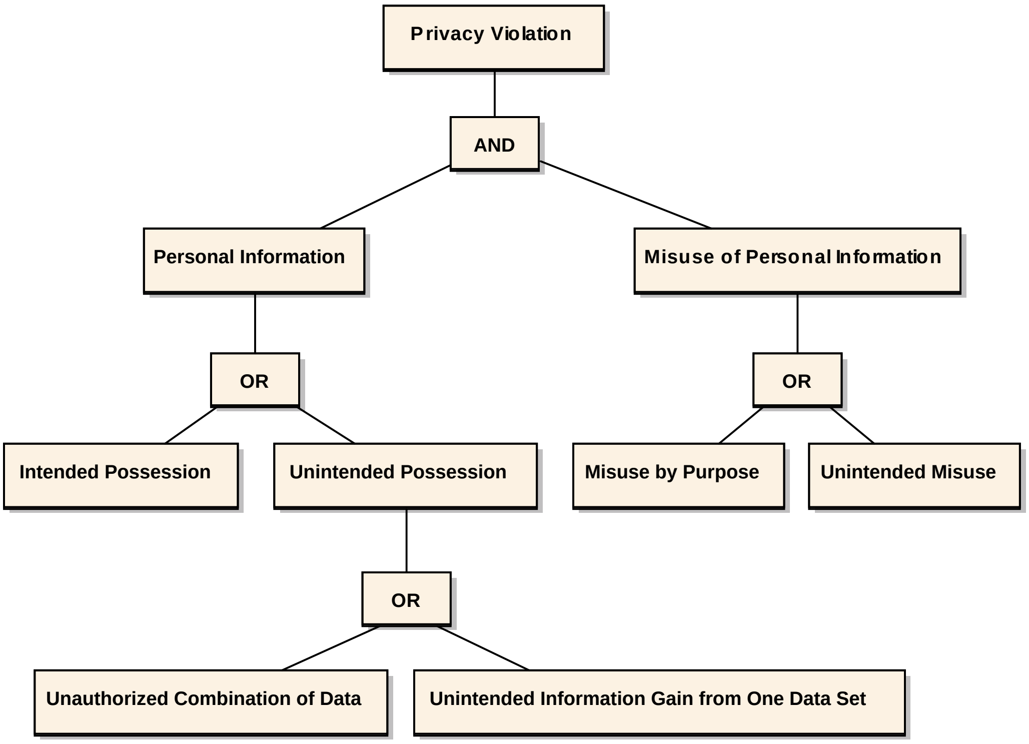

Figure 2.1: Graphical view of the definition of privacy and privacy violation. Figure adapted with modifications from [50].

thus provides it to another party, the *collector*, which stores or processes the data. *Third parties* eventually acquire data from the collector for a particular reason.

Terms that are related to privacy are *directly personal* data and *sensitive* data. These terms are legally defined by [61]. Directly personal data refers to data that allows to immediately determine the identity of a person. Indirectly personal data, by contrast, does not allow to determine that using legal procedures. Sensitive data comprises – but is not limited to – ethical meanings, origin, religious and political opinions and health records.

Mather et al. [43] discuss privacy impacts for cloud applications in depth. While the entire concept of the smart grid is much more wide-ranging than cloud computing, the data storage issues in specific are mostly the same. Some approaches, such as [25] and [58], even suggest cloud-hosted information repositories for smart grid data processing. The concepts mentioned can therefore be applied to privacy requirements in the smart grid in general. Data storage and processing is considered to be privacy-aware in accordance to [43] if the following is assured:

(i) **Compliance and data access.** Data usage must comply to the applicable regulations for the individual use case. It must be clarified who monitors compliance and who is responsible. Transparency and control over gathered data must be guaranteed at any time.

(ii) **Retention and ownership.** Data providers must be clearly identified. Further, policies that govern retention and responsibilities for this government must be available.

(iii) **Monitoring and data breach alerts.** All operations involved in storage and processing must be monitored at all time. Further, data breaches must be recognized and data providers must be notified immediately.

A comprehensive discussion about privacy in general and privacy in the smart grid can also be found in [73]. According to the authors there are various definitions of privacy that slightly differ. Generally, however, it can be seen as the right to maintain control over one's personal information, i.e., to whom it is revealed, for what purpose and how long the usage of information is granted. The following requirements and characteristics for privacy-aware design are concluded:

(i) **Full disclosure.** The extent to which the disclosure is communicated is an indication for the privacy risk. Further, a full disclosure is the basis for a comparison between services. A full disclosure consists of the following elements: a description including the type of data that is collected, a statement about how long data is retained and the intended use of the data.

(ii) **Consent to data collection.** Data collection must be acknowledged explicitly by users. Furthermore, the user must be notified when the underlying technology changes. For instance, when monitoring power usage and an analogous meter is replaced by a smart meter.

(iii) **Minimization of collection.** Collection is only justified as long as there is a functional requirement, thus the data collected is essential for the running of the system.

(iv) **Minimization of identification.** Analogously to the minimization of collection, identification – the linkage between data items and other personal information – must be tied to the operational needs of the system. This implies that data can either be collected about an equipment itself or about the user who owns or operates the equipment. The latter should be avoided.

(v) **Minimized and secure retention.** Data must not be retained for possible future usage, by contrast it must again be tied to the immediate use for the operation of the technology, hence there is a functional requirement for it. If it is inevitable to store data, storage must be protected. Especially inadvertent disclosure must be avoided to have privacy-aware data storage.

A thorough state of the art analysis of the understanding of the term privacy is conducted by Barker et al. and can be found in [5]. For the following taxonomy the authors identify four dimensions, each with a set of discrete points. The following is adapted from [5] where each point in the space is described as a four-tuple $P = \langle \text{Purpose, Visibility, Granularity, Retention} \rangle$:

(i) Purpose $\in$ {none, single, reuse same, reuse selected, reuse any, any}

(ii) Visibility $\in$ {none, owner, house, third party, all/world}

(iii) Granularity $\in$ {none, existential, partial, specific}

(iv) Retention $\in \{\text{date}, \infty\}$

The purpose dimension (i) refers to the way data is used. A *single* use is considered a one-time usage of data for a very specific purpose only. *Reuse same* widens this aspect to intend usage of data more than once. *Reuse selected* is a step further and additionally allows the usage for a wider set of selected, but foreseeable purposes. For *reuse any* the restriction to foreseeable purposes does not apply anymore. The last point, *any*, refers to a general usage or even publicly available data.
The visibility dimension (ii) refers to access policies. *Owner* indicates that only the owner has access. Note that Barker et al. distinguish between owner and provider. The ownership of data might move to the party storing the data and might not longer belong to the one originally providing it. *Housing* implies a visibility to the party housing the data, *third party* extends this visibility to externals that have been granted access to the data by the house. A visibility for *all/world* finally allows data access for anyone.
The granularity dimension (iii) refers to the fact that data can be exposed in various degrees of granularity to meet different access policies. *Existential* hereby refers to the fact, that privacy can – if at all – be revealed through existence tests only. *Partial* revelation of privacy is when techniques were applied to blur privacy sensitive data through aggregation, summarization or categorization. The most privacy sensitive case is when *specific* data is released; this is what most queries are doing when requesting data for a certain key.

All dimensions, but retention, have a common origin (*none*) where collected data is useless for any purpose, not visible to anyone or no information is captured at all. It will later be shown that P is equivalent to the description of a data item and can thus be used for classification.

By contrast to the pure description of privacy and to the development of taxonomies, a number of privacy assessment tools exist. Clarke [18] gives a thorough overview of the development of privacy impact assessment tools, their current application and references to assessment tools proposed in literature. In a smart grid, privacy plays a significant role and is crucial to leverage advanced concepts that require user participation such as demand forecast and demand response. The need for privacy-aware architectures and privacy-aware data retrieval is addressed by many authors. In [54], a thorough analysis of requirements for smart grid information and communication infrastructures is conducted, including privacy as a non-functional requirement. In [44], it is pointed out to what extent privacy influences end-customers. Further, the need for effective privacy-aware software infrastructures is confirmed.

These definitions and the requirements for privacy will be used to develop an ontology that captures expert knowledge about privacy in the smart grid. They are therefore used as a starting point for requirements engineering to subsequently determine the expert knowledge. Additionally, the following terminology will be used. A *threat* is a possible impact and thus a potential privacy violation. Privacy is actually *violated*, however, as soon as the threat becomes real. Recall the smart meter showcase, where one of the privacy threats is that meter data originally collected for billing is used to track the customers presence at home. As soon as this is actually done by either the billing company or any affiliates that legally have access to the data, the customer's privacy has been violated.

2.3 Ontology

The term ontology stems from a philosophical discipline that studies the nature and structure of reality [26]. The concept of ontologies, in specific computational ontologies, refers to the modeling and formal representation of a system. An ontology represents a taxonomy and as such captures all entities and relations that are of interest for a specific problem domain [26]. An attempt towards a taxonomy/ontology that models the term privacy – not formal yet, however – has already been made in the previous section. Figure 2.2 shows a formal representation of this taxonomy. This ontology includes classes and relationships as identified by Barker et al. [5].

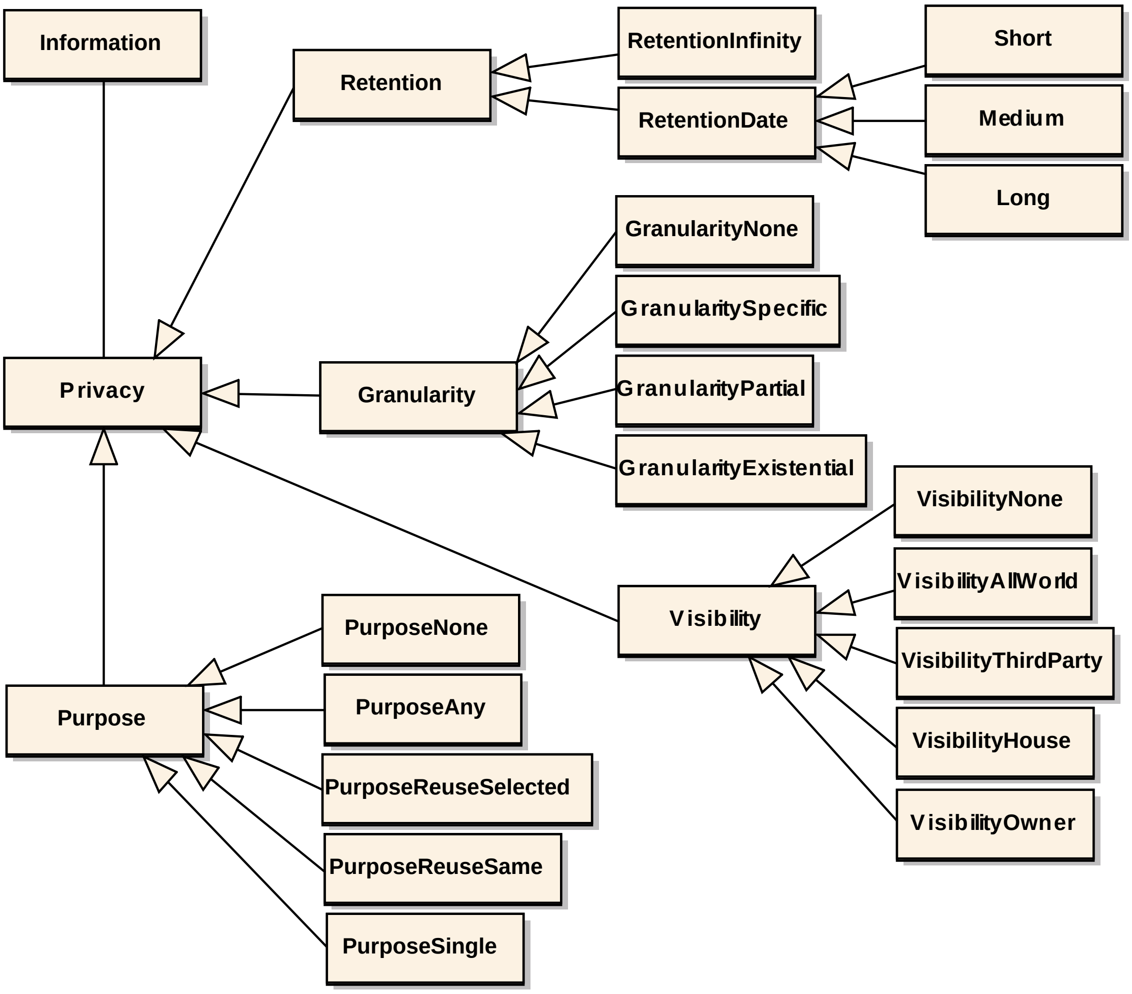

Figure 2.2: Simple ontology showing the classes and relationships of the privacy taxonomy by Barker et al. [5].

Even though this is simple, the building blocks for a use case that affects privacy, namely classes and their relations, are already identifiable: given a use case such as *person provides data*; *data includes name and address*; *data is used for billing*; *data retention is of indeterminate duration*; the ontology can be used to relate the information flows involved to privacy aspects.

Ontologies in computer science are often related to the concept of the semantic web. In a semantic web, information is augmented with semantics that allows automatic processing of information. Participating actors are able to find, combine and recombine information from different sources. Furthermore, inconsistencies are identified automatically [3].

The semantic web (and ontology modeling in general) gains its capabilities from a number of underlying and therefore fundamental concepts. First, the driving fundamental concept is known as the *open world assumption.* In formal logics, such as used for ontolgies, this assumption asserts the truth value of statements based on what is known at the time. Therefore it is not possible to draw conclusions on information that is not available. The information present is further not seen as all the information available. "Unknown" is a perfectly valid answer for a query that draws evidence from an open world. The closed world assumption, by contrast, emanates from the idea that all the information available is all the information available in total. Thus, in doubt queries will not return results [3]. Another concept that is essential, is the fact that ontologies are crafted for reuse and extensibility. Ontologies capture knowledge that cannot be fully anticipated at the time they are created initially [3].

A distinguishing feature of ontolgies is the degree of formalization. Guarino et al. [26] discuss this aspect in depth and present a continuum from informal to formal specifications. Informal specifications include the loose coupling of terms in ordinary glossaries. UML [55], database schemes and XML [66] are seen as more formal, whereas first-order, higher and modal logic rank at the top of formalization. Description logics (e.g., Web Ontology Language (OWL)) are seen as semi-formal compared to the others. The strength of a semi-formal specification such as OWL is its trade-off between efficiency and expressiveness. Thus, while allowing reasonable modeling capabilities, conclusions and consistence can still be drawn efficiently.

To link information from different, widespread sources a fundamental concept of the (semantic) web is applied to ontologies. The use of Uniform Resource Identifiers (URIs) that facilitates the distinct identification of objects. Information belonging to one single entity can be spread over multiple sources [3, 24]. To represent data in the semantic web, a formal model has been established, the Resource Description Framework (RDF). A resources is "a thing in the world" [3] and is uniquely identified with an URI. In RDF information is represented as triples in the *subject predicate object* pattern, e.g., `:Data :isUsedFor :Billing` or `:Data :isSubjectTo :Privacy`. This relationship is referred to as an object property, as the property is restricted to an object. Further, there are data properties where the property restricts to a literal such as string, double or boolean. Note that the full URI of `Data` in fact would look something like `http://owl.fh-salzburg.ac.at/odpe/samples/sample#Data`. The notation is shortened for readability here, by using a default namespace that is abbreviated with a colon. Alternatively, namespaces can be predeclared and abbreviated. Assume the resource `Privacy` is in the namespace `http://www.fh-salzburg.ac.at/odpe/samples/privacy#` abbreviated by `priv` and that all other resources are in the namespace `http://owl.fh-salzburg.ac.at/odpe/samples/others#` abbreviated by `oth`. The above triples would look like the following `oth:Data oth:isUsedFor oth:Billing` and `oth:Data oth:isSubjectTo priv:Privacy`.

The power of such a formalization is the ability to draw inferences. Inferences are drawn from explicitly modeled knowledge, also called axioms [3]. Given the following axioms (triples), it is possible to infer that `:Name` and `:Address` are sub classes of `:StoredData`: `:PersonalData rdf:SubClassOf :StoredData`, `:Name rdf:SubClassOf :PersonalData`, `:Address rdf:SubClassOf :PersonalData`. Hence, new, not explicitly modeled relationships are inferred.

To relate information to one other, there is a set of predefined identifiers. RDF has a limited set of identifiers and is extended by the Resource Description Schema (RDFS) and later by OWL. Today, most commonly OWL is used to model ontologies due to its comprehensive vocabulary [68, 69]. Predefined identifiers are located in default namespaces using the abbreviations `rdf`, `rdfs` and `owl`, respectively. The most important identifiers are presented here. Note that the OWL vocabulary includes all identifiers from RDF and RDFS [3, 65, 68, 69]:

(i) **rdf:type.** This predicate relates an individual to a class by stating that something is of the type of something else, e.g., `:SmartMeter rdf:type :Component`.

(ii) **rdf:Property.** Specifies a resource as a property that can be used as a predicate, e.g., `:isCompositionOf rdf:type rdf:Property` which can now be used as `:Name :isCompositionOf :Data`.

(iii) **rdfs:Class.** This identifier specifies a resource as a class, e.g., `:Component rdfs:type :rdfs:Class`.

(iv) **rdfs:subClassOf.** This predicate relates classes to each other and declares something as the subclass of something else, e.g., `:MeterData rdfs:subClassOf :Data`.

(v) **rdfs:subPropertyOf.** Analogously to the sub class predicate, this identifier relates properties to each other, e.g., `:isAggregationOf rdfs:subPropertyOf :isCompositionOf`.

(vi) **rdfs:range.** This identifier is borrowed from mathematics and defines how a property is to be used, i.e., the type of the object that has to be used in a triple, e.g., `:isSentBy rdfs:range :Actor`.

(vii) **rdfs:domain.** This identifier is the counterpart to the range and defines the subject of a triple, e.g., `:isSentBy rdfs:domain :Data`.

(viii) **owl:inverseOf.** This predicate makes an inverse relationship explicit to draw inferences on it. Given `:Data :isSubjectTo :Privacy` and `:Privacy :isOfInterestFor :Data`, this can be made explicit by `:isSubjectTo owl:inverseOf :isOfInterestFor`.

(ix) **owl:TransitiveProperty.** This identifier specifies that a property is transitive, e.g., `:isCompositionOf rdf:type owl:TransitiveProperty`.

As mentioned above, it is possible to define object properties as needed. These properties can be derived from other properties and may be characterized by the following important axioms in addition to the ones mentioned above [3, 65, 68, 69]:

(i) **Equivalent Object Property.** This axiom states that in a set of object properties each property is equivalent to the other. Note that a similar concept is available for equivalent classes.

(ii) **Disjoint Object Property.** This axioms states that in a set of object properties each property is pairwise disjoint with the other. Note that a similar concept is available for disjoint classes.

(iii) **Inverse Object Property.** This axiom makes the inverse of an object property explicit. This is commonly expressed in predicates such as for example `:isOwnedBy` is the inverse of `:owns`.

(iv) **Functional/Inverse-Functional.** This axiom borrows its semantics from mathematics and restricts the value of a property to at most one. Inverse functional is the counterpart. For instance, assuming that data is sent by one and only one sender, the property `:isSentBy` could be functional.

(v) **Reflexive/Irreflexive.** This axiom states that whenever two individuals are connected with a reflexive property, this means the individual is connected through this property with itself. Irreflexivity by contrast makes explicit that no individual is connected to itself.

(vi) **Symmetric/Asymmetric.** This axiom expresses symmetry or asymmetry between objects.

(vii) **Transitive.** This axiom states transitivity. Transitivity is crucial for the purpose of this work since data flow analysis draws to a great extent on the fact that data that is sent from A to B and from B to C is in fact also sent from A to C.

