THE IMPACT OF WEB SERVICES ON AUDIT

Cristina Buchholz
SAP AG

Abstract: The technological key to open business environments is the use of web services. Web services are a loosely-coupled, standardized way of linking business logic across borders and platforms. In order to cope with the new audit requirements in a web service environment, one needs to deliver a concept for a collaborative, flexible, legislation compliant, application independent audit system. The system needs to be employed for intrusion detection, as well as for internal revision and auditing.

Key words: web services, collaboration, standards, audit

1. INTRODUCTION

Companies collaborate in e-business, and thus the borders between the Internet and corporate intranets disappear. In an environment like this, in which open networks are increasingly replacing closed and monolithic systems, secure data transfer between the two worlds is a central aspect of doing business.

The technological key to open business environments is the use of web services. Web services are a loosely-coupled, standardized way of linking business logic across borders and platforms.

Two scenarios attract attention in the web services environment: Companies need to do business on marketplaces. That is, they offer services for a wide range of users and different degrees of anonymity can and must be accepted for these users. Strong authentication is needed for those scenarios where for example financial transactions are implied; for personalization only, some basic authentication is needed. For the inter-company relations, the quality of the authentication is different. In this case,

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35693-8_16
M. Gertz (ed.), *Integrity and Internal Control in Information Systems V*

the role of a user is important, and the identity of the company whose employee this user is, not the identity of the users themselves. In both cases, there must exist a certain level of trust between the party that does the authentication and the party providing the services.

The means of accomplishing this level of trust are either a common ownership of the authentication services, or a federation of trust providing services. Efficient trust relationship management offers authentication, single sign-on, and impersonation mechanisms, as well as integration of public-key infrastructures, bridging the gap at the interface between users, systems and applications. As new standards emerge, user management will evolve from proprietary, application specific solutions that contain only high-level role information to generic user stores that offer not only role data, but also detailed, ready-to-use authorization information. Applications will be able to use information supplied by the central user stores without additional processing or checking, eliminating the need for sophisticated user management functions within individual business applications.

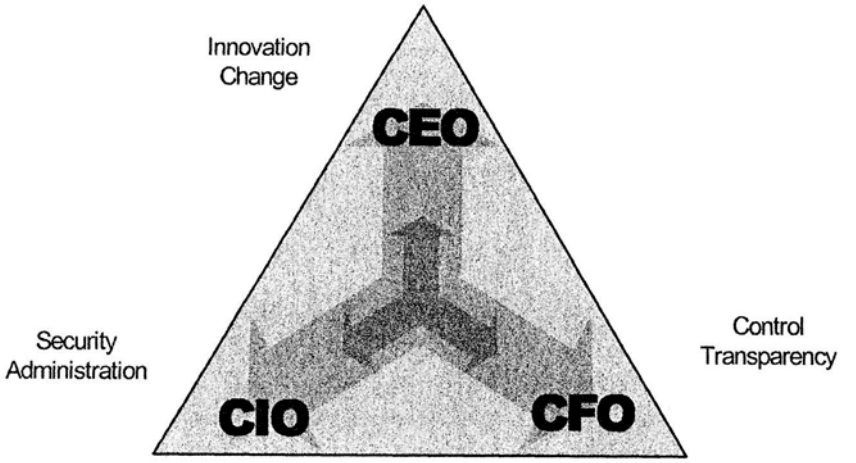

Figure 1. The audit challenge

The working environment of e-business is characterized by cross-company business transactions replacing closed internal business process loops. Various different applications are involved in one business process. With distributed processing, the data is successively changed and stored by different applications on different systems. The relevant data is distributed across multiple systems.

At present, it is virtually impossible to trace and analyze cross-application business processes. This is particularly true when a business process addresses several systems that are provided by different manufacturers. The situation becomes more problematic if the process runs

across the systems of different, collaborating companies. There are currently very few possibilities for auditing business processes between collaborating companies.

Audit is being made even more difficult by the traditional conflicting interest of the driving forces in an organization: Implementing new, open business processes based on web services causes changes in the administration landscape. Changes are the worst enemy of system administrators trying to keep a productive system running in a secure way. On the other hand, running business transactions across company boundaries, that is, outside the domain that can be controlled by the internal revision, raises the problem of auditing the process steps performed by external systems.

In order to cope with the new audit requirements in a web service environment, one needs to deliver a concept for a collaborative, flexible, legislation compliant, application independent audit system. The system needs to be employed for intrusion detection, as well as for internal revision and auditing.

2. SECURITY IN AN OPEN BUSINESS ENVIRONMENT

To realize thin Web services in such a way that they can be integrated seamlessly, the integration components are taken out of the application. Both user and process integration now take place in dedicated components:

One uses a portal for people-centric integration, and the other uses an exchange infrastructure for process-centric integration.

A first consequence of removing user integration is to begin to take away all user management from the applications. And since they no longer carry integration knowledge about users, there is no need to keep user information there. Information about people instead becomes part of the individual business objects. To exchange users between different trust domains, the concept of federated identities is currently being developed.

The implication of this shift is that there is also no longer any authorization administration within applications. Which makes sense, since authorizations can then be given to users in the portal framework on a business basis (and the application works with these values) instead of following the rules offered by the application for assigning rights to users. At present, there are a few early products that provide these kind of provisioning services.

