Using Linked Data for Modeling Secure Distributed Web Applications and Services

Falko Braune, Stefan Wild, and Martin Gaedke

Technische Universität Chemnitz, Germany
{firstname.lastname}@informatik.tu-chemnitz.de

Abstract. The increasing service orientation of today's Web applications enables swift reaction on new customer needs by adjusting, extending or replacing parts of the Web application's architecture. While this allows for an agile response to change, it is inappropriate when it comes to security. Security needs to be treated as a first thought throughout the entire lifecycle of a Web application. The recently proposed WAMplus approach does not only offer an expressive, extensible and easy-to-use way to model a Web application architecture, but also puts a strong emphasis on the security. In this paper we present an exemplary implementation of WAMplus using the Sociddea WebID identity management system known from prior work. There, we show how WebID is used to identify, describe and authenticate Web applications and services while taking their protection through WAC and fine-grained data filters into account.

Keywords: Modeling, Security, Identity, Protection, Linked Data, WebID.

1 Introduction

Modern Web applications are facing changing costumer needs, short time to market, and an increasing degree of distribution. The service-oriented architecture (SOA) design pattern supports developing such Web applications by providing a set of best practices for organizing distributed capabilities. For responding to change, agile methodology fits well in this context. When it comes to security, however, it is inappropriate to apply this approach without sufficient consideration. Security needs not to be treated as an afterthought, but as a first thought throughout the entire lifecycle of a Web application [5]. While there are various tools which support developers in modeling and building Web applications, they do not holistically address the security of the entire Web application architecture [6].

In a recent work, **(author?)** proposed the WAMplus approach [8]. It does not only offer an expressive, extensible and easy-to-use way to model a Web application architecture, but also puts a strong emphasis on the security.

This work describes a prototypical implementation of the WAMplus approach into an existing WebID identity provider and management system. By exemplarily integrating the approach into Sociddea (http://www.sociddea.com/), we show its applicability, demonstrate its use in practice and strive to increase its adoption.

S. Casteleyn, G. Rossi, and M. Winckler (Eds.): ICWE 2014, LNCS 8541, pp. 540–544, 2014.

The rest of the paper is organized as follows: Sect. 2 discusses related work. Sect. 3 demonstrates the WAMplus approach. Sect. 4 concludes the paper.

2 Related Work

Model-driven Web engineering (MDWE) approaches like OOWS, UWE or WebML aim at providing means for a systematic and efficient engineering of Web applications [3]. Yet, the variety of existing domain-specific and often proprietary model description languages reduces interoperability. An integral and widely adopted solution for addressing the security topic in MDWE is missing. Dealing with the interoperability concern, semantic vocabularies for describing Web services in RDF typically consider only one type of service aspect: SAWSDL or OWL-S for SOAP-based services; SA-REST or ROSM for RESTful services [2]. OpenID or Facebook Connect are widely adopted identity management systems [1], but their limited extensibility makes them inappropriate to directly identify and attach data to Web applications and services. The Access Control Ontology (ACO) semantically specifies role-based protection of URI-identifiable resources via RDF-based access control lists. Compared to WAC, it is not yet widely used [7].

3 Utilizing WAMplus to Model Secure Web Applications

WAMplus enriches the WebComposition Architecture Model (WAM), known from prior work [4], with 1) semantic descriptions, 2) universal identification, and 3) protection through fine-grained access control for Web application and services. The semantic description of Web services allows for dynamic adaption to interface changes or feature updates. The identification facilitates interconnecting components and establishing authentication through WebID. WAMplus suggests to rely on WAC and customized views for protecting resources, also including descriptions of Web applications and services, at different granularity levels [8].

For the *description* of SOAP-based and RESTful Web services and to maintain interoperability, we use the WSDL RDF mapping (`http://www.w3.org/TR/wsdl20-rdf/`) combined with WebID. Being an identity concept, WebID (`http://www.w3.org/2005/Incubator/webid/spec/`) enables *identification* and authentication, and facilitates more detailed description. It consists of three artifacts: The *WebID URI* refers to a subject, like a Web service, and links to a *WebID profile* storing the subject's identity data and public keys using Linked Data. WebID profiles rely on RDF to semantically describe a subject's attributes. The *WebID certificate* is an X.509 certificate that includes a WebID URI identifying the subject. Matching the public key in both WebID certificate and profile enables subject authentication.

