A two-level approach to ontology-based access control in pervasive personal servers

Mohamed Bourimi, Simon Scerri, Marc Planaguma, Marcel Heupel,
Fatih Karatas and Philipp Schwarte

IT Security Management Institute, University of Siegen, Siegen, Germany
Digital Enterprise Research Institute National University of Ireland, Galway, Irland
Barcelona Digital Technology Centre, Barcelona, Spain
{bourimi/heupel/karatas/schwarte}@wiwi.uni-siegen.de
simon.scerri@deri.org
mplanaguma@bdigital.org

Abstract. A new trend in pervasive personal server hosting is to enable the integration of a user's social spheres. Ideally, the design of access control to private data should be flexible and independent of the target host. Personal data should also remain independent of environmental constraints, e.g., in order to support easy migration to new deployment landscapes. Such information interoperability can be achieved by ontology-based personal information sphere representation and management. In the digital.me project, personal data is modeled using a comprehensive set of integrated, multi-domain ontologies.This paper addresses the design and first prototype of the digital.me Userware access control engine. Here, we introduce a two-level access control design in order to decouple the semantic core from the hosting web container, while ensuring that personal data and the associated ontology-based access rights remain flexibly decoupled from the underlying environment.

Keywords: Pervasive Computing; Authentication; Authorization; RBAC; Ontology-Based Access Control;

1 Introduction

New trends aim at integrating all personal data in a personal information sphere (e.g., interests, contact information) by a single, user-controlled point of access. For this, many examples could be cited such as:

- DIASPORA, a project to help people to own their social data in a decentralized way [1]
- pervasive/ubiquitous solutions for end-user-hosted remote online education (CURE[1]) [2, 3], support for mobile communities sensing [4, 5] or SocialTV [6, 7]

[1] Collaborative Universal Remote Education: http://cure.sourceforge.net/

- Eclipse's Higgins project [8]
- the EU FP7 digital.me project [9] itself

All these systems foresee the usage of pervasive personal servers for reaching their interaction goals. However and with respect to the context of this paper Higgins and digital.me leverage ontology modeling in order to provide intelligent management capabilities of the personal spheres [10]. From the security point of view, digital.me supports transparent anonymity at the network level [11] and does not target only cloud deployment as in the case of Higgins.

In general access control, namely, secure authentication and authorization as well as ensuring the confidentiality of the communication between clients and server is must-have. In this paper we present an approach describing details related to the advancements in the ontology-based access control approach employed in the project. The essential requirements (functional and non-functional) by considering technical agreements (e.g., usage of the Spring Framework[2], support of HTTP and OAuth[3]) are listed and their fulfillment is addressed. The presented two-layer design of the access control allows for flexible decoupling of the ontology access control from environmental security attributes (securing communications, RBAC[4] based authentication, authorization, and secure business logic access).

The paper is structured as follows. The next section addresses related work. Section 3 presents the digital.me related requirements analysis. Section 4 presents our approach and demonstrates its feasibility by providing technical details in section 5. Finally, the paper is concluded in section 6 and future directions are shortly mentioned.

2 Related work

An actual overview of current EU trust and privacy related projects can be found in [3] where recent EU FP6 (PRIME project[5]) as well as EU FP7 projects (e.g., PICOS[6] and PrimeLife[7]) are addressed. A relatively good overview of classical access control mechanisms, especially for collaborative/cooperative settings (sharing data in workspaces) is given in [12].

With respect to ontology-based pervasive personal servers, none of contemporary literature, however, addresses the combination of ontology modeling in combination with classical access control mechanisms (i.e., RBAC as described in this document) as initially introduced in [13] and detailed here based on the Privacy Preferences Ontologies [14]. Further we also address the inclusion of trust in those new mechanisms and thus present a contribution beyond the state-of the art.

[2] http://www.springsource.org/

[3] http://oauth.net/

[4] Role Based Access Control

[5] https://www.prime-project.eu/

[6] http://www.picos-project.eu/

[7] http://www.primelife.eu/

3 Problem and requirements analysis

The digital.me project aims at integrating all personal data in a personal information sphere by a single, user-controlled point of access: the digital.me Userware (s. Fig. 1). This tool shall be a user-controlled personal service providing intelligent personal information management and is targeted on integrating social web systems and communities. It realizes a decentralized communication to avoid external data storage and undesired data disclosure.

