Redbooks Paper

Jeff Crume
Axel Buecker
Keith Gordon
Jim Heid
Jatinder Pannu
John Sanders
Andreas Schmengler

On Demand Operating Environment: Security Considerations in an Extended Enterprise

This IBM Redpaper is intended for use by Software IT Architects (SWITAs) and Client IT Architects (CITAs) as a means to better understand some of the security issues introduced in an on demand environment. While the topic is very broad, this paper focuses only on software-related solutions specifically within the context of securing Web services transactions and managing identities within a dynamic, federated infrastructure.

This material is intended to serve as a primer only. A more comprehensive treatment of these and other subjects related to on demand security can be found in other documents referenced in “Appendix A: On demand logical security architecture” on page 51.

Organization

This document is comprised of four primary parts:

- Part one introduces some of the architectural issues and client pains involved in securing an on demand operating environment (ODOE).
- Part two discusses ongoing standards work with existing and future product capabilities.
- Part three combines Parts one and two by applying solutions from Part two to a set of fictitious client scenarios.
- Part four contains appendixes listing key concepts, acronyms used in this paper, and additional reference materials.

Part one: On demand and security

Part one discusses the increasing demands of an on demand operating environment (ODOE), and the role of security in that environment, as well as the major client pain points and the different models of on demand security.

Accelerating on demand

Today, enterprises are continuing the on demand evolution by developing the business model agility needed to create value and demand for their new products and services. For the past 10 years these enterprises have been hyper-connecting their businesses, processes, clients and suppliers to gain competitive advantage. This hyper-connectivity is now the life support system of business speed and value. This connectivity brings with it the administrative burden of connecting more users to previously guarded applications, with the pressure of smaller administrative staffs and lower expense targets.

As shown in Figure 1, risks and threats have similarly evolved that need to be cost-efficiently mitigated by the proper use of security services, hardware and software in order to take advantage of on demand functionality.

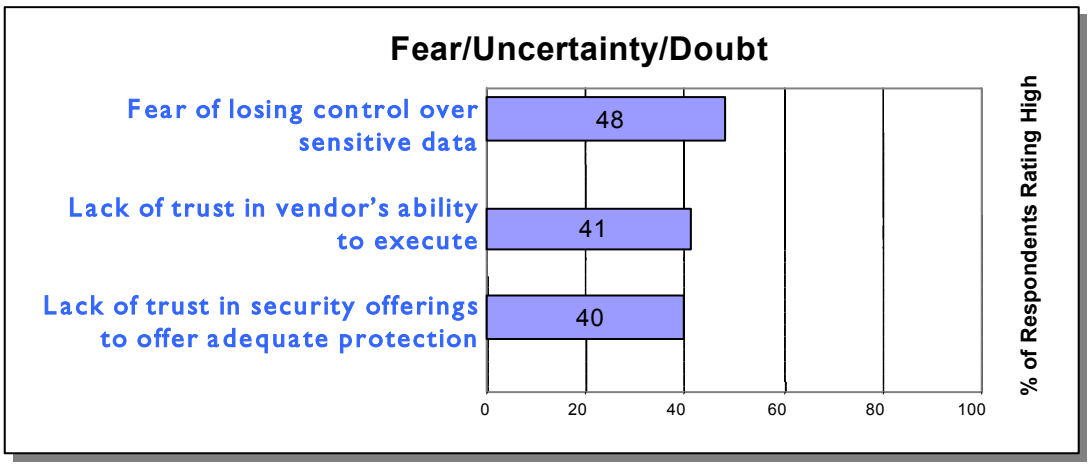

Figure 1 Inhibitors to becoming an on demand business

The role of security in an on demand environment

In an on demand world, change happens faster and more unpredictably than ever before. To stay competitive, an organization needs to gain the speed, flexibility and resilience to handle whatever the market does next.

On demand business™ allows organizations to lower costs, increase revenue and respond quickly to industry pressures. It requires changes in the way a company deals with technology, strategy, business practices and corporate culture. These transformations can run deep, but the potential rewards of operating on demand are enormous.

On demand business: A company whose business processes are integrated end-to-end across the company and with key partners, suppliers and customers and can respond with flexibility and speed to any customer demand, market opportunity or external threat. An on demand business has four key attributes: responsive, variable, focused and resilient.