But the application still has to check the validity of a request. This is handled by new protocols for exchanging credentials, known as assertion

handling protocols, such as the Security Assertion Markup Language (SAML), for example.

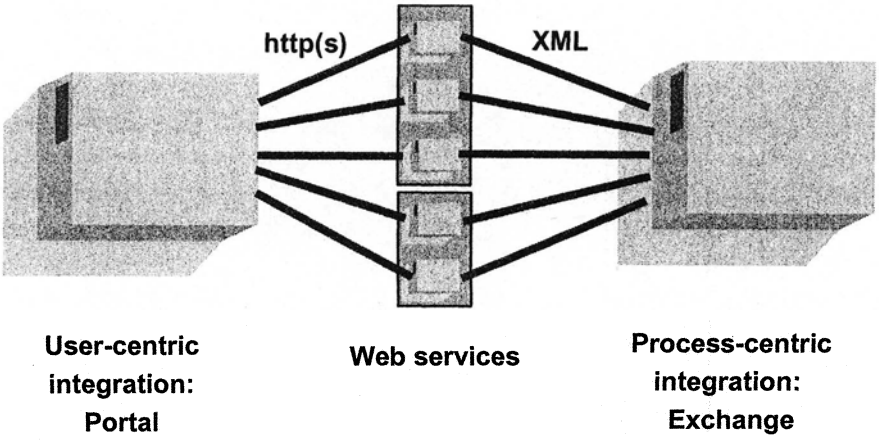

Figure 2. Web service integration

Removing process integration has a similar impact. First of all, the communication between different services no longer takes place within one closed system. So this communication has to be protected against manipulation, eavesdropping, and so on. Web Services Security Extensions provide a standardized framework for applying encryption and digital signatures to SOAP requests.

Secondly, the knowledge of a company's processes moves from the application server to the exchange infrastructure. This knowledge is crucial for a company's assets, and this component therefore has to be highly secured.

Finally, processes must be audited at some point in time, for legal or financial reasons. But since processes in a Web-services world are distributed by their nature, auditing becomes largely impossible. Which is why you need a framework for tracking processes across a broad landscape.

All of these new requirements show that it is no longer sufficient to rely on a perimeter type of security to protect your company's assets. Both firewalls and centralized user management have to make way for distributed security mechanisms.

3. BREAKPOINTS IN THE SECURITY CHECK

In performing a business security check, there are certain aspects that have to be considered, especially in the open network environment. Those are defined as breakpoint in the security check process.

3.1 Identity

In addition to the identification of the person, the connection to the company must be evident. For business-to-business scenarios, the connection to the company and the position in the company is much more important than the personal data of the user.

For business-to-customer scenarios, the purchase history for one customer is important. The history can be kept in the own system, for example CRM, or administered by the federation. In the latter case, it is questionable whether the history should be centralized and made available across all participating systems.

3.2 Roles and authorizations

Following the authentication, the employee is being assigned a role. The fact that an individual works as a purchaser for company A is an important trust information for company B. Based on that information, the role management system of company B, trusting the role management of company A, assigns the role of customer to the individual.

There must exist a mechanism for limiting authorisations to certain periods (day/night for resource-consuming tasks, end-of-month for closing balances) or issue authorisations dependent on the type of access (web or not), or the locality of the user.

Not only the roles derived from the organisational structure need to be considered, but also roles for partners, employees of partner companies and customers. Mapping of company relevant roles to external users is needed. Best approach for ease of maintenance is to automate that mapping. This means that one has to consider the following question: If an individual is purchaser for company A, does she get the same role for the vendors B and C?

The data users are allowed to access is not only derived from their role, but also from the responsibilities users have. The responsibilities and the roles generate the authorization for accessing an object, a service or performing an action.

A particular authorization is needed to allow the user to delegate some of their privileges. Delegation of authorization must conform to time and audit rules.

3.3 Privacy and non-repudiation

Today, transport layer security services such as SSL for HTTP connections or the GSS API for communications between SAP systems provide communication security from starting point to finishing point. But when data is passed on via intermediary systems outside of this transport layer, the integrity of the data might be lost.

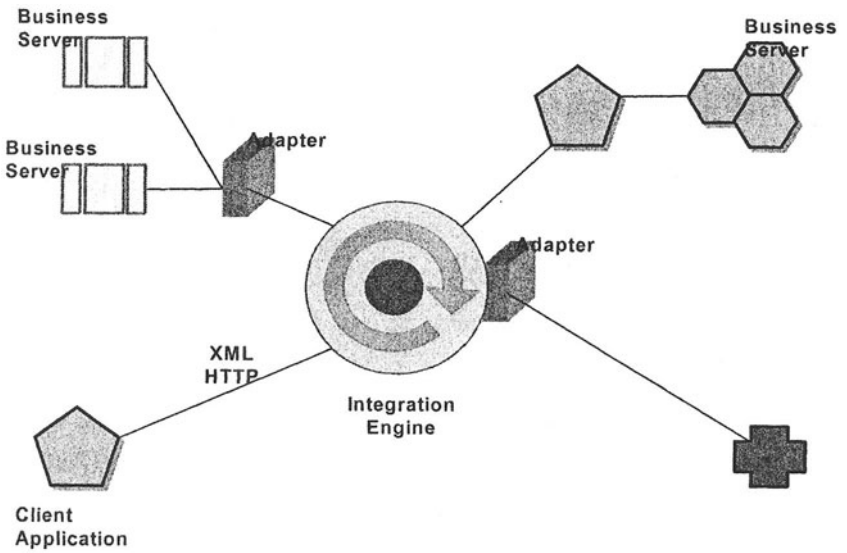

Figure 3. Integrating Systems in an Open Business Landscape

The recipient of the message has to rely on the security checks performed by the systems that have already processed the message – which means that a Web-service security architecture is needed to provide end-to-end security. Message confidentiality and integrity can be provided by using XML encryption and signatures as well as security tokens, such as digital certificates, to ensure that the message is transmitted without modification.