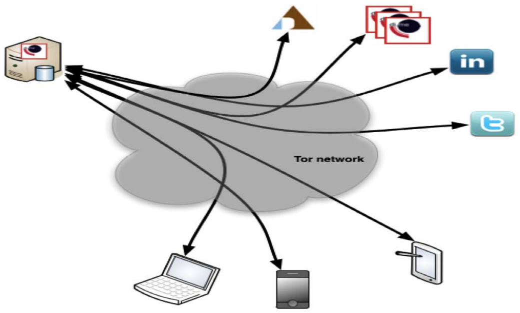


Fig. 1. Digital.me communication flows and involved parties

3.1 Role in the digital.me system architecture

In summary, the role of the Access Control Engine also by considering relationship to other components, especially the Trust Engine is highlighted in Fig. 2. The developed trust metric is being used for access control decisions. Therefore we have identified two main points of usage. First it helps the user giving access rights to the data under his/her control by providing privacy recommendations. Then, after a while of usage of the digital.me Userware, when the trust engine learned enough about the user to provide accurate trust values (or after the user might have set them manually), the user is able to define special rules to automatically share specific data to trusted contacts. Both of this access rules are defined with the help of a special ontology, the Privacy Preference Ontology (s. 4.3).

The numbered arrows (1, 2, and 3) highlight the principle flows taken place and involving directly or indirectly the Access Control Engine. Since the request broker forwards incoming calls either to the Trust Engine (step 1a) or to the Access Control Engine (step 1b). In both cases, the Access Control Engine is involved since the Trust Engine involves indirectly (step 2). The business logic is than carried out by involving the digital.me controllers layer hosting as well as storage layer (step 3 for persisting roles, permissions, roles and further access control data incl. those managed with the help of the ontology model). The

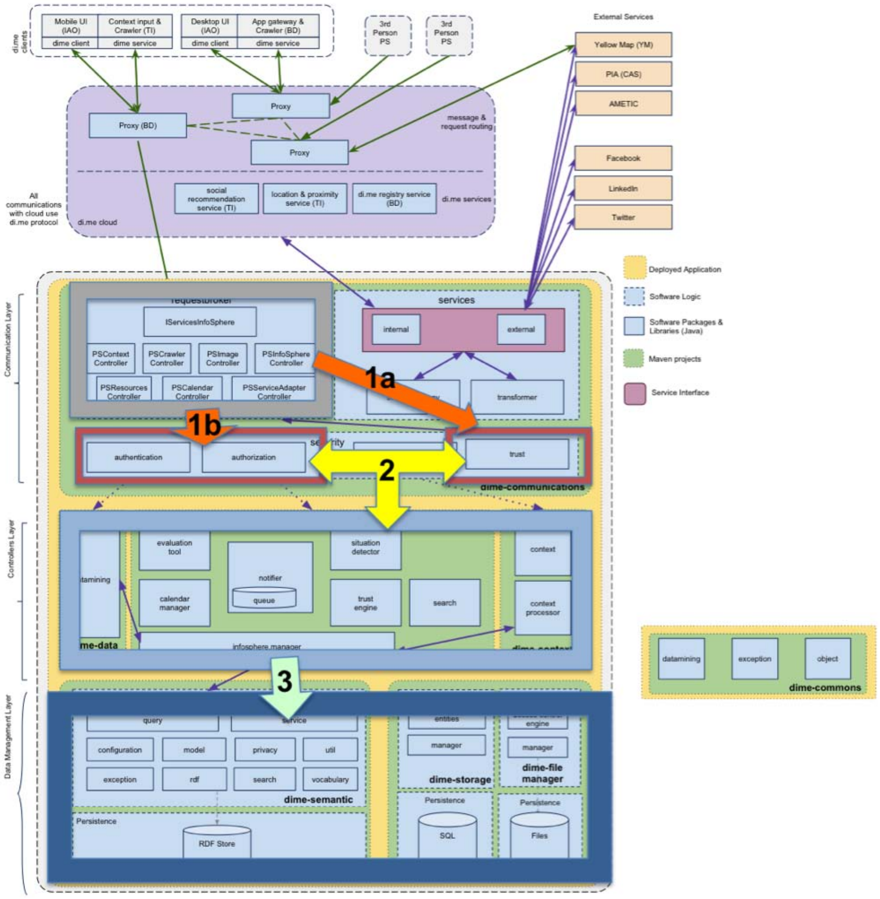


Fig. 2. Overall access control engine interdependencies

opposite direction is also supported which means that access control decisions could include the Trust Engine as described in Deliverable D4.01[8].

The objectives of the Access Control Engine target to:

– model Trust, Privacy and Security
– perform decisions based on ontology modeling
– enforce secure authentication and authorization
– ensure filtering Personal Information shared with 3rd parties
– enable User Controlled Data Sharing
– support On-the-go user -defined and customized privacy settings, by considering trust in involved networks, etc.
– take external (e.g., organizational) privacy policies into consideration
– drive the Privacy Advisory/Recommendations (warnings to user)

[8] http://dime-project.eu/publications/Items/ItemDetail.aspx?ID=4498

– return results of access decisions to the UI for considering usability issues

3.2 Gathered requirements and technical decisions

Based on relevant literature for access control and in the context of ongoing tasks in digital.me, the consortium partners agreed on supporting the following requirements:

- securing communication among involved components in the digital.me environment (Client to Server, Server to Server, and Server to External Services with different trust levels) (**R1**)
- securing the digital.me Web Layer (incl. references to any needed resource) (**R2**)
- securing the access to the digital.me Business Logic (incl. access to the ontology model) (**R3**)
- supporting security event logging if needed (**OPT_R4**)