Achieving leadership in the on demand world requires IBM to create cost-efficient enterprise integrity that mitigates new risks and threats.

In a utility-based model where systems and resources are dynamically shared, connecting the right people to the right resources at the right time is critical to the business process. Maintaining the integrity of who interacts with your business must be assured for this on demand model to work. In addition, the continuity of the information assets and access to those assets needs to be protected. Managing security processes and policies delivers an effective solution.

The promise of on demand added value is in enabling businesses with the capability to take advantage of dynamic market conditions and respond to new client needs.

On demand attributes

An on demand business exhibits the following four key attributes according to *On demand Operating Environment: An Overview and Implementation Guide,* IBM REDP-3858:

Responsive The ability to sense and respond to dynamic, unpredictable changes in demand, supply, pricing, labor, competition, capital markets, and the needs of its customers, partners, suppliers, and employees

Variable The ability to adapt processes and cost structures to reduce risk while maintaining high productivity and financial predictability

Focused The ability to concentrate on its core competencies and to differentiate and meet the needs of all of its constituents

Resilient The ability to manage changes and external threats while consistently meeting the needs of all its constituents

Resiliency, in particular, is directly affected by the extent to which the business is able to secure its critical IT assets.

IT security fundamentals

You can think of IT security in terms of three primary requirements: confidentiality, integrity and availability (CIA).

- Confidentiality, which is the requirement to have sensitive data protected from unauthorized disclosure. This requirement can be further deconstructed into the need for:
  - Encryption of data to guard against eavesdropping
  - Authentication of users to ensure that they are who they claim to be
  - Authorization to enforce access control policy, which determines what users are allowed to see and do
- Integrity, which is the requirement that data has not been tampered with and can be, therefore, relied upon for use in business transactions. Some definitions also include an assurance as to the source of the information, while others group this under the area of authentication.
- Availability, which is the requirement that systems are operational and data is accessible when it is needed.

Many IT architects only focus on the threat of accidental equipment or software failure; in other words, they deal just with the ramifications of Murphy's Law, which states that anything that can go wrong, will go wrong. This attitude ignores an entire class of possibilities that can be equally devastating: the threat due to intentional, nonrandom events caused by a

determined adversary. This latter issue has been referred to as *Satan's Law.* To read more about Satan's Law, refer to *Programming Satan's Computer* by, Ross Anderson and Roger Needham of the Cambridge University Computer Laboratory, see

http://www.ftp.cl.cam.ac.uk/ftp/users/rja14/satan.pdf

The ability to fail over to redundant infrastructure components, for example, can help mitigate the risk from Murphy but security countermeasures are typically needed to deal with Satan. This creates a link between security and availability which further relates to the resiliency attribute of an on demand operating environment.

Customer pain points

IBM is investing heavily in on demand computing. Clients need the flexibility and resilience such an environment offers, but they have concerns as to what this new computing model will do (or will not do). For example, the results of an on demand market intelligence survey found that security is the top concern on the minds of clients (see Figure 2).

> *More than any other single factor, the potential of e-commerce hinges on people's confidence that the network can keep confidential transactions confidential and private records private*.
>
> Lou Gerstner, Chairman, IBM Corp.

