J. Meinecke, M. Gaedke, F. Majer, A. Brändle; Capturing the Essentials of Federated Systems; Proceedings of the Fourteenth International World Wide Web Conference (WWW), Pages 895-896, Edinburgh, UK, 23-26 May, 2006.

Capturing the Essentials of Federated Systems

Johannes Meinecke[1] Martin Gaedke[1] Frederic Majer[1] Alexander Brändle[2]

[1]University of Karlsruhe
Engesserstr. 4
76128 Karlsruhe, Germany
+49 (721) 608-8076
{meinecke, gaedke, majer}@tm.uka.de

[2]Microsoft Research Cambridge
7 JJ Thomson Avenue
CB3 0FB Cambridge, United Kingdom
+44 (1223) 479-0
alexbr@microsoft.com

ABSTRACT
Today, the Web is increasingly used as a platform for distributed services, which transcend organizational boundaries to form federated applications. Consequently, there is a growing interest in the architectural aspect of Web-based systems, i.e. the composition of the overall solution into individual Web applications and Web services from different parties. The design and evolution of federated systems calls for models that give an overview of the structural as well as trust-specific composition and reflect the technical details of the various accesses. We introduce the WebComposition Architecture Model (WAM) as an overall modeling approach tailored to aspects of highly distributed systems with federation as an integral factor.

Categories and Subject Descriptors
D.2.11 [**Software Engineering**]: Software Architectures – *Languages*; H.3.4 [**Information storage and retrieval**]: Systems and Software – *Distributed systems*

General Terms
Management, Design, Security

Keywords
Security, Web Services, Architecture, Modeling, Federation

1. INTRODUCTION
Many complex tasks to be solved by technological disciplines arise from the needs of modern businesses that operate world-wide, cooperate with various partners, and deliver their services in real-time. Over the years, this trend of connected businesses has affected the Web and driven its transformation into a platform for distributed applications. Beyond offering means for merely supplying documents to users, it has recently been used as a communication infrastructure that links together applications, e.g. by exposing functionality through Web services. Now, a tendency can be observed towards a new class of applications: the federated portal respectively 4th generation portal [2]. Relationships within such federations of portals belonging to multiple organizations do not only consist of simple HTML links, but comprise the connection of the portal backbones to share for example data, functionality, or user accounts. In such federated solutions, the question of access control and security is especially important and requires the application of advanced concepts [1]. This is related to the circumstance that external users from cooperating organizations have to be granted access to local resources while preserving the autonomy of the individual federation partners. Consequently, the design and evolution of such systems calls for models that give an overview of the federation structure and reflect the technical details of the various accesses. At the same time, the dynamic nature of Web-based federations requires approaches that go beyond merely supplying one-time static descriptions in favor of dynamic machine-readable representations. Current Web Engineering approaches provide design methologies for dealing with various aspects of Web applications, including navigation, interaction and business processes. If architectural aspects are covered at all, as e.g. in the case of WebSA [3], then the focus lies on the structure of individual Web applications, while security and federated identities are treated outside the models. To complement the state of the art to this end, we propose the WebComposition Architecture Model (WAM).

Copyright is held by the author/owner(s).
WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

2. THE WAM MODELING APPROACH
In order to implicitly include security aspects, WAM adopts general concepts that also form the foundation for the state of the art of federated identity and access management protocols, like WS-Federation, SAML and the Liberty Alliance project. Central to the approach is the concept of the *security token*, which generally takes the form of a digitally signed XML document that contains security-relevant statements to be exchanged between different zones of control (so-called *security realms*). The statements are usually either a proof of identity (e.g. of a user that is authenticated with the help of an *identity provider* / IP) or a proof of privileges (like e.g. the set of roles belonging to a user). They are acquired by calling specialized Web services that in turn may demand other tokens or forms of authentication as a pre-condition. The stated proofs then provide a basis for access control decisions for the protection of Web services and Web applications. As a major advantage, this allows for authentication and authorization tasks to be distributed and delegated to individual system parts as needed.

As a brief introduction to the graphical modeling formalism, Figure 1 contains an example model, describing the design of two federated Web portals *Portal1* and *Portal2*. The autonomous organizations controlling the portals are represented by the two security realms *R1* and *R2*. Both portals use Web services as sources for their provided content, with *Portal1* also integrating the functionality of a service from the other realm. This is enabled by an explicit trust relationship running in the opposite direction, reflecting an agreement that has been established between the two organizations in advance. The realm-crossing invocation is further

characterized by a *profile* statement written in UML-OCL that specifies details about the security measurements applied to the exchanged SOAP messages. As an additional form of cooperation, this setup also allows users that have already logged in at the identity provider of realm *R1* to perform tasks at the portal in realm *R2* without any additional authentication steps (single sign on).

