

# Semantic modelling of policies for context-aware services

Sandford Bessler and Joachim Zeiss

**Abstract—** One of the architectures that help to realize the WWRF vision of I-centric communications is that of Personal Networks (PN), defined as a secure network formed by the devices both in the personal area of the user and in remote places (clusters) such as home, office, car, etc. An important component of PNs is a context management system that collects, maintains and derives context information from different sensors or from user actions, then makes it available to applications and services (context-aware applications). Within the Personal Network of a user, data can be freely accessed, since the user owns the devices, but when it comes to external services or resources of a PN belonging to another user, access control and privacy protection mechanisms are needed.

The work presented in this paper deals with the use of semantic web technology for modelling the distributed policies above, in the interaction with context-aware services.

**Index Terms:** policy, context-aware services, semantic web, access control, privacy

## I. INTRODUCTION

A lot of research has been focused on context-aware services in the recent years, because context can be used to make services personalized, adaptive, proactive and not only reactive. The software usability would increase as the configuration of the terminal would be drastically reduced. User context data, originally restricted to location, presence, time and identity information, has been extended to include any relevant user and environment information (also historical data, service subscriptions, biographical information, even intentions and desires) that may improve communication and the service quality.

Concerning the context modelling techniques, Strang and Linnhoff-Popien [1] evaluated several context models: key-valued models which are simple data structures for context modeling, missing most semantic information; markup scheme models which are based on hierarchical data structures that consist of markup tags with attributes and content; object-oriented models and graphical models which can be described using the Unified Modelling Language.

Finally, the ontology-based models are the most expressive models according to the study in [1] and [2]. This approach has been selected in the MAGNET beyond project [3],[13] to model the context of the user. According to the personal network concept, the user builds secure associations to a cluster of personal nodes and can freely access services and resources within that cluster. However, when interacting with other personal networks or with external services, access control and privacy mechanisms have to be in place. User context information that consists of dynamic data such as presence, location, user identities and history data, etc. is

quite sensitive, therefore its disclosure has to be carefully controlled by appropriate policy rules. On the other side, modern services become pervasive, run in user environments such as a personal network, are accessible via the web and need flexible rules for access control.

The interactions between the client application, the service and the context system can be designed according to different patterns. In this paper we consider a more advanced mode that allows the service to react when the context changes. In this case the service has to subscribe directly at the interface of the Context Management System (CMS) for events. Technically speaking, this asynchronous operation mode requires that CMS listens to events.

In Figure 1, we describe an architecture based on this interaction pattern. In the architecture we also address the problem of matching context variables and service parameters by using common ontology parts for both client and service sides, as well as symmetric policies. As explained in detail below, the policy checks are performed before the service invocation. Policy matches are performed both at the client and user side, requiring that policies are distributed and anytime available.

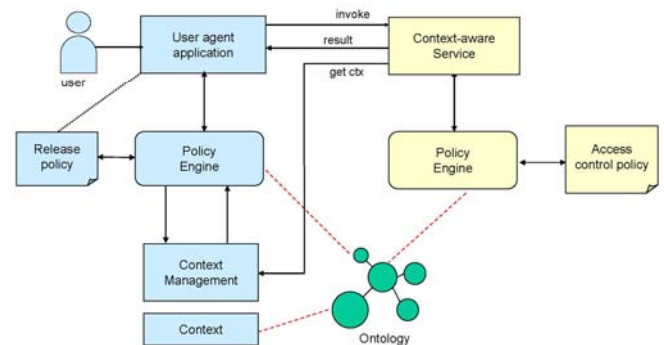


Fig. 1: Policy Architecture in a context aware service environment

The type of policies we consider in the following can be classified in:

- Access control policies: define authorization rules to access data and services. These policies are normally issued by the service. They specify the information required from clients to optimally deliver the service, and that may include private, context information such as user location or age. They may include obligations as well, specifying for example that the location data shall be deleted after the service session ends. Note, that these policies are equivalent to the privacy policies described by Platform for Privacy Preferences (P3P)

[4], [5].

- Release policies: define the user’s preferences regarding the release or disclosure of private data. More precisely, these policies specify to which user or service and under which conditions or obligations specific parts of the profile and context data can be disclosed.