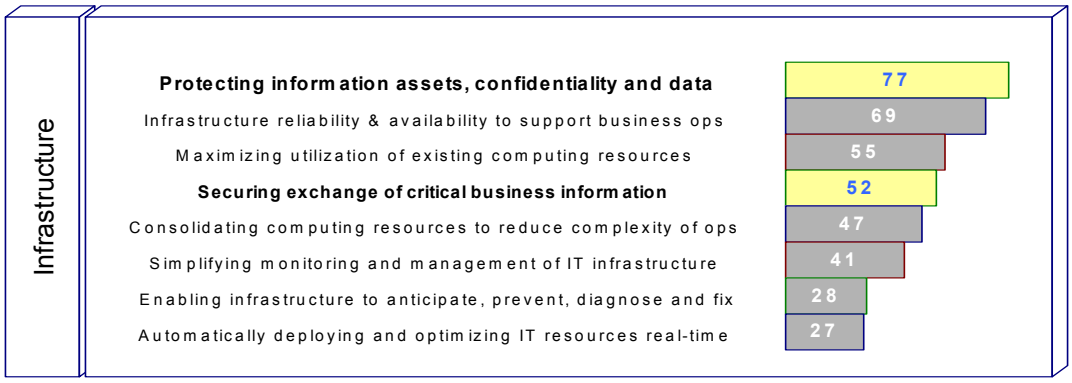

Figure 2 Importance of operating environment capabilities

The results of the survey were that two of the top four issues deal directly with IT security. Given the close relationship of security to availability, we suggest that actually three of the top four responses are related to security.

It has been widely reported that the lack of security in Web services standards is a primary reason that this functionality has been confined to small, intranet-based projects where risks can be minimized. Simply stated, if these and other security issues are not dealt with to the client's satisfaction, the promise of on demand computing will never be fully realized and IBM's business results will be seriously impacted.

A closer inspection of on demand computing reveals a few of these underlying client pain points. Pain points are discussed at length in “Identity pains” on page 5. While some of these are not entirely new to on demand, they become particularly acute in a cross-enterprise environment where disparate systems operate as one, despite being separated by an insecure, global Internet. Much could be done to assuage the client pains surrounding

security by addressing the problem of identity management, managing multiple user accounts across platforms and enterprises.

ODOE client goals

One of the biggest challenges facing ODOE clients today is the cost and complexity of user life cycle management. This is also referred to as a *multiple identity account problem*. This challenge becomes even more complicated when the problem space must scale up from that of a single enterprise's own internal systems to one that also includes interactions with the identity management and access control systems of other business partners, suppliers and customers. Customers need to:

- Improve and increase confidence in business transactions
- Establish identity as the basis for security

  Poor identity management means poor security.
- Lower administrative cost

  Costs soar with account and password administration, user registration and help desk support.
- Mitigate risk and meet compliance requirements by
  - Business, legal and privacy issues with user data (for example, Sarbanes-Oxley, HIPAA, Graham-Leach-Bliley, CA SB1386, Australia Privacy Act, and so on)
  - Management of unauthorized access from former employees, users and partners who no longer have a business need
  - Audit failures as a result of orphan account exposure
  - Identity theft
- Establish trust
  - No standard mechanism exists to trust identities from third parties.
  - There is high cost associated with integrating applications that deal with identities.

These issues all involve the subject of identity management.

Identity pains

The fundamental issue pervading identity management is that every time a user requests an account for access to an application or a system, an IT administrator must intervene manually. This results in a number of client pain points which can increase cost and dilute security:

- Account creation, deletion and update must be done separately in each target operating system, application, database, network access control system, and so on.
- User enrollment and registration must be handled manually.
- Determining the appropriate access rights for a given user on a particular system is handled on a case-by-case basis, increasing the likelihood of errors in application of the security policy.
- Additional help desk staff are required to deal with managing password resets and new access requests.
- A given user must remember many passwords because they are not synchronized across all systems.
- Users must manually sign on and off each system separately.

- The ability to track provisioning requests and approvals is lacking.
- Access rights are not promptly removed when users no longer have a business need for them.
- No single audit and enforcement point exists for ensuring that access rights comply with security policy.

All of these pains exist even within a single enterprise's own IT environment. However, when linking to the systems of other organizations, additional trust issues arise:

- Who is the authoritative identity provider for a given user?
- Do I trust that identity provider?
- What proof of identity will I require in order for a user to access my systems?
- Can I automate the process of provisioning accounts so that each user will not have to register separately?
- How will I know when to deprovision accounts for users for whom I am not the primary identity provider?

On demand security models

An important characteristic of an ODOE is the fact that processes and interactions are highly dynamic. The growth of partnerships in the on demand world is increasing steadily, leading to a corresponding increase in security challenges.

In a business-to-consumer (B2C) or business-to-employee (B2E) environment where consumers and employees communicate with one company as a focal point for multiple partners, it is important to secure access to all involved parties. In business-to-business (B2B) environments, partners and applications must also be used in a secure and reliable way.