Technical decision are related to the following and could be found in the technical specification of the digital.me infrastructure[9]:

- supporting HTTP as main communication protocol and supporting Restful web services in the first phase[10]
- using Spring framework as underlying development framework
- using ORM Technologies for accessing various storage technologies[11]
- using OAuth (along with Restful web services) as authorization protocol for specific scenarios (e.g., interaction with external services)[12]

The most important non-functional requirements (NFRs) are:

- flexible integration the ontology model access control facilities (**NFR1**)
- considering Trust and Privacy in the (meta) modelling of the ontology model (**NFR2**)

One additional NFR, which is related to architectural concerns targets leveraging advantages of aspect-orientated programming (AOP) for crosscutting concerns like security for simplifying the architecture and code maintenance (**NFR3**) [15–18]. Thereby fulfilling mainly **NFR1**. By using the Spring framework, AOP support is ensured (fulfilling so **NFR3**). Another important NFR is to support future authentication schemes by introducing an abstract authentication layer allowing for easy switching or parallel chaining of different authentication schemes (**NFR4**). The last important NFR is related to the secure data storage and migration of the data hosted in the PS (**NFR5**).

[9] http://dime-project.eu/publications/Items/ItemDetail.aspx?ID=4501

[10] XMPP remain a candidate for future extensions

[11] Hibernate and JPA in combination with relational databases such as MySQL

[12] Only client functionality is supported for now. Two external services where included, namely, LinkedIn and Twitter

4 Approach

4.1 Digital.me access control overview

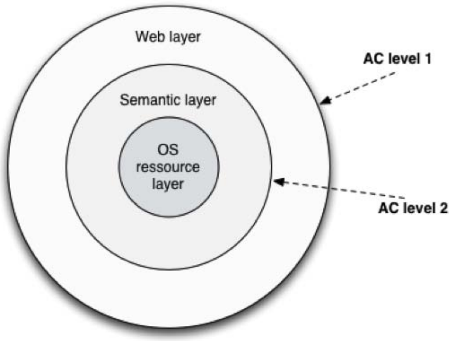


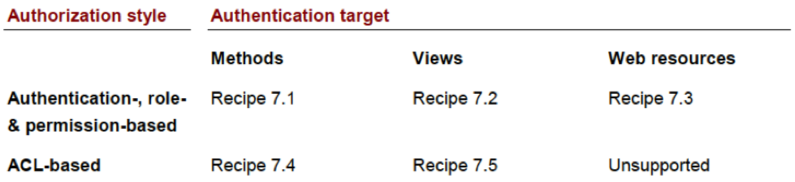
Fig. 3. Digital.me Access Control Layers (1st and 2nd level)

Digital.me access control is designed as two-level layer, which fulfills **NFR1**:

- Environmental Access Control Layer called 1st level layer, and
- Ontology-based Access Control Layer (2nd level layer).

Such separation between the environmental related access control attributes and those one stored and managed by the ontology model (2nd level layer) allows for:

- solving the problem of securing the access to the ontology model itself, which can therefore not be used for the access to the environment hosting it.
- keeping the digital.me ontology model independent from environmental attributes (in this case those ones related to the web container hosting the Userware software) and leveraging so the potential future hosting in other environments.



Authorization style	Authentication target		
	Methods	Views	Web resources
Authentication-, role- & permission-based	Recipe 7.1	Recipe 7.2	Recipe 7.3
ACL-based	Recipe 7.4	Recipe 7.5	Unsupported

Fig. 4. Spring Security Supported Authorization Styles (from [19])

In our web oriented current implementation, Figure 3 depicts the resulting access control (AC) layers. While the 1st level layer is related to the Web layer, the 2nd level layer is connected with the semantic model. Both layers need access to low level resources hosted on the underlying operating system (OS) such as files or any other resource that have to be persisted, referenced or used in digital.me. For this, both layer communicate with each other by means of an Access Decision Manager (**R2**). That component has in our case to be implemented separately since the Spring Framework does not support in an adequate manner until now as presented in the following table provided by Wheeler et al. in [19]. Wheelers book provides many practical recipes for classical security topics related to authentication, authorization as well as session management, secure business logic access and security oriented logging (**R1-3** and **OPT_R4**). For implementing the Personal Server (PS) security, trust and privacy functionalities, many of these recipes (s. Fig. 4) were adopted/adapted as we describe in the following (sub-)sections .

An Access Control interceptor is enabled at the level of the PS and is involved by any incoming call to it. This interceptor resides at the level of the single-entry-point of the whole digital.me environment (**NFR3**).