The main contribution of the paper is to apply the concepts of the policy aware web [7],[8] to a context-aware service architecture and to show the advantages of using one single language, symmetric policies and reasoning tools to express both a variety of privacy rules and access control rules. The next sections are organized as follows; in Section II we present more in detail the interactions for the service presented in the scenario, in Section III we give examples of release and access control rules, then we conclude and propose further research tasks.

## II. INTERACTIONS FOR ACCESS AND PRIVACY CONTROL

In order to better understand the service interactions and the corresponding policies, we describe a pervasive scenario in which a consultant visits a company and gets access to a certain project directory as long as he is located at the company premises. The file access operation can be seen as a service, a location based service in this case. The benefit of using declarative distributed policies here is, that visiting consultants need not have user accounts in the company, their statements (e.g they work in a certain joint project) can be verified by asking the project manager for example.

With this scenario in mind, we present a more detailed message passing diagram in Fig. 2.

The Context Management System may store also the release policies, preferences and other profile data in addition to context information. On the client application side, the policy engine verifies the access policy rules sent by the services and whether they match the release policy and the preferences of the user. Similarly, the engine on the service side verifies the statements (proof) of the agent against the access policy. The release policy of the agent is checked as well, obligations are considered (e.g. to delete location data after the service is closed).

We consider the area notification operation mode, in which the context-aware service is allowed to subscribe directly to location change events. Such functionality has been proposed as web service in the Parlay X standardisation group [11]. The interesting fact is that area notification functionality has been prototyped in a mobile terminal equipped with GPS, as an IMS native location service [12].

In order to avoid policy checks for each interaction during the service session, special session IDs are exchanged between client application and service. The main interactions are listed below (see Fig. 2):

- Based on the discovery of a service access point, the user agent starts a session with the service. The authentication may be performed later on. The service policy for access control is sent back to the agent.
- The policy engine at the client side checks the access policy and verifies it against the user release policy (e.g. user agrees to be localized when he visits the

company). The statement (proof) that entitles the user to access the service is returned. It may state the user is a consultant of the company xy and works in project z. The service performs the authentication and authorization steps based on this proof (asking company xy or the project manager of z).

- Since the agent has made available the interfaces of its context management system, the service, once it had successfully authorized the agent, can directly subscribe to location information. SessionID are sent in each interaction in order to avoid repeating policy checks.
- The agent may use a service method (we denote with useService()) to invoke the service. The service may, as an alternative, subscribe to location data, receive the first notification and then return a response to the agent.

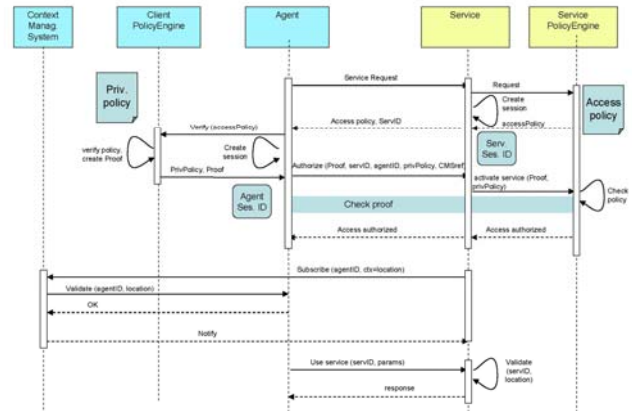


Fig. 2: Message passing diagram

## III. SEMANTIC MODELLING OF POLICIES

Previous works have often used markup language such as XACML or P3P [6] to express policy rules in context-based systems. The main research work on modeling policies with semantic methods has been performed at MIT where the Rei language has been developed [9] and where the policy aware web (PAW) initiative has been started [7],[8].

We adopted the Notation 3 (N3) language [10], which has been already proposed for expressing policies [9]. The language is very powerful and expressive. For example the statement `:locFileService a res:Service.` is to be read “`:locFileService` (the subject) is of type (predicate) `res:Service` (object). `:res` points to a name space (URI) in which the names resource and service are semantically defined.

In the first example we describe a client privacy rule (preference in P3P terminology for which the APPEL language would have been used.).

Coming back to our consultant scenario, we first would like to express the following release policy: the consultant (here the user `pdb:per1`) allows the service to track the changes in his location, however only for the area in which the visited company is located !

This rule is checked for each service and each of defined

areas, for our user (is an instance of a user). The expression ?r denotes a variable that has to be checked whether it represents a request to the file service (LOFS).