Managing identities in this dynamic environment, with many different organizations interlinked, becomes problematic when using today's traditional, static models. For this reason it is necessary to organize *federations* in order to propagate identities across multiple organizations dynamically in a seamless management infrastructure.

In such a dynamic environment, *trust relationships* between partners are essential. Traditionally, IT infrastructures have dealt almost exclusively with their own environments. Those environments have not necessarily reflecting the needs of interoperation and integration with other parties. In an ODOE, all parties must interact seamlessly to meet the requirements of a dynamic business. A representation of the Security Triangle (Integration, Interoperate and Trust) is depicted in Figure 3.

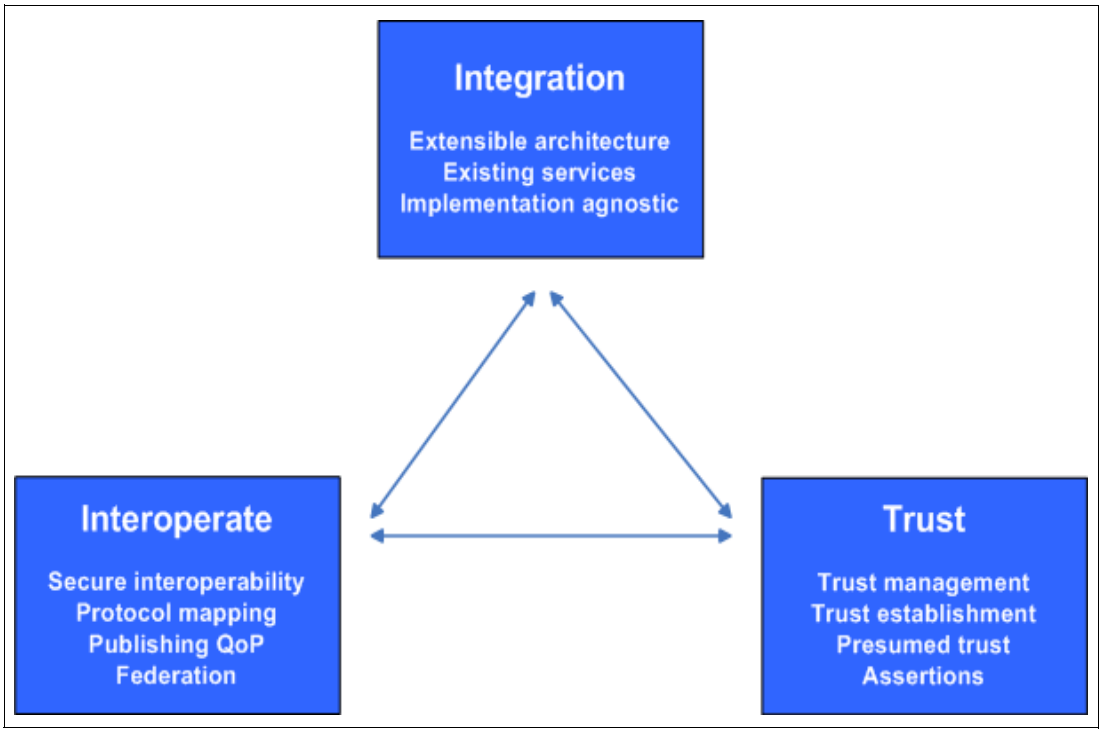

Figure 3 Trust as a fundamental requirement in a dynamic security environment

Traditional security issues still apply, but need to be expanded in many ways. In an on demand world, closer convergence of IT and interlocked business processes require flexible architectures to reflect the needs of these virtual organizations (see Figure 4).