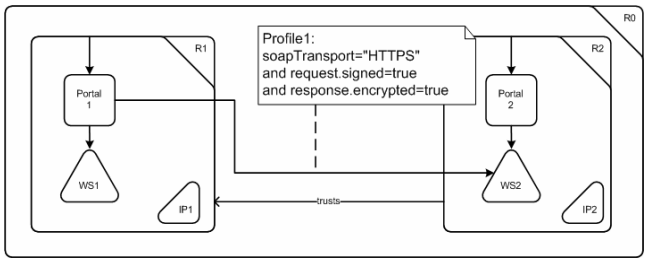

Figure 1: WAM example model of a federated portal scenario

3. WAM-XML MODELLING LANGUAGE

The graphical notation of WAM focuses on diagrams that are relatively easy to draw and comprehend. While the diagrams form the basis for communicating the models between stakeholders (e.g. during the design process), later at operation time, the changes that inevitably affect the evolving system cause a gap between the modeled world and the model. Although UML-OCL expressions help humans to add system details that are also relevant for the implementation at runtime (e.g. protocol restrictions), their complex syntax render them inappropriate for further machine-based processing and automation. Instead, we designed a language that is capable of describing the modeling concepts of WAM, and at the same time forming the basis for tools and support systems. The language (called WAM-XML) is implemented as an application of the XML, and as such profits from its manifold capabilities, especially in the context of large-scale heterogeneous systems. In order to define the exact notation, we designed an XML schema with the goal to incorporate existing XML-based specifications where appropriate to improve the overall interoperability and the applicability of standard tools. For example, WAM-XML includes a metadata concept that is based on the Dublin Core Metadata Initiative, a de-facto standard defining a set of common meta-level attributes. Additionally, the realization of the WAM relationship concept makes use of the XLink specification that describes standardized ways of linking resources in XML documents.

As a demonstration of how the schema types are instantiated in WAM-XML documents, we show an extract from an example model in the code listing below. With reference to the example diagram above, the document body contains elements representing a portal *P1* and a Web service *WS2*. As in the diagram, the entities are related via an invocation relationship. According to the XLink specification, the resources to be connected are not addressed directly, as e.g. by an identifier. Instead, the *from* and *to* attributes of the arc-element *invocation* contain XLink labels that serve as placeholders for separately declared groups of WAM entities. The mapping is achieved with an XPointer expression that in this case uses the identifier introduced by the Dublin Core metadata concept as a unique key. The flexibility of the XPointer addressing scheme has the advantage that the same relationship can have multiple origins and destinations (n:m relationships), e.g. allowing for a concise statement of the fact that all realms in a model trust a dedicated central realm. As the addressed resources may reside in different XML files, this also enables distributing the model on multiple documents, possibly owned by different federation partners. The profile describing the invocation details is referenced with a URI to be resolved outside the WAM document.

```
<wam:Application xmlns:wam="http://wsls.net…">
  <dc:Identifier xmlns:dc="http://purl.org…">
    http://mwrg.tm.uka.de/portal1
  </dc:Identifier>
  <dc:Title>Portal1</dc:Title>
  ...
</wam:Application>

<wam:Service>
  <dc:Identifier>
    http://mwrg.tm.uka.de/ws2
  </dc:Identifier>
  <dc:Title>WS2</dc:Title>
  ...
</wam:Service>

<wam:Invocation xlink:from="P1" xlink:to="WS2"
  xmlns:xlink="http://www.w3…">
  <core:Profile>uri:mwrg:profile1</core:Profile>
</wam:Invocation>

<core:Selector xlink:label="P1"
  xlink:href="xpointer(/core:Model/core:Body/*
  [dc:Identifier='http:...'])"/>

<core:Selector xlink:label="WS2"
  xlink:href="xpointer(/core:Model/core:Body/*
  [dc:Identifier='http:...'])"/>
```

While the XML format is not necessarily intuitive to write and read by humans, it provides a solid basis for applications that process the modeling information at operation time. Models can be transformed from WAM into WAM-XML and vice versa due to a similar expressiveness of both representations.

4. WAM INFRASTRUCTURE SERVICE

In order to expose the model and make its machine-readable representation available within the Web-based federation, we implemented a supporting infrastructure, with the *WAM Service* acting as the central component. The main idea behind this was to apply the same technology that is already in use for the functional parts of the architecture and provide a Web service for querying and changing the model. On top of the WAM infrastructure service, some applications have already been implemented that work directly on the provided model information. As a means for supporting model engineers in creating and modifying system descriptions, we customized Microsoft Visio with dedicated support for WAM diagrams. While the model is being updated, an automatically generated RSS-feed aids stakeholders in keeping track of the ongoing changes within the federation. The diagram authoring tool and other related infrastructure components can be downloaded at http://mwrg.tm.uni-karlsruhe.de/downloadcenter/.

5. REFERENCES

[1] Cameron, K., The Laws of Identity - MSDN Article (2005): http://msdn.microsoft.com/library/en-us/dnwebsrv/html/lawsofidentity.asp (29.03.2006).

[2] Gootzit, D. and Phifer, G., Gen-4 Portal Functionality: From Unification to Federation, Gartner SPA-20-7217. 2003: Stamford, CT.

[3] Meliá, S. and Cachero, C. An MDA Approach for the Development of Web Applications. in 4th International Conference of Web Engineering (ICWE 2004). 2004. Munich, Germany. p. 303-305.