It has already been mentioned that a powerful aspect of ontologies is to draw inferences and by doing so, to retrieve implicit knowledge. This process is known as *reasoning* and is done by a component called *reasoner*. Reasoners check the consistency of an ontology and are based on description logic. In literature reasoners are classified by their specific approaches towards logical analysis and the performance they achieve. Commonly used state of the art reasoners are HermiT [56], Pellet [59] and FaCT++ [62], all based on tableau calculus.

First, reasoners perform a consistency check. An ontology is consistent if it has no contradictory axioms or assertions. Besides the consistency check, Sirin et al. [59] define the following main objectives for an ontology reasoner: (i) concept satisfiability, which checks if a class is able to have any instances. If unsatisfiable classes have instances, the ontology is inconsistent; (ii) classification, the process of computing the complete class hierarchy and all sub class relations; and (iii) realization, which finds in a class hierarchy the most specific class an individual conforms to and further gets all the types of an individual.

Reasoning takes place on two different layers in an ontology: classes and individuals. The first is also referred to as the T-Box (terminology box), the latter as the A-Box (assertion box). While from a reasoner's point of view this is very similar, from a modeling perspective, classes and individuals are two different concepts and they closely resemble object oriented programming with classes and objects [3] or the even older distinction between type and token. Classes are an abstraction and define concepts, whereas individuals or objects are concrete, identifiable instances of classes.

In an ontology, a reasoner assigns sets of individuals to super classes based on conclusions drawn from class expressions. Class expressions are description logic statements, for instance formulated in Manchester Syntax [29, 69]. This syntax is designed to be easily readable by avoiding verboseness (such as in RDF/XML) and intricacy (such as OWL Abstract Syntax). Manchester Syntax is characterized by an infix notation and by a minimum number of brackets necessary and it supports the following, limited set of natural language keywords: `and`, `or`, `not`, `some`, `only`, `min`, `max`, `exactly` and `value`.

Given the privacy ontology depicted in Figure 2.2 a class `:CriticalInformation` could have the following equivalent class expression in Manchester Syntax: `Information and isSubjectTo some AllWorldVisibility`.

Manchester Syntax is useful to model expressions about equivalent classes, disjoint classes and super classes, it has, however, limited purpose for more complex queries. SPARQL is a query language that allows to retrieve information from RDF stores. SPARQL, currently in version 1.1, is described by Perez et al. [51], described in detail in [21] and it has been a W3C recommendation since 2008 and 2013, respectively [67, 70]. SPARQL is designed as a graph-matching language, i.e., a given graph pattern is matched against the data. Queries are structured in three parts, *pattern matching*, *solution modifier* and *output*.

It should be noted at this point, that SPARQL is in fact a more powerful language allowing not only queries to retrieve, but also to insert, modify and delete data. The latter is omitted here, since for this work only queries that retrieve information are of importance.

In general, the building blocks of a SPARQL query are as follows (i) prefix declaration (`PREFIX`); (ii) data set definition (`FROM`); (iii) result clause (`SELECT`); (iv) query pattern (`WHERE`); and (v) query modifiers (ORDER BY, LIMIT, . . .). The following simple example query is using a pattern that retrieves information about *something* that is subject to privacy. Note the namespace abbreviations `oth` and `priv`, these abbreviations must be declared in a preceding `PREFIX` clause that is omitted here: `SELECT ?data WHERE` { `?data oth:isSubjectTo priv:Privacy` }. This query can be made more precise by specifying that only something of type data that is subject to privacy should be retrieved: `SELECT ?data WHERE` { `?data oth:isSubjectTo priv:Privacy . ?data rdf:type oth:Data` }. The dot denotes an *and* condition, i.e., both expressions must be true.

SPARQL has been chosen as the query language for this work for the following reasons: (i) SPARQL matches patterns to triples in the RDF store. The more information provided in a query, the more precise are the results. However, if only little information is available all the triples in the store that match are returned. Thus, such a language is ideal for formulating expert knowledge; (ii) SPARQL provides powerful constructs for defining patterns, including advanced filters based on regular expressions [30]; and (iii) SPARQL syntax is based on natural language (and is similar to SQL [32]) which makes it easier for experts to formulate their knowledge and requirements. The arising issue of naming conventions is discussed later in this report.

For the evaluation of privacy an ontology driven approach is used for two main reasons: (i) expert knowledge from various domains is captured and integrated to make it explicit. Ontologies are a feasible way to maintain a formal representation of expert knowledge while still being capable to handle future extensions. OWL is used as a modeling language having a reasonable trade-off between efficiency and expressiveness; and (ii) ontologies and reasoners allow to draw inferences and to reveal implicit knowledge from explicitly stated axioms. Relationships that are identified can then be used for classification.

2.4 Classification

Classification is the process of assigning an object to a certain class, given a set of feasible classes. Classification is used in mathematics and engineering as well as in economics and social studies. Thus, there are a number of approaches for classification ranging from highly sophisticated self-learning algorithms based on numbers and probabilities to manual classification done by humans.

In mathematics it is often referred to the more general term pattern recognition that is concerned with the automatic discovery of patterns. This approach to pattern recognition and classification is well established and discussed thoroughly in literature [7, 22]. It is generally differentiated between unsupervised and supervised learning. In the latter, the system is first undergoing a training phase to adjust internal parameters to best fit a given and previously targeted training set. New, previously unseen data is then applied for actual classification. Unsupervised learning, by contrast, aims to find structures in given – but untargeted – data without undergoing prior training phases. For supervised learning there are a number of established and well understood algorithms, including deterministic and stochastic approaches. Common examples include Support Vector Machines and Neural Networks [22]. When facing a classification problem, it is beneficial to integrate as much knowledge as available about the problem domain. This is referred to as *a priori* knowledge and in the context of this work later as *expert knowledge*. Depending on the algorithm a reasonable amount of expert knowledge can be applied, e.g., using a priori probabilities in a Bayesian decision.

A considerably distinct approach to the classification problem is addressed by ontologies. Classifiers based on ontologies are subject to current research in various domains. In [6], building types are detected using ontologies by Belgiu et al. Therefore, a set of classifiers are defined to describe the equivalence of classes in formal description logic, e.g., in Manchester Syntax. A reasoner is able to determine the most specific class an individual belongs to. Text classification is a different field that applies ontologies, such as presented by Bloehdorn and Hoto in [8]. The authors propose a machine learning technique that uses features with a higher degree of semantics than plain document based approaches. Based on these features, an ontology is used to represent background knowledge about certain concepts for improving the classification performance.

Since ontologies formally capture present knowledge about a domain, they allow the drawing of conclusions. Besides the knowledge that is explicitly modeled, appropriate logic will reveal implicit facts through reasoning. For the purpose of classification, however, it is evident that ontologies allow to draw inferences for given individuals, simply due to the fact that all relationships are present, either explicit or implicit.

2.5 Classifier

For the following work the definition of classification is specified as follows: Classification is the process of assigning a use case UC to a certain class c from a given set of classes C. Formally, a classifier function α is defined, so that a class c is assigned by $c = \alpha\,(UC)$. The decision is made using an ontology, the ontology being finally based on explicit and implicit expert knowledge. The aim of the classification process is to identify potential threats for use cases. A class therefore represents a certain subset of threats T^* where $t \in T^*$ and $T^* \subseteq T$. T is the set representing all potential threats. The output displayed to the user is a table showing the threats as well as a numeric risk value. This is referred to as the *threat matrix*, which is discussed in detail later. Note that the threat matrix is not a matrix in the mathematical sense, but a table listing threats, such as commonly used in security analysis.

The principle classification rule α for a use case is as follows:

$$\text{Assign } UC \text{ to } c_i \text{ if } t \in T_i^*, \forall t \in T, 1 \leq i \leq C$$

This section outlined the principal classification process and its main components including input (UC), output (T^*) and the classification procedure itself (α) as well as the base set of threats (T). Figure 2.3 shows the four main components and their interrelation.

The input UC of any classification is a use case of a specific smart grid communication scenario. Use cases are modeled as Data Flow Graphs and transformed to XML. The classification is based exclusively on the data modeled in the graph.

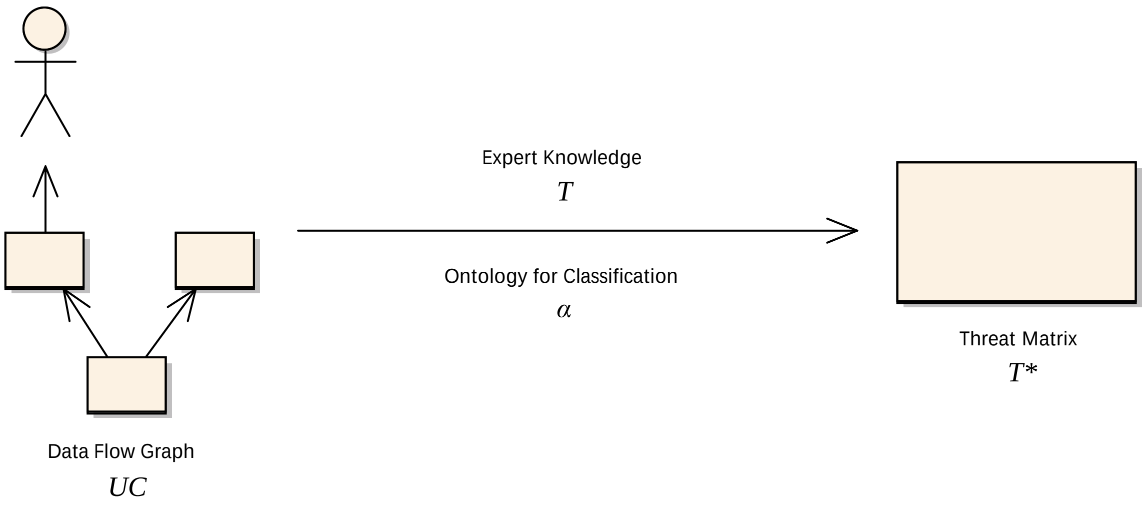

Figure 2.3: The four main components for the principal classification process, Data Flow Graphs, expert knowledge, ontology and threat matrix.

The classifier α consists of two main components: (i) expert knowledge, which is derived through the investigation of best practices and a thorough requirements analysis; and (ii) the ontology for classification.

The classification is carried out as a discriminative process using both, a pre-classification based on the graph itself and an identification of threats based on expert knowledge.

The output c is represented by a threat matrix. A threat matrix contrasts the known list (through experts) of potential threats with the identified threat level. The threat level is determined by the classification.

2.6 Policy Decision Point

This section briefly introduces the principal concept of a Policy Decision Point (PDP). The PDP is chosen as an example for a practical application that immediately benefits users. The PDP-PEP architecture is standardized as Extensible Access Control Markup Language (XACML) in [52]. An application for this architecture in the smart grid is presented by Jung et al. in [33]. The authors describe how to use the pattern to equip a middleware in the smart grid with an access control mechanism that incorporates distributed services and access policies. The work presented here builds on that concept, however defers in terms of how policies are created.

In general, a PDP is a component that evaluates access requests and issues some authorization. The PDP therefore provides some mechanisms to authenticate users, usually by prompting credentials such as username and password. The PDP then checks in a repository (policy store) if a certain user is granted access to a certain resource. The assessment framework presented in this work is used as a PDP in order to allow privacy-aware data retrieval in the smart grid. The scenario at hand is as follows: a user wants to access a new application or service in the smart grid. This application or service has a certain privacy impact that has been assessed with this framework. Additionally, the application or service is governed by a Policy Enforcement Point (PEP). The PEP redirects the user to a PDP that displays to the user a list of privacy implications associated with this particular application or service. The user is then requested to confirm the intention to use the application or service. If the user accepts, the PEP grants access.

The scope of this work is not on providing an authentication mechanism, but on providing a transparent way for users to manage the privacy impact of their applications and services. In addition to a PDP, a typical scenario found in practice comprises the following components:

(i) **User.** An individual or an external application that attempts to access an application or service that is governed by a set of policies. Here, end users are addressed and thus these users are interested in using a certain application or service.

(ii) **Application/Service.** The application or service that is attempted to be accessed. In this setting, this might include services for remote monitoring of energy consumption or tools for managing preferences for electric vehicle charging. Each application or service corresponds to a use case and is described formally as a Data Flow Graph.

(iii) **Policy Enforcement Point.** The PEP guards access to a certain application or service and asks the PDP for authorization. For this setting, the PEP grants or denies access based on the users decision after being displayed the privacy impacts.

(iv) **Policy Store.** Policies are stored in a database. The PDP forms a query for that database in order to retrieve if a certain user has access to a certain application or service.

3

Related Work

This chapter is a thorough investigation of related work in the field of service integration, semantic access and semantic classification and security analysis. Furthermore the concept of Data Flow Graphs and their relationship to standardized smart grid reference architectures is described.

3.1 Software Architectures

The smart grid relies to a great extent on software architectures to effectively handle the vast amount of information that flows from one actor to another. As information processing and exchange is one of the key success factors, research groups in the US, e.g., [58, 25], and in the EU, e.g., [53, 64], have ongoing work in this domain. Approaches range from low-level architectures focusing on transportation and network protocols to high-level, cloud-hosted information repositories. This section is dedicated to the description of services in the smart grid and state of the art software architectures. The focus is on approaches integrating semantics in services and data for privacy-aware retrieval.

The development of information architectures in the smart grid is conducted on different levels. Apart from the typical OSI-like structuring [1] with levels – for this applications mainly including physical, network, transport and application layers – most middleware architectures are multi-tier architectures themselves. This typically comprises data model, business logic control and data view and is therefore often referenced as the model-view-controller pattern.

In [34], an approach for a decentralized data-centric information infrastructure is proposed. This work mainly deals with the low-level communication of sensors and meters.

The focus of this work is on the investigation of network and transport protocols with respect of reliability and latency, as well as on an appropriate security architecture. Security issues are addressed profoundly on the network layer with a focus on reliability and protection of data exchange, especially when being transmitted through public networks. This approach, however, is not concerned with privacy and the storage and processing of immediate end-user data.

When the link is drawn from the network layer to middlewares and finally to the end-user, new challenges and objectives become evident [74]. These challenges and objectives can be summarized as follows: There is a need for a clear specification of the relationship between what the user needs and the middleware provides, a number of heterogeneous clients and devices need to be supported and for the sake of extensibility and portability the independence of the types of devices is crucial. Additionally, service quality assurance metrics are a crosscutting concern of increasing interest. Bridging from data sources to customers is also the aim of the work presented in [64]. In this paper, a smart grid middleware that deals with the uniformization and aggregation of data from heterogeneous sources is presented. Privacy is enforced by using rigorous, but adaptive access policies and by providing full transparency to data owners. This work, however, has only limited semantics included in data storage so far.

A markedly broader approach towards a semantic service integrating infrastructure is developed by Rohjans in [53]. The author investigates available standards and specifications, as well as potential interoperability integrations in order to allow automated communication processes in the smart grid. First, a detailed analysis of requirements for a future power systems is conducted. Therefore, a set of requirements for an information and communication infrastructure is identified by Rohjans et al. [53, 54]. The objective of the work presented, is to model the information and the services that exchange the information. Finally, the underlying semantics is used for automated discovery, aggregation and requests of data in a service oriented middleware. This approach is not broadly taking into account privacy aspects.

A very high-level approach towards a smart grid middleware is presented in [58, 75] and [76]. This work, which is part of the Los Angeles Smart Grid Model Region, aims to foster the integration of heterogeneous data sources for demand response. Demand response deals with the curtailment of peak loads and builds on the interaction with customers. Customers are offered incentives and as a result reduce energy consumption during peak periods. The University of Southern California campus in Los Angeles serves as a show-case project for demand response in the model region. Another objective of the project is to forecast demand prior to the occurrence of peak loads and to react preemptively. The proposed solution uses semantics to integrate information from widespread sources. Besides actual power consumption this also includes mobile phones, electric vehicles, weather, traffic and social networks. An ontology serves as the basis for the semantic information integration. Even though the authors mention the need for privacy, the approach does not include privacy features in the semantics.

An approach focusing explicitly on privacy-aware data retrieval is described in [60]. The authors propose an architecture that brings together service provider and service user in a policy matching process. This work heavily relies on semantics and is based on semantic web technologies. The implementation uses ontolgies and additional security features to match policies and to control adherence. Policies can be defined with respect to the purpose of data retrieval, the perspective and the receiving agent. Further, an example taxonomy for purposes has been defined that includes the following: a purpose can either be (i) non-commercial; (ii) consulting; (iii) commercial; or (iv) statistics. In summary, the work described in [60] contains a taxonomy and a framework for policy definition.

3.2 Privacy and Security Analysis

There are a number of tools for the assessment of privacy and security features of systems. This set of tools ranges from purely conceptual and more management oriented frameworks, e.g., ISO 27001 [31], to fully integrated and automated analysis systems, e.g., presented by Chen et al. [17]. Even though the focus of this work is on privacy, security related work in the domain of analysis frameworks provides valuable insights.

Conceptual frameworks such as standards and recommendations for effective management systems provide a source of expert knowledge for classification. Focusing on management aspects of information security, the ISO 27001 is the basis for many security assessment tools.

In [45], McKenna et al. give a general overview of privacy issues related to smart metering. However, the authors also stress to what extent other applications in the smart grid, such as demand response, rely on the data gathered by smart meters. Especially the metering frequency is identified as one of the crucial indications for privacy critical settings. While higher frequency enable additional services, also more information about customers is revealed. Lower frequencies, by contrast, are privacy preserving, however, additional applications might not have appropriate data for their effective operation.

A comprehensive study of privacy and security requirements in the smart grid is carried out by [57]. The authors focus on potential issues in the Smart Grid Model Region Los Angeles and identify specific roles and their particular requirements. Roles in a smart grid that are identified include utilities, consumers and third party service providers. Privacy and security aspects that must be taken into account are data leakage, data modification, data/platform sharing and access diversity. This is also reflected in the concrete security and privacy issues that stem from data diversity, data granularity, data transformation, data access and sharing and data archival. All together, a comprehensive taxonomy including potential threats and a profound basis for a security assessment is presented, also including the ramifications on system engineering.

Automated assessment frameworks are fully integrated approaches. A workflow-oriented security assessment tool is presented in [17]. The framework proposed in this paper is based on the evaluation of argument graphs. The systems input are security goal, workflow description, system description, attacker model and evidence. The assessment itself applies a discriminative set of graphs, labeled as G-graph, GS-graph and GSA-graph. Each graph captures logical relations and is later evaluated through the solving of boolean random variable equations. Based on a workflow description, such as on-demand reading from smart meters, the G-graph contains the workflow goal, the actors involved and the messages exchanged. The GS-graph is a more detailed view on the system taking into account the system description, including physical components such as data concentrator units and head end systems. The GSA-graph finally includes the attacker model which consists of attacker action templates, e.g., the physical access to devices. The result of the assessment process is quantitatively presented as an availability score and a confidentiality score. Both are plugged into the system by the evidence, which is based on (statistical) data about the devices.

A project proposing a security analysis that is based to a great extent on semantic web concepts can be found in [2]. The SemanticLIFE framework by Ahmed et al. is a considerably broader approach with the objective to organize collaboration more effectively. The system architecture is highly modular and based on a service-oriented design. To assess attacks on the system, an ontology driven approach is suggested which consists of the following three models: (i) a user environment ontology capturing the kind of operating system, software and hardware configurations of users; (ii) a project ontology describing projects and related concepts such as tasks, resources and costs; and (iii) an attack ontology describing different kinds of possible attacks. For an ontology based risk assessment attack patterns and specific preconditions (e.g., operating system version, open ports) are matched. Further a quantitative analysis is provided by calculating the Annual Loss Expectancy (ALE) by $ALE = \sum_{i=0}^{n} I(O_i)F_i$. $I(O_i)$ denotes the impact of harmful outcomes in a monetary quantity and F_i is the frequency of an outcome i.

