A federation of web services for Danish health care

Esben Dalsgaard
Chair, SOSI steering committee
Digital Health Denmark (SDSD)
Rugaardsvej 15
DK-5000 Odense C, Denmark
ead@sdsd.dk

Kåre Kjelstrøm
Solution Architect
Silverbullet A/S
Skovsgaardsvaenget 21
DK-8362 Hoerning, Denmark
(+45) 2092 8244
kkj@silverbullet.dk

Jan Riis
Solution Architect / Project Manager
Lakeside A/S
Aabogade 15
DK-8200 Aarhus N, Denmark
(+45) 2160 7252
jri@lakeside.dk

ABSTRACT

Having relevant, up-to-date information about a patient's health care history is often crucial for providing the appropriate treatment. In Denmark, it-systems have been built to support different work flows in the health sector, but the systems are rarely connected and have become islands of data.

To remedy this situation, a service-oriented architecture based on web services for online exchange of health care data between the vast array of heterogeneous it-systems in the sector is being built.

The architecture forms a federation of web services and enables secure and reliable authentication of end-users and systems in the Danish health sector. The architecture is based on national and international standards and specifications. Yet it defines its own profile for secure interchange of data due to a lack of available international profiles that could handle the special need of the health sector at the time of project inception.

The architecture has been tested through a pilot project from mid 2005 to the end of 2007. This paper aims to convey experiences from the project, so rich in benefits that the architecture has been accepted and standardized as the foundation for the future of system integration in the health sector in Denmark.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed applications

D.2.11 [**Software Architectures**]

D.2.12 [**Interoperability**]: Distributed Objects

D.2.13 [**Reusable Software**]: Reusable Libraries

General Terms

Performance, Design, Reliability, Experimentation, Security, Human Factors, Standardization, Legal Aspects, Verification.

Keywords

Federated Identity Management, Web Services, SOA, SAML, WS-Trust. Single-Signon, X509 Certificates, Digital Signatures, SOAP, Security Token Service, Health Care, Electronic Patient Records.

1. INTRODUCTION

The it-system landscape in the Danish health care sector contains a plethora of different systems targeting various needs: patient administration, general practitioning, specialized care, electronic health recording, citizen access through web based health portals, etc.

The systems fall more or less uniformly into three classes:

1) Off-the-shelf systems typically obtained by privately held companies (e.g. health centers)

2) Tender based regional systems (e.g. for hospitals) and

3) National systems, typically tender based systems hosted by health care related departments.

Some of these systems are integrated today, but typically integration has been done locally, with the aim to reduce information redundancy. The real benefit in terms of quality of patient treatment and care, however, lies in a deeper integration of health care systems across organizational boundaries, such that *all relevant* information for treatment and care is made directly available in the systems that the health care professionals use in their daily work.

Founded in the strategic vision to strive for better quality in patient treatment, better systems for health care professionals, and the optimization of resources, the health care sector in Denmark has started the work on a national health care architecture that supports this vision.

The quest for universal availability of relevant and up-to-date information has been *the* most important force, shaping the architecture. There are, however, many other premises that govern this work, for instance the fact that in this domain, the "business" is never closed even if some or all of its it-systems become unavailable: People will still need treatment and care.

Therefore the architecture *must* minimize the impact caused by parts of the system becoming unavailable, and *must* support that systems can go into special emergency states. The more systems are unavailable the higher the risk of inefficiency or failure in treatment, and hence the higher the risk of physical harm, adverse effect or permanent maladies.

On the technological side, most health care applications in Denmark are non-browser based. In most cases users need specialized and highly supportive systems, something which until very recently was not feasible to build with web browser technology.

In Denmark, the Ministry of Science, Technology and Innovation has defined reference models and reference architectures for public it-systems. All new public it-systems must to some extent adhere to the principles of a "service oriented architecture" (SOA), and systems, legacy or otherwise, should be integrated using web services (WS). This, of course, is true for the health care sector as well.

As a consequence, the infrastructure that is currently being built in Denmark is based on a highly integrated, reliable, and fault tolerant SOA/WS architecture, where most existing applications are not based on web technology.

In such an architecture it is vital to be able to identify the end-user and/or the end-user system with various degrees of certainty. It is equally vital that all systems agree upon authentication credentials and semantics.

In 2005, the Danish health care sector launched an initiative with the purpose of analyzing and testing a combination of national and international standards surrounding federated identity management.

This initiative was coined the SOSI project for "Service Oriented System Integration". It was initiated by the Capital Region of Denmark, The Region of South Denmark, and the Danish Medicines Agency. Present in the steering committee was also the Danish Ministry of Science, Technology and Innovation. The project was funded by Danish Regions and is now governed by the Danish National eHealth initiative: Digital Health Denmark (SDSD).