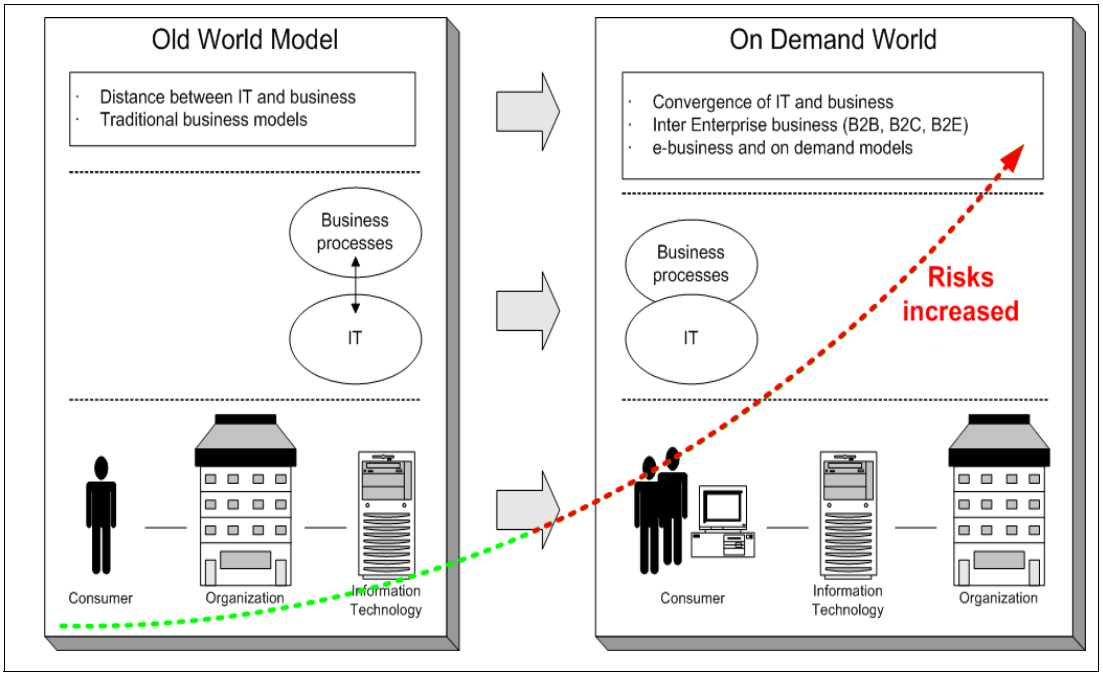

Figure 4 New on demand world paradigm for business process

Perhaps the most significant change is the move from a static security environment to a highly dynamic environment reflecting fast changes in this world. These new security

challenges span multiple organizations and are no longer bound to persons, but extend to applications and devices, as well.

Figure 5 and Figure 6 illustrate some of the differences between traditional identity management and federated identity management from a user's perspective.

In the traditional identity management environment shown in Figure 5:

- The user has to sign on to each system separately.
- Complexity grows with the number of back-end systems.
- The inconsistency of back-end systems inhibits dynamic processing.
- The approach is exclusively individual user-centric, with no application-to-application security and account management.