To protect resources and descriptions, the WebAccessControl (WAC) (`http://www.w3.org/wiki/WebAccessControl`) is a RDF-based vocabulary that defines access rules for URI-addressable resource. The WebID Profile Filter Language enables specifying fine-grained filters to protect data *within* user profiles [7].

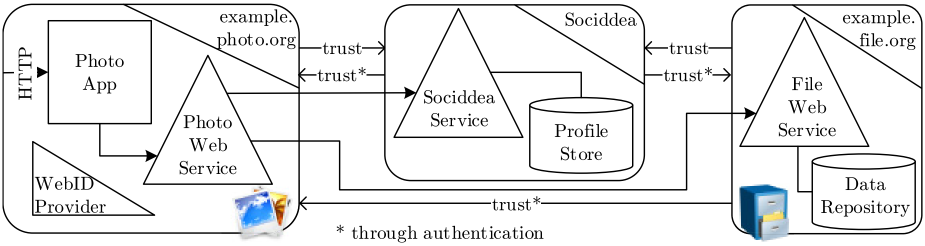
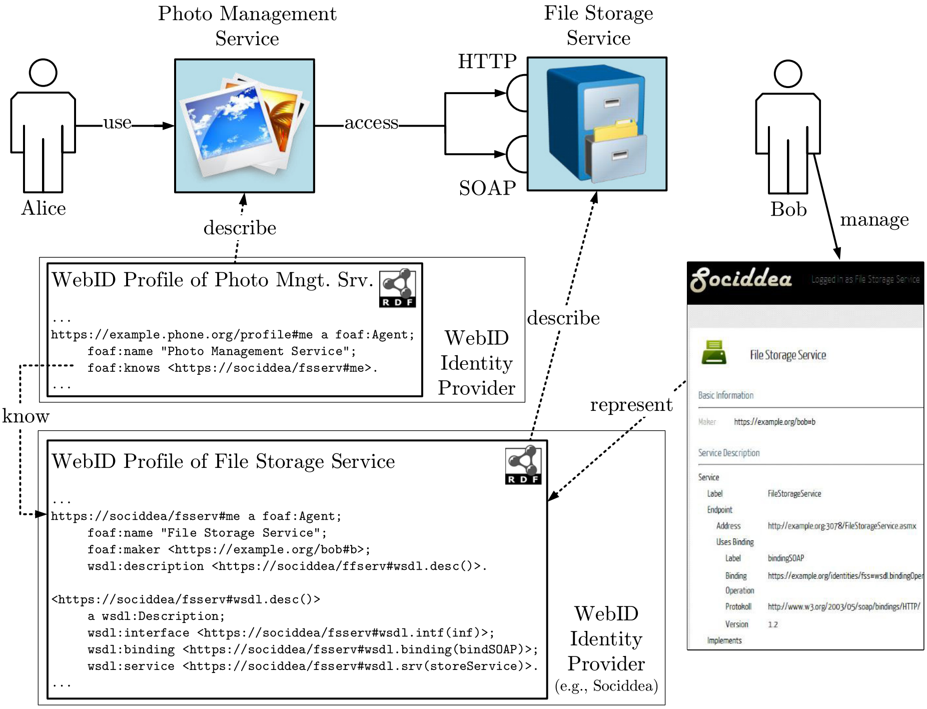


Fig. 1. WAM example

Combining these technologies facilitates a security-focused modeling of Web applications and services that also takes interoperability into account.

To demonstrate WAMplus in practice, we prototypically implemented it in the Sociddea WebID identity provider proposed by **(author?)** in [7]. Yet, WAMplus is not limited to this system. Fig. 1 illustrates an example Web application modeled with WAM. There, a file Web service contains data a photo Web service intents to use. Sociddea is used here to publish Web service descriptions as WebID profiles.