Variable ?r points a remote N3 document that describes the privacy policy of some service.

```
# accept if location is tracked only at
the company premise:
@forAll :service, :area.

{
  :service a res:Service.
  :area a ont:Area.
  ?r a ont:LOFSRequest.
  ?r.log:semantics log:includes {
    :service ont:privacy_policy
[ont:tracks [ont:subscribedArea :area]].
  :area a :companyPremise.
}
} => {
  :service ont:is_accepted_by pdb:per1.
}.
```

The statement

```
:service ont:privacy_policy [ont:tracks
[ont:subscribedArea :area]].
```

verifies that the privacy policy of the considered service contains a location tracking, but constrained to the given location *area*. *subscribedArea* is defined as follows:

```
:subscribedArea a rdfs:Property;
  rdfs:comment "defines the area inside
which location might be tracked";
  rdfs:domain :Location;
  rdfs:Range :Area.
```

At the time of processing the request the instances at the service side are as follows:

```
:companyPremise a ont:Area.

:priv_pol a pol:PrivacyPolicy;
  ont:tracks [ ont:subscribedArea
:companyPremise];
  ont:obligation :loc_del;
  ont:stores ont:AuthData,
ont:Preferences.

:locFileService a res:Service;
  mag:name "LOFS";
  ont:privacy_policy :priv_pol.
```

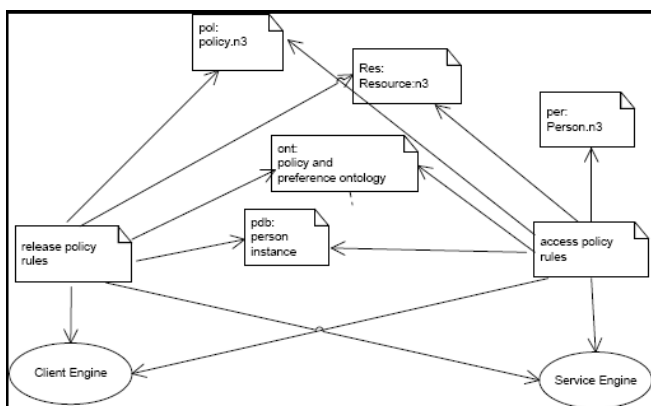


Fig. 3: Distributed access on policy and ontology files.

The second example describes the **access control** (or authorization) policy of location oriented file service (LOFS). If a certain person (instance *pdb:per2*) is for example a manager that basically confirms the authorization, then the policy rule reads as follows:

```
# grant service access for
# visitors of pdb:per2
@forAll :person, :manager.
{
  :service a res:Service.
  :service mag:name "LOFS".
  :person ont:auth_policy
[ont:consultant_of pdb:per2;
  ont:employee_of pdb:partner1;
ont:for_service :service].
  ?r a ont:LOFSRequest.
  ?r.ont:personDB.log:semantics
log:includes {
  pdb:per2 per:takes_role pdb:manager
}
} => {
  :person pol:is_granted_to_use :service.
```

The instance of the authorization policy, *:auth\_pol* for a certain user *pdb:per1* is shown below:

```
:auth_pol a pol:AuthorizationPolicy;
  ont:consultant_of pdb:per2;
  ont:employee_of pdb:partner1;
  ont:for_service :locFileService.
pdb:per1 ont:auth_policy :auth_pol.
```

This code shows that for the user (instance) *pdb:per1*, the authorization credentials for using a certain service stem from being a *consultant\_of* the manager *pdb:per2* and an employee of the company *partner1*. The last statement binds the user to the authorization policy. Note that *ont:consultant\_of* showed below as a part of the policy ontology is **common** to both client and server.

```
:consultant_of a mag:relation_1to1;
  rdfs:comment "determines a work
relationship as part of authorization
policy";
  rdfs:domain pol:AuthorizationPolicy;
  rdfs:range per:Person.
```

The ontology /semantic web approach is completely distributed: requests, policies, ontologies refer and are related to each other as illustrated in Fig. 3. The arrows show that the files have to be fetched and fed into the reasoning engine. Thus, web connectivity and open access to the ontology files, independently of any authentication and authorization step are prerequisites for such a system to function.

#### IV. CONCLUSION

In this paper we have applied the ideas of the policy aware web to a context-aware service scenario. This analyzed

setting has certain symmetry, since both the service and the application client present policies that have to be matched. The approach seems flexible and powerful. For the realization of the concepts presented, we used the CWM general semantic data processing tool written by Tim Berners Lee of W3C. Towards the realization of a service prototype, more work has to be done for the integration of context ontology with the policy ontology, the mapping of authentication enforcement schemes to the semantic assertions and proof statements. Another important engineering task is to evaluate the performance of reasoners such as CWM on target platforms such as PDAs and mobile terminals.