4.2 Environmental access control: 1st Level authentication and authorization

The 1st level Access Control Layer addresses the following points:

- securing the digital.me communications layer incl. external services (securing the web container communication means[13] and 4.2/5.2 for external services). (**R1**)
- securing di.me Web Layer and allowing for multiple authentication provider at the same time by using AuthenticaionVoter[15] . (**R2**)
- allowing for RBAC based authorization in the 1st level layer by using self implemented RBAC functionality, which could be combined with Spring AbstractAclVoter[16] classes. An example is depicted for our current implementation in Figure 4. (**R1-2**)

[13] The Secure Sockets Layer (SSL[14]) protocol and its successor, Transport Layer Security (TLS), are used to provide transport level security for HTTP transactions over the webthese are known as HTTP Secure (HTTPS) transactions. Hashing and salting passwords describe enabled additions.

[15] In Wheelers book Spring Access Control lists are explained in various recipes for empowering different authentication schemes by voting involving different AuthenticationProvider. Some AuthenticationProvider are provided by the Spring Security framework: AnonymousAuthenticationProvider, CasAuthenticationProvider, DaoAuthenticationProvider, JaasAuthenticationProvider, LdapAuthenticationProvider, OpenIDAuthenticationProvider, RemoteAuthenticationProvider, PreAuthenticatedAuthenticationProvider.

[16] In Wheelers book Spring Access Control Lists and Voters are explained in various recipes for empowering access control.

– securing Business Logic and Ontology Model by using JSR-250 annotations in code[17] with combination of the RBAC configuration implemented. (**R3**)

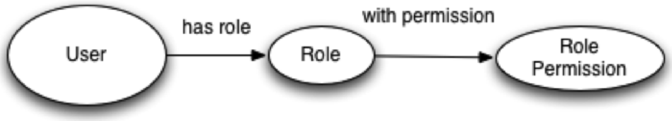


Fig. 5. RBAC 1st level implementation for digital.me according to a Wheelers recipe (from [19]).

Adding adapters for wrapping authentication and authorization through the ontology based model is the connection point between the both levels as we show here concretely with the help of the file upload scenario (involving the semantic crawler, PSresourceController, the AccessDecisionController and FileManager as well as the Access Control Repository[18]).

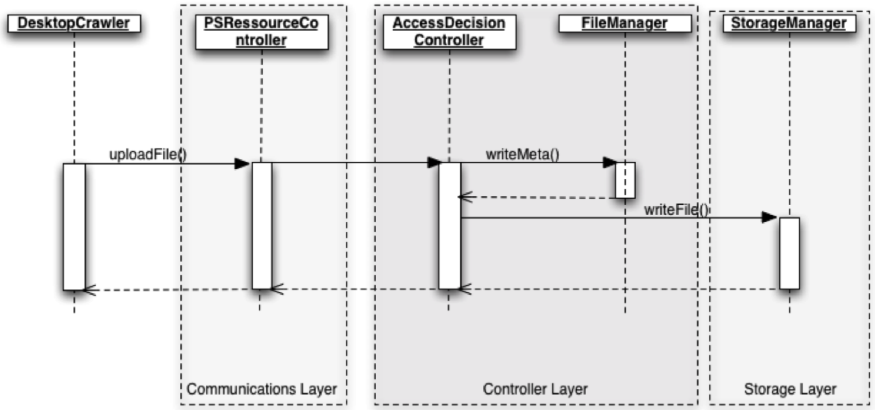


Fig. 6. Sequence diagram for a typical flow taking place when involving all layers with respect to resources usage (e.g., Files).

[17] @PreAuthorize annotation defines a default denyAll rule for the methods in the class and override later with @PreAuthorize(”hasRole(’PERM_READ_FORUMS’)”).

[18] Adding adapters for wrapping authentication and authorization through the ontology based model is the connection point between the both levels as we show here concretely with the help of the file upload scenario (involving the semantic crawler, PSresourceController, the AccessDecisionController and FileManager as well as the Access Control Repository).

A similar sequence diagram emerges when the OAuthController is accessing an external services by using access tokens. In that case the attributes depicted in the following ER diagram shows the tables structure of the used entities for supporting the management of digital.me multiple service-accounts (e.g., various LinkedIn accounts of the same user).

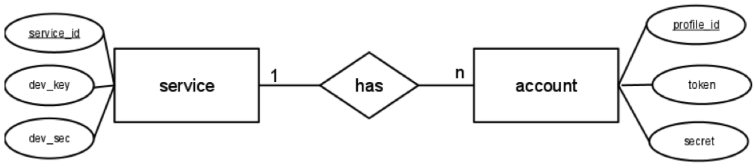


Fig. 7. ER diagram for the service-account relationship managing external services tokens.

The fulfillment of **NFR4** along with **NFR3** (enabling architectural support for **R1-3** and **OPT_R4**) is automatically reached by using Spring Security as shown in Fig. 8.

