Scope-Aware Delegations in Distributed Social Networks

Anna Scholtz(✉), Stefan Wild, and Martin Gaedke

Technische Universität Chemnitz, Chemnitz, Germany
{anna.scholtz,stefan.wild,martin.gaedke}@informatik.tu-chemnitz.de

Abstract. Swiftly meeting challenges by distributing tasks to the most suitable knowledge workers is an important matter, especially for network-centric organizations. In such distributed work environments delegations help to solve tasks faster, but also require measures to prevent delegatees from exceeding assigned competencies. By providing universal identification, WebID by W3C can assist in establishing a basis for distributed collaboration. Yet, it does not allow users for delegating access rights to others in a controlled way to act on their behalves. This paper presents the DASC approach to enable scope-aware delegations in distributed social networks using WebID. We introduce a vocabulary to describe delegations including associated constraints and demonstrate a prototypical implementation of DASC within an existing WebID identity provider.

Keywords: Security · Delegation · Semantic web · Social web · WebID

1 Introduction

Network-centric organizations have been rapidly growing in number since the advent of the 21st century [1]. As things are going increasingly faster in our globalized world, it is vital to meet challenges in time by breaking complex activities down into smaller, more manageable tasks and swiftly distribute them to the knowledge workers who are best-suited for carrying them out. Enabling to organize loosely-coupled teams without implying further system dependencies, distributed social networks are well-suited for the knowledge work domain.

A distributed online social network can be implemented using W3C's WebID specification [2]. With WebID, users get a globally accessible and platform-independent identity they can use for authentication and building connections. Therefore, WebID employs three artifacts: A user, like Alice, authenticates herself using a *WebID certificate*. It contains her *WebID URI* and her public key. This *WebID URI* represents a specific identity of user Alice and refers to a *WebID profile* that stores her identity data in a machine-readable way as RDF triples.

Despite the benefits of delegation, it is not an intrinsic part of WebID yet. To prevent delegatees, like Bob, from improperly using profile data and access rights, such WebID extension would need to be aware of the delegation scope defined by a delegator, e.g., restricting the delegation to a specific service and time period.

P. Cimiano et al. (Eds.): ICWE 2015, LNCS 9114, pp. 709–712, 2015.
DOI: 10.1007/978-3-319-19890-3_63

Based on the conceptual foundation for preventing exploitation of personal user data established in [4], this paper discusses how to realize such security feature in practice. We provide three main contributions: First, a process model and semantic vocabulary that specify delegations with associated constraints to define the delegatee's scope, called DASC. Second, a proof-of-concept implementation of DASC and third, the integration into an existing WebID identity provider.

The paper is organized as follows: We discuss related work in Section 2, present and demonstrate DASC in Section 3 and conclude the paper in Section 4.

2 Related Work

While delegation in a closed context is not a new extension to existing identity systems, scope-aware on-behalf-of delegation for distributed social networks is.

OAuth (`http://tools.ietf.org/html/rfc6749`) is a widely used open standard to authorization. It enables users to grant third-party services access to their personal resources, with a *Resource Owner* delegating rights to a *Client* requesting access to a protected resource. While OAuth facilitates restricting the *Client*'s scope of action, it does not directly integrate with existing authentication routines.

XML-based languages such as SAML and XACML allow for detailed specification of authorization and delegation aspects, yet they lack semantic features including high expressiveness, self-descriptiveness and machine interpretability.

Tramp et al., discuss another approach for delegation based on WebID in [3], which distinguishes involved agents by their roles as *secretary* (delegatee) and *principal* (delegator). A WebID URI stored in the principal's WebID profile denotes the secretary, which in turn adds to each HTTP request issued on the principal's behalf the `X-On-Behalf-Of` HTTP header field containing the principal's WebID URI. Extending each HTTP request in such way does not only increase complexity, but also decrease interoperability and applicability.

3 Scope-Aware Delegations Through DASC

Devising DASC, an approach for Delegations Aware of SCope, we aimed at making as few changes as possible to the original WebID identification and authentication mechanism [2]. The conceptual model of DASC defines a delegation as a set consisting of the delegatee's WebID URI, a task to be fulfilled and constraints to determine the scope of action as a whitelist. A delegation-enabled WebID certificate is issued to delegatee Bob. It enables him to authenticate to services and act on Alice's behalf. To identify the real subject that is using the service, such certificate contains both of their WebID URIs. When delegatee Bob authenticates to a service using this certificate, the service can retrieve delegator Alice's WebID profile, which describes the delegation by referring to delegatee Bob by his WebID URI and by specifying other parameters, like constraints.

Transforming the conceptual into a physical model, the delegation-enabled WebID certificate is an X.509 client certificate that contains the *Subject Alternative Name* (SAN) denoting delegatee Bob and the *Issuer Alternative Name*

(IAN) denoting Alice as delegator. To define scope-aware delegations, we extend the delegator's WebID profile by specific RDF statements: The `delegatee` is represented by his WebID URI. The `task` is a URI pointing to the description of the work to be done. Delegators can define constraints, e.g., regarding `domain` or `validity`, from an extensible set. A thus specified delegation is shown below:

```
<WebID URI delegator> sociddea:delegate [
  sociddea:delegatee <WebID URI delegatee>;
  sociddea:task <URI to task description>;
  sociddea:delegationConstraints [
    sociddea:delegationValidity DEADLINE;
    sociddea:delegationDomain SERVICE; ] ]
```

Listing 1.1. Structure of the delegation parameters

As we cannot prevent attackers from creating a delegation-enabled WebID certificate on their own, it is mandatory to include additional protective measures in the delegator's WebID profile and verify them whenever required. Assuming an attacker creates a WebID certificate containing his WebID URI as SAN and Alice's WebID URI as IAN, he would not be enabled to authenticate to a service as Alice's delegatee. This is because the delegation parameters of Alice's WebID profile, retrieved by the service, do not contain the attacker's WebID URI.