Another ontology driven approach for privacy classification in the smart grid is presented in [38, 39]. The authors propose a privacy by design framework based on a privacy assessment cycle. The cycle integrates high level privacy statements and technical privacy statements, which are further used to evaluate system behavior and system properties. The approach takes into account both, a formal specification of requirements and a formal system description. For verification, the privacy requirements are matched to the system model using an ontology based mapping. Privacy violations which the system aims to detect are (i) unpermitted operations on data; (ii) individual privacy preferences violations; (iii) absence of security mechanisms; and (iv) violation of retention or minimization. Privacy is evaluated using domain and application independent system properties, called privacy indicators. They are derived from the system model using logic rules. Privacy risk is calculated by combining a set of privacy indicators. The system model is designed to show components, operations, data items and information flows. The ontology defines meta-information and refines abstract concepts to concrete data items (e.g., address) [38].

3.3 Data Flow Graphs

A Data Flow Graph (DFG) is a structured way to represent a smart grid UC according to the Smart Grid Architecture Model as presented in the European standardization mandate M/490 [15].

A two dimensional representation of the model, omitting domains and zones, but including the components of the use case mapping process is shown in Figure 3.1. Due to its origin from the SGAM Toolbox[1] it is referred to as the SGAM Meta-model.

In [15], the SGAM Methodology is described in detail. The aim is to assist in the development of smart grid architectures and to map use cases in order to identify gaps with respect to existing standards. The suggested approach consists of preceding use case analysis and the subsequent development of the five layers.

For information security matters an associated standard for Smart Grid Information Security [14] has been established and is under development. However, it does not address privacy issues sufficiently yet. The NIST Guidelines for Smart Grid Cyber Security [46], by contrast, provide a thorough analysis of privacy impacts.

[1] http://www.en-trust.at/downloads/sgam-toolbox/

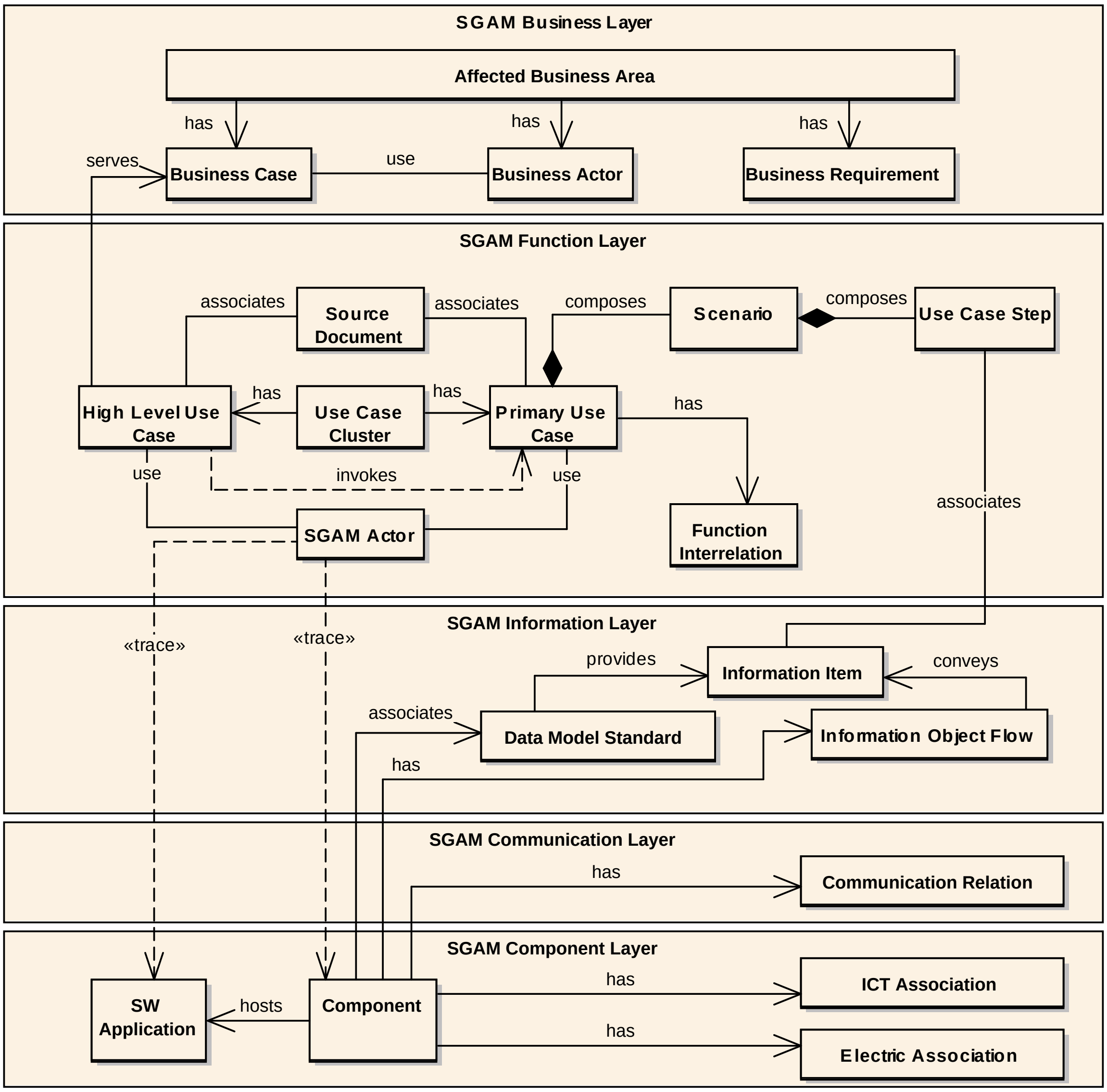

Figure 3.1: The SGAM Meta-model as proposed in and adapted from [20, 49]. The model involves the five layers from the reference architecture as well as the use case mapping process.

For the following work, the M/490 reference model has been chosen for two reasons as the basis for evaluation of use cases: (i) the NIST model is mainly a subset of the European reference model, thus all insights gained in the US can be adapted to the European model; and (ii) the SGAM Toolbox is a powerful modeling tool based on this model.

The SGAM Toolbox is a set of objects for the development and modeling of smart grid systems in adherence to the reference architecture described above. The toolbox itself is based on the Unified Markup Language (UML) [55] and integrates in Enterprise Architect[2]. The development process for smart grid use cases is based on the SGAM Methodology. The process is described in detail in [20] and consists of three phases: (i) System Analysis; (ii) System Architecture; and (iii) Platform Specific Model and Platform Specific Implementation. In the system analysis phase a computational independent model is designed. The purpose of this model is the description of the system uncoupled from implementation considerations. With a sufficient understanding of the functionality of the system, the system analysis phase yields a platform independent model. This model is finally refined to a platform specific model in the system architecture phase. In [20], this approach is proposed for the use case mapping process from the SGAM Methodology. The work presented in this report is not only set on the resulting use cases, but also provides an entire framework for the development of use cases. Thus, in accordance with the model driven design of smart grid solutions, it is feasible to evaluate possible impacts and potential threats at design time.

DFGs can be modeled graphically with the SGAM Toolbox. The toolbox provides a set of actors, connectors and components in adherence to both, UML and the SGAM Methodology. During the three phases of the development process, each layer is developed iteratively. Relationships between objects in each layer are further maintained consistently.

Formally, DFGs are directed graphs and thus representing actors and data items that flow from one actor to another, as such DFGs are a set of nodes (or vertices) and edges. They can be represented in their simplest form by $DFG = (A, D)$, where A denotes the set of actors and D denotes the set of ordered data items. This notation can be extended by allowing labeled graphs. A labeled graph is given by $DFG_l = (\Sigma_A, \Sigma_D, A, D, A_{source}, A_{target}, l_D, l_A)$. Σ_A and Σ_D are the finite alphabets for the vertices (actors) and the edges (data items). $A_{source} : D \to A$ and $A_{target} : D \to A$ indicate the direction of the edge. The labels are given by $l_D : D \to \Sigma_D$ and $l_A : A \to \Sigma_A$. A natural language interpretation is as follows: $l_A : A \to \Sigma_A$ denotes that

[2] `http://www.sparxsystems.com.au/products/3rdparty/frameworks.html`

an actor is characterized by a set of attributes and $l_D : D \to \Sigma_D$, denotes that a data item is characterized by a set of attributes; $A_{source} : D \to A$ and $A_{target} : D \to A$ are indicating the information flows from a source actor to a target actor. It will later be shown that the finite alphabets Σ_D and Σ_A are crucial, as they directly correspond to attributes in the ontology and therefore provide essential information to the classifier.

3.4 Scope of This Work

In the previous section, state-of-the-art approaches for smart grid middlewares and privacy and security frameworks have been presented. Furthermore, an overview of the smart grid reference architecture and how to map use cases to this architecture has been given.
Based on the above investigations of state of the art approaches, the following aspects, that have not yet been considered sufficiently, are seen as of high relevance for smart grid software architectures:

(i) **Structured meta-data inclusion.** State of the art approaches allow no or very limited inclusion of structured meta-information only. Data items are annotated with additional information about sender, receiver and properties describing the data item itself. This allows a detailed analysis of data flows in the smart grid.

(ii) **Design-time impact analysis.** Building complex and heterogeneous systems are expensive in practice. Design-time analysis for the entire system or for certain services helps to reduce the costs of the development process. Impacts and potential threats can be analyzed prior to implementation and testing.

(iii) **Privacy-aware data retrieval.** Privacy-aware data retrieval is meant to provide full transparency for the actors involved, especially for customers. For the latter, the representation of privacy impacts has to be easily understandable.

In summary, these approaches for IT infrastructures either integrate no semantics at all or semantics for service discovery and aggregation only. Existing smart grid middlewares do not use additional meta-information to evaluate the privacy impact of services or more general influences to the infrastructure. These middlewares further provide limited semantics for privacy-aware data retrieval. Automated security and privacy assessment tools provide valuable results in the early design phase, however they do not take into account a privacy impact analysis that is designed to integrate in the smart grid reference architectures or is primarily based on meta-data. Use cases that are mapped to the reference architecture using the SGAM methodology, by contrast, provide a sufficient amount of semantic annotations for data items and actors. This work bridges the gap between the design-time analysis of data flows and operational issues of privacy-aware data retrieval. An outline of the work discussed in this report is also presented by Knirsch et al. in [36]. When dealing with heterogeneous systems and user data it is evident, that finding approaches to classify the impact of use cases on such architectures – even prior to implementation – is important. This work focuses on the development of a classification framework for use cases that facilitates the evaluation of privacy impacts on services. Classification can be performed (i) at early design time by systems engineers; or (ii) before subscription by customers. System engineers have a powerful tool to do low-cost analysis of projects and customers have full transparency of their data processing. As a proof-of-concept implementation, an existing smart grid IT infrastructure [64] is enhanced with a policy decision point taking into account the classification results.

4

Classification Ontology

This chapter describes the design of the ontology that is used for classification. First, the design principles are discussed in detail. This section focuses on design decisions, naming conventions and the principal approach towards an ontology for classification. Further, the modeling of particular features of the ontology is discussed as well as the objective and the implementation of pre-classifiers. Finally, the information that can be extracted, either implicitly or explicitly is described in this chapter.

4.1 Design Principles

In this section the design principles as well as the naming conventions used for the proposed ontology are discussed in detail. When it comes to the development of ontologies it is usually about capturing (domain) knowledge. Further, the concept of ontologies is tightly related to the principles of the semantic web. Whenever new concepts are to be formalized, it is therefore advisable to browse for existing ontologies intending to model the same concept. Allemang and Hendler [3] propose the following methods for getting started in ontology design: (i) finding models that meet the requirements; (ii) making use of less-formal semantic models such as thesauri, organized vocabulary or information organizations; and (iii) engineering models from scratch.

The purpose of the ontology that is modeled for this project, is to allow classification for use cases based on DFGs. In specific, a DFG is mapped to the ontology so that all the relevant information is represented as individuals. DFGs themselves are based on the European Smart Grid Reference Architecture. To the best of the author's knowledge, there is currently no ontology for the European Smart Grid Reference Architecture. However, the European SGAM provides an organized (and standardized) vocabulary.

Furthermore, DFGs extend this vocabulary. The ontology presented here is therefore modeled from scratch, but using known sets of vocabulary whenever applicable.

Another issue that arises is the decision if something is modeled as a class or as an individual. This issue is especially interesting when it comes to reusing ontologies, as different applications might have different ideas of what is an instance [3]. Given a use case, something as concrete as a customer can be seen as a class (something is an instance of the conceptual customer) or as an individual (customer is an instance of an actor). This poses the question of the level of detail in contrast to the degree of reusability. In [3], it is suggested to model a concept as a class if there is at least one possible instance. For the purpose of this work most abstract concepts – but still allowing provision for a profound analysis – have been modeled as classes and the components of DFGs are modeled as individuals.

Recalling the triple pattern *subject predicate object*, naming conventions follow this pattern in order to make reading easier and relations more clear. Subjects and objects are nouns starting with a capital letter (e.g., Actor, Data). Predicates are verbs (e.g., owns) or a combination of verbs and nouns (e.g., isOwnedBy). This yields a very natural description for triples in the known form `:Actor :owns :Data` and its inverse `:Data :isOwnedBy :Actor`. Note that this is principally true for object and data properties. Given a class `:Data` with a property `:Frequency` of type int is described by a triple `:Data :hasFrequency 100`. While this notation is common and suitable for ontology modeling, it looks somehow unnatural when modeling DFGs in UML since data properties belong to a class and prefixes such as *has* or *is* are redundant.

For the most suitable notation a combination of both is used, object properties feature a verbose, but natural notation to make integration of expert knowledge easier and data properties feature the less verbose notation to make the modeling of DFGs easier.

4.2 Modeling

The objective for the model is to have an ontology that allows full representation of all actors, information objects and data flows of a DFG. Therefore, having all classes, object and data relations for a lossless bidirectional mapping from a DFG to a set of individuals in the ontology is the intended outcome. The TBox of the ontology represents the following main concepts:

(i) **Actor.** The `:Actor` class is an abstraction of any kind of actor in the DFG. The class is specialized by disjoint `:BusinessActor`, `:SystemActor` and

`:Component`. Actors have at least one reflexive and transitive object property `:isRealizationOf` to represent actor mappings. The class is further specialized by the disjoint classes `:SingleActor` and `:MultipleActor` and by the non-disjoint classes `:DataProvider`, `:DataCollector` and `:ThirdParty`. In summary, the following information from the DFG is contained in this concept: any component is a realization of a system actor which is again a realization of a business actor (note that due to reflexivity each actor is a realization of itself). Further, an actor can either be a single actor or a multiple actor and its role is either data provider, data collector, third-party or a combination.

(ii) **BusinessCase.** The `:BusinessCase` class is a simple concept designed to represent business case relationships between actors. This class covers the business layer of a DFG and has the object properties `:hasBusinessCase` and `:hasBusinessActor`, respectively.

(iii) **Data.** The `:Data` class is the most powerful concept as most of the information in a DFG is somehow mapped to this class. The class is specialized by two disjoint classes `:Event` and `:Series` and by a class `:PersistentData`. The latter is introduced in order to denote if data is persisted by an actor. This can be expressed by the `:isPersistedBy` relationship. Further, the class is specialized by `:PersonalData` which has been dedicated a section below due to its importance. The `:Data` class comprises the following reflexive and transitive object properties: `:isCompositionOf`, `:isAggregationOf` and the specializations `:isCompositionOfOne`, `:isCompositionOfMany`, `:isAggregationOfOne` and `:isAggregationOfMany`. Finally, the class is related to `:Actor` with the `:isSentBy` and the `:isReceivedBy` object property. In summary, this allows to represent arbitrary data flows from one actor to another including complex compositions and aggregations.

(iv) **PersonalData.** Even though `:PersonalData` is a subclass of `:Data`, it is one of the crucial concepts taken into account for privacy evaluation and therefore emphasized at this point. This class is specialized by disjoint `:DirectlyPersonalData` and `:IndirectlyPersonalData` and by disjoint `:SensitiveData` and `:InsensitiveData`.

Each class has a set of attributes (representing Σ_D and Σ_A of the DFG). This set is generally not predefined in order to have a generic solution that can be adapted to any application scenario. However, there is a limited number of attributes that is mapped to object properties for information objects and therefore has predetermined semantics:

(i) **PersonalData.** This property expects a value that is either `:DirectlyPersonal` or `:IndirectlyPersonal` and describes the personality of the information object.

(ii) **SensitiveData.** This property expects a value that is either `:Sensitive` or `:Insensitive` and describes the sensitivity of the information object.

(iii) **Multiplicity.** For actors, this is the multiplicity of expected instances of this actor. Valid values are either 0, 1 or n. While 0 is only considered for the sake of completeness it does not have a practical application. 1 and n, however, indicate if an instance of an actor occurs only once or if there are multiple realizations of that actor behaving similarly.

(iv) **Type.** The type of an information object is either `:Actor`, `:Data` or a specialization of one. Currently `:Event` and `:Series` are defined as sub classes of `:Data`. An event occurs only once (following a probability distribution) whereas a series is a deterministic occurrence described by a frequency.

(v) **Owner.** This property refers to the owner of an information object and is usually a business actor. This attribute is mapped to a `:isOwnedBy` object property that relates `:BusinessActor` to `:Data`.

(vi) **Persistence.** This property describes if an information object is persisted and is optionally denoted by a value indicating the expected retention. This attribute is mapped to a `:isPersistedBy` object property that relates `:Actor` to `:Data`.

Note that transitivity and reflexivity axioms are widely used for object properties in order to represent the complexity and characteristics of data flows. Further, properties follow a hierarchy to allow the specification of high level threat patterns matching a wider range of specific relations.

The following provides an overview of property super and sub class relationships. There exists a corresponding inverse property for each of the following which is not shown in this list. Note that `:a` ▷ `:b` is an abbreviating notation for `:a rdfs:subPropertyOf :b`.

(i) (`:isAggregationOfOne, :isAggregationOfMany`) ▷ `:isAggregationOf` ▷ (`:isCompositionOfOne, :isCompositionOfMany`) ▷ `:isCompositionOf`

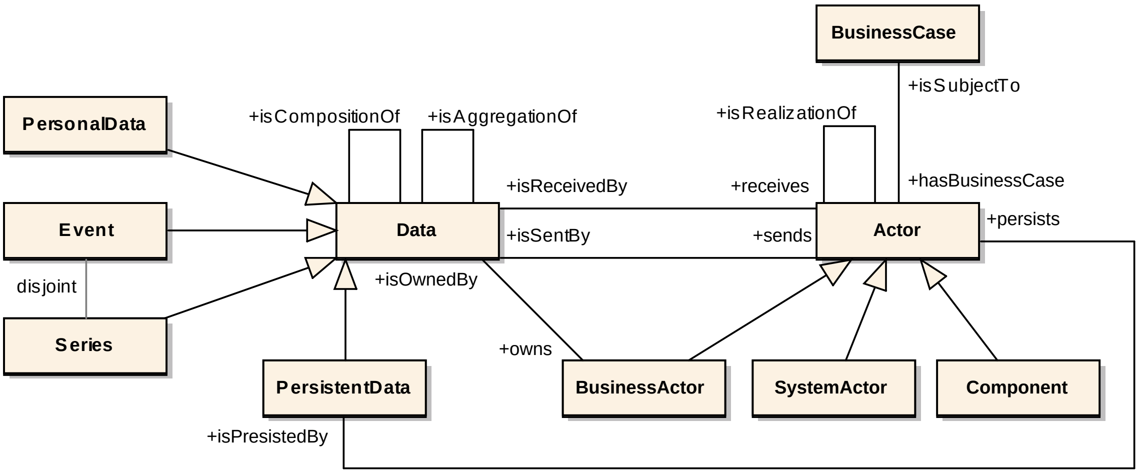

Figure 4.1: Simplified view of the ontology TBox showing the main concept and how the classes actor and data are related to each other.

(ii) (`:isSentBy`, (`:isPersistedBy` ▷ `:isReceivedBy`)) ▷ `:isTransmittedBy`

Figure 4.1 shows a simplified view of the most relevant concepts. Altogether, the ontology comprises more than 60 classes, object properties and data properties in the TBox. When a DFG is mapped to the ontology, individuals (ABox) are created and assigned the appropriate type.