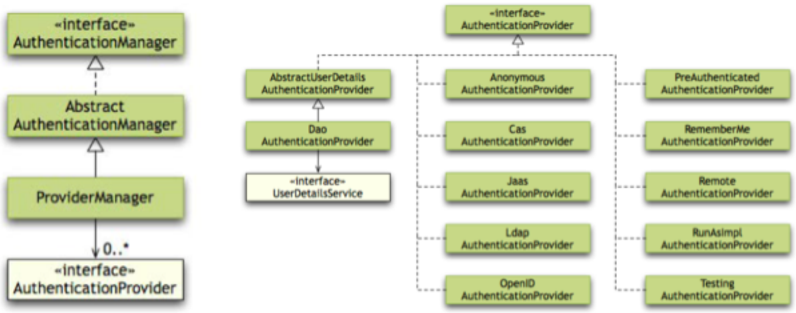


Fig. 8. Abstract Authentication Manager (left) and Authentication Provider (right) (from [19]).

In summary, the realization of the current RBAC implementation includes the following steps:

- segmentation of users of the system into user classes
- assigning levels of authorization to user roles : The first step is the mapping of an authenticated principal to one or more authorities (often called roles).

- assigning user roles to user classes (Authorities class)
- applying authentication rules globally across application resources: The second step is the assignment of authority checks to secured resources of the system (AccessDecisionManager and ontology access control facilities involvement).
- applying authorization rules at all levels of the application architecture (Securing the business logic by considering RBAC design).
- preventing common types of attacks intended to manipulate or steal a user's session (SSL, Hashing and session management facilities provided by Spring).

4.3 Domain model access control: Ontology-based authorization (2nd Level) (NFR2)

In digital.me, different kinds of information in the integrated personal information sphere are subject to different access rights, as defined and controlled by the user. This means that different subsets of this integrated information should be made available to different types of agents, be they other users or service accounts. To address this requirement, we pursue an ontology-based approach to flexible authorization system, which enables the users authorization preferences to be stored adequately and separately to the information resources to which they provide or restrict access. Thus, personal information is not replicated into multiple subsets, each of which is targeted at different agents. Instead, multiple access right preferences are stored as metadata alongside the unique personal information representations. To achieve this ontology-based authorization system, we adopt the Privacy Preference Ontology (PPO), introduced next, in digital.me.

The PPO [14] is a lightweight vocabulary that was originally intended to enable users to create fine-grained privacy preferences for their data. In digital.me, it is useful for the same purpose, albeit with a few restrictions in regards to its use. The PPO was designed with an Open World view to personal information in mind, where unless specified otherwise, all personal information is assumed to be accessible by everyone. This approach is aligned with the original domain for which the PPO was designed i.e. the Web Linked Open Data (LOD) cloud. Within digital.me, the required approach is for a closed worldview to personal information, i.e., all such information is inaccessible, unless the contrary is stated. Thus, whereby in the LOD the PPO was meant to to restrict any resource to certain attributes, which a requester must satisfy, within digital.me it is meant to give access to resources represented in the personal information sphere.

Figure 9 demonstrates the adoption of the PPO within the digital.me Ontology Framework (refer to Deliverable 03.01[19]). In fact, the PPO is only one of an integrated set of independent ontologies covering a wide variety of domains relating to the user's personal information. This ranges from less abstract information such as different kinds of native resources present on the user's devices and online accounts, user presence derived from device sensors and activities, to higher-level representations of user annotations and resource relationships in

[19] http://dime-project.eu/publications/Items/ItemDetail.aspx?ID=4496

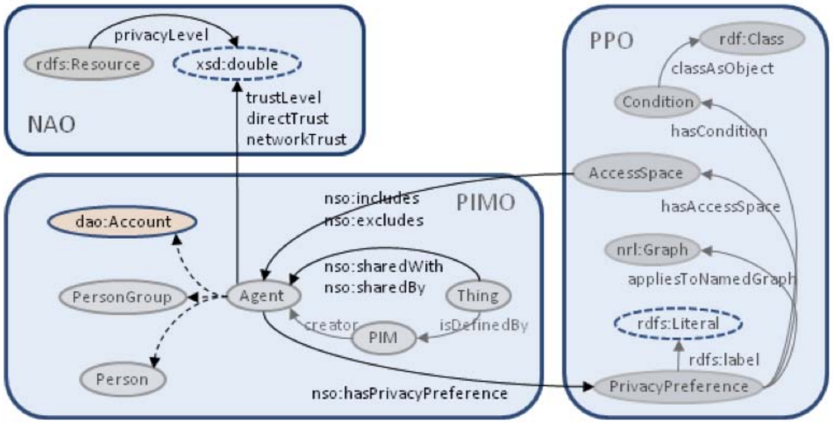


Fig. 9. Reusing the PPO in digital.me.

user's mental model of the personal information itself. The ontology framework adopts and extends established standards[20], including the entire set of ontologies used for the Social Semantic Desktop [20].