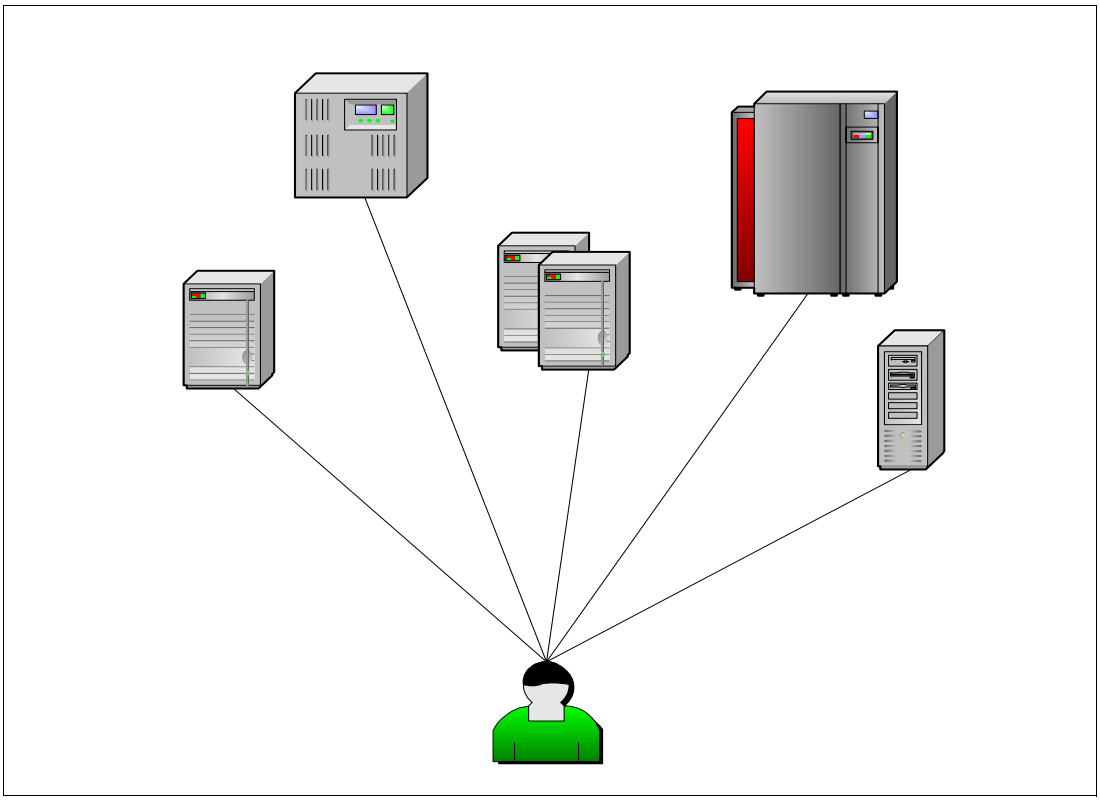

Figure 5 Traditional identity management environment

In the dynamic ODOE federated identity environment shown in Figure 6 on page 9:

- The user is only required to sign on once.
- A focal point acts as an Identity Provider (IdP) to other systems.
- Flexible provisioning and deprovisioning of user accounts is enabled.
- Application-to-application security is supported.

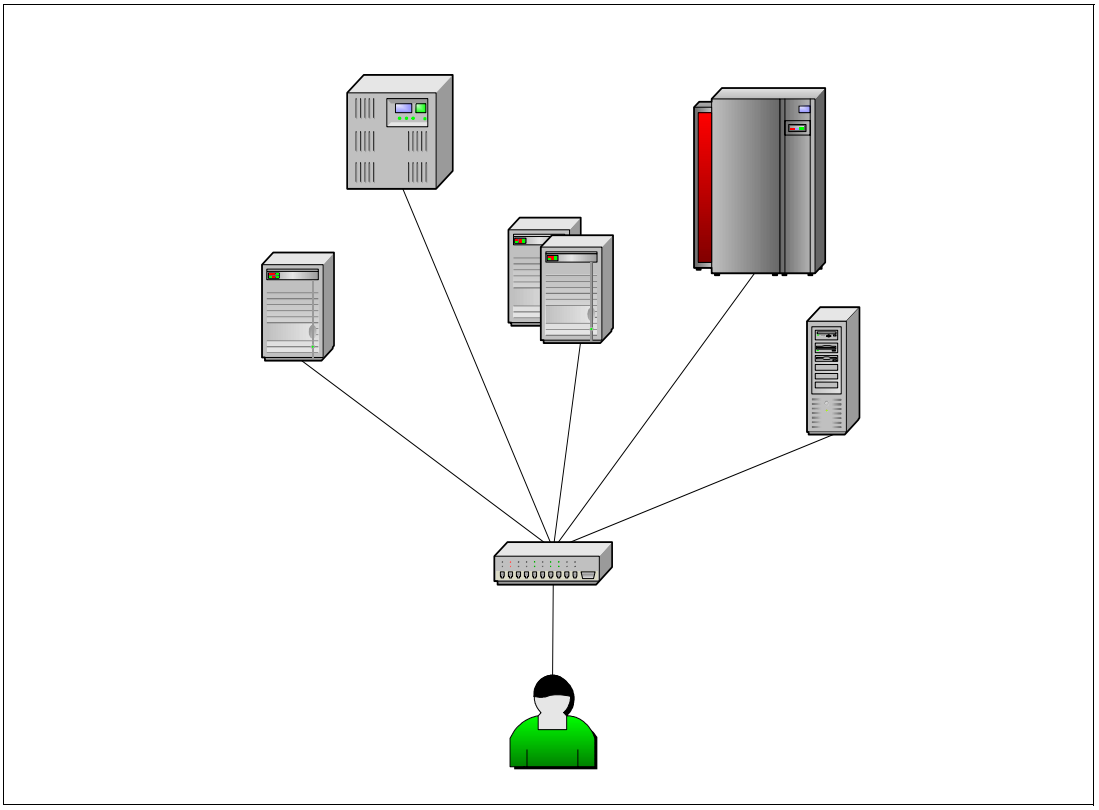

Figure 6 Dynamic environment - federated identity management

New federated identity management specifications that extend existing Web services standards form the basis for a solution to the new identity management issues that arise in an ODOE. These solutions are discussed in more detail throughout the remainder of this paper.