4.3 Pre-Classification

This section introduces the term pre-classification and describes the purpose of the pre-classifiers. Pre-classification refers to the fact that ontology reasoning is applied to get a general set of classes that provide additional information. Pre-classifiers are OWL classes consisting of equivalent class expressions. Hence, a reasoner assigns individuals to these classes and can therefore be seen as a classifier.

Pre-classifiers are utilized in two significantly differing ways: First, pre-classification is not essential since the same amount of information can be retrieved by defining queries in the threat patterns, however, it makes writing threat patterns (and therefore queries) easier and more natural, which subsequently helps to bridge the semantic gap. For instance, pre-classifiers for typical, privacy critical aggregations are defined. For example compositions of location information or compositions of directly personal data. The set of pre-classifiers can be modified and extended to meet the needs for arbitrary classification problems and is not limited to privacy issues.

Another application for pre-classifiers is to serve as an abstraction layer. Even though the framework is designed to work with meta-data and abstract components, it is also possible to model highly particular scenarios. Instead of describing a smart meter as a component in the customer premises, for some applications it might be useful to describe a concrete smart meter with a concrete meter number (e.g., 90089) at a particular location (e.g., University Park Campus). For evaluation this particularity needs to be brought to a more abstract level in order to have reusable queries. This is where pre-classifiers are plugged in. Equivalent class expressions for pre-classifiers might then transform this into triples such as `:Component :hasNumber true` and `:Component :hasLocation true`. Again, this is to make modeling expert knowledge more natural and easier and therefore bridging the semantic gap.

4.4 Information

This section is dedicated to a detailed description of the information that can be retrieved explicitly or implicitly from the ontology. Explicit information is anything that is modeled immediately in the DFG and therefore represented by explicit axioms in the ontology. Implicit information, by contrast, is gained through ontology reasoning. Since components are a realization of system actors and system actors are a realization of business actors, the term *actor* is used in place of any kind of node in a DFG. Actors are generally characterized by the SGAM zone and domain they are located in. This gives additional information about the role, e.g., if some device that realizes a system actor is in the customer premises. Further, actors may store data encrypted and they might have a certain processing delay. Processing delay refers to internal storage of data due to buffering or due to legal requirements, e.g., the submission of meter value from the previous day. Actors are characterized by their multiplicity m where $m \in \{0, 1, n\}$. A multiplicity of 1 refers to one single, clearly identifiable realization of an actor (depending on the use case), e.g., a smart meter head end system in a smart metering scenario. A multiplicity of n, by contrast, refers to an actor that is realized more than once and that is acting interchangeably among other instances, e.g., an electric vehicle charging station. The multiplicity value 0 has no practical meaning, however, it is included for the sake of completeness and reserved for future usage. Table 4.1 shows a summary of information that is explicitly contained in individuals of type Actor.

In addition to actors, information objects are another source of information. Information objects are characterized by either being a series or an event and by either being

Table 4.1: Explicit information from actor.

Name	**Description**
Data Access	immediate or restricted data access by actor
Multiplicity	multiplicity of actor
Domain	SGAM domain of actor
Zone	SGAM zone of actor
Encryption	actor processes encrypted data
Delay	processing delay by actor

Table 4.2: Explicit information from information objects.

Name	**Description**
Aggregation/Composition	data is aggregated or composed of other data
Event/Series	data is sent once or regularly
Directly/Indirectly Personal	data is directly or indirectly personal
Sensitive/Insensitive	data is sensitive or insensitive
Frequency	transmission has a certain frequency
Timestamp	data has a known timestamp
Persistent	data is persistent with an expected retention

directly personal data, indirectly personal data, sensitive data or insensitive data. Further, information objects can be an aggregation or a composition of other information objects what might have significant impact on the classification. A composition is a set of information objects combined to one, new information object, without losing any information from the containing objects. An aggregation, by contrast, creates a new information object that either generates new, additional data or reduces data from the containing information objects, e.g., by summarization or minimization. In both cases, for the receiving actor there are no pointers to the original data items. Information objects of type series can have an additional attribute denoting the frequency at which they are sent from the sender to the receiver. A timestamp denotes if their ending receiving time is identifiable for the receiving actor. The persistence attribute finally indicates if an information object is stored persistently at a certain actor with a certain retention (e.g., storage in a database) or if it is just processed temporarily. Table 4.2 shows a summary of the information that is explicitly contained in individuals of type data.

Table 4.3: Implicit information derived from actors and information objects.

Name	**Description**
Transitivity	transitive data flows
Multi-domain/-zone	data flows across multiple domains/zones
Multi-event/-series	aggregations/compositions of multiple events/series
Directly/Indirectly Personal	aggregation/composition of directly/indirectly personal data
Sensitive/Insensitive	aggregation/composition of sensitive/insensitive data
Business Actors	business actors that persist data
Data Access	additional data access by actors

Once the individuals are created in the ontology a reasoner is applied to draw conclusions and to retrieve implicitly contained information and relationships. One of the main objectives is to determine implicit data flows that can be revealed due to transitivity. Furthermore, it is possible to determine if an aggregation or composition comprises directly personal data or sensitive data. Especially for compositions, the resulting information object is not only of type directly personal/sensitive but it might also imply more severe privacy issues due to the combination of such data with other information. Based on the domain and zone attributes of actors, it is determinable if data flows are within a certain domain/zone or across multiple domains/zones. By analyzing the actor mappings business actor to system actor and system actor to component, it is possible to retrieve which business actor actually persists certain information objects and which actors might have data access in addition to the ones explicitly modeled. Table 4.3 shows a summary of information that can be derived from ontology reasoning.

The information presented in this section is used for matching the threat patterns. This solution, however, is generic and new, additional patterns can be defined in order to meet the requirements of specific applications. This is discussed later within the scope of the framework's ability to be generalized.

4.5 Attack Vector Assessment

Having a set of individuals placed in the ontology, a first generic assessment can be applied. For this purpose, the concept of attack vectors is adapted from security analysis, since this terminology is well understood and widely applied. An attack vector describes how an attack reaches its target [27]. In privacy analysis and for this work *attack* is defined as the potential misuse of information gained from data provider by a data collector. This yields three components describing an attack in terms of privacy: $\langle$data access, privacy asset, attack resources$\rangle$. Data access describes which actor has access to what data; privacy asset indicates if data is privacy critical or not, e.g., if it contains sensitive data; and attack resource defines if the attacker can actually access/misuse the information.

For each information object, the data provider and the data collector is determined (according to the terminology defined in [5]) and it is assessed who has access to the data. This yields a list of three-tuples in the form $\langle$information object (IO), data provider (DP), data collector (DC)$\rangle$. Then it is determined if an information object either contains sensitive or directly personal data (according to the terminology defined in [61]). This yields another three-tuple in the form $\langle$information object (IO), sensitive (S), directly personal (DP)$\rangle$. Finally it is determined if the attacker has actual data access, yielding one more three-tuple in the form $\langle$information object (IO), data collector (DC), access (A)$\rangle$. Data access depends on the relationship of actors, on data resolution, retention and encryption. Matching these tuples to each other results in the components of the attack vector, recalling $\langle$data access, privacy asset, attack resources$\rangle$ yields $\langle\langle \mathrm{IO}, \mathrm{DP}, \mathrm{DC}\rangle, \langle \mathrm{IO}, \mathrm{S}, \mathrm{DP}\rangle, \langle \mathrm{IO}, \mathrm{DC}, \mathrm{A}\rangle\rangle$. An exemplary attack vector for a DR use case where DR preferences are sent to the utility is $\langle\langle \mathrm{DRPreferences}, \mathrm{Customer}, \mathrm{Utility}\rangle, \langle \mathrm{DRPreferences}, \mathrm{false}, \mathrm{false}\rangle, \langle \mathrm{DRPreferences}, \mathrm{Utility}, \mathrm{true}\rangle\rangle$.

This already provides thorough qualitative analysis. It is possible to determine which actor can potentially threaten the privacy of another actor. It is even possible to conclude how and where this might happen.

5

Expert Knowledge

This chapter introduces the term *expert knowledge* and describes how to capture, collect and process expert knowledge for classification. The two main sources for expert knowledge, requirements engineering and available knowledge, are discussed in detail. Further, this chapter describes specific privacy requirements for the classification of use cases with respect to privacy. This chapter concludes with a description of the quantitative analysis, i.e., the risk estimation, and how the results of the qualitative and the quantitative assessment are combined in one single threat matrix.

In systems engineering, experts play a major role in design and development of successful and sustainable systems. Experts are called to contribute their profound knowledge about a specific topic or specific circumstances. For the privacy assessment framework that is described in this report, expert knowledge is primarily represented as threat patterns. Threat patterns describe a set of vulnerabilities and a set of countermeasures that are used for the quantitative risk assessment. While for the modeling of DFGs experts may also be an influential source of knowledge, in this chapter the term explicitly refers to the definition of threats.

5.1 Threat Patterns

Threat patterns are used to quantitatively assess the impact on privacy based on the qualitative attack vectors. A threat pattern comprises a uniquely identifiable name, a set of conditions that can either be exploited for an attack or that mitigate an attack. Each threat t therefore consists of a one or more vulnerabilities t_{vul} and of no, one or more countermeasures t_{cnt}. Each vulnerability and each countermeasure is assigned an expected loss value that is positive or negative, respectively. A negative loss value which is assigned to countermeasures represents a mitigation of threats. In the pattern matching process these threat patterns are compared to the explicit and implicit information in the data flow graph and based on the resulting matches, the probability of occurrence and the expected loss are used for calculating a risk value. Threat patterns consist of the following components:

(i) **Name.** A unique name describing the pattern. This is name is shown in the threat matrix and compared with the risk value.

(ii) **Vulnerability.** A set of one or more vulnerabilities. Each vulnerability describes a concrete scenario, e.g., unlimited data retention and access by external business partners. Each vulnerability is tagged with a value $\Phi(t_{vul})$ with $\Phi(t_{vul}) \in \mathbb{R}^{+}$ stating the expected loss for this vulnerability being exploited by an attacker.

(iii) **Countermeasure.** A set of no, one or more countermeasures. Each countermeasure describes a concrete scenario that mitigates one or more vulnerabilities, e.g., homomorphic encryption to prevent data access by intermediary components. Each countermeasure is tagged with a value $\Phi(t_{cnt})$ with $\Phi(t_{cnt}) \in \mathbb{R}^{-}$ stating to what extent the expected loss is reduced by this countermeasure.

Vulnerabilities and countermeasures are defined as patterns in fractions of SPARQL ASK queries (in literature also referred to as *graph patterns*). The pattern matching implementation uses these fractions to build a full query that returns either true/false for calculating the probability of occurrence or the expected loss/zero, respectively. Implementation issues concerning SPARQL ASK queries are discussed in Chapter 6. Formally, this is later expressed as the function Ψ for calculating the probability of occurrence and the function Φ for calculating the expected loss. Both is explained in greater detail in the following sections.

An example for a pattern describing a vulnerability is shown in Listing 5.1. The pattern shown here, detects if some device is in the customer premises and if data is sent to another business actor. Lines in square brackets are not part of the pattern, but added for processing in the implementation. A natural language interpretation of this pattern is *some sender in the customer premises is a realization of some business actor, that sends an information object to some receiver, that is a realization of a different business actor.* The prefix `x` is used for brevity and refers to the namespace `http://owl.fh-salzburg.ac.at/dfg` and the predicate `a` is a synonym for `rdf:type`. Patterns can be as simple as a single triple or of arbitrary granularity and complexity, including advanced expression for filtering, sub selections, union and optional expressions [67, 70].

```
[ASK WHERE {]
    ?sender x:isRealizationOf ?ba1 . ?ba1 a x:BusinessActor .
    ?sender x:Zone "Customer Premises"^^xsd:string .
    ?sender x:sends $io .
    ?receiver x:receives $io .
    ?receiver x:isRealizationOf ?ba2 . ?ba2 a x:BusinessActor .
    FILTER (ba1 != ba2)
[}]
```

Listing 5.1: Example of a vulnerability condition in a threat pattern. This pattern matches any device in the customer premises that sends data to another business actor.

Generally, this matches any triples in the ontology that have the same pattern; for a more precise analysis, however, this is not enough. The above pattern evaluates to true for any kind of information objects, not taking into account the underlying semantics and the privacy impact of the information object itself or whether the information object is even prone to an attack. Such additional conditions are considered using variable bindings. Note the different notation for variables, e.g., `?sender` and `$io`. While both is a valid and interchangeable notation for variables in SPARQL, the latter notation has an additional meaning for the assessment: variables marked with $ can be bound to an individual. The current prototypical implementation supports two types of variables that can be bound: (i) `$io` can be bound to an information object (individual of type data); and (ii) `$ba` can be bound to a business actor (individual of type business actor). The binding of variables never increases the number of false positives, thus the number of patterns that apply without having an actual impact; by contrast this number can be reduced effectively for the information object or business actor that is subject to investigation.

The range of individuals is based on the qualitative assessment and represented by the attack vectors. An exemplary attack vector for a DR use case was given by $\langle\langle \text{DRPreferences}, \text{Customer}, \text{Utility}\rangle$, $\langle \text{DRPreferences}, \text{false}, \text{false}\rangle$, $\langle \text{DRPreferences}, \text{Utility}, \text{true}\rangle\rangle$. Values for the bindable variables for `$io` are `DRPreferences` and for `$ba` they are `Customer` and `Utility`.

The application of such a binding combines the qualitative assessment and the threat patterns to get a quantitative result. In the following section, privacy requirements are identified, that are taken into account for the definition of patterns.

5.2 Privacy Requirements

In order to define threat patterns which reflect actual and representative privacy impacts, two sources for expert knowledge are taken into account. First, for specific requirements existing knowledge in the domain is used, which has been gathered by investigation of literature and state of the art approaches. Secondly, a set of general privacy requirements is formed for the evaluation of use cases and based on the threat tree approach.

Specific requirements are based on investigations from state of the art approaches, literature reviews and insights gained in both model regions in the US and in the EU. In [50], Neureiter et al. present six high-level threat patterns: (i) information abuse; (ii) data abuse; (iii) unintended data aggregation; (iv) security violation; (v) criminal intention; and (vi) unclarity. Based on that, for this work, the following concrete threat patterns are developed, aiming to represent a typical set of privacy threats:

T_1 **Customer Presence at Home.** This privacy concern is discussed by Cavoukian et al. in [13] and by Knapp and Samani in [35]. To potentially determine a person's presence at home, some device in the customer premises is needed. This device collects data at a certain frequency, high enough to have a resolution that allows to draw conclusions on the energy usage of specific devices. Furthermore, data collected from that device needs to be sent to another actor (i.e., an utility). At the utility an individual or a system needs to have access to the data in an appropriate resolution. Since we always assume that data is accessed legally, we do not focus on unallowed data access. Additionally, the total delay of the data transmission is of relevance. If data is collected and transmitted in almost real time the presence at home can be determined immediately. If data is available with a certain delay only, the analysis of past events and predictions might be possible. If this information is published, an attacker might exploit this vulnerability in order to break into the house.

T_2 **Tracking Customer Position.** This threat is especially interesting for electric vehicle charging. Assuming the customer has some identification towards the charging station, at least the location, a timestamp and the amount of energy consumed will be recorded for billing. Depending on the design of the infrastructure only little information will be sent to the operator or a very detailed profile of the customer is maintained. Here, the multiplicity of the actors is crucial and the fact that different actors have access to the same data. Potential attacks for this threat are described by Langer et al. in [40], e.g., using information for targeted ads, to determine the customers principal residence, for tracking movements to certain places or to infer the income based on the amount of recharges.

T_3 **Identification of Customer Habits.** Customer habits is a general term and refers to anything that allows to potentially determine or predict certain regular actions of a customer. In fact, customer presence at home can be seen as a sub case of customer habits. However, for this threat more information is taken into account and the focus is not limited to the customer premises. Relevant information objects include the number of actors involved in the use case and their multiplicity, transmission delays, frequency and whether data is an event or a series. Furthermore data access and data compositions are taken into account as indicators. This threat is mitigated by aggregations of data (i.e., information about multiple customers is minimized) or by having multiple senders and/or receivers of data that are not connected or affiliated. A severe breach of privacy occurs, if such data becomes publicly available for some reason without the consent of the owner. This vulnerability can be exploited by an attacker to deliver targeted ads, tailored to customer preferences.

T_4 **Public Availability of Data.** This threat is severe and implies that some data is disclosed by intention or inadvertently. Research has shown that even anonymous data (i.e., indirectly personal data) can reveal significant details about an individual [37], hence this threat is not limited to directly personal data. Once data is publicly available, a number of threats (e.g., the ones mentioned above) become evident even if not present otherwise. If data is available for external parties, all new methods to combine and aggregate data are feasible. This includes both, information and stakeholders, not being otherwise related to that business case.

General privacy requirements are determined following the threat tree approach presented in Chapter 2, i.e., privacy is violated if “some has personal information” and a “misuse of personal information” occurs. While both can be determined in the qualitative analysis, threat patterns define concrete vulnerabilities and countermeasures and therefore define where and how privacy critical data acquisition, processing or combination occurs. In [50], Neureiter et al. further present a set of seven high-level privacy requirements. Six of these requirements are rephrased in the following in order to represent a generic set of vulnerabilities (t_{vul}) and countermeasures (t_{cnt}). One of the requirements is risk awareness, meaning that there must be a clear communication of risk for specific applications in the smart grid. This requirement is fulfilled by the framework by design and therefore not present in the following list.

$t_{vul,1}$ **Information Awareness.** If the information that can be extracted from various data sets is either not clearly communicated to the user, if information flows are not obvious, e.g., if transitive relationships and a high number of actors are present, or if data access is not restricted, this vulnerability applies.

$t_{vul,2}$ **Data Control.** This vulnerability is applicable if the user cannot actively control which information is released to the collector, e.g., if data is sent to collectors with high multiplicity.

$t_{vul,3}$ **Data-Information Link.** This vulnerability applies if data is linked in order to retrieve additional and implicit information. Linkage can happen at collectors or intermediary components and is usually represented by a composition.

$t_{vul,4}$ **Data Aggregation.** Data composition or aggregation is crucial, since the combination of data items might imply a severe risk for privacy violations. In this framework this is detailed by distinguishing between compositions and aggregations, where the latter always include some reduction or minimization of data.

$t_{vul,5}$ **Situation Dependency.** This vulnerability is applicable if data is not released situation specific, e.g., if data is captured on a regular basis (series) where single, non-deterministic samples (events) would be sufficient or if the metering frequency is higher than necessary.

$t_{vul,6}$ **Data Overview.** If there is no transparency which data is released to whom and for what reason, this vulnerability applies; e.g., if two parties share a common business case which implies an exchange of data which has not been clearly communicated to the customer.

$t_{vul,7}$ **Data Minimization.** This vulnerability applies if data is collected, stored or processed that has no immediate use for the specific application.

$t_{vul,8}$ **Release Expiry.** This vulnerability applies if expiry (retention) of data is either not considered at all or handled improperly.

Each vulnerability described above is mitigated by a an appropriate countermeasure:

$t_{cnt,1}$ **Information Awareness.** This countermeasure implies transparent information flows, restricted data access and only a limited number of actors involved.

$t_{cnt,2}$ **Data Control.** This countermeasure is applicable if data is only sent to a few, clearly identifiable collectors with little multiplicity.

$t_{cnt,3}$ **Data-Information Link.** If data from different sources related to a single person is not composed for storage or further processing or if instead of a composition some aggregation is performed that minimizes information.

$t_{cnt,4}$ **Data Aggregation.** A countermeasure for aggregations and compositions is the aggregation of data from a provider with high multiplicity, i.e., a single data item can not be tracked.

$t_{cnt,5}$ **Situation Dependency.** This countermeasure applies if data is not collected on a regular basis, but in a granularity and resolution that is immediately bound to the application.

$t_{cnt,6}$ **Data Overview.** An effective countermeasure is a clear separation between parties, i.e., data that is collected by one party and is not sent to another one without notifying the original provider.