The use of the PPO relies on two other ontologies in the framework, the Personal Information Model Ontology PIMO (shown in the figure) and the Nepomuk Sharing Ontology NSO. As a special instance of a PIMO Agent, the user can define multiple privacy preferences (through nso:hasPrivacyPreference in the extended Nepomuk Sharing Ontology), each of which identifies a reference to an informational resource that can be shared with a specific set of agents (ppo:AccessSpace). The access space can be defined through two new NSO properties nso:includes and nso:excludes. These can be mapped to the UI level, whereby the user can provide both access whitelists and blacklist to easily determine who should be given access to a resource. The privacy preference grant access to specific resources, or to all instances of a specific class having specific conditions. The first option is more fine-grained, and states exactly which piece of information (e.g. the users first name, personal email address, file) is being made accessible. The second option is more generic, and can be used in cases where all instances of a class (e.g. all resources of type pimo:Document that are related to a project) can always be shared with the same set of agents.

Two additional properties in the NSO ontologies mark which resources have actually been shared with other agents. This is to be differed from the PPO representations, since the latter define with whom resources can be shared, rather than with whom they have already been shared. Also shown in Figure 9 is a subset of the Nepomuk Annotation Ontology NAO, which in the context of the digital.me project has been extended to support the automatic/manually-adjusted specification of trust values for a user's contacts and groups, as well as privacy levels for a user's resources.

[20] For more information please refer to di.me deliverable D03.01

5 Selected technical details

We based on Spring technologies for implementing AC functionalities along with classes from the ontology framework used. Spring security[21] provides a sophisticated authentication and access control system and became widely adopted as the standard solution for securing Spring Framework based applications used in critical applications [21]. Spring Security 3 provides a bundle of resources that allow for many common security practices to be declared or configured in a straightforward manner[22].

According to various technical literature, standards such as Java Authentication and Authorization Service (JAAS[22]) or Java EE Security do offer some ways of performing some of the same authentication and authorization functions, but the Spring Security module packages up implementations in a concise way and offers powerful baseline configuration features available out of the box, e.g., for various security topics such as authentication and authorization. Furthermore, a big community (also from the industry[23]) is continuously contributing and improving this framework to cover new security topics and fix detected issues [22]. Furthermore, the fulfillment of our requirements **R1-R3**, **OPT_R4**, and **NFR1-NFR4** are efficiently supported in terms of development costs.

Even though Spring Security's, application specific implementation concerns, architecture limitations, and infrastructure integration requirements are likely to complicate implementations also in the case of using Spring Security. However, Spring Security is a "hands-on" framework where developer are able to customize or extend the code to fulfill requirements that go beyond the basic out of the box options [22].

We used two libraries for integrating/supporting OAuth at the level of the PS:

- Spring Social[24] (initial implementation)
- Scribe[25] (current implementation)

Although all candidates considered for the retrieval of data from the external services supported OAuth, the Scribe Library was preferred for various reasons. In comparison to other candidate libraries such as OpenSocial , its advantages are that it is able to retrieve raw data (XML, JSON[26], etc.), it supports the majority of the popular social networks, and, from an architectural point of view, is

[21] http://static.springsource.org/spring-security/site/

[22] http://www.oracle.com/technetwork/java/javase/jaas/index.html

[23] Spring was recently acquired (for 420 million dollars, http://www.readwriteweb.com/enterprise/2009/08/vmware-acquires-springsource-for-420-million.php) by VMware Inc., the leading company for virtualization technologies. With this, the deployment of a Spring based PS into the cloud us assured since VMware is part of the Cloud Alliance targeting inter-operability.

[24] http://www.springsource.org/spring-social

[25] https://github.com/fernandezpablo85/scribe-java

[26] http://www.json.org/

more decoupled than for example OpenSocial. However, an advantage of the latter candidate library was that it introduced an abstract interface to the different external services APIs and supported the persistence of the used access tokens and secrets. Therefore, since Scribe does not include persistence, this functionality had to be coded separately and integrated in the DecisionAccessManager.

The current PS implementation includes an OAuth controller at the level of the digital.me_communications package. The supported interaction flow is shown in the following sequence diagram depicted in Figure 8.