2. CHALLENGES AND PRECONDITIONS

No it-system exists in a vacuum. Any attempt at creating a federation of heterogeneous systems that exchange sensitive information between disparate organizations will be bound by prerequisites given by the operating environment. For the SOSI project, it was necessary to take into account national standardization initiatives, and existing infrastructure components.

The Ministry of Science, Technology and Innovation drives much of the standardization effort in the Danish public sector as well. It does so in part by evaluating international specifications and classifying them in an interoperability framework [12]. For Single-Signon, SAML 2.0 is classified as the preferred framework of choice.

Any Danish SSO architecture should hence build on SAML, and attention was turned to profiles that were based on this specification.

There exists within the context of SOAP based web services a profusion of specifications aimed at solving various well-known issues from the world of computing: security, reliability, messaging, addressing, transactions, etc. Each specification adds levels of complexity and typically provides not just one, but multiple ways of achieving the same overall goal. Add to this the fact that often times, specifications from different bodies compete to become the de-facto standard, each attacking the problem at hand in slightly different ways: There's the recipe for non-interoperability.

The solution to this problem comes in the shape of profiles that cut through the stack of specifications, paving a narrow path of design choices for specific usage scenarios.

In the world of federated single-signon over the Internet, a number of such choices exist. OASIS defines the SAML specification [8], which is implemented by the Internet2 initiative Shibboleth [3]. A large group of non-Microsoft companies drive The Liberty Alliance Project [10], whose specifications extend SAML. IBM and Microsoft push the WS-Federation [4] specification and implements support in a range of products.

When the SOSI project was initiated in mid 2005 none of the existing single-signon projects gave good solutions to the particular needs for the project. Although there was a SOAP binding for SAML, no profile existed that laid out a complete protocol stack for exchanging SOAP messages with SAML assertions, while achieving single-signon to services.

There was and still is a heavy bias towards providing SSO for browser-based clients, with specifications relying on facilities such as HTTP redirect and cookies. In the SOSI federation, clients are typically client/server solutions, or stand-alone systems and almost never browser based.

The lack of a useful profile brought out the first reluctant thoughts of creating one for SOSI.

A large part of the health sector organizations in Denmark are connected to the same VPN network known as "SDN". The network was originally planned for teleconferencing, exchanging large amounts of data e.g. x-ray images, and accessing web based applications in a secure manner.

Any organization that wants access to services on SDN is evaluated for relevancy and must sign a mutual agreement per system-to-system connection. Although cumbersome, this procedure provides a certain degree of certainty that the network is primarily made up of organizations with legal business in the health sector.

The Danish national it-strategy for the health sector 2003-2007 [13] positions SDN as *the* communication channel for health care data. By supplying an integrity and confidentiality protected transport mechanism, which is immune to replay and man-in-the-middle attacks, and which has many of the relevant organizations connected already, SDN is useful for web services as well.

Also part of the it-strategy is the mandated use of digital signatures for secure identification of health care personnel. An important precondition in the design of a solution would

therefore be to leverage the Danish national certificate initiative, OCES.

OCES provides a number of important infrastructural properties including an embedded identifier, which can be translated to personal identification numbers through a secured service for authorized organizations. Since all citizens are provided with such a personal identification number at birth or naturalization, it has grown to become the primary key for identification. For instance, given a personal identification number, it is possible to check whether that person is a health care professional approved by the National Board of Health.

Finally, yet another standardization effort from The Ministry of Science, Technology and Innovation, "OIO", aims at providing a repository of reusable XML Schemas that follow predefined structural and naming conventions. The idea is to promote reuse and increase the chance of interoperability at a model level.

The web service body data, the actual business model of a SOAP envelope, should hence follow OIO guidelines and reuse schemas as appropriate or define new ones when needed.

In summary, the infrastructure should be built on:

1) Ratified international standards with SAML 2.0 as a cornerstone.
2) SDN, a VPN based health care network for secure transport.
3) OCES Digital signatures for identification of health care personnel.
4) OIO XML Schemas for promoting reuse and interoperability at the model level.
5) Sound design principles in particular those laid out by the Ministry of Science, Technology and Innovation for SOA.

3. THE SOSI DESIGN

A real world national Health Care architecture must be highly available, efficient, stable and tamper resistant to be useful. Hence, the first phases of the SOSI project put a lot of energy in analyzing real architectural needs for WS integration.