$t_{cnt,7}$ **Data Minimization.** This countermeasure applies if only data is collected that is necessary for the immediate purpose.

$t_{cnt,8}$ **Release Expiry.** An effective countermeasure includes that data retention is lowest or that there is no retention at all.

Available knowledge is any knowledge already captured and formalized in a way suitable for further automatic processing. Such knowledge is especially valuable when enhanced with semantics. In [75] and [76] a number of sources for such knowledge are identified, analyzed and integrated. The authors focus on the analysis of events for demand response and provide a semantically integrated model for data from various sources, such as consumption, infrastructure, individual behavior, schedules and natural conditions. Further, in [53], a work is presented building on the IEC Common Information Model and thus also provides a terminology drawing on already formalized knowledge.

5.3 Estimating Risk

The risk assessment builds on the results of the qualitative analysis. The objective of the risk estimation is to perform a quantitative analysis, i.e., to find a number that represents to what extend a threat is potentially harmful. In this way, the complexity of the attack vector can be reduced to one, single number. The advantage of this approach is to have a representative value that gives system engineers and customers an immediate feedback on the use case and further makes the comparison of use cases easier and more efficient. It is therefore possible to assess a set of use cases in comparison to one another in order to find the one with the least privacy critical impact. Recalling that T^* is the subset of threats T that applies for the particular use case, risk is a number $R \in \mathbb{R}^+$ calculated as a function of the probability of occurrence $PO(T^*)$ and the expected loss $EL(T^*)$, hence $R = f(PO, EL, T^*)$ with $f : T^* \rightarrow \mathbb{R}^+$.
The function f is defined taking into account the following considerations:

(i) **Applicability.** The risk assessment is intended to represent the potential harm of a threat and to be applied as the basis for the comparison of use cases. The result is defined to be realistic if $f(PO, EL, T_i^*) > f(PO, EL, T_j^*)$ is true for each pair of a subset of threats (T_i^*, T_j^*) where T_i^* is considered to be worse than T_j^*. What in fact is said to be *worse*, however, depends on the scenario at hand.

(ii) **Coherence.** The risk assessment is coherent, if an additional threat t in T that is not a member of T^* does not affect R. Hence, a threat that is not part of the class that the use case is assigned to, does not affect the risk for that class.

(iii) **Stability.** The risk assessment is stable, if the value of R is not affected by changes in the use case that do not affect a threat in T. This means, adding, removing or changing actors or information objects in the use case definition that have no privacy impact (i.e., that are not matched by a threat pattern) also do not affect the risk.

For the functions $PO(T^*)$ and $EL(T^*)$ additionally applies: $PO : T^* \rightarrow [0, 1]$, i.e., it represent a probability; and $EL : T^* \rightarrow \mathbb{R}$, which is a number indicating the loss. This might be mapped to a monetary unit where positive numbers can be seen as a loss and negative numbers as a gain.
Recalling that each t in T^* consists of a set of vulnerabilities t_{vul} and a set of countermeasure t_{cnt}, for classification each vulnerability and each countermeasure is matched

to the ontology and the corresponding condition Ψ is either true (1) or false (0), i.e., the vulnerability or the countermeasure applies for this use case or not.

In order to have a value for PO in the range $[0, 1]$, a (preferably linear) function needs to be found that satisfies at least the conditions defined in Equations 5.1, 5.2, 5.3. The linear model is preferred over higher-degree models due to it's simplicity and might later be replaced by more complex approaches.

$$C_1 : PO\left(\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n}) = |t_{vul}| \text{ and } \sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n}) = |t_{cnt}|\right) = \frac{1}{2} \tag{5.1}$$

$$C_2 : PO\left(\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n}) = |t_{vul}| \text{ and } \sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n}) = 0\right) = 1 \tag{5.2}$$

$$C_3 : PO\left(\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n}) = 0 \text{ and } \sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n}) = |t_{cnt}|\right) = 0 \tag{5.3}$$

Given these points and a three dimensional space for the number of vulnerabilities that apply, the number of countermeasures that apply and the probability of occurrence, it can be shown that a plane that passes through these points (i) satisfies the above conditions; (ii) gives reasonable results for other pairs of vulnerabilities and countermeasures that apply; and (iii) the approach works for an arbitrary total number of vulnerabilities and countermeasures.

To determine a plane that passes through the points given by $C_1 \ldots C_3$, the following vectors that represent the conditions are used $\overrightarrow{C_1} = (|t_{vul}|, |t_{vul}|, \frac{1}{2})$, $\overrightarrow{C_2} = (|t_{vul}|, 0, 1)$ and $\overrightarrow{C_3} = (0, |t_{cnt}|, 0)$. Next, the vectors $\overrightarrow{C_1C_2}$ and $\overrightarrow{C_1C_3}$ are determined in order to calculate the normal vector $\overrightarrow{C_1C_2} \times \overrightarrow{C_1C_3}$ as in Equation 5.4.

$$\begin{aligned} (a, b, c) = \overrightarrow{C_1C_2} \times \overrightarrow{C_1C_3} &= (-|t_{vul}|, |t_{cnt}|, -1) \times \left(-|t_{vul}|, 0, \frac{1}{2}\right) \\ &= \left(-\frac{1}{2}|t_{cnt}|, \frac{1}{2}|t_{vul}|, |t_{vul}||t_{cnt}|\right) \end{aligned} \tag{5.4}$$

Given the cartesian equation of a plane, $ax + by + cz + d = 0$, where x is a shorthand notation for $\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n})$, y is a shorthand notation for $\sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n})$ and z is the function value for PO, this yields Equation 5.5:

$$-\frac{1}{2}|t_{cnt}|x + \frac{1}{2}|t_{vul}|y + |t_{vul}||t_{cnt}|z + d = 0 \tag{5.5}$$

After determining d by inserting in any of the above conditions (C_3, for the sake of simplicity) and after again replacing x by $\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n})$, y by $\sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n})$ and z by PO, this gives the function for $PO(T^*)$ as defined in Equation 5.6:

$$PO(T^*) := \frac{1}{2}\left(\frac{1}{|t_{vul}|}\sum_{n=1}^{|t_{vul}|}\Psi(t_{vul,n}) - \frac{1}{|t_{cnt}|}\sum_{n=1}^{|t_{cnt}|}\Psi(t_{cnt,n}) + 1\right) \tag{5.6}$$

In case that there are no vulnerabilities, i.e., $|t_{vul}| = 0$ or that there are no countermeasures, i.e., $|t_{cnt}| = 0$, the above equation is undefined. These cases are defined by Equations 5.7 and 5.8. In case that both, the total number of vulnerabilities and the total number of countermeasures defined are zero no value at all is returned; this case does not allow any statement about the probability of occurrence for a threat.

$$PO_{|t_{vul}|=0}(T^*) := \frac{1}{2}\left(-\frac{1}{|t_{cnt}|}\sum_{n=1}^{|t_{cnt}|}\Psi(t_{cnt,n}) + 1\right) \tag{5.7}$$

$$PO_{|t_{cnt}|=0}(T^*) := \frac{1}{2}\left(\frac{1}{|t_{vul}|}\sum_{n=1}^{|t_{vul}|}\Psi(t_{vul,n}) + 1\right) \tag{5.8}$$

This results in high probability of occurrence values for a high number of vulnerabilities and a low number of countermeasures that apply, in low probability of occurrence values for a low number of vulnerabilities and a high number of countermeasure and for a value of $\frac{1}{2}$ if vulnerabilities and countermeasures apply equally. The case where no vulnerabilities and no countermeasures apply at all is undefined, i.e., no statement is possible. Figure 5.1 shows PO for values for $\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n}) = 1 \ldots 30$ and $\sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n}) = 1 \ldots 30$.

$EL(T^*)$ is defined as in Equation 5.9:

$$EL(T^*) := \frac{1}{|t_{vul}|}\sum_{n=1}^{|t_{vul}|}\Phi(t_{vul,n}) - \frac{1}{|t_{cnt}|}\sum_{n=1}^{|t_{cnt}|}\Phi(t_{cnt,n}) \tag{5.9}$$

By contrast to the definition of the probability of occurrence where Ψ is true only if the pattern applies, this definition uses values predefined in the threat patterns, hence the function Φ returns the value defined in the pattern if it applies or zero otherwise.

This equation is again undefined in case that there are no vulnerabilities or that there are no countermeasures. These cases are defined by Equations 5.10 and 5.11. Also, in

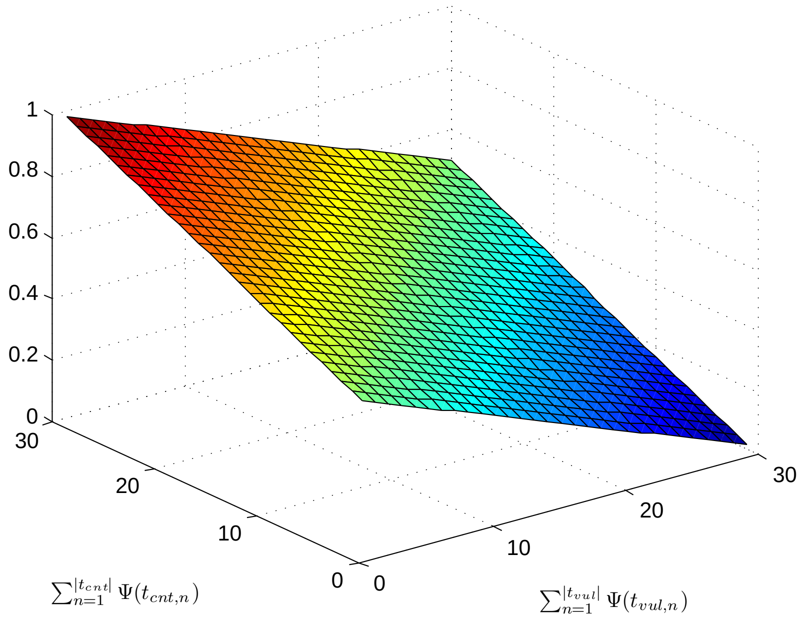

Figure 5.1: PO for values for $\sum_{n=1}^{|t_{vul}|} \Psi(t_{vul,n}) = 1 \dots 30$ and $\sum_{n=1}^{|t_{cnt}|} \Psi(t_{cnt,n}) = 1 \dots 30$.

case that both, the total number of vulnerabilities and the total number of countermeasures defined are zero no value at all is returned.

$$EL_{|t_{vul}|=0}(T^*) := -\frac{1}{|t_{cnt}|} \sum_{n=1}^{|t_{cnt}|} \Phi(t_{cnt,n}) \tag{5.10}$$

$$EL_{|t_{cnt}|=0}(T^*) := \frac{1}{|t_{vul}|} \sum_{n=1}^{|t_{vul}|} \Phi(t_{vul,n}) \tag{5.11}$$

In summary, risk is defined in Equation 5.12:

$$R(T^*) := PO(T^*)\, EL(T^*) \tag{5.12}$$

This definition meets the requirements for applicability, coherence and stability as stated above.

5.4 Threat Matrix

The threat matrix is the method envisioned to display threats. This concept is commonly used in information security analysis and often is referred to the more general concept of a *control matrix* [72]. A control matrix compares a set of threats and the particular assets endangered. For this work, the threat matrix is defined as follows:

Personal attitudes	0	very low
Individual person presence at home	0.65	high
Some person presence at home	1	very high

Table 5.1: Threat matrix showing three threats and the related risk including a numeric value and a textual interpretation.

given the set of all threats T, the threat matrix compares a subset T^* with the value indicating the risk R for that threat. Optionally, the probability of occurrence and a detailed listing of vulnerabilities and countermeasures including their particular expected loss can be displayed.

In order to get a risk value in that range the linear mapping shown in Equation 5.13 is applied. The functions min and max return smallest and largest value for each R in the threat matrix. Note that this mapping can be improved by using minimum and maximum values from a larger set of risk evaluations that have been performed for this use case or within similar settings.

$$R' = \frac{R - \min\{R\}}{\max\{R\} - \min\{R\}} \tag{5.13}$$

A textual interpretation of this value is also given to improve readability of the results. For this purpose the following mapping is used $R' \leq 0.2 \rightarrow$ very low, $0.2 < R' \leq 0.4 \rightarrow$ low, $0.4 < R' \leq 0.6 \rightarrow$ medium, $0.6 < R' \leq 0.8 \rightarrow$ high and $R' \geq 0.8 \rightarrow$ very high.

For instance, given three threats with the risk values $R(T_1) = 3.7$, $R(T_2) = 9.1$ and $R(T_3) = 12.0$, the value for $min(R) = 3.7$ and for $max(R) = 12.0$ and hence $R'(T_1) = 0$, $R'(T_2) \approx 0.65$ and $R'(T_3) = 1$.

Table 5.1 is a sample threat matrix showing three different threats and the related risk R' as a value ranging from 0 to 1.

6

Implementation

This chapter describes the prototypical implementation of the use case assessment tool as well as the prototypical implementation of the policy decision point. First, the representation of DFGs in XML and export and import issues are shown. Further, implementation aspects of the classification process are addressed and finally the integration of the framework as a policy decision point in a working smart grid middleware is described.

6.1 Data Flow Graph Export

DFGs are the basis for the evaluation of use cases. DFGs are modeled in UML and for further processing converted into a format more applicable for the classification process. For immediate machine readability, DFGs are exported to Extensible Markup Language (XML) for further processing. XML is a broadly supported standard for data storage and exchange [66]. XML itself is generic and can be molded to a specific format by XML schemes. For an XML based representation of graphs a number of formats are available. Most notable Graph Exchange Language (GXL)[1], Scalable Vector Graphics (SVG)[2] and Graph Markup Language (GraphML)[3]. For this project, the latter has been chosen due to its simplicity, while still being capable to annotate nodes and edges with attributes and to draw directed and nested graphs as well as its broad support from the community. A detailed description about the format itself, as well as background information on its establishment can be found in [11]. A primer is also available online in [12].

[1] `http://www.gupro.de/GXL/`

[2] `http://www.w3.org/Graphics/SVG/`

[3] `http://graphml.graphdrawing.org/`

A GraphML file is characterized by the root tag `graphml` and the namespace `http://graphml.graphdrawing.org/xmlns`. Following the root tag, a set of graphs (tag `graph`) is defined and given an `id` attribute. Each graph contains `node` tags, including an `id` attribute, and `edge` tags, including `source` and `target` attributes, as well as a `directed="true"` attribute. A node may contain a sub graph following the same pattern. Listing 6.1 shows the principal structure of a DFG in XML, later referred to as DFGX. Note that some parts are omitted for conciseness and only the component layer is shown in this example.

```
<graph id=”Container” edgedefault=”undirected”>
    <node id=”Container:ComponentLayer”>
        <graph id=”ComponentLayer” edgedefault=”directed”>
            <node id=”...:SmartMeter”/>
            <node id=”...:AMIHeadEnd”/>
            <edge source=”...:SmartMeter”
                         target=”...:AMIHeadEnd”
                         directed=”true”>...</edge>
        </graph>
    </node>
</graph>
```

Listing 6.1: Principal structure of DFGX. Only the component layer is shown and other parts are omitted for conciseness.

In addition to the layers defined by the SGAM, two additional layers are introduced: (i) *Business Actor to System Actor Mapping*; and (ii) *System Actor to Component Mapping*. Both layers comprise the mapping from one type of actor to another, again represented as a directed graph since each business actor is mapped to one or more system actors and each system actor is mapped to one or more components. In the SGAM this mapping is not a layer of its own, however, for coherence in the XML schema and the implementation it is advantageous to have an explicit graph. The information in these layers is used in order to map back from components (that either send or collect data) to the business actor.

For ids and keys throughout the graph a single notation is used. This notation, however, is a recommendation only and not enforced for validity. The values are a combination of the layer, e.g. Business Layer, their role, e.g., Actor and the individual name separated by a colon, i.e., *BusinessLayer:BusinessActor:Customer*. Information objects follow the same pattern, i.e., *General:InformationObject:MeterNumber*.

In accordance to the GraphML specification, the DFGX schema has been extended with custom sets to fully meet the requirements for DFGs. Especially the concept of attributing elements in the graph has been abstracted to information objects in

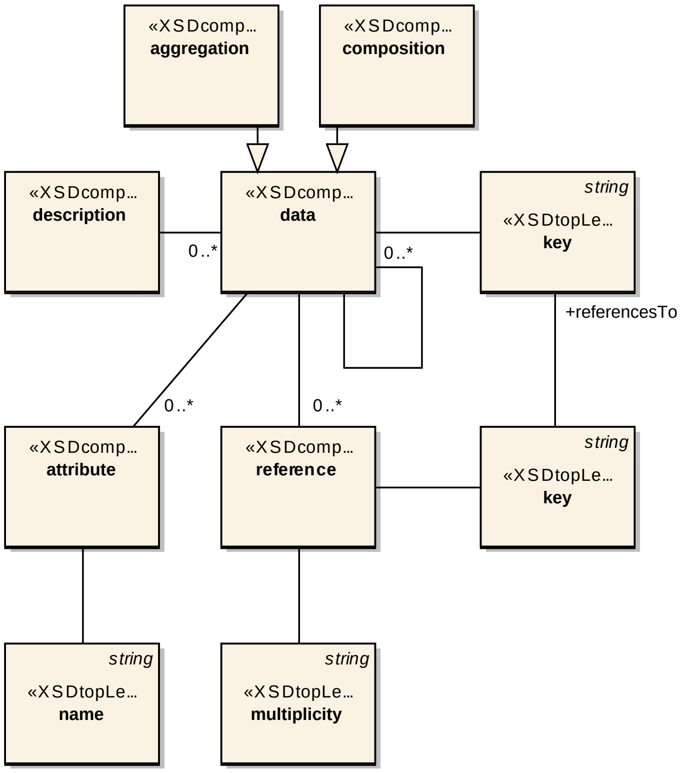

Figure 6.1: Building blocks for the information object meta-model. This model shows all valid relationships between objects.

order to handle complex relationships, aggregations and references. All extensions are located in the `http://en-trust.at/odpe` namespace. The extension meta-model is designed to meet the following requirements: (i) data items with attributes for nodes and edges; (ii) aggregations and compositions of data items and other aggregations and compositions; and (iii) references to data items, aggregations and compositions. The model itself is generic and does not specify the attributes and values itself in order to keep the framework as a whole extensible. Figure 6.1 shows the building blocks of the information object meta-model. The model outlines valid relationships and references. Information objects must be declared in the header `declaration` tag. Nodes and edges can reference to previously declared objects using an `information` tag and an appropriate `key` attribute.

In addition to the tags described above, plain text descriptions (UC name, UC description, author, ...) can be added to the graph. Using the GraphML `desc` tag, this allows accessory descriptions of nodes and edges. These tags do not carry computational semantics, but are intended to support human readability and understanding.

In summary, DFGX requires the following three components to be present in XML:

(i) **Interoperability Layers.** According to [15], the five interoperability layers of the Smart Grid Reference Model are part of a DFG. Each layer is a graph on its own describing the layer's actors (nodes) and data flows (edges). Besides the five layers defined in the standard, for exporting the graph two additional layers, the *actor mapping layers*, are introduced. These layers are necessary, in order to keep the relationship between business actors, logical actors and physical components as described in [20].

(ii) **Actors and Components.** Both, actors and components are nodes in the graph. Nodes are uniquely identified with an id and edges are identified by their source and target relation and direction. The position within the SGAM layer, which is given by zone and domain, is encoded as an information object.

(iii) **Information Object.** An information object is anything describing a node or an edge, an actor/component or a relationship, respectively. Information objects are declared in the header and then nodes and edges reference to a subset of them. Information objects follow the information object meta-model as described above.

One intention of this ontology driven approach is to have a tool that assists in the early design phase of the system engineering process. It is therefore valid to omit information in the graph that is not available at the point of creation or for a very first impact analysis. Subsequently, both, the classification process as a whole and the XML schemes are designed to be equally valid for subsets (e.g., one layer only) and for complete models.
A tool for exporting Data Flow Graphs to DFGX that have been modeled in Enterprise Architect is developed and contributed by Norbert Egger.