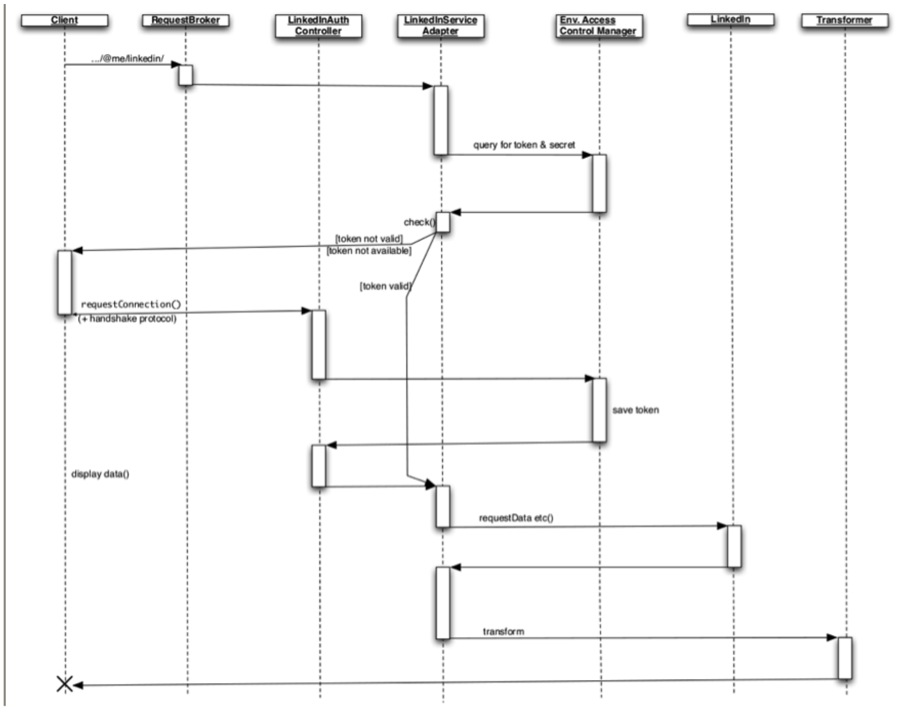


Fig. 10. Agreed flow for OAuth external services based communication.

The sequence diagram show how the PS checks if there were valid credentials acquired before in a previous interaction with a given external services provider (here LinkedIn in our current implementation). The access tokens are then either stored for the first time for future use or used if existing. In the case they were invalidated (according to the OAuth invocation procedure), the OAuth controller ensures their update in the storage. For the moment, the PS has access to two external services, namely, LinkedIn and Twitter, both by using OAuth 1.0A[27].

[27] OAuth 1.0A addresses the session fixation attack in OAuth 1.0. However, the digital.me project is targeting supporting OAuth 2.0 Authorization Protocol, which is more easy to use but still a draft and therefore not supported by many providers.

The OAuth protocol enables third-party applications to obtain limited access to an HTTP service on behalf of the resource owner. This access is established by orchestrating an approval interaction between the resource owner and the HTTP service. It is also possible to allow a third-party to obtain access to an HTTP service on its own behalf. In the traditional model, the resource owner shares its credentials with a third party, which leads to several problems and limitations:

- for future use, the third-party is not able to save the resource owner's credentials a hashed form. The credential has to be saved in clear text, which leads to the problem, that after one third-party is compromised the resource owner password is compromised, too.
- the third-party gains full access to the resources. The resource owner has no ability to restrict the access to a limited subset or to restrict the duration of the access.
- revoking the access of one third-party means to revoke the access of all others, because the only possibility the resource owner has, is to change his password.
- it is hard for the user to keep an overview over all third parties he has granted access to his resources.

OAuth uses a new layer of authorization to address these issues. The new layer separates the role of the resource consumer from the role of the resource owner. The resource consumer is issued a different set of credentials than those of the resource owner. This new credential, called the Access Token, indicates access attributes like the scope or the lifetime of the access, and is the only secret a third-party needs to access the users protected resources. Because every third-party has to request a different Access Token, it is easy for the user to maintain a list of entities consuming his resources and revoke access on/at the smallest possible level.

With OAuth, the digital.me Access Control Engine (Authentication Layer) will be able to distinguish among different devices as well as applications (e.g., web browser) of the same user without collecting privacy sensitive data about these devices. This is due to the fact that OAuth authorizes separately the access to each application (running on also on different devices of the same user). However, an extension for specific scenarios needing to distinguish between those devices remain possible and has to be analyzed from the privacy point of view for the intended scenarios.

6 Conclusions and future work

This paper provided presented the two-layer/-level design of the Access Control Engine in the digital.me project. The gathered functional and non-functional requirements were listed. The approach concretely fulfills gathered requirements with two layers which separates between the environmental related access control attributes (1st layer) and those one stored and managed by the ontology model (2nd layer). The main advantage of keeping the digital.me ontology model independent from environmental attributes is easing potential future migration to other environments as well as excluding the risk of non-intentional sharing of access control data (i.e., used credentials). The contribution represents an added value when considering related work for ontology based pervasive personal servers. The technical solution is based on cutting-edge technologies such Spring Security for implementing authentication and authorization along with the ontology based access rights for our target scenarios in digital.me.

With respect to further security related concerns, future works go towards design and implementation as well as the integration of anonymous credential systems such as Idemix by considering usability and trust advisory for identity management in general [23, 24] as well as fulfilling NFR5.

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007- 2013) under grant agreement no. 257787. Thanks to Ismael Rivera, Andreas Schuller, Sophie Wrobel as well as all our digital.me partners who reviewed this paper.