WebID Profile of Photo Mngt. Srv.

```
...
https://example.phone.org/profile#me a foaf:Agent;
     foaf:name "Photo Management Service";
     foaf:knows <https://sociddea/fsserv#me>.
...
```

WebID Profile of File Storage Service

```
...
https://sociddea/fsserv#me a foaf:Agent;
     foaf:name "File Storage Service";
     foaf:maker <https://example.org/bob#b>;
     wsdl:description <https://sociddea/ffserv#wsdl.desc()>.

<https://sociddea/fsserv#wsdl.desc()>
     a wsdl:Description;
     wsdl:interface <https://sociddea/fsserv#wsdl.intf(inf)>;
     wsdl:binding <https://sociddea/fsserv#wsdl.binding(bindSOAP)>;
     wsdl:service <https://sociddea/fsserv#wsdl.srv(storeService)>.
...
```

Sociddea Logged in as File Storage Service

File Storage Service

Basic Information

Maker https://example.org/bob#b

Service Description

Service
Label FileStorageService
Endpoint
Address http://example.org-3078/FileStorageService.asmx
Uses Binding
Label bindingSOAP
Binding Operation https://example.org/identities/fss#wsdl.bindingOper
Protokoll http://www.w3.org/2003/05/soap/bindings/HTTP/
Version 1.2
Implements

Fig. 2. Usage scenario of photo management system showing WAMplus in practice

As an example, Alice wants to use this photo service to create a new album, as illustrated by Fig. 2. Her photos are stored on the file service, which is operated by Bob. When deploying the file service, Bob also published the service description as a WebID profile using Sociddea. He uses the provided graphical user interface to create a new profile and to add the WSDL RDF description. All following steps are executed during the run-time. Requesting the description of the file service through the photo service invokes the WebID authentication process of Sociddea. In the Sociddea system, every request to a resource has to pass the WAC-type access control rules. Assuming that WAC only allows authenticated users to access profile resources, the photo service has to authenticate with its own WebID. Having access to the profile, its WSDL RDF description provides all necessary information for addressing and binding the service to interact with it. Consistent linking between the resources is established by the use of predicates like *foaf:maker* or *foaf:knows*. If the photo service encounters a problem, Alice or other services could retrieve the operator's contact data by following the WebID URI to Bob's profile being linked to in the WebID profile of the file Web service.

Demonstration. For a live demo and further information about Sociddea and WAMplus visit: `http://vsr.informatik.tu-chemnitz.de/demo/sociddea/`

4 Conclusion

Combining semantic description, universal identification, and access control with WAM's modeling capabilities, WAMplus contributes to designing and managing secure distributed Web applications and services. Our approach assists Web engineers in modeling SOA-based Web applications by creating machine-readable big pictures of their architecture, by using WebID to identify and describe Web services with WSDL+RDF, and by protecting resources with WAC and fine-grained filters. In future work we intend to apply the WAMplus approach in more scenarios to discover patterns relevant to the evolution of Web applications. There we will research the topic of dynamic service replacement and delegation.

References

1. El Maliki, T., et al.: A Survey Of User-centric Identity Management Technologies. In: The International Conference on Emerging Security Information, Systems, and Technologies, SecureWare 2007, pp. 12–17. IEEE (2007)
2. Kim, C.S., et al.: Building semantic ontologies for RESTful web services. In: CISIM, pp. 383–386 (2010)
3. Koch, N., et al.: Model-driven Web Engineering. Upgrade 9(2), 40–45 (2008)
4. Meinecke, J., et al.: Enabling Architecture Changes in Distributed Web-Applications. In: Web Conference, LA-WEB 2007, pp. 92–99. IEEE (2007)
5. Papazoglou, M.P., et al.: Service-Oriented Computing: State of the Art and Research Challenges. IEEE Computer 40(11), 38–45 (2007)

6. Saleem, M.Q., et al.: Model Driven Security Frameworks for Addressing Security Problems of Service Oriented Architecture. In: ITSim, vol. 3, pp. 1341–1346. IEEE (2010)
7. Wild, S., Chudnovskyy, O., Heil, S., Gaedke, M.: Protecting User Profile Data in WebID-Based Social Networks Through Fine-Grained Filtering. In: Sheng, Q.Z., Kjeldskov, J. (eds.) ICWE Workshops 2013. LNCS, vol. 8295, pp. 269–280. Springer, Heidelberg (2013)
8. Wild, S., Gaedke, M.: Utilizing Architecture Models for Secure Distributed Web Applications and Services. Information Technology Special Issue on Architecture of Web Applications (Accepted Journal Paper, Published Q2/2014)