As of 2005 none of the SSO projects available were well suited for the specific needs of the Danish health care sector. All of them were in some way or other aimed at browser based applications and not at pure WS integration of stand-alone systems. For instance many of them included services or components that increased system dependencies instead of reducing them, thereby introducing potential single points of failure.

Although many of the profiles had elements that could be reused, the use-cases and interaction schemes were off target. A basic SAML and WS-Trust based profile [5] was therefore created based on the following principles:

1. A user should be able to authenticate with the federation once and then be able to use any service for which she has authorization for as long as she can present a valid federated security token. The design should, in other words, help reduce the number of sign-ons to the federation.
2. Using a client initiated authentication scheme, a WS client (WSC) system should be responsible for logging the user into the federation before starting to interact with any WS provider (WSP).
3. Inspired by current work on short-lived PKI certificates [9] the security token must have a limited lifetime and hence eliminate the need for revocation checks by WSPs.
4. Security tokens must be verifiable "out-of-band" by WSCs and WSPs, i.e. without having to communicate with any third party.
5. Security tokens should be able to carry basic end-user and client-system attributes that most WSPs use for logging and/or authorization. The design should support trust delegation/re-use, such that when the credentials within the security token have been verified, the embedded attributes can also be trusted. In effect this reduces the effort that WSPs must put into implementing web services. It also stabilizes the entire architecture by reducing system dependency to a minimum.

The proposed solution consists of:

- A trusted Security Token Service (STS)
- Security tokens as SAML Assertions
- Client initiated authentication that results in STS signed SAML assertions
- Core attributes embedded in the SAML security token

Figure 1 shows a simple interaction between a WSC, an STS, and a number of WSPs:

Step 1. The user authenticates with the federation either just-in-time before calling a service or as part of the local log-on to the WSC system. The WSC builds a SAML assertion with core attributes and user credentials, in this case a digital signature.

Step 2. The STS checks that

a. the WSC is on the white-list of systems and that that the user is not on the black-list of users not allowed to enter the federation

b. the user's digital signature is valid

c. the user's certificate is valid and not revoked

Step 3. The STS now seeks to verify that the client-specified core attributes are valid by using backend attribute services.

Step 4. If everything is OK, the security token is signed by the STS and returned to the WSC.

Step 5. The security token can now be used in interactions with different WSPs until it expires.

Step 6. Upon receipt, the WSPs validate the security token and leverage the embedded attributes for logging and authorization.

Step 7. Finally a result, i.e. business information or an error is returned.

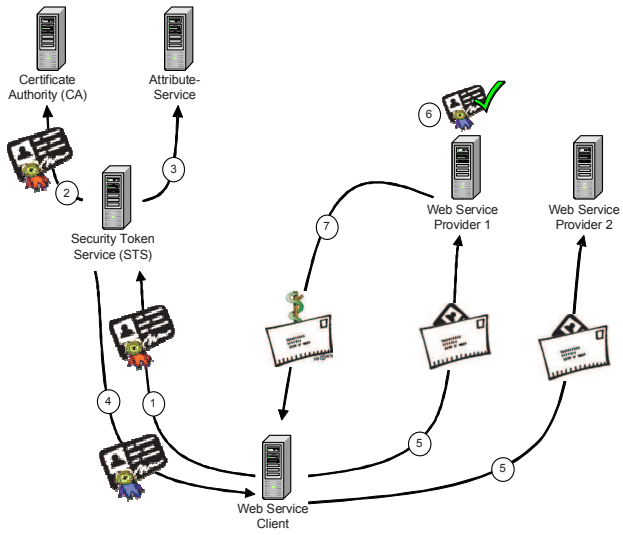

Figure 1: a simple WSC/WSP interaction

It is important to note the temporal flexibility between steps 1-4 and steps 5-7: The authentication request for the STS could be executed as part of the user's log-on to the WSC system. They could even be performed asynchronously and would only become blocking if the user entered a step in a workflow where entrance to the federation was needed, for instance in order to gather information from outside the system.

The maximum validity of security tokens is 24 hours in the SOSI proposal. However, the amount of trust a WSP can put into the security token depends on how old the token is. In other words the level of trust degenerates over time.

If the token is 5 minutes old when received by a WSP, the WSP can be pretty confident that it is still the same user operating the console. The SOSI proposal opens up for the possibility that the WSP can choose to reject security tokens that are "too old" at its own discretion.

It is worth noting that this mechanism is in contrast to the "single-sign-on" requirement: If all WSPs reject security tokens that are more than 5 minutes old, the user will be forced to re-login to the federation every 5 minutes. This should, however, only happen for services, which provide very sensitive information and hence demand very rapid time-outs.