References

1. Foundation, D.: The diaspora project. http://diasporafoundation.org/ (December 2012)
2. Haake, J.M., Schümmer, T., Haake, A., Bourimi, M., Landgraf, B.: Supporting flexible collaborative distance learning in the cure platform. Volume 1., Los Alamitos, CA, USA, IEEE Computer Society (2004)
3. Bourimi, M., Kühnel, F., Haake, J.M., el Diehn I. Abou-Tair, D., Kesdogan, D.: Tailoring collaboration according privacy needs in real-identity collaborative systems. In: CRIWG, Springer-Verlag (2009) 110–125
4. Bourimi, M., Ueberschaer, B., Ganglbauer, E., Kesdogan, D., Barth, T., Dax, J., Heupel, M.: Building usable and privacy-preserving mobile collaborative applications for real-life communities: A case study based report. In: Information Society (i-Society), 2010 International Conference on, IEEE (June 2010) 435–442
5. Bourimi, M., Haake, J.M., Heupel, M., Ueberschär, B., Kesdogan, D., Barth, T.: Enhancing Privacy in Mobile Collaborative Applications By Enabling End-User Tailoring Of The Distributed Architecture. Journal for Infonomics (IJI) **3**(4) (2011) 563–572
6. Bourimi, M., el Diehn I Abou-Tair, D., Kesdogan, D., Barth, T., Hoefke, K.: Evaluating potentials of internet- and web-based socialtv in the light of privacy. In: Social Computing (SocialCom), 2010 IEEE Second International Conference on. (aug. 2010) 1135 –1140
7. Bourimi, M., Tesoriero, R., Villanueva, P.G., Karatas, F., Schwarte, P.: Privacy and security in multi-modal user interface modeling for social media. In: Social Computing (SocialCom), 2011 IEEE Third International Conference on. (oct. 2011)
8. Eclipse.org: Higgins 2.0 personal data service. http://wiki.eclipse.org/Higgins_2.0 (2011)

9. digital.me Consortium, E.F.: Integrated digital.me userware. http://dime-project.eu/ (2011)
10. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufman (2008)
11. Bourimi, M., Heupel, M., Westermann, B., Kesdogan, D., Rafael Gimnez, M.P., Karatas, F., Schwarte, P.: Towards transparent anonymity for user-controlled servers supporting collaborative scenarios. In: Information Technology: New Generations (ITNG), 2011 Eighth International Conference on. (april 2012)
12. Haake, J.M., Haake, A., Schümmer, T., Bourimi, M., Landgraf, B.: End-user controlled group formation and access rights management in a shared workspace system. In: CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work, Chicago, Illinois, USA, ACM Press (November 6-10 2004) 554–563
13. Scerri, S., Gimenez, R., Herman, F., Bourimi, M., Thiel, S.: digital.me towards an integrated Personal Information Sphere. http://d-cent.org/fsw2011/wp-content/uploads/fsw2011-digital.me-towards-an-integrated-Personal-Information-Sphere.pdf (June 2011)
14. Sacco, O., Passant, A.: A privacy preference ontology (PPO) for linked data. In: Linked Data on the Web Workshop at 20th International World Wide Web Conference, ACM Press (2011)
15. Bourimi, M., Lukosch, S., Kuehnel, F.: Leveraging visual tailoring and synchronous awareness in web-based collaborative systems. In Haake, J.M., Ochoa, S.F., Cechich, A., eds.: CRIWG. Volume 4715 of Lecture Notes in Computer Science., Springer (2007) 40–55
16. Safonov, V.O.: Using Aspect-Oriented Programming for Trustworthy Software Development. Wiley & Sons (2008)
17. Lukosch, S., Bourimi, M.: Towards an enhanced adaptability and usability of web-based collaborative systems. International Journal of Cooperative Information Systems, Special Issue on 'Design, Implementation of Groupware (2008) 467–494
18. Bourimi, M., Barth, T., Haake, J., Ueberschär, B., Kesdogan, D.: AFFINE for enforcing earlier consideration of nfrs and human factors when building socio-technical systems following agile methodologies. In Bernhaupt, R., Forbrig, P., Gulliksen, J., Lárusdóttir, M., eds.: Human-Centred Software Engineering. Volume 6409 of Lecture Notes in Computer Science. Springer-Verlag (2010) 182–189
19. Wheeler, W., Wheeler, J., White, J.: Spring in Practice. Manning Publications (2012)
20. Sintek, M., Handschuh, S., Scerri, S., van Elst, L.: Technologies for the social semantic desktop. In Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.C., Schmidt, R., eds.: Reasoning Web. Semantic Technologies for Information Systems. Volume 5689 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2009) 222–254
21. Walls, C.: Spring in Action. 3rd edn. Manning (2010)
22. Mularien, P.: Spring Security 3. Packt Publishing (2010)
23. Heupel, M., Kesdogan, D.: Towards usable interfaces for proof based access rights on mobile devices. In: iNetSec 2011. Springer-Verlag (2011)
24. Bourimi, M., Heupel, M., Kesdogan, D., Fielenbach, T.: Enhancing usability of privacy-respecting authentication and authorization in mobile social settings by using idemix. (2011)