In addition, the web of trust established by a PKI must be extended to include both direct and brokered trust relationships. Existing trust relationships can be used as a basis to broker trust by creating services that issue and administer security tokens.

3.4 Audit

In the type of system landscapes that have been common until now, business processes are each processed on a single computer system.

As open landscapes replace closed systems, processes are no longer limited to one application server and may flow through the entire IT environment. In situations like this, the auditing capabilities of each system – even sophisticated ones – no longer suffice. System interaction in collaborative processes increases complexity. In addition, some components of the open landscape may contain sensitive information that belongs to different organizations. In virtual marketplaces, exchange infrastructures, or application service provider scenarios, this may even include data from competitors.

4. IDENTITY MANAGEMENT – REQUIREMENTS FOR BUSINESS SCENARIOS

In a Web-service environment, different systems operating on different platforms and in different technology environments can come into play in one business process. Web-service security is about making sure that the integrity and confidentiality of messages is ensured as they are passed on via the Simple Object Access Protocol. It is imperative that the service called acts only if the requests come from an authorized source, and this stresses the importance of user and authorization management, in short, identity management.

We shall in the following concentrate on the identity management. Based on the above analysis we derive the following central requirements for a functioning business-to-business authentication and authorization framework.

4.1 Authentication and Single-Sign-On

Authentication must be realized towards a primary identity management system. The users should not be obliged to log in again for accessing web services, not even across company borders. The trust in the existing authentication must be provided.

4.2 Authorization management

There must exist a trusted role correspondence between the systems of the two companies, stating that for example every authorized purchaser of company A is allowed to act as a customer towards company B.

4.3 Coordinated policy management

There should be common rules and policies established for maintaining the user data and authorizations across companies. Thus, synchronization and accuracy of data are uniformly guaranteed across participating systems.

4.4 Privacy

The rules and policies defined in the previous paragraph should also guarantee that user can choose to participate in a service, the amount of data they share and the data shared is accurate.

4.5 Audit

Auditing is a major requirement for business environments. There are two major situations where auditing is a must:

- Provable audit trails for business scenarios are a legal requirement;
- Intrusion detection relies on access logs.

5. FRAMEWORKS FOR FEDERATED IDENTITIES

There are at present several initiatives trying to solve the previously outlined challenges of a business-to-business scenario.

Microsoft started to work on a universal authentication and authorization framework, now spreading in the world of end-users: the .Net Passports. This doesn't really address the world of business-to-business relations; nevertheless, it has already a wide acceptance in the consumer world.

Sun reacted to this by forming a consortium of industry under the name Liberty Alliance. The declared scope of the Liberty Alliance is to define a vendor independent framework for federated authentication and authorization, addressing consumers' as well as businesses' needs.

With the special focus on Web Services, IBM, Microsoft and Verisign formed an alliance to promote Web Service Security. The roadmap of this alliance promises to address many of the issues outlined in this document

and is certainly answering some open questions in the matter of providing document authenticity and non-repudiation.

5.1 Microsoft .NET Passport

To benefit Web users, Microsoft has developed .NET Passport, a simple single sign-in authentication solution for the Internet. .NET Passport stores credentials for each user, including a user name and password. When users logs on to any .NET Passport-enabled Web service, the .NET Passport service performs the authentication process, verifying the user's credentials and notifying the service of the results.

.NET Passport has a two-year track record and more than 200 million .NET Passports issued. The Microsoft initiative holds thus a very strong position on the front-end market.

