

Flexible Services

EDEN

Phase 2 Technology studies D1.2.4

Document Name:	Phase 2 Technology studies D1.2.4
Project/WP Title:	EDEN/WP1
Document Type, Security	P (Public)

Document Title:	Phase 2 Technology studies D1.2.4
Agreed date of delivery	
Actual date of delivery	30.9.2010
Editor	Sanna Suoranta
Version and	version 1.0
Date Last Change	
File:	Phase 2 Technology studies D1.2.4.doc

Participants	Name	e-mail
Aalto	Sanna Suoranta Laura Marcia Villalba Monné Pihla Maria Viitanen Sandeep Tamrakar Pekka Silvekoski Markku Laine André Palas de Andrade	
Elisa	Ilpo Mäntykangas Olavi Karasti	
TeliaSonera	Olli Jussila	

1 Table of contents

1	Table of contents.....	2
2	List of Figures	3
3	Introduction	4
4	Identity and Access Management (IAM) Technology Studies	5
4.1	IAM Architecture for Service Ecosystems	5
4.1.1	Federation Technologies	7
4.1.2	Access Control Technologies	8
4.2	IAM Pilots in EDEN	9
4.2.1	OtaSizzle - Authentication and Single Sign-on	9
4.2.2	LUCRE's uSpace	10
4.2.3	OpenID for XFormDB	10
4.2.4	Authentication Session Migration	11
4.3	Conclusions	12
5	DM Technology Pilot.....	13
5.1	Overview.....	13
5.2	Use Case	13
5.3	Setup	14
5.3.1	Portal Implementation	14
5.3.2	Backend	15
5.4	Security Aspects	16
5.5	Relationship of the stakeholders	16
5.6	Conclusions	16

2 List of Figures

Figure 1 Federated service ecosystem environment.....	6
Figure 2. EDEN Device Management architecture.....	14
Figure 3. Screen capture of DM demo portal	15

3 Introduction

This deliverable presents key technological enablers found in Flexible Services program. There are number of enablers which are important for different pilots but here there are presented those which are the most interesting for different pilots in Flexible Services program.

There are two enablers presented: Identity and Access Management and Device Management

4 Identity and Access Management (IAM) Technology Studies

When all services are moving into the online environment, also supporting services such as user authentication must be implemented online. Often services use usernames and passwords for authentication, but the actual authentication process is only part of Identity and Access Management (IAM) system. Roughly, it has two parts: management of **digital identity**: creating a digital identity, verifying the corresponding real world identity, establishing required credentials, handling of lost credentials, removing the digital identity etc., and

access management: checking that the holder of the digital identity has right to access a service or a resource. There are several techniques for authentication (verifying the identity) and authorization (assigning access rights for a subject). Some techniques provide stronger and better-verified proof of clients' identity than others. For example, above-mentioned passwords are a weak authentication method.

In the EDEN technology pilots, aspects of IAM are studied in service ecosystems. Example target services were the OtaSizzle [Sisslelab, 2010] social media service that allows users to discuss and change services and sell things and LUCRE's uSpace [2010] that is another social media service where user can gather widgets that can share information. These kinds of service ecosystems benefit of using single sign-on (SSO) technologies such as OpenID [2010]. Moreover, often users are using the services with different devices and seamlessly changing the device even during the service usage should be possible. Some services may need strong authentication that can be provided by mobile phone technologies. The EDEN IAM pilots concentrate mainly on access management and leave the management of digital identity out of deep investigations. Before presenting the technology pilots, we discuss the IAM architecture for service ecosystems next.

4.1 IAM Architecture for Service Ecosystems

In a service ecosystem, identity and access management can be implemented as a separated service that other services use as is depicted in Figure 1. Services can have different needs: sometimes identity of a client must be verified reliably and sometimes it is enough to know that the client is same as before. Moreover, all services or users do not necessarily trust to the same identity provider. Thus, the ecosystem can have several IAM services that can provide either weak or strong (or both) authentication verification of clients, and clients and services can choose one of the identity services that both find trustworthy. In addition to the identity providers that take care of the identity management, a service ecosystem needs a component that act as a common service for all services that needs authenticate the users, especially if the ecosystem's users can create their own services. This service can be part of the service ecosystem composer.