6.2 Data Flow Graph Import

For linking the components for classification into a single, self contained software package, Java 1.7[4] has proven to be the most suitable framework. A number of open source libraries for reading and writing XML, as well as to integrate ontologies are freely available for Java. XML interoperability is realized with the XOM XML library[5]. This library provides interfaces for creating, reading and modifying XML files.

[4] http://www.java.com/en/download/

[5] http://www.xom.nu/

For parsing XML and adding objects into the ontology, two principal implementation strategies are investigated. First, a non-persistent approach that does not attempt to build an internal data structure, but immediately creates the instances for the ontology from the nodes which have been read. And second, an implementation that does build an internal persistent data structure.
One of the characteristics is the fact, that DFGX file parsing is not order invariant. Due to the pre-declaration of information objects, these nodes need to be parsed first prior to any other element referencing to them. For that reason the implementation that uses an internal persistent object model for the DFG has been chosen as this allows a linear forward run through the file.
The parser reads a well formed XML document, does a principal validation and builds an internal data structure. The data structure consists of a graph for each layer (including the actor mapping layers) and a class hierarchy corresponding to the information object model.

6.3 Classification and Threat Pattern Matching

Once a DFG is imported and valid, it is mapped to the ontology as described in Chapter 4. The TBox – i.e., the classes and relationships – are loaded from an OWL file. This OWL file has previously been modeled with Protege[6] and OWLGrEd[7]. The latter is a graphical tool designed to visualize and graphically edit ontologies. It further provides capabilities for domain-specific extensions and a tight integration with Protege [16, 41]. In order to create the ontology ABox – i.e., the individuals – the internal object model representing the DFG is mapped in the ontology. The mapping itself does not affect the amount of information nor are any semantics changed. However, some attributes in the DFG have predetermined semantics that are taken into account at this point of the classification process. Finally, to the resulting ontology, i.e., TBox and ABox, a reasoner is applied to make sure the outcome is consistent and to compute the complete class hierarchy including all sub class relations. At this point also pre-classification is performed, since the reasoner computes the members for the classes defined by expressions in Manchester Syntax. The reasoner applied in this implementation is HermiT 1.3.8[8], which has been chosen due to its efficiency in reasoning and its powerful and easy integration in Java.

[6] http://protege.stanford.edu/

[7] http://owlgred.lumii.lv/

[8] http://hermit-reasoner.com/

For the qualitative and quantitative analysis an abstract interface for querying the ontology is designed. The interface is following a factory design pattern where a query host of a specific type is instantiated which creates and evaluates queries on demand. Further, two concrete implementations for that host are evaluated:

(i) **Description Logic Queries.** Description logic queries allow to form requests in Manchester Syntax and to retrieve a list of classes and individuals that match the query. This is a powerful query interface for simple requests, e.g., to determine all individuals for a given class. In order to evaluate threat patterns, however, this has shown to be limited. Manchester Syntax does not allow variables and more elaborate constructs such as optional conditions.

(ii) **SPARQL Queries.** SPARQL queries allow to define a pattern including variables and for each variable a set of classes, individuals or properties that match is returned. SPARQL queries are very powerful since they allow the formation of elaborate queries e.g., including optional matching, unions and sorting.

The assessment is performed in two steps linked interdependently. First, the qualitative analysis returns a set of attack vectors. Second, the quantitative analysis assesses the risk for a specific setting. The qualitative analysis uses a set of SPARQL SELECT queries to determine each component of the vector. Listing 6.2 shows the query used for the data access component. All information objects that are sent by a business actor and received by another business actor are returned. Listing 6.3 shows the query that determines the privacy assets, i.e., all information objects that are either of directly personal or sensitive data. Note that only the query for directly personal *and* sensitive data is shown. Listing 6.4 shows the query that determines the attack resources. Hence it returns whether a business actor has actual access to data or not.

```
SELECT DISTINCT ?io ?baprovider ?bacollector
WHERE {
    ?io x:isSentBy ?provider; x:isReceivedBy ?collector .
    ?provider x:isRealizationOf ?baprovider .
    ?collector x:isRealizationOf ?bacollector .
    ?baprovider a x:BusinessActor .
    ?bacollector a x:BusinessActor .
    FILTER (?baprovider != ?bacollector)
}
```

Listing 6.2: The SPARQL query that is used to determine the data access component of the attack vector.

```
SELECT DISTINCT *
WHERE {
    ?io a x:DirectlyPersonalData .
    ?io a x:SensitiveData
}
```

Listing 6.3: The SPARQL query that is used to determine the privacy asset component of the attack vector. Note that there are corresponding queries for directly personal or sensitive data and indirectly and/or insensitive data.

```
SELECT DISTINCT ?io ?bareceiver
WHERE {
    ?io x:isReceivedBy ?receiver .
    ?receiver x:isRealizationOf ?bareceiver .
    ?bareceiver a x:BusinessActor
}
```

Listing 6.4: The SPARQL query that is used to determine the attack resource component of the attack vector.

For the quantitative analysis threat patterns defined by experts are used. Threat patterns describe vulnerabilities and countermeasures with SPARQL ASK queries, taken together in an XML file. Patterns, vulnerabilities and countermeasures are assigned a name. Each vulnerability and each countermeasure is further assigned an expected loss value (EL). For the description of conditions for vulnerabilities and countermeasures SPARQL has been chosen. SPARQL resembles natural language queries and due to its pattern matching syntax it is easy to plug in results from the qualitative analysis. Listing 6.5 shows a fraction of such a threat pattern.

```
<Pattern name=”tracking customer position”>
  <Vulnerability name=”direct personal data”>
        <EL>5</EL>
    <Condition>
            ?io a x:DirectPersonalData
    </Condition>
  </Vulnerability>
  <Countermeasure name=”...”>
        ...
  </Countermeasure>
</Pattern>
```

Listing 6.5: Fraction of a threat pattern including vulnerabilities and countermeasures.

As a result of the qualitative analysis, business actors and information objects that are subject to privacy issues are known. Hence, for a detailed analysis variables in the conditions marked with a $ can easily be bound to concrete values. The implementation provides mechanisms for this kind of binding.

6.4 Policy Decision Point

This section briefly describes the prototypical implementation of the PDP-PEP pattern as introduced in Chapter 3. The implementation is based on Java 1.7 Servlets running on Apache Tomcat 7[9]. The Servlets represent PDP and PEP, respectively. A user request for an application is guarded by a PEP and forwarded to the PDP, including information about the intended application and the sending party. The PDP performs an ontology driven privacy assessment for the particular use case with a predefined set of threat patterns and displays the result to the user. The result shown includes (i) a summary for the overall privacy impact (low, medium or high) in appropriate colors for immediate recognizability; and (ii) an optional detailed view showing the full threat matrix. The user is requested to either *continue* or *cancel*. If the user decides to continue, the browser is forwarded to the application. In case of a cancel, the user is directed back to the PEP which displays that access will not be granted. For the prototypical implementation, the set of applications is given by the use cases defined in Chapter 7. As the focus is on demonstrating the PDP-PEP pattern for ontology driven privacy assessment there is no actual implementation of the use cases, i.e., no application that actually performs demand response or the like.

In practical use the formal use case description will be provided by either third-parties or the providers of the application themselves.

[9] `http://tomcat.apache.org/`

7

Evaluation

This chapter describes in detail how the system has been evaluated and which insights have been gained from the results. First, the methodology for the evaluation of privacy impacts based on the ontology driven assessment tool is explained. The assessment is then applied to three use cases and the resulting classification is discussed. Further, the system's ability to generalize to other threats is addressed. Finally, one of the main issues, the semantic gap, and how this work aims to bridge this gap is outlined.

7.1 Methodology

For the development of the system a thorough state of the art analysis about privacy requirements, requirements engineering approaches and existing knowledge served as the basis for ontology design. The aim of the system is to be a qualitative and quantitative assessment tool for both, system engineers and customers.

For evaluation, the framework is applied to the use cases in the following sections. These use cases are previously unseen and are intended to be a representative set of typical smart grid applications. *Unseen* refers to the fact, that these use cases have not been taken into account explicitly at the time the system was designed in order to gain justifiable results. The use cases are chosen to illustrate a representative set of typical and real-world smart grid applications that affect customer privacy. Furthermore, the use cases are based on insights from the Los Angeles Smart Grid Model Region and the Salzburg Smart Grid Model Region.

For each of the following use cases, first a general overview and related literature is provided. The principal actors, information objects and data flows are outlined and finally a detailed qualitative and quantitative assessment is conducted. An outline for each use case sketching business case, business actors, system actors and principal information flows is shown in Appendix A. Finally, the results are summarized and discussed with respect to insights gained in model regions and state of the art literature reviews.

7.2 Smart Metering

The smart metering (SM) use case has been chosen as smart metering is one of the first services that is rolled out and a key enabling technology in order to adapt infrastructures towards a smart grid [4]. Furthermore, business processes for smart metering are either already established or about to be rolled out and there is ongoing research on smart meter privacy and security issues [9, 44, 45]. In this smart metering scenario, the residential building of a customer is equipped with a device that collects data about the current energy consumption. This data is sent to the utility for billing and optionally for more advanced applications such as demand response. Investigations in both model regions in the US and the EU have shown different approaches towards the setup and operation of the smart metering infrastructure. In the US model region this use case (denoted as SM_1) is given as follows: Smart meters are installed by the utility in a residential building. The smart meter maintains a wireless connection to a central base station (router) which is responsible for a number of surrounding meters. Whenever the utility wants to read data, an on-demand read request is sent to the router and subsequently to the corresponding smart meter. It is assumed that the returned values are temporarily stored in an internal database. For billing, data from the billing system is linked to meter data. Meter data is given by triples of meter number, energy consumption and timestamp, whereas the billing software maintains a complete customer profile, including name address and past payment behavior and pointers to the customer's meter.

In the EU model region this use case (denoted as SM_2) is given as follows: smart meters are installed in a building and data is further collected at a fixed rate (e.g., 96 values per day, which is most common due to regulatory provisions). In contrast to the US, data of one day is summarized and sent to the utility on the subsequent day. Multiple smart meters are connected to a data concentrator that does not only map from power line communication to an IP based protocol, but also collects (modeled as a composition) data from the attaches meters. At the utility, data is processed in a head-end system and stored in a database. Billing processes follow the same information flows as in the US.

SM$_1$ – Actors. Business actors are the *user* and the *utility*. The user is mapped to the system actor *smart meter*. The utility is mapped to a *router*, connecting to a number of surrounding smart meters, to a *database* that stores retrieved values, to a *billing system* containing information about the customer and to a *processing system* that combines meter values, hence energy consumption, and billing information.

SM$_1$ – Information Objects. Information flows in this use case are characterized by on-demand request/response message from the utility premises to the customer premises. Within the utility's premises the requested meter value is sent from the router to the database and further to the processing system. The processing system also receives customer data from the billing system and triggers meter value requests at the router.

SM$_2$ – Actors. Analogously to SM$_1$, business actors are the *user* and the *utility*. Again, the user is mapped to the system actor *smart meter*. The utility, however, is mapped to a *data concentrator*, collecting data from a number of assigned smart meters, to a *head end system* that stores retrieved values, to a *billing system* containing information about the customer and to a *processing system* that combines meter values, hence energy consumption, and billing information.

SM$_2$ – Information Objects. The principal information flows are similar to SM$_1$. A meter value is sent at a fixed frequency from the customer premises to the utility's data concentrator and forwarded to the head-end system. The processing system sends meter value requests to the head-end system and receives data from both, the head-end system and the billing system.

The risk values for the following threats are determined for the binding `$ba` ← `BusinessLayer:BusinessActor:Customer`.

SM$_1$ – Customer presence at home ($R(T_1)$). A smart meter is a device in the customer premises that collects data on a regular basis. The qualitative analysis shows that on demand read requests allow the utility to determine the current consumption at arbitrary points in time (depending on the frequency of the meter itself). Further, meter values that have been retrieved are stored in a database and sent for further processing. For this threat four vulnerabilities and one countermeasure are identified, resulting in a PO of 0.7, an EL of 6.0 and a risk value of 4.2.

SM$_1$ – Identification of customer habits ($R(T_3)$). In this use case metered data is stored in a database for further processing. The intended usage is billing, however, data can be used to run statistics on previous customer behavior and therefore predict future actions. For this threat eight vulnerabilities and two countermeasures are identified, resulting in a *PO* of 0.65, an *EL* of 7.5 and a risk value of 4.875.

SM$_2$ – Customer presence at home ($R(T_1)$). Smart metering is especially prone to reveal information about customer presence when metering is done on a regular basis. The qualitative analysis shows that meter values are sent from the customer premises to the data concentrator and further to the head-end system. The latter is critical, since meter values sent at a certain frequency are persisted. For this threat four vulnerabilities and one countermeasure are identified, resulting in a *PO* of 0.9, an *EL* of 11.5 and a risk value of 10.35.

SM$_2$ – Identification of customer habits ($R(T_3)$). In this use case data metered at a fixed frequency is stored persistently. The intended usage is billing, however, data can be used to run statistics on previous customer behavior and therefore predict future actions. For this threat eight vulnerabilities and two countermeasures are identified, resulting in a *PO* of 0.75, an *EL* of 11.5 and a risk value of 8.625.

Evaluation of the smart metering use case is performed under consideration of typical threats as discussed in literature [9, 44, 45]. Typical threat patterns are used and the smart metering approaches for the US and the EU are compared. Customer presence at home is seen more critical in SM$_2$, since metering is done at a fixed frequency, whereas in SM$_1$ particular values are requested. Accordingly, the threat of identifying of customer habits is ranked higher in SM$_2$. Based on the use case outlines described above, the threats of SM$_2$ also apply for SM$_1$, if metering would be done at a fixed frequency.

7.3 Demand Response

In this section a real-life use case from the University of Southern California microgrid is evaluated as an example. This use case has been chosen as it is (i) simple enough to verify results based on literature reviews; and (ii) complex enough to have an interesting combination of actors and information flows. The focus is on a demand response (DR) scenario similar to the one described in [58] and briefly discussed in [45]. A customer interested in demand response creates an online profile stating which demand response actions the customer is interested to participate (e.g., turning down air condition). When the utilities want to curtail load with demand response, a customer whose profile fits the current requirements is sent a text message to, e.g., turn down the air condition. This message is acknowledged by the customer and the utility further reads the meter values to track actual power reduction. Besides the data flows mentioned, this further involves the storing of the profile and the past behavior of the customer for a more accurate prediction. For modeling this use case as a DFG, the following actors and IOs are identified.

DR – Actors. Business actors are the *user* and the *utility.* The user is mapped to the system actors *smart meter*, *device* and *portal.* demand response requests are sent to the user device (e.g., a cell phone) and the user's demand response preferences are set in the portal (e.g., a web service). The smart meter is used to measure actual curtailment. The utility is mapped to a *DR repository*, containing preferences for each user and past behavior, to a *prediction unit* predicting demand response requests based on the preferences and a *control unit* to meter user feedback and actual curtailment.

DR – Information Objects. Cross-domain/zone information flows include user preferences sent to the utilities, demand response requests sent to the user from the utility and both, the user acknowledge/decline and the meter values sent back to the utility. Information flows within the utility premises are from the demand response repository to the prediction unit and from the control unit to the demand response repository. Given the threat patterns introduced above, the framework is used to determine the privacy impact of this use case which provides the following results.
The risk values for the following threats are determined for the binding `$ba` ← `BusinessLayer:BusinessActor:Customer`.

DR – Customer presence at home ($R(T_1)$). The qualitative analysis shows that in the demand response repository of the utility information about both, past customer behavior and customer data is brought together, i.e., directly personal data is composed with a detailed history of a person's actions. Furthermore, the customer's acknowledge/decline and the measured curtailment reveal if a customer (i) responded to the demand response request; and (ii) actually participated in DR; both is a indication for the presence at home. For this threat four vulnerabilities and one countermeasure are identified, resulting in a *PO* of 0.9, an *EL* of 11.5 and a risk value of 10.35.

DR – Tracking customer position ($R(T_2)$). This threat might apply in two different scenarios: First, it is immediate if the acknowledge/decline response to demand response requests contains the customer position (e.g., if sent by a cell phone or other mobile device). This does not only show the customers past and present position, but also if the customer is able to remotely control devices in his or her premises. Secondly, this threat applies when the customer is represented by an additional component *electric vehicle charging station.* Assuming that demand response requests are also sent with respect to the charging behavior. Based on the amount of energy the customer is willing to provide for demand response, it might be possible to estimate the consumption of the electric vehicle and subsequently the traveled distance. For this threat two vulnerabilities and one countermeasure are identified, resulting in a PO of 0.66, an EL of 5.0 and a risk value of 3.33.

The demand response use case draws on immediate insights from the University of Southern California microgrid. The assessment is performed for the threat revealing the customer's presence at home and for tracking the customers' position. While the risk for the latter is comparably low, the former has a high risk due to immediate responses of the customer to requests.

7.4 Electric Vehicle Charging

Electric vehicle charging (EVC) is an upcoming issue that is addressed in both model regions, in the US and the EU. While a global roll-out for electric vehicles is still pending, in urban areas considerable effort is made in recent years to establish appropriate infrastructures [63].

A use case for electric vehicle charging and corresponding privacy issues are addressed by Langer et al. in [40]. The authors identify four principal use cases: (i) controlled customer premises charging; (ii) controlled foreign premises charging; (iii) uncontrolled customer premises charging; and (iv) uncontrolled foreign premises charging. For evaluation use cases (i) (denoted as EVC_1) and (ii) (denoted as EVC_2) are investigated more closely as they have the most privacy impacts. Controlled customer premises charging describes a scenario where the customer plugs in the vehicle in his or her charging station and optionally sets some preferences, such as a deadline for the charging process. Here, the utility has detailed information about the date and time when the customer arrives and departs as well as information about the energy used for charging the car. The latter can be used to estimate travel distances. Controlled foreign premises charging describes a scenario where the customer is charging an electric vehicle at a foreign charging station. The customer is further using authentication for billing purposes. This allows an even more detailed analysis about the customer's routes and behaviors.

EVC$_1$ – Actors. Langer et al. [40] describe the following set of actors. Business actors are the *customer* and the *utility.* The user is mapped to the system actors *electric vehicle* and *charging station*, which is the gateway to the grid operator. The utility is mapped to *energy supplier*, *grid operator*, *energy management system* and *customer interface.* The customer interface is allowing the user to set preferences and the energy management system is collecting information and manages the charging process.

EVC$_1$ – Information Objects. The energy demand of the vehicle is reported to the charging station, both processes result in an information flow within the customer premises. Cross-domain/zone information flows include the reporting of the maximum energy need to the energy management system and the sending of a charging start/stop signal from the energy management system to the charging station. Within the utility premises the energy management system reports the energy usage to the energy supplier for billing and the grid operator reports the current grid load to the energy management system.

EVC$_2$ – Actors. In addition to the actors described for EVC$_1$, Langer et al. [40] identify another business actor *service provider* that is mapped to the system actor *charge service provider.* This actor supplies the charging services. For authentication the business actor *customer* is additionally mapped to a system actor *ID card.*

EVC$_2$ – Information Objects. The information flows in this use case are more complex than in EVC$_1$. The charging station receives information about the energy need from the electric vehicle and information about the customer (customer ID) from the ID card. The charging station and the charge service provider exchange information about customer identification (request/response). Maximum energy demand and customer ID are then reported to the energy management system, which in return sends a charging start/stop signal. The energy management system receives charging preferences from the customer interface and the current grid load is reported by the grid operator. Energy usage and customer ID are finally sent by the energy management system to the energy supplier for billing.
The risk values for the following threats are determined for the binding `$ba` ← `BusinessLayer:BusinessActor:Customer`.

EVC$_1$ – Customer presence at home ($R(T_1)$). The qualitative analysis shows that plugging in the electric vehicle (or at least initiating the charging process) and plugging it out triggers an event that can be recorded. In the utility premises, in addition, two different parties interact and exchange information, the grid operator an the energy supplier, both connected with the energy management system. For this threat four vulnerabilities and one countermeasure are identified, resulting in a PO of 0.6, an EL of 2.0 and a risk value of 1.2.