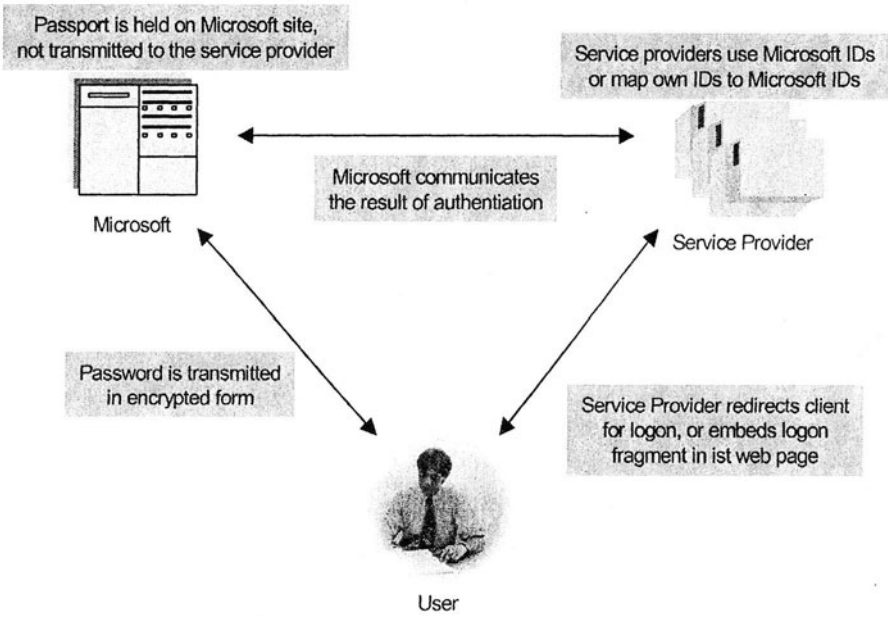

Figure 4. Authentication performed based on .NET Password

When a user signs in to an outward-facing Web service that uses the .NET Passport authentication service, .NET Passport verifies their identity and securely transmits an encrypted cookie that contains the user's ID to the service. An enterprise can rely on this ID to identify an individual consistently across resources at their company, as well as across resources operated by business partners using the .NET Passport service.

In order to build bridges between the islands of authentication that exist within enterprises, between enterprises, and across the Internet, Microsoft has announced the Internet Trust Network, in short ITN.

The ITN is composed of .NET operators, each performing the set of operations described. Each operator is the holder of the entire user data for the associated users. The participating non-operator entities need to trust the authentication performed by an operator.

The level of privacy depends on the policy of the operators. It is not clear at the moment, whether the user has any influence on the level of privacy for certain transactions, or the choice of the information transmitted to the participating entities.

5.2 Liberty Alliance

The Liberty Alliance Project is a business alliance formed to deliver and support an identity solution for the Internet. Its purpose is to enable single sign-on, for consumers as well as business users, in an open, federated way.

The strength of the Liberty Alliance is given by its broad range of participants. They are business-to-customer, as well as business-to-business interested.

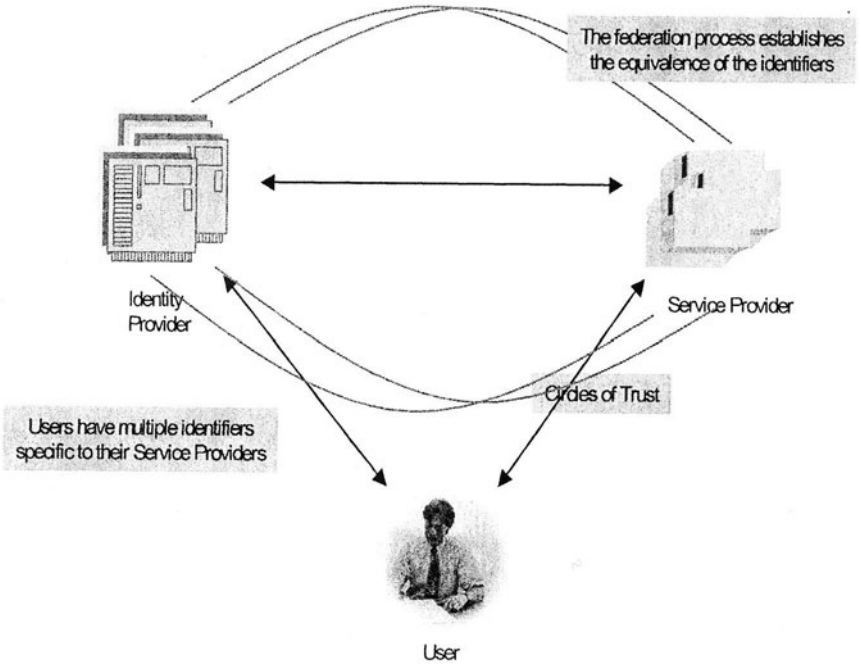

Figure 5. Mode of operation in a Liberty federation context

The primary goals of the Liberty Alliance Project are:

- To allow individual consumers and businesses to maintain personal information securely.
- To provide a universal open standard for single sign-on with decentralized authentication and open authorization from multiple providers. To provide an open standard for network identity spanning all network devices.

Liberty shall include a set of protocol specifications and design patterns enabling distributed single sign-on, authenticated transactions, distributed user management, identity representation, identity verification and bilateral trust agreements.

The solution shall be based on open standards (such as XML, SOAP, PKI, SAML) and deliver a set of open specifications that can be implemented on a range of platforms. Liberty shall support the notion of multi-tiered authentication so that simple to difficult means of authentication may be required depending on the purpose of the transaction.

The solution shall support role-based and group-based authorization where a user can access a service on the basis of the role the user is playing without necessarily disclosing identity information.