Part two: Standards and products

This section discusses the key concepts and needs of security reference architecture, IBM products and services that meet those needs and different organizations that have developed security standards.

On demand security reference architecture

Some initial work has been done to define a comprehensive on demand security reference architecture in the white paper, *On Demand Functional Security Architecture*, by Sridhar Muppidi of IBM. Figure 7 illustrates this work. A complete discussion of this architecture is beyond the scope of this paper, which focuses specifically on the areas of Web Services Security (WSS) and Federated Identity Management (FIM), circled in the diagram in Figure 7. However, brief descriptions of the background concepts of architecture are included in, “Appendix A: On demand logical security architecture” on page 51.

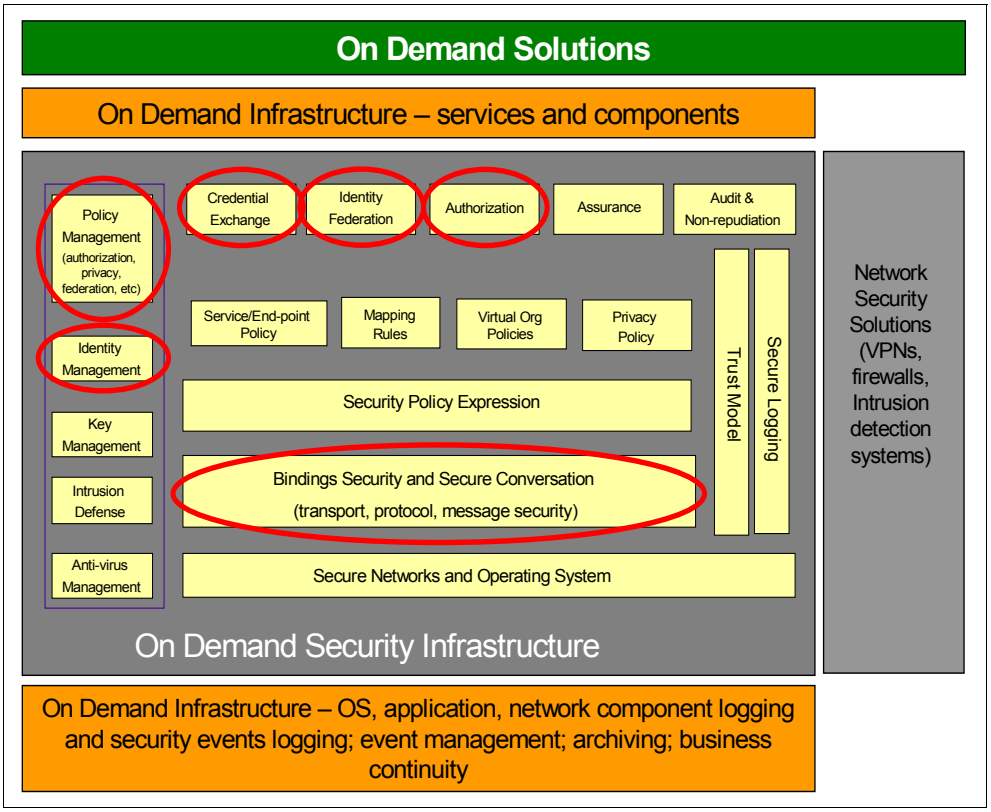

Figure 7 On demand security architecture (logical)

IBM's identity management offerings

IBM offers a scalable, standards-based identity management solution which can speed deployment and help reduce costs. Integration is provided with other IBM core technologies, such as IBM WebSphere Application Server, WebSphere MQ and Lotus Domino. Security can be extended to UNIX and Linux environments, providing a consistent security implementation across the organization. Figure 8 on page 11 illustrates how the products in this family build to form a comprehensive identity infrastructure.