Service Ecosystems are challenging environments for a common IAM service not only because of the different levels of authentication requirements but because, for example, services can use other services on behalf of a client, and a clients would like to authenticate themselves for all the services once. Single sign-on (SSO) provide

means to access several services with one identity verification process. There are several technologies and products that offer SSO. However, there is no possibility to find one single SSO solution that suits all situations. Especially hard is to define the level of authentication verification in a situation where a client first uses a service that requires strong authentication and then another service that requires weak authentication or vice versa.

In the service ecosystem, service providers and identity providers form federations where the entities trust each other. Agreements in the federation define used identity federation technologies and protocols, details of authentication levels, establishment of identities e.g. how to verify user's authentication etc. Depending on the chosen federation technology, the sessions between user and service provider, and session between user and identity provider are managed as is agreed. A federation can have several identity providers and several service providers.

In Figure 1 there is federated service ecosystem. In that picture dashed arrows are authentication sessions and full arrows are service sessions between user's device and other parties of the federation.

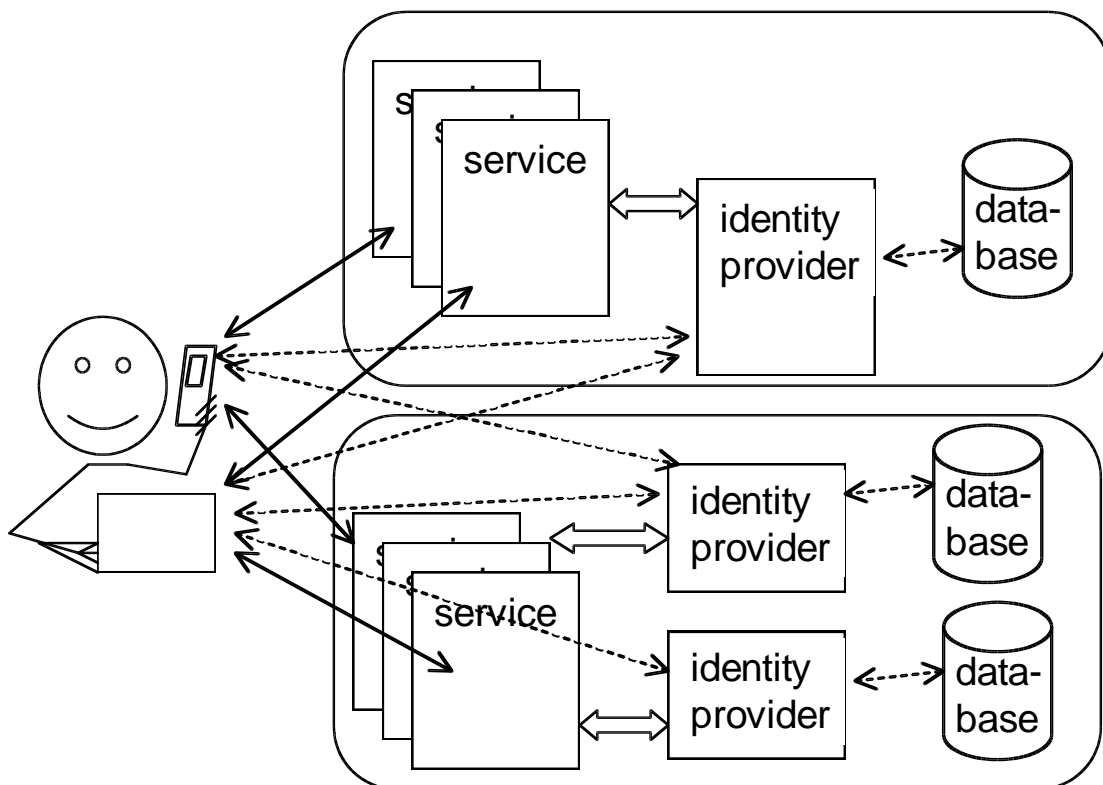


Figure 1 Federated service ecosystem environment.