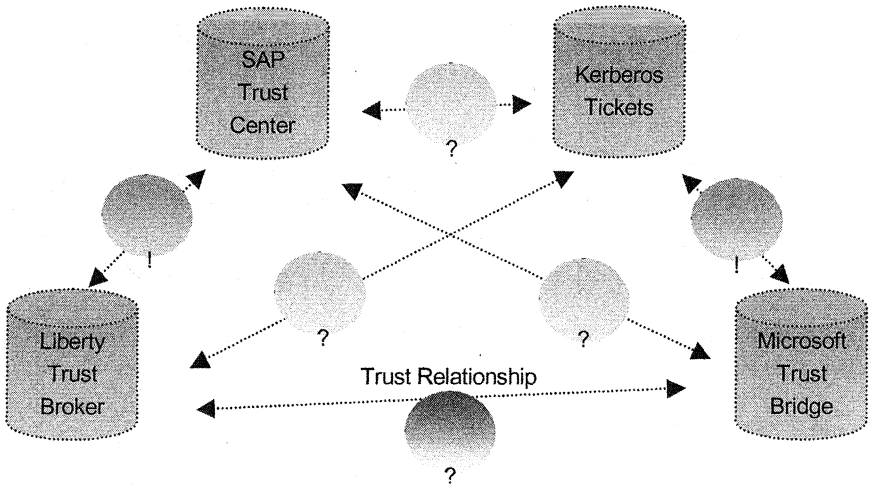

Figure 6. Future Vision on Trust Relationships

The solution needs to support multiple, independent profiles. For instance, a user may want a "home" profile and a "work" profile, and perhaps one or more "role" profiles such as "purchasing agent" profile.

The authentication is performed against one of the participating entities of the federation, but this entity is not the holder of all the user attributes.

Given that the user data is not centrally held by an operator, like the .NET approach proposes, the privacy issues to be solved are much more

complex. On the other hand, no participant in the federation can get possession to all the data, and thus the danger of a monopolic mode of operation is banned.

The framework will have a truly federated approach to privacy. The adhering organizations have the possibility to specify a required, as well as a desired level of privacy protection, and the level of protection adopted for the communication will thus be negotiated.

The open question is, when will all this become available. Liberty is a new initiative, the first specification has been released in July. Given the market pressure and the competition by Microsoft with the .Net Passports, the alliance has set itself the goal to finalize the specification until the end of 2003.

5.3 Web Services Security

Following the path designed by the Passport initiative, Microsoft, IBM and Verisign joined in defining an open Framework for establishing Web Services Security.

The intent is to describe a general-purpose mechanism for associating security tokens with messages. No specific type of security token is required by WS-Security. It is designed to be extensible (e.g. support multiple security token formats). For example, a client might provide proof of identity and proof that they have a particular business certification.

The technical layer is the SOAP protocol and the SOAP extensibility model. SOAP-based specifications are designed to be composed with each other to provide a rich messaging environment.

WS-Security relates to standards for XML messaging, and a central role is taken by the XML signatures. Although they envisage a common trusted management of identities and authorisations, it is not yet clear how this level of trust can be achieved or maintained.

Among the further specifications that are announced, is the one concerning privacy. It is desirable, in this context that they will go beyond giving an answer to how to technically ensure data privacy and tractability of transactions, and also address the business scenarios involved. A model very similar to the one proposed by the Liberty Alliance is planned: the users and organizations should be able to express privacy preferences and requests at their respective level.

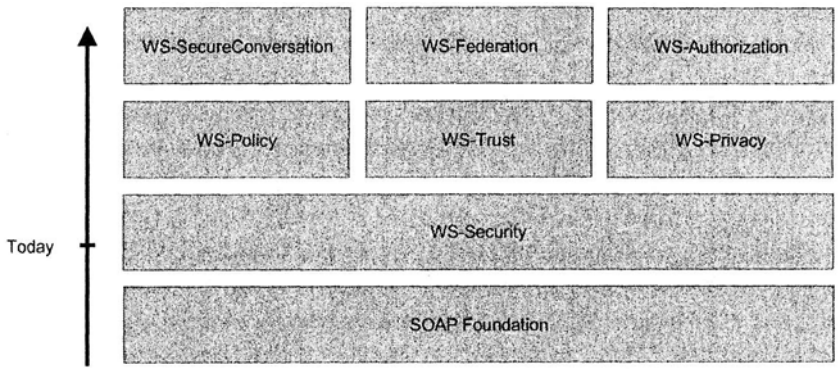

Figure 7. WS-Security planned content and specification roadmap

The position towards federation and trust policy is only to be published at a later time. These papers will address the modalities for organizations to manage and broker trust relationships in a federated environment, including federated identities.

Again, still to follow is the description of authorization and authorization policy management.

6. AUDIT – SETTING THE PICTURE FOR WEB SERVICES

Web services provide a way of linking applications not only within an enterprise, but also across company boundaries.

The connections are loosely coupled, and language- and platform-neutral, which allows greater flexibility in collaborating with customers and partners.

However, it also means that such security functions as managing users and trust purely within an enterprise, or providing non-repudiation information using digital signatures, are no longer sufficient and need to be enhanced by Web-service security features that transcend the boundaries of the closed enterprise IT environment.

The new security models needed can be added to existing functionality, to protect your investments as business processes are turned into Web services.

The main task of these new models is to secure the integrity and confidentiality of messages sent via the Simple Object Access Protocol (SOAP), and to ensure that the services that are called act only if the request is properly authorized and can provide proof of this authorization.

6.1 Things that Make Audit Difficult

We are used to the picture of users accessing the business systems directly. The actors involved are the users and the systems. In order to facilitate access, the applications have been web-enabled. We had users accessing the systems through browsers, supported by web servers.

But the present-day system landscape is becoming tremendously complex, access to the systems needs to be granted within companies, but also for business partners. We experience the change between webifying an application, and implementing access points to be used in a collaborative scenario.

The user does not access a web service directly, but through the mediation of an application.