EVC$_1$ – Tracking customer position ($R(T_2)$). According to Langer et al. [40] it is possible to determine an estimation for the route of an electric vehicle based on the amount of energy needed. For this threat two vulnerabilities and one countermeasure are identified, resulting in a PO of 0, an EL of 0 and a risk value of 0.

EVC$_2$ – Customer presence at home ($R(T_1)$). In this use case, charging takes place in a foreign premises. The customer presence at home can thus only be detected indirectly, i.e., by assuming that the customer is not at home when using the ID to register at a foreign charging station. For this threat four vulnerabilities and one countermeasure are identified, resulting in a PO of 0.6, an EL of 2.0 and a risk value of 1.2.

EVC$_2$ – Tracking customer position ($R(T_2)$). The assessment reveals that a charging station is attributed with a location information and the customer is identified with an ID. Both, customer ID and the location of the charging station are transmitted to energy management system for billing purposed. It is therefore possible to track both, the location used for charging and the amount of energy needed and hence determine an estimation for the route. For this threat two vulnerabilities and one countermeasure are identified, resulting in a PO of 0.66, an EL of 5.0 and a risk value of 3.33.

For the electric vehicle charging use case a thorough analysis of privacy issues has been conducted by [40]. The tracking customer position threat described in literature for EVC$_1$ was not confirmed by the assessment. Investigating both, the use case description and the threat patterns, shows that the threat pattern describes a composition of location and timestamp which is not present in the use case, however location and timestamp are sent in a single IO. This is an example of the semantic gap and described in the corresponding section below. The other threats show comparably low risk values, however, all with a probability of occurrence greater or equal 0.6.

7.5 Generalization

In this section the system's ability to be applied to other, more general threats is discussed. Even though the primary focus of this work and the classification framework is on privacy, during design and development the aspect of generalization has been widely considered. First, the ontology driven approach allows extensibility by design. Ontologies allow to be combined, merged and revised. Secondly, expert knowledge changes and evolves and is enriched by new insights. In summary, the system is extensible in three major dimensions:

(i) **Ontology.** Ontologies are designed for extensibility and for linking knowledge from different sources. For specific issues based on the existing ontology, it is sufficient to adapt the pre-classifiers by modifying the appropriate class expressions. Whenever the framework needs to be extended to new fields of application or adjusted to new taxonomies apart from the smart grid, it is reasonable to start extending the ontology. Existing ontologies can be integrated and merged in order to meet domain specific requirements. Merging ontologies, however, is not always a trivial task. The challenges and opportunities of merging ontologies are discussed in depth and formalized by Euzenat and Shvaiko [24]. The authors present the matching problem as a function that returns an alignment (the correspondence between different entities) for a pair of ontologies. Finding that function is an iterative and evolving process consisting of a feedback loop and space for enhancements.

(ii) **Threat Patterns.** Threat patterns are how expert knowledge is primarily plugged in the system. Therefore, threat patterns are most suitable for extending the system for new threats in the smart grid, where underlying semantics of the model are not changed. New, application specific patterns can be formed in order to address – for example – high level business cases, security issues, or specific communication scenarios in the smart grid. This does neither affect the system architecture as a whole nor the assessment process, yet it adds all new perspectives for evaluation.

(iii) **Information Objects.** Information objects are the counterpart to threat patterns. Information objects are one of the major characteristics of a use case, hence one of the main sources that vulnerabilities and countermeasures are matched to. The system can easily be extended by defining additional attributes for actors or data flows. There are limited semantics for information object attributes that are predefined (e.g., the type), which is in practice not a limitation for extensions.

Investigations in the Los Angeles Smart Grid Model Region have shown possible applications for the analysis of use cases apart from privacy assessments.

Communication Bottlenecks Assessment. In the smart grid services and networks often deal with huge amounts of data. That data is composed or aggregated and processed by multiple actors. Given known attributes about both, information object properties and the component's processing capabilities, it is possible to identify such bottlenecks. The issue of data delaying and processing is mainly modeled on the information, communication and component layers. The framework can easily be adapted for such an assessment by defining appropriate properties and additional threat patterns. Existing properties are delay and frequency, new properties might include package size for information objects and buffer size for components.

Security Assessment. While the focus of this work is explicitly on privacy, security assessment procedures do not principally defer and the framework is fully applicable. In addition to a set of security specific threat patterns, the framework can be extended with properties that cover security specific aspects, such as encryption standards and algorithms. Even entire security protocols can be modeled. The issue of security evaluation is mainly modeled on the information layer.

7.6 Semantic Gap

Whenever semantics is an integral part of a solution, the problem of the semantic gap is evident. This problem is well known in other domains aiming to integrate high-level knowledge for automated retrieval or evaluation, such as search engines for text and multimedia content [28] or recommender systems [42].

Threat patterns are supposed to describe high-level threats as naturally as possible, just as experts do it. However, what in fact is posed to the system is a reasonably concrete query for retrieving a set of individuals from the ontology. The missing link between high-level statements and low-level queries is known as the semantic gap. In practice, there is the need to keep as much semantics of the original statement as possible, i.e., to make the semantic gap as small as possible. An example for such a semantic gap is the following statement describing the threat *customer presence at home*. Experts can detail this statement by claiming that a vulnerability is *if information is collected* *that allows immediate conclusion about current energy usage* or *if some device is in the customer premises that meters data at a certain frequency.* Both statements might be considered as being semantically equivalent, the latter, however, resembles a query, that better matches the ontology as the appropriate vocabulary is used (device, customer premises, . . .). For bridging the semantic gap, experts are therefore encouraged to stick to the taxonomy provided by standards as close as possible – or, if necessary – the ontology can be adapted to better meet the particular vocabulary of experts. Crapo et al. [19] address this issue and the authors present a Semantic Application Design Language (SADL) that aims to be more natural for domain experts to describe and model systems and their behavior. SADL has further shown to be applicable in industrial practice. Considering this, SPARQL, which similarly resembles natural language, has been chosen as a promising starting point. Practical evaluation with a wide range of domain experts will show the applicability.

For modeling expert knowledge in a way that minimizes the semantic gap, the following approach is suggested. This approach is based on the discernment gained from working with domain experts in smart grid model regions for the design and evaluation of use cases in the smart grid.

1. Making domain experts familiar with the taxonomy used for modeling the ontology. The ontology is based on DFGs which are based on international standards, hence a common vocabulary is provided by design. If practical application reveals that experts are using terms or concept in a different manner throughout, adapting the ontology should be considered.

2. Starting with high-level concepts (threats) that are detailed iteratively (vulnerabilities, countermeasures). Threat patterns are designed to be extensible and to capture different levels of granularity. A first high-level description of threats gives a common understanding of the problem domain and a set of vulnerabilities and countermeasures, weighted with their expected loss value, explains that problem.

3. Improving the system based on past results. The system evaluation process is an iterative approach and assessment results should be validated. Past results then influence state of the art expert knowledge in order to improve the system's performance.

The latter is also a valuable source for measuring the semantic gap, i.e., to what extent the assessment results meet the actual threats as described by experts or observed in practice.

8

Conclusion

In this chapter the work is summarized and the main achievements and outcomes are discussed. This chapter concludes with an outlook on future work in the domain of use case assessment and privacy evaluation in the smart grid.

8.1 Summary

In this report an approach was presented that allows the assessment of use cases in the smart grid. The focus of the assessment tool is on privacy, however, the system's ability to generalize on a wider range of threats was discussed. First, relevant terms and definitions were outlined. This included the concept of the smart grid and the need for changing existing infrastructures towards a more intelligent, connected and interacting network with bidirectional flows of electricity and power. Further, the term privacy was defined and potential threats for privacy were discussed in detail. The motivation for and the benefits of an ontology driven approach for classification, as well as the basics of ontologies, description logic and query languages were presented.

A number of approaches in the field of smart grid software architectures were evaluated, with respect to their ability to integrate privacy-aware data retrieval and state of the art privacy and security analysis tools were reviewed. Investigations have shown that there is currently no ontology driven approach focusing on a qualitative and quantitative assessment in the smart grid, which is based on international standards. Therefore, the concept of Data Flow Graphs, modeling actors and information flows in adherence to the European Smart Grid Architecture Model, was utilized in order to have a standardized input for the use case assessment.

Further, in this report the design and implementation of a classification ontology that draws on the concepts of the European Smart Grid Architecture Model was discussed thoroughly. Based on that, the concept of pre-classifiers using description logic expressions was introduced and the attack vector model, borrowed from security analysis, was adapted in order to be applied for privacy assessment. This finally yielded a qualitative analysis of use cases. The results of this analysis were plugged into the quantitative analysis that gives a numeric value for risk. In this report both was discussed, how to integrate expert knowledge from various domains and how to determine a numeric value. Results of the qualitative and quantitative assessment were combined in one single threat matrix for simplicity and brevity.

Implementation issues were briefly discussed, focusing on the export and import of Data Flow Graphs, the classification process and the generic representation of threat patterns. As a practical proof-of-concept application a Policy Decision Point was presented that is based on the assessment tool and thus allows privacy-aware data retrieval.

This report concludes with an evaluation that draws on use cases from two model regions in the US and the EU: the Los Angeles Smart Grid Model Region including the University of Southern California microgrid and the Salzburg Smart Grid Model Region. A set of use cases representing typical scenarios was chosen, this includes smart metering, demand response and electric vehicle charging. Evaluation provides a qualitative and quantitative assessment for these use cases and has shown promising results. Finally, the system's ability to generalize an arbitrary set of threats in the smart grid was analyzed and the issues of the semantic gap and how to bridge this gap were discussed.

8.2 Main Achievements

For this work, a set of requirements and objectives has been identified in Chapter 1. Each requirement R_i has a corresponding outcome O_i. In this section it is first outlined how these requirements are met by their corresponding outcomes and finally the main achievements are summarized.

O_1 **Identify meta-information for classification.** State-of-the-art approaches are evaluated to identify relevant meta-information for privacy assessment. Both, the Data Flow Graphs and the ontology are designed with respect to their ability to capture that information. The ontology driven approach is chosen in order to reflect both, explicit and implicit information.

O_2 **Design of an ontology for classification of use cases.** The design of the ontology is based on international standards and reflects a typical taxonomy that is used by experts from multiple domains. Ontologies are extensible by design and new, additional knowledge is easily integrable. Classification is divided into a qualitative assessment and a quantitative assessment. The latter yields a numeric value determining the risk.

O_3 **Use ontology based classification in a policy decision point.** For a concrete implementation a policy decision point is provided as a proof of concept realization for the assessment tool. This allows privacy-aware data retrieval for smart grid IT infrastructures.

O_4 **Evaluate the system's ability to generalize and to allow service classification.** Evaluation draws on use cases from model regions in the US and the EU. Therefore a representative set of use cases has been chosen that reflects typical applications. Use cases are evaluated with respect to their privacy impact.

O_5 **Evaluate the ability of this system to generalize to allow service classification.** The system is designed to evaluate the privacy impact of use cases. However, the ability of the system to be applied to more general use cases is assessed. Therefore additional application from the University of Southern California microgrid are investigated.

O_6 **Implementation is open source.** The assessment tool as well as the Policy Decision Point is provided as a prototypical open source implementation and licensed under the terms of the MIT license[1]. Further, an appropriate documentation and all depending files are made publicly available.

The solution presented in this report primarily addresses two different kinds of stakeholders and also aims to move these diametrical views closer together. One point of view encompasses system engineers who want to assess their applications and services in the smart grid with respect to privacy, in order to meet statutory or business requirements. On the other hand, end customers are addressed, who want to assess a service or application before using it, in order to make sure the service meets certain personal requirements for privacy. While both are achieved with the same core technologies, the way system engineers and users access and view the system greatly differs. System engineers approach the system from a modeling point of view and they define

[1] `http://opensource.org/licenses/MIT`

use cases from scratch. Customers, by contrast, are concerned with the assessment of existing use cases they want to participate in. The latter is addressed by integrating the assessment tool as a recommender system in a Policy Decision Point.

8.3 Future Work

Future work in the domain of this framework includes the investigation and development of additional threat patterns and the application of the assessment process to future use cases. The application scenarios of these use cases are not limited to privacy, but will also include the assessment of communication bottlenecks and security assessments. Work on this framework has also shown that in most of the cases the operational capabilities of systems are limited when adding too many privacy constraints, e.g., when allowing metering only at certain, comparably low frequencies, the effectiveness of demand response will decline as data for suitable predictions is not available anymore. Future work in the domain of automated privacy assessments will therefore focus on extending the system to automatically determine an applicable trade-off between privacy requirements and operational requirements. This will be achieved by allowing to not only define patterns for privacy threats, but also for operational requirements. In a first step, the concept of DFGs will be brought to a more abstract level that allows to define a mathematical model to capture dependencies more accurate. It will then be possible to perform an automated data flow analysis based on a set of privacy and operational requirements for certain nodes.

References

[1] ITU-T X.200 (07/1994): Information technology – Open Systems Interconnection – Basic Reference Model: The basic model, July 1994.

[2] M. Ahmed, A. Anjomshoaa, T.M. Nguyen, and A.M. Tjoa. Towards an ontology-based risk assessment in collaborative environment using the semanticlife. In *Proceedings of the The Second International Conference on Availability, Reliability and Security*, ARES 07, pages 400–407, Washington, DC, USA, 2007. IEEE Computer Society.

[3] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist – Effective Modeling in RDF, RDFS and OWL*. Morgan Kaufmann Publishers, 2012.

[4] G.W. Arnold. *Smart Grid*. Green IT: Technologies and Applications. Springer Berlin Heidelberg, 2011.

[5] K. Barker, M. Askari, M. Banerjee, K. Ghazinour, B. Mackas, M. Majedi, S. Pun, and A. Williams. A data privacy taxonomy. In *Proceedings of the 26th British National Conference on Databases: Dataspace: The Final Frontier*, BNCOD 26, pages 42–54, Berlin, Heidelberg, 2009. Springer.

[6] M. Belgiu, I. Tomljenovic, T.J. Lampoltshammer, T. Blaschke, and B. Höfle. Ontology-based classification of building types detected from airborne laser scanning data. *Remote Sensing*, 6(2):1347–1366, 2014.

[7] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] S. Bloehdorn and A. Hotho. Boosting for text classification with semantic features. In B. Mobasher, O. Nasraoui, B. Liu, and B. Masand, editors, *Advances in Web Mining and Web Usage Analysis*, volume 3932 of *Lecture Notes in Computer Science*, pages 149–166. Springer B, 2006.

[9] J.-M. Bohli, C. Sorge, and O. Ugus. A privacy model for smart metering. In *Proc. IEEE Int Communications Workshops (ICC) Conf*, pages 1–5. IEEE, 2010.

[10] M.H.J. Bollen. *The Smart Grid – Adapting the Power System to New Challenges*. Synthesis Lectures on Power Electronics. Morgan & Claypool Publishers, 2011.

[11] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M.S. Marshall. Graphml progress report, structural layer proposal. In P. Mutzel, M. Junger, and S. Leipert, editors, *Graph Drawing - 9th International Symposium, GD 2001 Vienna Austria*, pages 501–512. Springer Verlag, 2001.

[12] U. Brandes, M. Eiglsperger, and J. Lerner. Graphml primer. Online, Retrieved: 03.03.2014 11:57, `http://graphml.graphdrawing.org/primer/graphml-primer.html/`, 2001.

[13] A. Cavoukian, J. Polonetsky, and C. Wolf. Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation. *Identity in the Information Society*, 3(2):275–294, 2010.

[14] CEN, Cenelec, and ETSI. Smart Grid Information Security. Technical report, CEN/Cenelec/ETSI Smart Grid Coordination Group Std., November 2012.

[15] CEN, Cenelec, and ETSI. Smart Grid Reference Architecture. Technical report, CEN/Cenelec/ETSI Smart Grid Coordination Group Std., November 2012.

[16] K. Cerans, R. Liepins, A. Sprogis, J. Ovcinnikova, and G. Barzdins. Domain-Specific OWL Ontology Visualization with OWLGrEd. In *ESWC Poster Session*, 2012.

[17] B. Chen, Z. Kalbarczyk, D.M. Nicol, W.H. Sanders, R. Tan, W.G. Temple, N.O. Tippenhauer, A.H. Vu, and D.K.Y. Yau. Go with the flow: Toward workflow-oriented security assessment. In *Proceedings of New Security Paradigm Workshop (NSPW)*, Banff, Canada, September 2013.

[18] R: Clarke. Privacy impact assessment: Its origins and development. *Computer Law & Security Review*, 25(2):123–135, 2009.

[19] A. Crapo, W. Xiaofeng, J. Lizzi, and R. Larson. The Semantically Enabled Smart Grid. Technical report, GridWise Architecture Council, 2009. Retrieved: 01.09.2013, 11:54.

[20] C. Dänekas, C. Neureiter, S. Rohjans, M. Uslar, and D. Engel. Towards a model-driven-architecture process for smart grid projects. In P.-J. Benghozi, D. Krob, A. Lonjon, and H. Panetto, editors, *Digital Enterprise Design & Management*, volume 261 of *Advances in Intelligent Systems and Computing*, pages 47–58. Springer International Publishing, 2014.

[21] B. DuCharme. *Learning SPARQL – Querying and Updating with SPARQL 1.1.* O'Reilly, 2nd edition, 2013.

[22] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, 2nd edition, 2001.

[23] European Commission Directorate-General for Energy. M/490 en smart grid mandate – standardization mandate to european standardisation organisations (esos) to support european smart grid deployment. Ref. Ares(2011)233514 - 02/03/2011, March 2011.

[24] J. Euzenat and P. Shvaiko. *Ontology Matching.* Springer Publishing Company, Incorporated, 2nd edition, 2013.

[25] X. Fang, S. Misra, G. Xue, and D. Yang. Managing smart grid information in the cloud: opportunities, model, and applications. *Network, IEEE*, 26(4):32–38, 2012.

[26] N. Guarino, D. Oberle, and S. Staab. *What Is an Ontology?* Handbook on Ontologies – International Handbooks on Information Systems. Springer, 2nd edition, 2009.

[27] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31 – 43, 2005.

[28] J.S. Hare, P.A.S. Sinclair, P.H. Lewis, K. Martinez, P.G.B. Enser, and C.J. Sandom. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In P. Bouquet, R. Brunelli, J.-P. Chanod, C. Niederée, and H. Stoermer, editors, *Mastering the Gap: From Information Extraction to Semantic Representation / 3rd European Semantic Web Conference*, Budva, Montenegro, June 2006.

[29] M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H. H. Wang. The Manchester OWL Syntax. In *In Proc. of the 2006 OWL Experiences and Directions Workshop (OWL-ED2006)*, 2006.

[30] ISOIEC. Information technology – Portable Operating System Interface (POSIX) – Part 2: Shell and Utilities. Online, Retrieved: 21.05.2014 10:55, `http://www.iso.org/iso/catalogue_detail.htm?csnumber=17841`, 2002.

[31] ISOIEC. ISO/IEC 27001:2013 - Information technology - Security techniques - Information security management systems - Requirements. Online, Retrieved: 21.05.2014 10:55, `http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=54534`, 2011.

[32] ISOIEC. ISO/IEC 9075:2011 - Structured Query Language (SQL). Online, Retrieved: 21.05.2014 10:54, `http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=53681`, 2011.

[33] M. Jung, T. Hofer, S. Döbelt, G. Kienesberger, F. Judex, and W. Kastner. Access control for a smart grid soa. In *Proceedings of the 7th IEEE Conference for Internet Technology and Secured Transactions*, pages 281–287, London, UK, December 2012. IEEE.

[34] Y. Kim, M. Thottan, V. Kolesnikov, and W. Lee. A secure dezentralized data-centric information infrastructure for smart grid. In *IEEE Communications Magazine, vol. Energy Efficiency in Communications*, pages 58–65. IEEE, November 2010.