4. AILMENTS AND CURES

Faced with a lack of product support due to a lack of profiles for SAML based web service interactions, it became clear that support for the SOSI profile would have to be implemented into every it-system in the federation in a custom manner.

While the SAML AttributeStatement although somewhat verbose in its syntax is not hard to implement, creating XML digital signatures is an entirely different story.

A programmer whose development platform does not support the XMLDSig [14] standard out-of-the-box will have to piece signing and verification functionality together e.g. from a crypto API. This includes creating secure hashes of data, implementing canonicalization algorithms, encrypting and decrypting, base 64 encoding and decoding, manipulating XML structures, and more.

As a remedy to this ailment it was decided early on to build a Java based library, "Seal.Java" [1] that would provide an abstraction, which would allow a developer to work with high-level primitives and not worry about envelope formats, digital signatures, or the darker secrets of the base-64 algorithm.

The EHR systems that entered into the SOSI project from the hospital side were mainly Java based, and while Seal.Java was relevant here, it could not be used with the EHR systems from the GP side that are mostly rich Win32 or .NET applications. This fact spawned Seal.NET [6], with the exact same purpose as its Java sibling.

Both projects have been constructed on an Open Source license and are available for general scrutiny via the web.

Third party software suffers from the "not invented here syndrome", a problem which the library projects sought to address by going to great lengths in testing, tuning, and publishing quality reports. When response times are high, multi-threading issues fixed, code coverage of the test suite well above 95%, and long term endurance testing of all API methods does not show any leaks; when the entire library is built from scratch, and all tests exercised on a nightly basis with fresh results published online in the morning [2], chances are others will accept it as stable and useful as well. Adoption of both libraries has proven to be high with most peers using them.

During development of the libraries, the idea surfaced that it would be useful to implement XML Schema validation for the XML, SAML, SOAP, WS-Trust, etc. that was passed around. Validation would improve overall quality and general faith that standards were followed.

Unfortunately that proved to be very difficult.

A profile that cuts across specifications is in effect limiting the number of possible choices a developer can make. Wouldn't it be great if it were possible to express the new set of limited choices in supporting schemas as well? It isn't! For instance, how do you express that it is a requirement to have an enveloped signature inside a SAML Assertion if the user authenticated using PKI?

Expressing such complex conditions is beyond and above what you can do with XML Schema. Even if it were possible, the problem of how to version a set of XML Schemas in concert arises: There is no great way today in which existing schemas can be narrowed under the same name space.

For development purposes, it was decided to modify the original schemas, SAML, SOAP, etc. to allow only those elements that were mandated by the profile. While helpful for testing, these schemas would not be used for production because they were overly strict and hence not compatible with off-the-shelf products that will attach extra non-critical SOAP headers, id's, etc.

Recently, a central test center for web services in the Danish health sector has been launched. The test center is capable of emulating clients and servers for various concrete services to a certain point not including too much business logic. It is manned by staff that can monitor requests and responses, and aid in debugging. The center provides value in ensuring that all parties wishing to implement a service will get past syntactical obstacles with the profile as well as with the model of the service in question.

The OIO initiative mandates that web services should be designed in a contract-first manner, where the service interface, the WSDL, including data models and service end-points, is defined independently of the code that implements it.

Unfortunately not that many off-the-shelf toolkits give good support to such a development paradigm. Now that tooling was already being implemented, it was decided to craft a contract-first WSDL tool that would allow for the easy creation of service interfaces as well.

Tooling is an important mechanism to help bridge the gap between specifications and products. Tools can make the difference as to whether a particular it-system will be able to participate in a certain scenario or not and without them, the SOSI project would not have been possible.

While providing tools and libraries to lower the threshold of integrating existing systems, there is also a risk associated with such a strategy: Source code, no matter how well written, will always have flaws, errors, or lack a feature for a given situation. Without an organization to maintain the code, it will eventually fail to be helpful. Such an organization is currently being formalized.

On the other hand, it is actually possible to tune the profile over time or align it with coming standards, when all parties rely on a few infrastructure components. Given the volatility of the current specifications for federations of services, this might prove to be a crucial strength.

5. LOOKING FORWARD

Federated identity management has evolved over the past few years, and there are now a couple of frameworks that might address the needs in the SOSI architecture. Most notably, the Liberty Alliance recently published version 2.0 of its Liberty ID-WSF, which defines interaction scenarios for web service clients with SAML via SOAP over HTTP. Future work will examine Liberty and alternatives in order to evaluate whether it would be feasible to align the SOSI project without critical impact.