For incoming SOAP requests, one must establish the calling user (Authentication) and then map this information to a system user able to request the web service. Authentication is performed based on credentials on the message level (for example, SAML Assertions/WS Security Claims/SAP Logon Ticket, certificates, Username/Password) or on transport level (for example, http Basic).

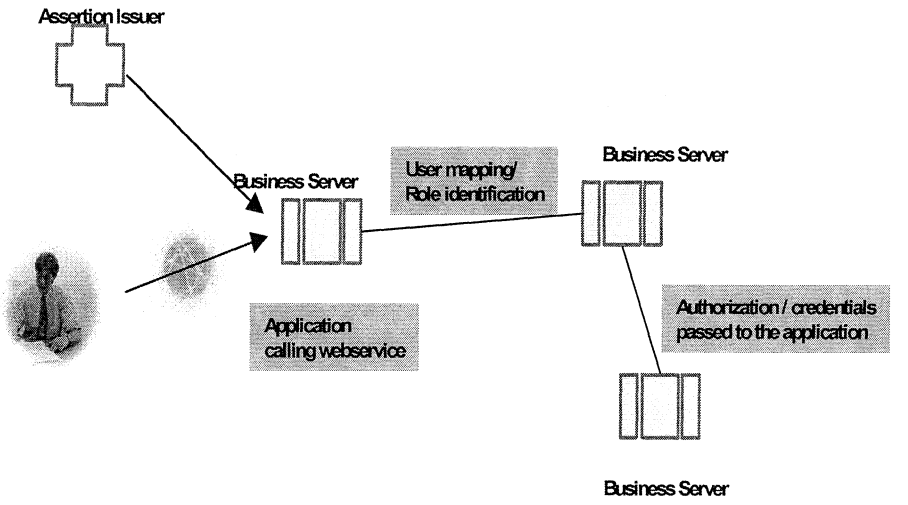

Figure 8. Auditing user information with web services

For subsequent calls to web services, the authentication information should be reused if possible (Single-Sign On). This implies a user mapping for incoming request. For outgoing requests, one has to specify which information (username, role) should be sent as assertion to the web service.The web service acts in the name of the user. Nevertheless, the

information regarding the authenticated user, the service performing the authentication and its authority should be available for the auditing system.

The access to web services must be restrictable depending on the authenticated user, the role and the credentials available. Beside authentication and relating a user to a role, further credentials may be available and should be passed on to the application.

When they are used, logs should be kept with the information who used them, for which purpose, and who was the issuer of the credentials. This means that both te security information regarding the credentials has to known to the application, and that it is not sufficient to provide the audit capabilities on the middleware level.

The communication has to be encrypted for preventing eaves dropping. This is challenging for the audit system, because of the choice to be made: it is either possible to check the content for potential intruders or malicious code, or preserve the required end-to-end security by decrypting the message only at the recipient. In the latter case, the recipient is also responsible for content checking and providing the audit information. The messages have to be signed in order to guarantee the authenticity and non-repudiation. It is not sufficient in his context to provide the name of the signer, but a comprehensive audit trail requires that also the authority of the signer, and the authority of the credential issuer are being logged.

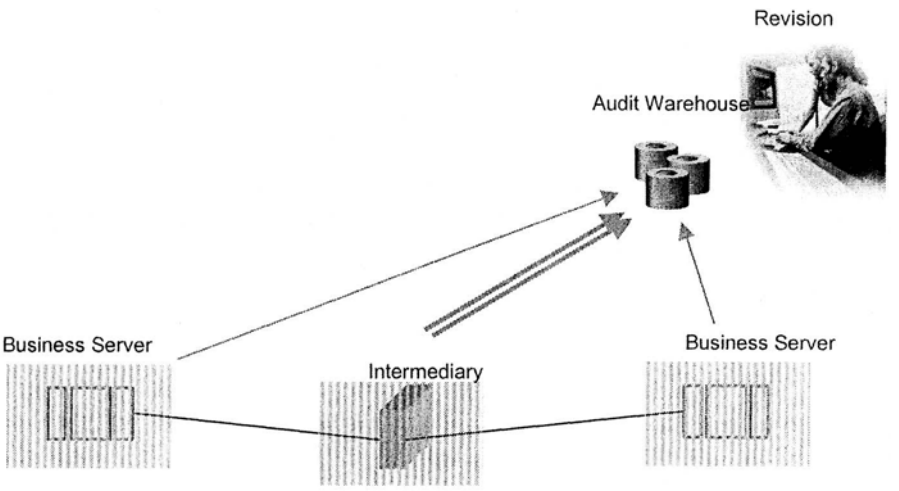

Figure 9. The intermediary - key component for audit

The concept of the intermediary poses problems for the auditability of applications. The intermediary should have access to enough information for

routing and addressing the message, but sensitive data (for example the Personal Identification Number PIN), should not be part of any logs on the intermediary. This is particularly difficult to account for when the intermediary is responsible for content checking. Nevertheless, it is a requirement for the revision.

Messages being passed through multiple systems – it is vital to ensure a sound audit trail: who sent what, when to whom. Consequently, it is needed to log the routing information and the decision making process when multiple choices are available.

6.2 Use Cases

6.2.1 Intrusion Detection

The system detects inappropriate, incorrect, or anomalous activity. The Audit Framework operates on multiple hosts, and also supervizes network data flow. The system immediately reports misuse and intrusion detection. Here, the term intrusion is used to describe attacks from the outside; whereas, misuse is used to describe an attack that originates from the internal network.