Identity provider manages users' identity information and can offer different methods for authentication. User identities are created and her true-life identity is verified as is agreed on the federation. If the user forgets her password or loses her credentials, she contacts directly to the identity provider or service provider redirects her to it

since IdP manages the identity. When a user is logging in to a service, IdP can send information about the used authentication method to the service provider together with user's other information if the service provider needs the information for making the access control decision. IdP keeps information if the user has already authenticated or not - this information can be stored on user's side in form of a credential e.g. session cookie in the user's web browser. This means that IdP and the client have a session. When the user accesses another service, re-authentication is not needed. Usually IdP does not keep track of the services that the authentication information is sent. This means that single logout cannot be implemented since there is no central entity keeping track of all services where the user has logged in. Logging out from the identity provider requires closing the web browser or deleting the session cookies.

Service provider gets information, for example user's name, affiliation, role, the used authentication method, and if the authentication was successful, from the identity provider. Based on the information, the service provider either grants or denies the user's accessing the service, and correct view on the service is shown to the user (e.g. role of student or teacher). If the access is granted, session between service provider and the user is formed and it can be stored as a session cookie in the user's web browser's memory. The connection with the service provider can be closed if the session expires or the user logs out from the service. Often the service session has shorter validity than the identity session.

From the users point of view, federation offers SSO for a group of services and separates authentication from services. Often a service allows user to choose a suitable one from a list of identity providers. Then, the identity provider verifies user's identity (if asked, using the requested authentication method). Some federation techniques shows the attributes that are sent to the service provider to the user that can double-check if she want to give the attributes to the service (of course, without the permission, the authentication usually fails). If the user wants to use another service in the same federation, the service forwards the user to the chosen IdP (or to the list of IdPs if several are accepted) as in the first service request, but IdP does not require re-authentication since the authentication session is still valid. IdP directly informs the service provider that the user is authenticated (it may show the attributes to the user again since different services can require different attributes). From the users point of view, separating the actual service from identity service can be either confusing ("why am I now on a different service's website") or trust-inspiring ("I trust more to PayPal than to this new online store").

4.1.1 Federation Technologies

IAM can be implemented using centralized system where all services trust to one identity provider. Traditionally this kind of IAM is used in an enterprise that has its own user database against which the users are authenticated and to which all the services can trust. For example, the Kerberos technology[Neuman, 2005] is used for centralized identity management. Federated identity management system offers similar service but it allows use of several identity providers. Service providers have to agree which identity providers they trust to authenticate users on their behalf. If the service accepts several identity providers, the user can choose which identity

provider she uses among them. Identity providers can use different means to authenticate the users and provide attributes for the service provider to define what kind of authorization (access rights) the user has in the service. There are several federation technologies that are shortly introduced next.

Shibboleth [Internet2, 2010] is SAML [Oasis, 2008] based technology that allows users to log in to several web services after authentication. Shibboleth is developed by Internet2 and it is freely available. For example, the universities in Finland form the HAKA federation [CSC, 2010]: each university has its own identity provider and both common and own services. Example of a common service is the Nelli portal that offers access to all scientific databases and example of an own service is the Noppa portal that offers web pages to all TKK's courses. CSC takes care of metadata information that tells which SPs accepts which IdPs and what information is needed for access control decisions in the services. Metadata is available as an XML file in CSC's webpages for all services in the federation. Similar federation is under construction for governmental use (it is called Virtu). The federation agreement defines how user's identity is established and how the authentication is done. Both of the HAKa and Virtu federations use the Shibboleth technology. Similarly than Shibboleth, Liberty Alliance Project [2010] use SAML in the messages that service providers and identity providers exchange in the circle of trust they form, and its specifications are also openly available.