#### ACKNOWLEDGMENT

This work is partially funded by the IST Project Magnet Beyond.

**MAGNET Beyond** is a continuation of the MAGNET project. MAGNET Beyond is a worldwide R&D project within Mobile and Wireless Systems and Platforms Beyond 3G. MAGNET Beyond will introduce new technologies, systems, and applications that are at the same time user-centric and secure. MAGNET Beyond will develop user-centric business model concepts for secure Personal Networks in multi-network, multi-device, and multi-user environments. MAGNET Beyond has 32 partners from 15 countries, among these highly influential Industrial Partners, Universities, Research Centres, and SMEs. [www.ist-magnet.org](http://www.ist-magnet.org)

#### REFERENCES

- [1] Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey. In First International Workshop on Advanced Context Modeling, Reasoning And Management, UbiComp 2004.
- [2] Panu Korpipää, Jani Mntyjrvi, Juha Kela, Heikki Kernén, Esko-Juhani Malm. /Managing Context Information in Mobile Devices/. IEEE Pervasive Computing, 2003 21
- [3] IST Project Magnet Beyond, <http://www.ist-magnet.org/>
- [4] Platform for privacy preferences (P3P), <http://www.w3.org/P3P/>
- [5] Marc Langheinrich, A Privacy Awareness System for Ubiquitous Computing Environments, Ubicomp 2002.
- [6] M. Zuidweg, J. Filho, M. van Sinderen, Using P3P in a web services-based context aware application platform, Ninth EUNICE Workshop on Next Generation Networks, Hungary, Budapest, September, 2003.
- [7] Daniel J. Weitzner<sup>1</sup>, Jim Hendler<sup>2</sup>, Tim Berners-Lee<sup>1</sup>, Dan Connolly, Creating a Policy-Aware Web: Discretionary, Rule-based. Access for the World Wide Web. To appear in Web and Information Society.
- [8] V.Kolovski, Yarden Katz, James Hendler, Daniel Weitzner and Tim Berners Lee, Towards a Policy-Aware Web, AND <http://www.policyawareweb.org/>
- [9] L Kagal and Tim Berners Lee, Rein: Where policies meet rules in the semantic web
- [10] T.B. Lee N3 Language Tutorial, <http://www.w3.org/2000/10/swap/doc/>
- [11] Parlay X specifications, <http://www.parlay.org>
- [12] R. Pailer, F. Wegscheider, S. Bessler A Terminal-Based Location Service Enabler for the IP Multimedia Subsystem, IEEE Wireless Communications & Networking Conference, Las Vegas (NV), USA, April 3-4, 2006
- [13] H. Olesen et al., Magnet Beyond Deliverable D1.2.1- The conceptual structure of user profiles, WP1, Sept 2006, [www.ist-magnet.org](http://www.ist-magnet.org)

(1983) and Dr.Tech. degree with focus on Operations Research (1988), both from the Technical University of Vienna. Between 1980-2001 he was with the austrian telecom company Kapsch AG, where he conducted industrial R&D in the fields of packet network optimization, distributed multimedia and CSCW systems. He participated in several european projects such as FORECAST (Eureka), web4groups (Esprit) and acted as coordinator in the ACTS project DIANE. Since 2001 Dr. Bessler works as a key researcher and project manager at the Telecommunications Research Center Vienna (ftw.)

His major interests are telecommunications service architectures and platforms, context-awareness and privacy enhancement schemes. Dr. Bessler has (co-)authored over 20 conference and journal papers.

**Joachim Zeiss** has an MSc degree in Telecommunications from Anglia Ruskin University in Chelmsford, UK and a BSc in computer science from university of applied science in Fulda, Germany. He has been with several joint ventures of Nortel Networks in Germany and Austria since 1995 where he was working on Telecommunications services architecture and Design. After 2001 he worked for Kapsch AG, a major telecommunications system vendor in Austria, as a lecturer for software design and engineering at the university of applied science in Eisenstadt, Austria and as a senior researcher at FTW in the area of service platforms and distributed computing. He is currently working on his PhD thesis dealing with semantic web policy aware mobile clients.