6.2.2 Plausibility Audit

The system detects and reacts to unusual actions – actions that do not match the pattern. This is usually to be accomplished through the mediation of an expert system gathering information during the normal functioning of a system and building up a knowledge database about the business processes. E.g. if a clerk pays bills up to 10$ every day, but on Sunday checks out 10k, then something is wrong and the system should report this event.

6.2.3 Revision / Configuration Audit

The configuration information is periodically retrieved and analyzed for detecting unsecure settings, e.g. extensive authorizations in a productive system or insufficiently protected resources.

6.2.4 Process and Performance Audit

Every step of a distributed process is audited put in a central context. The process flow is analyzed based on these data in order to determine who is in

charge of what process step, whether revision requirements ("four-eyes" principle) are met and which process steps are bottlenecks during processing.

7. SOLUTION: COLLABORATIVE AUDIT FRAMEWORK

Existing audit solutions work for business processes that are run on a single computer system. The data for a business process are then stored in a single database. This individual system can be audited. The "SAP - Audit Information System" is an example of this type of local audit system. None of the existing audit systems is capable of doing a process audit across heterogeneous systems. Therefore, a new Audit Framework needs to be developed. The most important goals for this framework are:

- It should allow an audit of processes in distributed application landscapes.
- In this context, it should be possible to perform a comprehensive audit of various applications and systems. This includes systems from different manufacturers.
- It should provide a central, tool-supported audit of cross-enterprise business processes.

The actions to be performed have been previously detailed – see Use Cases.

7.1 Architecture

As previously stated, it must be possible to audit all applications and systems that process business tasks. This also includes infrastructure elements such as directories that are important for user administration and the "Integration Middleware", which provides the communications flow

It is only possible to obtain an overview of the cross-application and cross-enterprise business processes if the data can be analyzed centrally. This central overview can be created using an "Audit Warehouse". It is therefore necessary to create a unified standard for an audit data interface. The "Audit Warehouse" can access the application data through this interface.

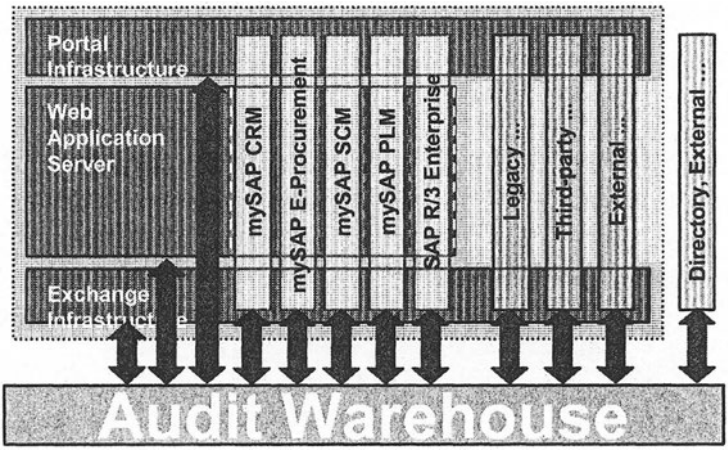

Figure 10. Collaborative Audit Framework

How does the data get into the Audit Warehouse? There are basically three possible types of access:

- First, "Push": The applications write the relevant data to the central audit pool.
- The second type of access is "Pull": The central audit warehouse regularly collects the data using the interface and stores the data in the central audit pool.

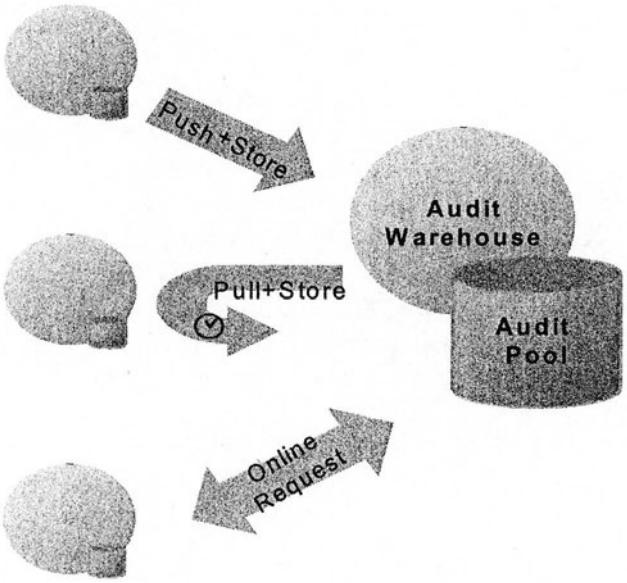

Figure 11. Data Access for the Audit Framework

- The third type of access is the "Online Request". With this type of access, the Audit Warehouse can address the interfaces of the relevant systems and collect the data at the time of the audit. In order to do this, the Audit Warehouse must, of course, know where the relevant data is (roughly comparable to a metadirectory).

The audit data pool can be analyzed using either SAP applications (comparable to the Audit Information System and the Business Warehouse) or with standard analysis programs.

7.2 Prerequisites

A comprehensive audit should include not only audit data from within the company, but also extend to the business partners. Subject to business agreements between peers, some part of the internal process knowledge of the company should be made available to the holder of the next (external) step of the process. Thus, the revision security can be proven for the whole process. Auditing frameworks are an essential tool for ensuring audit-proof processes in collaborative scenarios.