OpenID [2010] is an open technology for federated identity management. Anyone can establish an identity provider but, of course, the service providers define which identity provider they accepts to authenticate their users. There exist many OpenID identity providers, for example Google. The identity providers can have different means for establishing user's digital identity and authenticating user, but mainly working email address and password based authentication is used. OpenID does not provide any attributes, only that the user possesses an OpenID identifier (an URL form username). Another technology, OAuth, complements OpenID but it is distinct. It is an open standard for social media services to share user's private content, attributes such as e.g. photos and contact lists, among the services. For example, Facebook uses OAuth.

In addition to open technologies, several vendor specific solutions for federated identity management in an organization are available. IBM Tivoli [IBM, 2010] provides identity and access management and SSO using centralized user access control with chosen authentication method. Novell Identity and Access Management [Novell, 2010] has also a centralized database that integrates information of service specific databases, and access to information can be given based on the role of a user in this centralized access control service. Also others, for example CA Technologies [CA, 2010] and Oracle [2010], have their own IAM products.

4.1.2 Access Control Technologies

Federated identity management defines ways for services to communicate with identity providers. Often the identity provider itself can define what kind of authentication verification methods it uses. Authentication can be based on something known (password, PIN), something possessed (one time password list or

device), or something biometrical of the user. When at least two of the above mentioned are used together, authentication method is considered to be a strong one, otherwise the authentication method is weak. Traditionally, each service has had its own user database and the users have been authenticated using passwords.

Most of the OpenID identity providers use weak password based authentication method. Moreover, they often do not even verify user's real world identity, just that the user has a valid email address during the identity creation and that the user posses a URL. On the other hand, the technology itself does not prevent using strong identity. For example in Estonia, mobile phone operator providers OpenId identity provider service.

In Finland, banks offer user identity service for other services. The mechanism is called TUPAS [FK, 2010] and it uses strong authentication method normally used in accessing online bank. For the other services, the bank can verify the user's attribute such as identity number. However, TUPAS does not provide single sign on since all services must separately ask the user verification for themselves because the bank does not form any authentication session with the user.

Publik Key Infrastructure (PKI) is often suggested to provide identity verification in various services and protocols. In PKI, a trusted certification authority (CA) verifies the identity of an entity, e.g. user or a service, and creates a certificate for it. The certificate consists of a public key of the entity, validity period, and it is digitally signed by the CA. The entity can proof her identity with the certificate to other entities that trust to the same CA, and the communicating peers can use the public keys to create secure channel for their communication. There exits many implementations with different kind of certificate authority hierarchies and the details of the certificates varies, too. PKI is mostly used in secure web services that use HTTPS, but the problem is that really trustworthy common CAs does not exist. Next, we discuss how to create PKI based IAM mechanism using mobile phones.

4.2 IAM Pilots in EDEN

4.2.1 OtaSizzle - Authentication and Single Sign-on

OtaSizzle is creating "an open environment for mobile social media services" firstly for students and staff of Aalto University in Otaniemi [OtaSizzle, 2010]. Currently it has three services: Kassi for exchanging services or loaning equipments, Ossi for mobile chatting and Facebook-like service, and the NordSecMob social study guide service for the students of that degree programme [Sizzlelab, 2010]. Service developers can create their own applications using OtaSizzle's REST API called Aalto Social Interface (ASI) [Aalto, 2010]. ASI mediates data between the social media applications and the backend data structures that include user profiles, friend lists, groups, locations, and more. Moreover, ASI provides user and application specific sessions e.g. who has logged in to the system.