To illustrate our argumentation, Figure 2 depicts the delegation process as BPMN. Figure 1 shows DASC integrated into Sociddea - a WebID identity provider. On the left in Figure 1 Alice's view is featured when creating a delegation whereas the right part shows Bob's view on her WebID profile with delegations only concerning him. The numbers in Figure 2 correlate with those in Figure 1.

When Alice initiates the delegation, she has to describe the delegation parameters and optionally define constraints of domain and validity (cf. ①). After these parameters are added to her WebID profile (cf. ②), she can inform Bob. When Bob

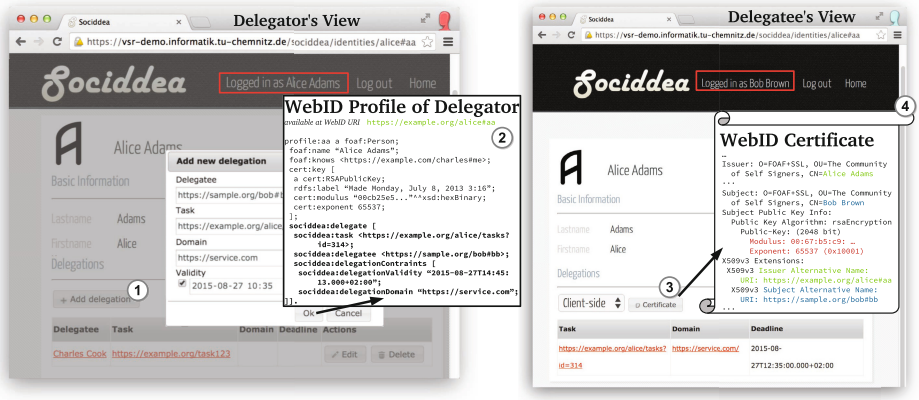

Fig. 1. Delegation creation in Sociddea – Delegator: ① Defining delegation parameters and ② storing in WebID Profile. Delegatee: ③ Viewing delegations on the delegator's WebID profile and ④ creating a delegation-enabled WebID certificate.

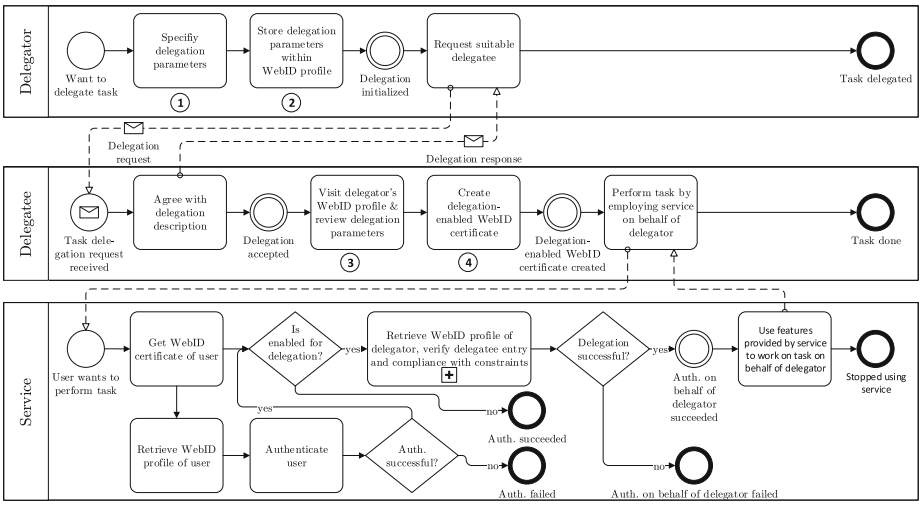

Fig. 2. Delegation process as business process model

visits Alice's WebID profile, he sees relevant delegations only concerning him (cf. ③). Once he decides to work on one task, he creates a delegation-enabled WebID certificate (cf. ④). He can then use this certificate to authenticate to a service that supports the delegation approach. The service requests Bob's and Alice's WebID profile to check the delegation constraints in Alice's WebID profile.

Further information about DASC and Sociddea, including screencasts and a demo, is available at: https://vsr.informatik.tu-chemnitz.de/demos/sociddea

4 Conclusion

This paper presented DASC, an approach for scope-aware delegations in distributed social networks using WebID. We introduced a vocabulary to specify delegations, modeled the DASC delegation process using BPMN, and showed a prototypical implementation of DASC integrated into a WebID identity provider.

Future work will focus on delegating a task to a group of subjects and to provide more precise restrictions, further analysis is needed on the constraints.

References

1. Abrams, R.S.: Uncovering the Network-Centric Organization (2009)
2. Story, H.: WebID Specifications (2014). http://www.w3.org/2005/Incubator/webid/spec/
3. Tramp, S., Story, H., et al.: Extending the WebID protocol with access delegation. In: COLD2012 (2012)
4. Wild, S., et al.: ProProtect3: An Approach for Protecting User Profile Data from Disclosure, Tampering, and Improper Use in the Context of WebID. TLDKS **19**, 87–127 (2015)