[35] E.D. Knapp and R. Samani. *Applied Cyber Security and the Smart Grid: Implementing Security Controls into the Modern Power Infrastructure.* Syngress Publishing, 2013.

[36] F. Knirsch, D. Engel, C. Neureiter, M. Frincu, and V. Prasanna. Model-driven Privacy Assessment in the Smart Grid. Technical report, Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, January 2014.

[37] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180, Madrid, Spain, April 2009. International World Wide Web Conference Committee (IW3C2).

[38] M. Kost and J.-C. Freytag. Privacy analysis using ontologies. In *CODASPY '12 Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 205–2016, San Antonio, Texas, USA, February 2012. ACM.

[39] M. Kost, J.-C. Freytag, F. Kargl, and A. Kung. Privacy verification using ontologies. In *Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security*, ARES '11, pages 627–632, Washington, DC, USA, 2011. IEEE Computer Society.

[40] L. Langer, F. Skopik, G. Kienesberger, and Q. Li. Privacy issues of smart e-mobility. In *Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, pages 6682–6687, Nov 2013.

[41] R. Liepins, K. Cerans, and A. Sprogis. Visualizing and editing ontology fragments with owlgred. In S. Lohmann and T. Pellegrini, editors, *I-SEMANTICS (Posters & Demos)*, volume 932 of *CEUR Workshop Proceedings*, pages 22–25. CEUR-WS.org, 2012.

[42] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.

[43] T. Mather, S. Kumaraswamy, and Latif S. *Cloud Security and Privacy – An Enterprise Perspective on Risks and Compliance.* O'Reilly, September 2009.

[44] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *Security Privacy, IEEE*, 7(3):75–77, May 2009.

[45] E. McKenna, I. Richardson, and M. Thomson. Smart meter data: Balancing consumer privacy concerns with legitimate applications. *Energy Policy*, 41:807–814, 2012. Modeling Transport (Energy) Demand and Policies.

[46] National Institute of Standards and Technology. Guidelines for smart grid cyber security: Vol. 2, privacy and the smart grid. Technical report, The Smart Grid Interoperability Panel – Cyber Security Working Group, August 2010.

[47] National Institute of Standards and Technology. NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0. Technical Report NIST Special Publication 1108, National Institute of Standards and Technology, January 2010.

[48] National Institute of Standards and Technology. NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0. Technical Report NIST Special Publication 1108R2, National Institute of Standards and Technology, February 2012.

[49] C. Neureiter. Introduction to the 'SGAM Toolbox'. Technical report, Josef Ressel Center for User-Centric Smart Grid Privacy, Security and Control, Salzburg University of Applied Sciences, November 2013.

[50] C. Neureiter, G. Eibl, A. Veichtlbauer, and D. Engel. Towards a framework for engineering smart-grid-speficic privacy requirements. In *Proc. IEEE IECON 2013, Special Session on Energy Informatics*, Vienna, Austria, November 2013. IEEE.

[51] J. Perez, M. Arenas, and C. Gutierrez. Semantics and complexity of sparql. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L.M. Aroyo, editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 2006.

[52] E. Rissanen. eXtensible Access Control Markup Language (XACML) Version 3.0. Online, Retrieved: 20.05.2014 10:50, `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf`, January 2013.

[53] S. Rohjans. *(S2)In - Semantic Service Integration for Smart Grids.* PhD thesis, Carl von Ossietzky Universität Oldenburg, September 2012.

[54] S. Rohjans, C. Dänekas, and M. Uslar. Requirements for smart grid ict-architectures. In *Proc. IEEE Int. Conf. on Innovative Smart Grid Technologies (ISGT Europe)*, Berlin, Germany, 2012. IEEE.

[55] J. Rumbaugh, I. Jacobson, and G. Booch. *The Unified Modeling Language Reference Manual*, volume 2. Addison-Wesley, 2005.

[56] R. Shearer, B. Motik, and I. Horrocks. Hermit: A highly-efficient owl reasoner. In C. Dolbear, A. Ruttenberg, and U. Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[57] Y. Simmhan, A.G. Kumbhare, C. Baohua, and V. Prasanna. An analysis of security and privacy issues in smart grid software architectures on clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 582–589. IEEE, 2011.

[58] Y. Simmhan, Q. Zhou, and V. Prasanna. Semantic information integration for smart grid applications. In J. H. Kim and M. J. Lee, editors, *Green IT: Technologies and Applications*, pages 361–380. Springer, Berlin Heidelberg, Germany, 2011.

[59] E. Sirin, B. Parsia, B.C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics*, 5(2):51–53, June 2007.

[60] S. Speiser, A. Wagner, O. Raabe, and A. Harth. Web technologies and privacy policies for the smart grid. In *Proc. IEEE IECON 2013, Special Session on Energy Informatics*, Vienna, Austria, November 2013. IEEE.

[61] The European Parliament and the Council. Official Journal L 281, 23/11/1995 P. 0031 - 0050 – Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995. Online, November 1995.

[62] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In U. Furbach and N. Shankar, editors, *Automated Reasoning*, volume 4130, pages 292–297. Springer Berlin Heidelberg, 2006.

[63] U.S. Department of Energy. Los Angeles Department of Water and Power Smart Grid Regional Demonstration Program. Online, Retrieved: 21.05.2014 12:05, `https://www.smartgrid.gov/project/los_angeles_department_water_and_power_smart_grid_regional_demonstration`, August 2013.

[64] A. Veichtlbauer, D. Engel, F. Knirsch, O. Langthaler, and F. Moser. Advanced metering and data access infrastructure in smart grid environments. In *SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications*, Barcelona, Spain, August 2013. SENSORCOMM.

[65] W3C. Resource Description Framework (RDF). Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/RDF/`, February 2004.

[66] W3C. W3C Recommendation: Extensible Markup Language (XML) 1.1 (Second Edition). Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/TR/2006/REC-xml11-20060816/`, August 2006.

[67] W3C. W3C Recommendation: SPARQL Query Language for RDF. Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/TR/rdf-sparql-query/`, January 2008.

[68] W3C. W3C Recommendation: Web Ontology Language (OWL). Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/TR/owl-features/`, December 2012.

[69] W3C. W3C Working Group Note: OWL 2 Web Ontology Language Manchester Syntax (Second Edition). Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/TR/owl2-manchester-syntax/`, December 2012.

[70] W3C. W3C Recommendation: SPARQL 1.1 Query Language. Online, Retrieved: 21.05.2014 10:56, `http://www.w3.org/TR/sparql11-query/`, March 2013.

[71] Merriam Webster. ”privacy”, merriam-webster.com. Online, Retrieved: 21.05.2014 10:56, `http://www.merriam-webster.com/dictionary/privacy`, January 2014.

[72] M.E. Whitman. Enemy at the Gate: Threats to Information Security. *Commun. ACM*, 46(8):91–95, August 2003.

[73] S.B. Wicker and D.E. Schrader. Privacy-aware design principles for information networks. *Proceedings of the IEEE*, 99(2):330–350, Feb 2011.

[74] L. Zhou and J.J.P.C. Rodrigues. Service-oriented middleware for smart grid: Principle, infrastructure, and application. *Communications Magazine, IEEE*, 51(1):84–89, January 2013.

[75] Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna. Semantic information modeling for emerging applications in smart grid. In *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on*, pages 775–782. International Conference on Information Technology : New Generations (ITNG), 2012.

[76] Q. Zhou, Y. Simmhan, and V. Prasanna. Incorporating semantic knowledge into dynamic data processing for smart power grids. In *The Semantic Web – ISWC 2012*, pages 1–16. International Semantic Web Conference (ISWC), 2012.

A

Appendix: Use Case Outline

In this appendix a sketch of the use cases that are discussed for evaluation is shown. Each figure shows business case, business actors, system actors and principal information flows. Properties and attributes of actors and information objects are not represented in this figure.

Figure A.1 outlines the smart metering use case for the US (SM_1), Figure A.2 outlines the smart metering use case for the EU (SM_2), Figure A.3 outlines the demand response use case (DR), Figure A.4 routines the electric vehicle charging use case with charging in the customer premises (EVC_1) and Figure A.5 outlines the electric vehicle charging use case with charging in a foreign premises (EVC_2).

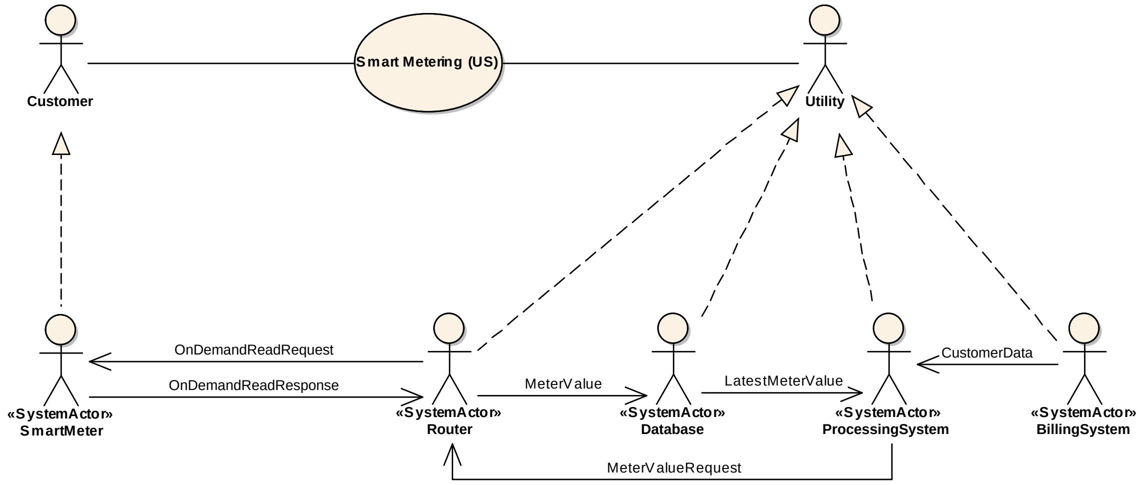

Figure A.1: Outline of the smart metering use case for the US (SM_1).

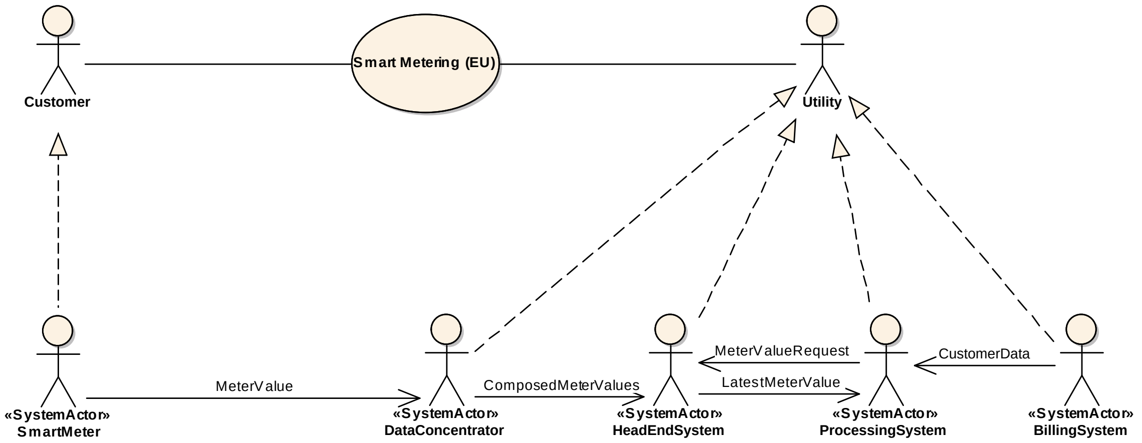

Figure A.2: Outline of the smart metering use case for the EU (SM_2).

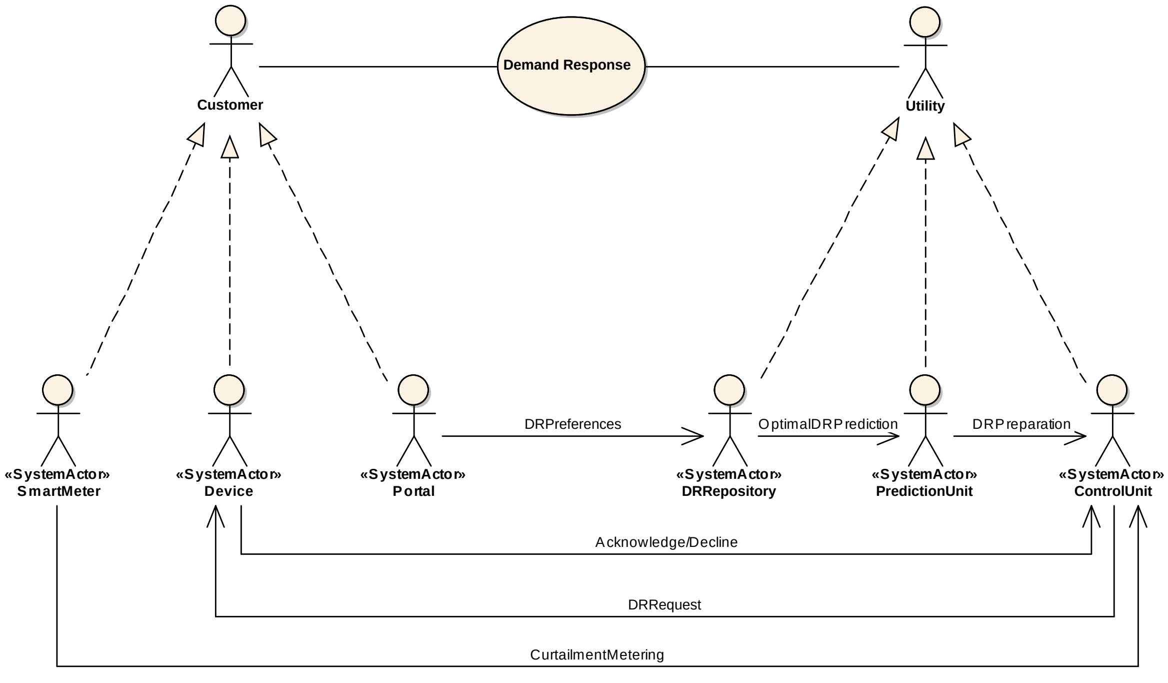

Figure A.3: Outline of the demand response use case (DR).

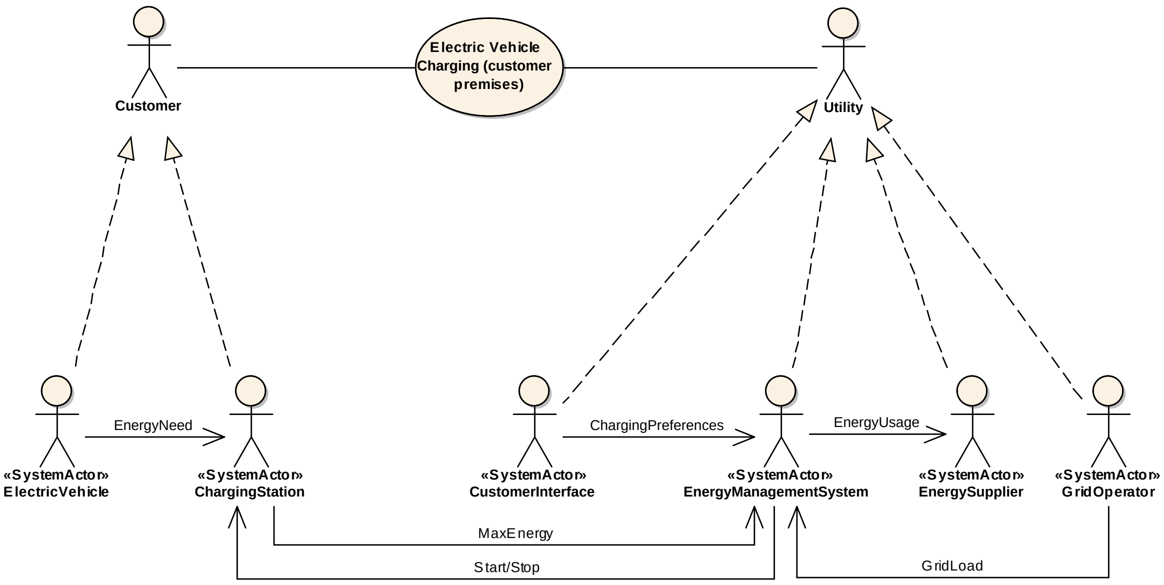
Customer
Electric Vehicle Charging (customer premises)
Utility
EnergyNeed
ChargingPreferences
EnergyUsage
«SystemActor» ElectricVehicle
«SystemActor» ChargingStation
«SystemActor» CustomerInterface
«SystemActor» EnergyManagementSystem
«SystemActor» EnergySupplier
«SystemActor» GridOperator
MaxEnergy
Start/Stop
GridLoad

Figure A.4: Outline of the electric vehicle charging use case with charging in the customer premises (EVC_1).

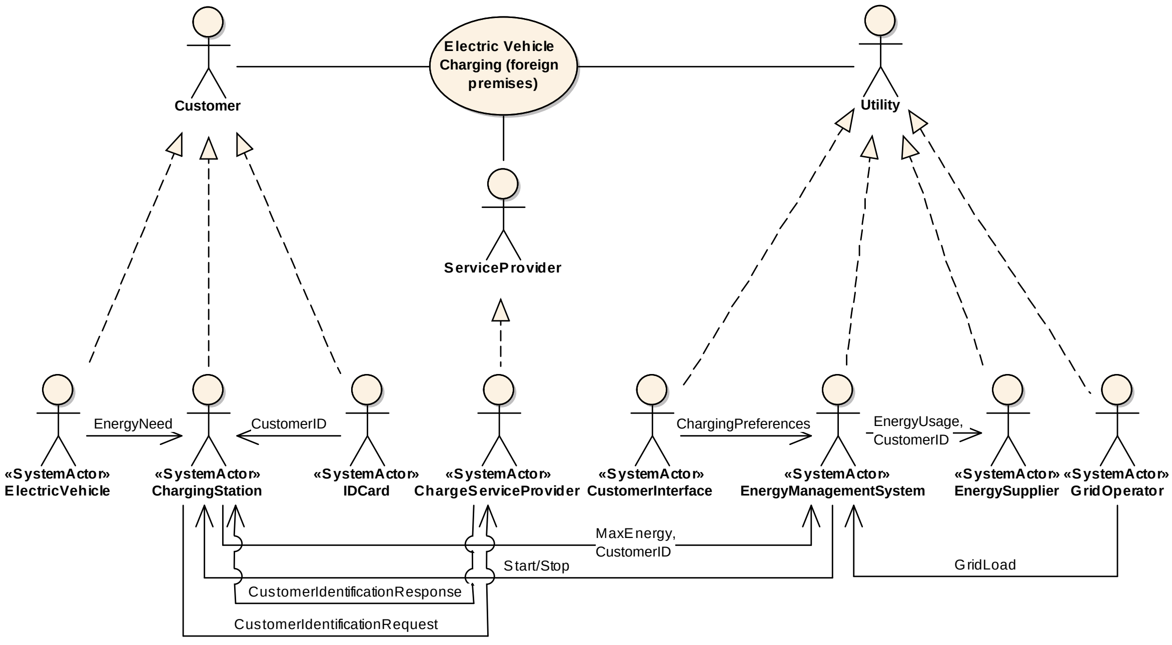
Customer
Electric Vehicle Charging (foreign premises)
Utility
ServiceProvider
EnergyNeed
CustomerID
ChargingPreferences
EnergyUsage, CustomerID
«SystemActor» ElectricVehicle
«SystemActor» ChargingStation
«SystemActor» IDCard
«SystemActor» ChargeServiceProvider
«SystemActor» CustomerInterface
«SystemActor» EnergyManagementSystem
«SystemActor» EnergySupplier
«SystemActor» GridOperator
MaxEnergy, CustomerID
Start/Stop
GridLoad
CustomerIdentificationResponse
CustomerIdentificationRequest

Figure A.5: Outline of the electric vehicle charging use case with charging in a foreign premises (EVC_2).