ASI works as a centralized identity provider for all the services of OtaSizzle and offers single sign on (SSO) and single log off. Authentication and SSO is based on Centralized Authentication Service (CAS) protocol developed by Yale University and later maintained by Jasig [2009]. When a user logs in to an OtaSizzle service, she actually logs in to CAS and gets a ticket that is given for the service telling that she is authenticated to use the service. Following the CAS protocol, the service and ASI create an user-specific, application session between them. When a user logs out, the CAS can notify all services that the application session has ended for the user.

While doing centralized ticketing, CAS mediates between client applications and authentication backends. One of these authentication backends can be a Shibboleth [CSC, 2010] identity provider. This means that the user's account is verified against Shibboleth. Thus, all the web SSO capable OtaSizzle services can be used with e.g. TKK's IT centre's service password. Later on, the system can be joined with HAKA (a trust network of Finnish universities using Shibboleth). Moreover, any OpenID enabled OtaSizzle service can use CAS as an OpenID provider. Still, OtaSizzle uses a centralized system and all user authentications despite the authentication method are stored in CAS.

4.2.2 LUCRE's uSpace

LUCRE's uSpace [2010] is a web application, where users can create shared workspaces for collaboration and communication. A user can have several workspaces and they can be shared with other chosen users. Functionality of the workspaces is provided by widgets. In the future, users can create new widgets or modify existing ones either using web based visual Integrated Development Environment called XIDE or coding by hand. The widgets can use information about user's location. uSpace has two modes: editing and normal. In the editing mode, the user can drag and drop widgets into the workspace from a list, and in the normal mode just use the applications in the workspace. Each widget can access only its own data currently. The creator of a collaborative workspace can define which users have what right in each widget of the workspace. Obviously, the workspace benefits user authentication and single sign on technology since widgets often handle user's private information and the workspace consists of several widgets. For example, the LUCRE ecosystem provides group and contact management for all widgets. LUCRE's uSpace uses either own username - password authentication or OtaSizzle accounts described above. New accounts can be created in the first log in, and no verification of identity is required. Finer grade authentication, meaning which widgets can access what information, is done using group management.

4.2.3 OpenID for XFormDB

XFormsDB is XML based technology for creating easily services that use web forms. In his master's thesis, Markku Laine presents proof-of-concepts implementation that integrates common server-side functionalities to the XForms markup language. The implementation allows applications to form sessions for users in the server side either using cookies or with URL rewriting if cookies cannot be used. In URL rewriting, unique session identifier is added to URL of every HTTP request. This also allows multiple simultaneous sessions for a user. The sessions are essential if the service require user authentication. Currently, the user authentication is implemented using

traditional username-password authentication where hashes of the passwords are stored in a database. Another pilot aimed to replace the authentication with external single sign on authentication, namely with OpenID that would allow users to sign once into all applications that are composed together in the service. In her master's thesis, Pihla Viitanen has planned a user interfaces for applications that use OpenID for user authentication. She has taken an existing OpenID API and simplified it for web service developers who do not know security requirements but who want to add user authentication into their applications. The new API hides details of OpenID, and provides simple "log in", "get user information" and "update user information" methods. Unfortunately the work did not reached its goals, integrating OpenID into XFormsDB applications.

More information

Markku Laine. XFormsDB—An XForms-Based Framework for Simplifying Web Application Development, Master's thesis, Aalto University School of Science and Technology, 2010

Pihla Maria Viitanen. Single sign on for digital ecosystem based on XFormsDB language. Master's thesis, Aalto University School of Science and Technology, 2010.

4.2.4 Authentication Session Migration

User authentication should not depend on the device the user is accessing a service which is often the case nowadays. In order to allow seamless service usage, the user should be able to move the active service session from one device to another one without re-authentication. In his master's thesis, Pekka Silvekoski implemented a prototype for client-side authentication session migration where the Shibboleth authentication cookies were moved from laptop to an internet tablet or vice versa in a way that allows user to continue using the service without new authentication. Only authentication session cookie (IdP's cookie) was moved, but the session migration delivered also the service URL to the target device, and the user could then continue using the service. The work was implemented as a plugin for Mozilla Firefox and Fennec browsers. Fennec could not open the given URL, but it is still under development. The only change for the server side is that requirement of using the same IP address for the connection is dropped. Usually, the IP address is used for replay attack protection, but here the IP address changes since the connection is moved from one device (having an IP address) to another one (having another IP address) and thus other replay attack protection mechanisms must be used instead.