To support the audit system, it is important that the applications provide an interface. This interface should be standardized, provide online information as well as permit configuration and statistical analysis, and be part of applications as well as middleware and communication infrastructure.

7.3 Implementation Steps

The interface is to be developed in 2002 together with the interested auditors and will be based on XML. This interface cannot be SAP-specific, but must be generally available, and become established as an official standard. It is important that the interface is supported by all participating applications.

The next step is to develop an analysis tool capable of handling the amount of data susceptible for audit. A good candidate is the SAP Business Warehouse. The information should be analyzed online in order to provide intrusion detection alerts, but also stored in a reduced format for later proof and revision of the business process.

Finally, the policy handler will be implemented as an expert system, capable both of recognizing patterns, and to learn new patterns during operation.

8. ANALYSIS

Web services need to be part of the collaborative business landscapes. At present, they are being tested in internal processes, before launching them against the openness of the market. The fact that they are part of the business reality cannot be ignored anymore, and we have thus to provide a comprehensive audit solution to meet the challenge.

To make things even more difficult, the architecture of a webservice based collaboration is under discussion. There are several initiatives trying to get out in the open and gain market momentum. The pitfall is that, if there is anything worse than no standard, then it is to have many standards. The key to success is for the existing federations to work together in providing a trust relationship.

In this respect, SAP is working together with both the Liberty Alliance and the WS Security Alliance. In the definition phase, SAP is providing input for the definition and support of the business-to-business scenarios.

At present, we are facing the following situation: Microsoft committed to be compatible with the Kerberos tickets, thus establishing a trust relationship. It is likely that the SAP Trust Center and other PKI-based approaches, as well as Liberty, will commit to support and accept Kerberos tickets as well. Due to their strong market presence, SAP is going to support the .Net Passports, in the future ITN. While Liberty is open to all supportive organizations, it is at present questionable whether Microsoft is going to join this alliance, thus providing the trust relationship between Liberty and .Net trust brokers.

WS-Security would be a good forum for working towards the standardization of the audit provision. Unfortunately, no such specification has been announced yet.

At the end, we will probably have a network of authentication mechanisms, all compatible, and relying on the principle of federated identification. The purpose of this paper is to analyze, whether the form and complexity of the existing services is sufficient for a realistic collaborative business scenario.

8.1 Pitfalls

Given the described strategies, they have common strengths, being driven by a common model, and unfortunately, also have common challenges. I shall evidentiate this with respect to the wish list defined for business-to-business scenario: single sign-on, authorization management and coordinated policy management.

Microsoft Passports contains the user's personal data, not the role. Besides, the identification is performed externally and centralized, by .NET operators – this is difficult to conciliate with the company policies. You don't want to increase the distance between you and your customers. On the other hand, the auditing of such a centralized approach is not much more difficult than dealing with a single system in the intranet. Regarding the .NET operators as an outsourced resource, and provided that enough transparency is given over the processes hosted by it, the audit problem can be reduced to the auditing of intra-company processes.

Liberty is driven by powerful business-to-consumer interested organisations. SAP, and other Liberty members with similar interests, play an active role in creating a definition that sets the scene for business-to-business. However, there is no support for policy coordination planned at the moment.

The privacy concerns with the .NET approach are being addressed by Liberty. There is no specification available yet, however, the distributed framework for authentication and attribute specification promises to exclude any monopoly on the possession of user data.

Given the centralized architecture of the .NET approach, auditing is much more feasible. The policy rules are static, this alone takes one dimension away from the audit complexity. With Liberty, the privacy level can be mostly automatically negotiated. The data is held by different entities and the notion of a principal actor becomes confuse. Thus, the answer to the auditing problem might prove difficult.

9. CONCLUSION

Companies must be able to check the security of not only their own IT architecture, but also that of their business partners using technical revision procedures. Contracts and laws often demand proof of a revision-secure environment. Comprehensive auditing frameworks are an essential tool for ensuring audit-proof processes in collaborative scenarios. A central element of this is the Audit Warehouse, which preserves the overview of distributed processes. To support the Audit Warehouse, it is important that the applications provide an interface. This interface must be an integral part of SAP applications.

None of the presented approaches is dealing with role definition and distribution across companies. The matter of establishing trust and realising corresponding access policies has not been solved yet. As a consequence, the auditing has to be subject to business agreements as well.

Concluding we can say that the techniques for realising secure Web Services are being standardized at present. In establishing a business relationship however, the most time is spent working on agreements, contracts and liability sharing. Still under research, is a way of automating the human actions, so that as little interference as possible I needed for configuring the secure world of e-business. The presence of web services thus imminent, the audit problem needs to be addressed swiftly and be given due attention by the standardization committees.

REFERENCES

[1] Cristina Buchholz, Digital Identities and Federation, DuD 9/2002
[2] Liberty Alliance Project http://www.projectliberty.org/
[3] Microsoft .NET Passport http://www.microsoft.com/netservices/passport/
[4] Cristina Buchholz, Policy Based Authorization Management, ISSE 2002
[5] Cristina Buchholz, Web Services Security, SAP TechEd 2002-10-14
[6] Web Services Security (WS-Security), Bob Atkinson a.o, http://www-106.ibm.com/developerworks/webservices/library/ws-secure/