Privacy-aware Semantic Service Discovery for the Smart Energy Grid *

Andreas Wagner, Martin Junghans, Sebastian Speiser, and Andreas Harth
Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany

1 Introduction

The Smart Grid aims to profoundly change the way how energy is created, distributed and consumed [1] – using information technology to enable data exchange. As more data is being recorded, more personal data is handled by the various participants. Traditional means for privacy protection are not sufficient in the Smart Grid setting, where data needs to be shared in a flexible manner (thereby, e.g., enabling new business scenarios). Thus, Smart Grid service providers may gain crucial insights into the personal life of their users, when exploiting their collected data. As a result, users may want search for providers matching their privacy constraints.

Given a user request, we outline an approach for service selection, taking into account privacy settings specified in the users' policies (we assume users retain control over their data via policies). That is, in contrast to previous work, we aim to combine current service discovery approaches with work on policies, resulting in a two-step process: (1) a service retrieval phase (at design time) and (2) a service refinement phase (at runtime) based on privacy constraints. Note, while we present the Smart Grid as an application in this paper, our approach is generic in nature. As future work, we aim at providing a scalable privacy-aware service discovery approach for vast domains such as the Smart Grid.[1]

2 Two-Step Semantic Service Retrieval in the Smart Grid

We assume the use of (Semantic) Web technologies, where URIs identify participants, infrastructure components and appliances. HTTP covers data transfer. Data is represented as RDF(S) and published adhering to Linked Data principles.

Service Retrieval at Design Time. Service retrieval is based on a semantic matchmaker identifying services applicable for given behavioural (functional) and quality-related (non-functional) requirements. While policies are attached to the data and have to be verified at runtime, service properties can be matched with a given user request at design time. Semantic service descriptions and matchmaking allows reasoning on service properties and handling of different vocabularies (employed by varying providers). We model functional and non-functional attributes. The former description captures atomic and complex services. Atomic services do not model (or abstract from) any intermediate operations and conversations during the execution (e.g., described by a OWL-S profile). Complex service descriptions model their internal behaviour and conversations. Such complex services are needed in a Smart Grid as we aim for automated service provision of policy compliant services. Extending the matchmaker with a light-weight service composition enables us to, e.g., make a given service policy compliant by prefixing an additional service (e.g., a data anonymizer). We employ the π-calculus process algebra to model the

* This work was in part supported by the German government (MeRegioMobil (Grant 01ME09005) and WisNetGrid `http://wisnetgrid.org`).

[1] Examples: `http://people.aifb.kit.edu/awa/2011/eswc/meregiomobil/`.

behaviour, description logics $\mathcal{SHIQ}(\mathbf{D})$ to model process resources, (e.g., messages, actors as ontology individuals and changes over the individuals), and a semantic version of a public key infrastructure to certify correctness of properties [3] in service descriptions. Non-functional properties are modelled in an ontology as key-value-pairs [4], potentially with ranges as values. On the request side, the temporal logic μ-calculus (in combination with semantic description of resources) is used to constrain the behaviour; non-functional requirements are expressed by desired value ranges [4]. The service matchmaker is based on an ontology reasoner and determines whether the set of desired property values contains the property value of an offered service. That is, there is a match w.r.t. the behaviour if there is an interpretation of the process model, which matches the offer [2]. In next step, we extend our retrieval approach with a policy-based refinement.

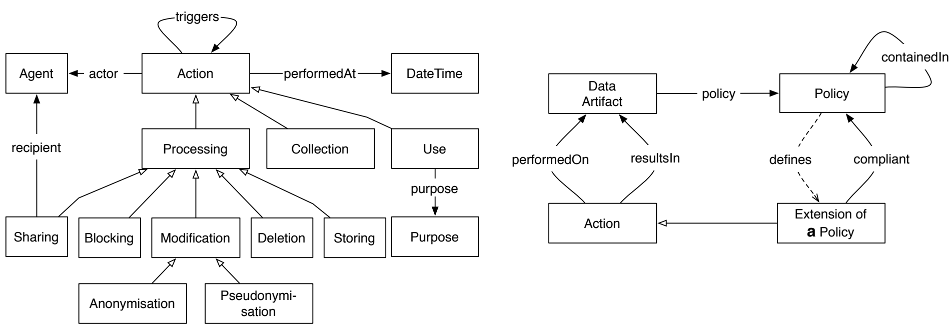

Fig. 1. Action model for privacy policies (left); top-level policy ontology (right).

Service Refinement at Runtime. Secondly, we wish to check whether the matching services comply with the user's privacy constraints. Privacy constraints are expressed as polices, which a user attaches to his data artefact. A policy models the subclass of actions, which are allowed to be performed on the artefact (see Figure 1) [5]. Data demand and intended data usage of a service is modelled according to the ontology visualised in Figure 1, which is based on the German privacy law. Policies are modelled as unions of conjunctive queries with the semantics, that an action is compliant with a policy, if its description is an answer to the policy's query. Our model does not impose purpose constraints on storage and collection of data artefacts. Instead we allow a policy to restrict the policy of the stored (or collected) data artefact in such a way that usage only for the specific purpose is allowed [5]. The triggers property can be used to express that an action is only allowed if its execution is accompanied by another action (e.g., for coupling storing and deletion actions).

References

1. NIST Framework and Roadmap for Smart Grid Interoperability Standards. National Institute of Standards and Technology, 2010.
2. S. Agarwal. Model Checking Expressive Web Service Descriptions. In *ICWS*, 2007.
3. S. Agarwal, S. Rudolph, and A. Abecker. Semantic Description of Distributed Business Processes. In *AAAI Spring Symposium*, 2008.
4. M. Junghans and S. Agarwal. Web Service Discovery Based on Unified View on Functional and Non-Functional Properties. In *ICSC*, 2010.
5. S. Speiser and R. Studer. A self-policing policy language. In *International Semantic Web Conference*, 2010.