More information:

Pekka Silvekoski. Client-side migration of authentication session. Master's thesis, Aalto University School of Science and Technology, 2010.

Sanna Suoranta, Jani Heikkinen, Pekka Silvekoski, Authentication Session Migration, To be appear in proceedings of the NordSec2010 conference, October 28-29, 2010.

4.3 Conclusions

In EDEN project AM has been regarded as one of the most important enablers of digital service ecosystem. Without good working IAM users are not eager to consume services and therefore there will not be an ecosystem of this kind.

In network of actors it is vital that federation of identity is possible and in a way where user is convinced that he is capable to manage his digital identity. This federation is also vital for service providers because there will not be one source where all identities are checked but there has to be federation between entities.

Flexible Services program is not continuing and therefore further necessary studies are not possible to make in this framework. Next steps for further studies include user experience studies and pilots which shows actual bottlenecks when building real ecosystems.

5 DM Technology Pilot

5.1 Overview

Device Management (DM) technology pilot studied device management as enabling technology for the flexible services ecosystem. As technology device management is well known and widely used. The study focused on aspects that are relevant for service ecosystems, namely service integration in a value network and the relationships of different actors in it. In particular it was studied how in a flexible services ecosystem a group of stakeholders having different functional responsibilities can combine their assets to a solution that solves particular end user problem. This problem setting is fundamental issue in service ecosystems and hence relevant study item for the flexible services program.

Approach for the work was pragmatic and concrete. Idea was to build an end-to-end proof of concept around the DM theme. The assumption was that there would be several specific roles and tasks in the value network that are realized through application services and accessed through well defined service interfaces. Together the distinct application services would form a complete functionality targeted at solving a particular end user need. Development was carried out mainly independently by participating companies / universities within the limits of responsibility. Meetings were organized when needed to go through the concept, solve interface related opens issues and for other relevant reasons. After defining the concept the work proceeded mostly very well. There were some delays in setting up the DM server and creation of the API description for the DM backend. Thereafter development proceeded as planned. Proof-of concept has been finalized during spring 2010 and demo shown in the SHOK Summit on April 20th.

5.2 Use Case

Device management technology is used for remote management of mobile terminals. Purpose for the remote management can be e.g. firmware update, provisioning of various parameters, removing sensitive information from lost device and so on. In the EDEN DM Proof-of-Concept the use case was provisioning of WLAN access point parameters. The basic storyline of the demonstration was as follows: owners of WLAN access points, such as universities, cafeterias, kiosks etc, could report their access point parameters to a common database. People who stay in the vicinity of the access point can order access point parameters through a simple web application for easy configuration of the device. Parameters are provided through configuration SMS sent by the DM system. This way it is possible to utilize available free access points and functionalities of WLAN capable terminal without possibly complicated parameter configuration. It is believed that the access point owner benefits from this as people like to spend more time in the coverage area of the WLAN AP, say a cafeteria or a kiosk. Furthermore, it is beneficial also for the network operator as part of the traffic is offloaded to WLAN to decrease the load of the mobile network . Finally it is

beneficial for the end user as his/her service experience is better. The actual goal of the work was, however, to consider broader targets than these i.e. by using this particular case as an example study how services and functionalities can be integrated in a service ecosystem and how DM as an enabling technology fits into this kind of environment.