Parallel to the initiatives in the health sector, The Ministry of Science, Technology and Innovation is driving other pilot projects that address slightly different needs, but define similar architectures. The OIOSI [11] project for instance is being pushed for secure asynchronous business document exchange via the internet using PKI and web services.

The health sector specific infrastructure must to be aligned with a future national infrastructure for all of the public sector without violation of the identified design criteria.

While digital signatures are currently being touted in Denmark as *the* technology to identify citizens and professionals alike, it is loved more by engineers than by end users. A digital signature is cumbersome to deal with and certificate management is not mature from an end-user's perspective.

At the time of writing two initiatives that extend the SOSI architecture are in the crucible:

First, a security gateway, SOSI-GW, is being developed that enables trusted domain cross-over. This vastly reduces the effort in implementing SOSI support for web service clients.

Secondly, a "logging and control" attribute hub and monitor is in an early design phase. The tool monitors and maintains attribute reliability, in effect taking over attribute management and resolution on behalf of all WSPs in the federation.

On the longer term, biometrics could have a place as the identifying technology, which would release the private key of a certificate instead of a password. The driving force for biometrics will, however, not be the increased security, but the fact that identification will become easier.

6. CONCLUSION

The proposed architecture has been developed and tested in real life, and the results are very promising with respect to both the development process as well as the implementation effort.

At the time of writing end-user feedback has not been gathered yet, but purely from a technical perspective the proposed architecture exhibits a set of nice qualities that support the special requirements for the health sector:

- **Single-Sign-On** to Web Services within the federation / trust domain.
- **Authentication levels**. Users and systems can be authenticated with different degree of certainty, depending on the credentials that the principal presents. This is in accordance with the guidelines [7] from NIST on which the Danish Ministry of Science, Technology and Innovation has based their authentication guidelines.
- **Reduction of impact** of unavailable of services. If, for instance the STS is unavailable, only users without a security token or with an expired security token, will be hindered in their treatment. All other users can continue to treat patients until their security token expires.
- **Reduction of the effort** that WSCs and WSPs must put into implementing web services. WSPs only have to trust/check *one* certificate (the federation certificate owned by the STS). Core attributes are available together with the security token.
- **Maximum performance**. The number of requests/messages is minimized. When the trust has been established (the user has logged in to the federation), the WSC and WSP communicate directly with no third party involved.
- **Reuse of existing infrastructure**. The design reuses existing infrastructure for establishing secure channels that takes care of confidentiality and stream integrity and prevents known cryptographic attacks.

The positive experiences with the architecture and profile outweigh the downside of not yet having international standards that fit the requirements of the Danish health sector.

SOSI is currently acknowledged as the best solution to the integration challenge, and at the time of writing, multiple projects that implement modules and systems based on the SOSI design, its standards and the associated Open Source tools are in the making.

7. REFERENCES

[1] Danish Regions, 2006-2007, SOSI Components, http://www.sosi.dk/twiki/bin/view/ProjectManagement/SOSIProducts

[2] Danish Regions, 2006-2007, SOSI Seal Component, http://www.sosi.dk/sosi/seal/

[3] Internet2/MACE, 2007, Shibboleth Project – Internet2 Middleware, http://shibboleth.internet2.edu/

[4] Lockhart et al., 2007, Web Services Federation Language, http://www.ibm.com/developerworks/library/specification/ws-fed/

[5] MedCom, 2006, Den Gode Webservice 1.0, http://www.medcom.dk/wm110102

[6] MedCom, 2006-2007, Den Gode Webservice Tools, http://www.medcom.dk/wm110344

[7] NIST, Electronic Authentication Guideline, 2006, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf

[8] OASIS, 2007, SAML 2.0, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security#samlv20

[9] PGP Corporation, 2006, PGP White Paper – Revocation Made Simpler, http://download.pgp.com/pdfs/whitepapers/Revocation-SLCs_060104_F.pdf

[10] The Liberty Alliance, 2007, Liberty Alliance Project, http://www.projectliberty.org/

[11] The Ministry of Science, Technology and Innovation, 2006, OIO Serviceorienteret Infrastruktur, http://www.oio.dk/arkitektur/soa/infrastruktur

[12] The Ministry of Science, Technology and Innovation, 2006. The Interoperability Framework. http://standarder.oio.dk/English/

[13] The National Board of Health, 2003. National it-strategy for the health sector. http://www.sst.dk/upload/nat_itstrategi03_07.pdf

[14] W3C, 2002, XML-Signature Syntax and Processing, Recommendation. http://www.w3.org/TR/xmldsig-core/