5.3 Setup

Architecture was built around a 3-party value network, see Figure 2. First, there is the provider of the web application. In the PoC there was only one web application but in real life there could be almost infinite number of such applications, limited only by the capacity and performance constraints of the used HW. Web application is linked with the backend system through service interface defined by the DM provider. This interface contains relevant information, such as MSISDN number, so that the provisioning process can be carried out. DM backend system in its turn is connected with an SMS gateway provided by a wireless network operator. This connection is made using specification provided by the gateway owner. In the EDEN DM PoC Aalto university created the web application using xForms technology. NSN provided RESTful interface specification for the backend system and hosted the DM server. TeliaSonera offered their content gateway as SMS center routing the configuration messages of the DM over the air to the end user's mobile terminal. Beside the WLAN parameter provisioning there was also another RESTful service for uploading and managing WLAN access point information. This was used by Aalto University for loading test WLAN AP information to the access point database that was part of the DM backend system.

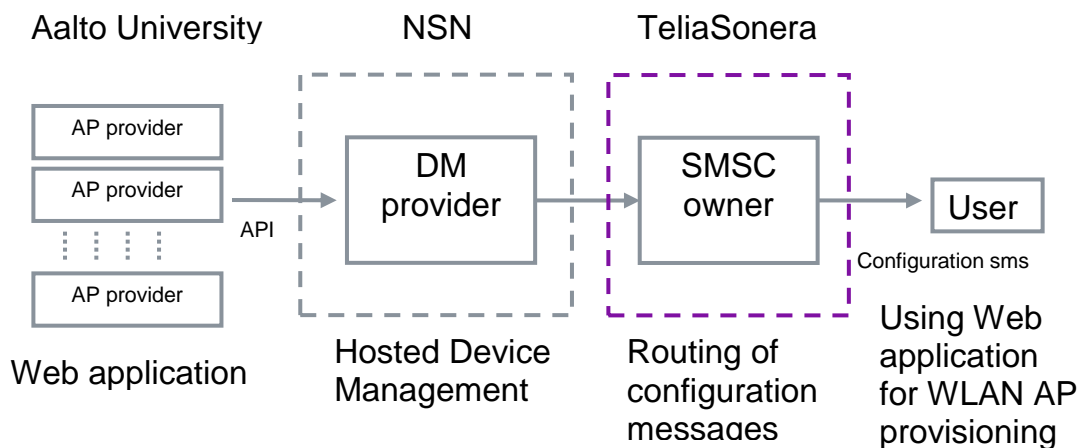


Figure 2. EDEN Device Management architecture

5.3.1 Portal Implementation



Figure 3. Screen capture of DM demo portal

Figure 3 shows a screen capture of Device management Portal. It is available at http://www.tml.tkk.fi/~mplaine/wififinder_english_v2.mov

It shows how user can get parameters easily from selected access point.

5.3.2 Backend

Backend system consisted of Nokia Siemens Networks device management server and some middleware components. API key based authentication for the HTTP messages and IP domain check was done in order to make sure that only authorized applications connect to the service interface. Device management server used UCP protocol to connect to TeliaSonera's content gateway which routed the messages further to a SMS center for transmission.

5.4 Security Aspects

Device management server interacts with end user's terminal. This poses some very relevant security risks which must be taken into account in the architecture. First, it must be ensured that the MSISDN number for which configuration messages are sent is the number of the user who initiated the request. This is because a malicious user might want to send spam messages to any valid MSISDN number or try to hack into the DM system. Therefore, the user of the web application must be authenticated. In addition, it must be controlled which applications can access the API as the authentication of the user is under the responsibility of the web application. For demonstration purposes basic authentication was used but it is believed that in a fully open system security aspects need to be considered with even more carefully, meaning strong authentication and further control mechanisms in the backend.

5.5 Relationship of the stakeholders

Creating shared responsibility to provide solution bringing value for the end user requires agreements between the stakeholders on the offered service, service level, monetary compensation, liabilities etc. This is common business-to-business integration problem that by nature is essential also for the flexible services ecosystem. As a starting point, in commercial system we would need clearly defined compensation agreement, liabilities between the parties and likely auditing procedure for the web application as malicious usage of the service must be prevented as far as practical. These are issues to be considered if some party sees it feasible to start this kind commercial service.

5.6 Conclusions

Device Management technology pilot studied how DM would work as enabling technology in flexible services ecosystem. The integration problem as such is similar to B2B integration but the nature of service ecosystem and involvement of end users as part of the process make security and usability aspect even more important than before. During the project the basic integration of the demonstration proceeded without any bigger problems, although in a commercial system a more complex implementation would be clearly needed. In addition, issues related to compensation, responsibilities and liabilities, as mentioned above, would require thorough study. This part was left out from the current project due to limited resourcing.

References

- Aalto Social Interface. Getting Started Tutorial. <http://cos.sizl.org/doc/tutorial> 2010 [Referred 15.6.2010]
- CA Technologies. CA SiteMinder. <http://www.ca.com/us/internet-access-control.aspx> [Referred 15.6.2010]

CSC. Haka-käyttäjätunnistusjärjestelmä. <http://www.csc.fi/hallinto/haka> [Referred 15.6.2010]

FK, Finanssialan Keskusliitto. Palvelukuvaukset: Tietoturva ja asiakasyhteydet. (TUPAS) http://www.fkl.fi/www/page/fk_www_3830 18.5.2010 [Referred 15.6.2010]

IBM. Tivoli Federated Identity Manager. <http://www-01.ibm.com/software/tivoli/products/federated-identity-mgr/> [Referred 15.6.2010]

Internet2. Shibboleth. <http://shibboleth.internet2.edu/> 2010 [Referred 15.6.2010]

Jasig. Central Authentication Service (CAS). <http://www.jasig.org/cas>. 2009. [Referred 15.6.2010]

Liberty Alliance Project. The Liberty Alliance. <http://www.projectliberty.org/> [Referred 16.5.2010]

Neuman, C., Yu, T., Hartman, S., Raeburn K. The Kerberos Network Authentication Service (V5) FC 4120, July 2005

Novell. Novell Access Manager. <http://www.novell.com/products/accessmanager/> [Referred 15.6.2010]

OASIS. Security Assertion Markup Language (SAML) 2.0 Technical Overview. <http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf> 25.3.2008. [Referred 15.6.2010]

OpenID Foundation. OpenID is a safe, faster, and easier way to log in to web sites. <http://openid.net/> [Referred 15.6.2010]

Oracle. Oracle Identity Management. <http://www.oracle.com/us/products/middleware/identity-management/index.html> [Referred 15.6.2010]

OtaSizzle. Ubiquitous Social Media for Urban Communities. <http://mide.tkk.fi/en/OtaSizzle> [Referred 15.6.2010]

Sizzlelab.org. OtaSizzle. <http://sizl.org/> [Referred 15.6.2010]

uSpace. Welcome to the uSpace demo. <http://testbed.tml.hut.fi/ospace/login> [Referred 15.6.2010]

Master's thesis:

André Palas de Andrade. Strong authentication using mobile phone. Master's thesis, Aalto University School of Science and Technology, under work 2010.

Markku Laine. XFormsDB—An XForms-Based Framework for Simplifying Web Application Development, Master's thesis, Aalto University School of Science and Technology, 2010

Pekka Silvekoski. Client-side migration of authentication session. Master's thesis, Aalto University School of Science and Technology, 2010.

Sandeep Tamrakar. Phone as a smart card. Master's thesis, Aalto University School of Science and Technology, under work, 2010.

Pihla Maria Viitanen. Single sign on for digital ecosystem based on XFormsDB language. Master's thesis, Aalto University School of Science and Technology, 2010.

Laura Marcia Villalba Monné, Remote credential management tool for an On-board credential application, Master's thesis, Aalto University School of Science and Technology, under work, 2010.