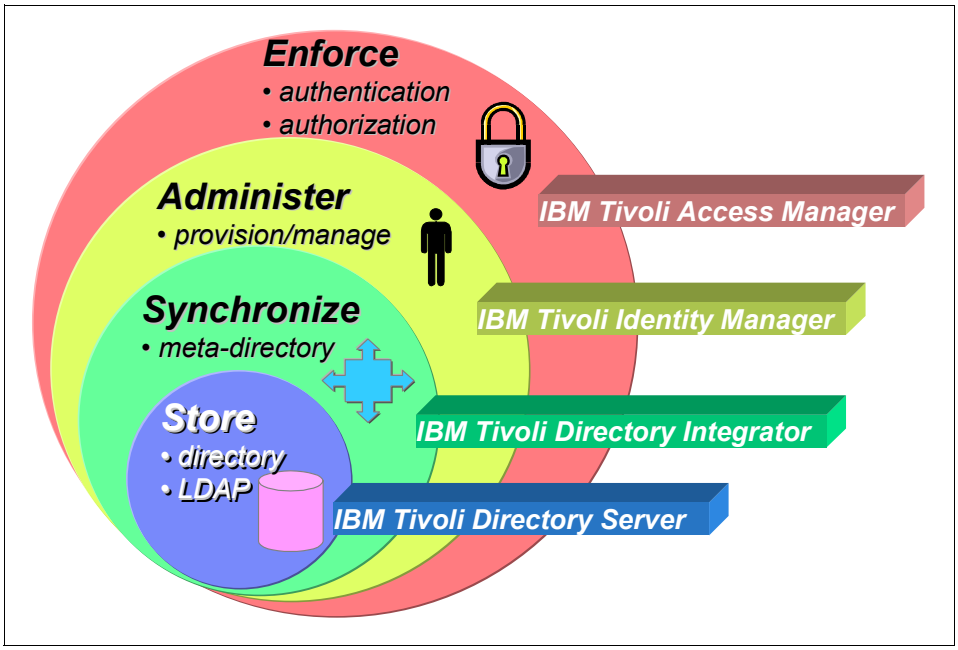

Figure 8 IBM's identity management offerings

The following IBM product descriptions were adapted from the Web site:

`http://www-306.ibm.com/software/tivoli/solutions/security/`

IBM Tivoli Directory Server

IBM Tivoli Directory Server provides a powerful *Lightweight Directory Access Protocol* (LDAP) V3 compliant directory which can serve as the foundation for an enterprise identity infrastructure. Directory Server includes strong management, replication and security features making it well-suited for mission critical, 24x7 installations.

IBM Tivoli Directory Integrator

IBM Tivoli Directory Integrator synchronizes identity data residing in directories, databases, collaborative systems, applications used for human resources (HR), customer relationship management (CRM), Enterprise Resource Planning (ERP) and other corporate applications.

By serving as a flexible synchronization layer between a company's identity structure and the application sources of identity data, IBM Tivoli Directory Integrator eliminates the need for a centralized datastore. For organizations that choose to deploy an enterprise directory solution, IBM Tivoli Directory Integrator can help ease the process by connecting to the identity data from the various repositories throughout the organization.

IBM Tivoli Identity Manager

IBM Tivoli Identity Manager provides a secure, automated policy-based user-management solution that helps address key business issues across both legacy and e-business environments. Intuitive Web administrative and self-service interfaces integrate with existing business processes to help simplify and automate managing and provisioning users. It incorporates a workflow engine and leverages identity data for activities such as audit and reporting.

Tivoli Identity Manager interacts directly with users and with two external types of systems: *identity sources* and *access control mechanisms*. The identity sources deliver authoritative information about the users that need accounts. The provisioning system communicates directly with access control systems to create accounts, supply user information and passwords and define the entitlements of the account. In reverse, local administrative changes made to an access control system are captured and reported to the provisioning system for evaluation against policy.

IBM Tivoli Access Manager

IBM Tivoli Access Manager for e-business integrates with e-business applications out-of-the-box to deliver a secure, unified and personalized e-business experience. By providing authentication and authorization APIs and integration with application platforms such as Java 2 Enterprise Edition™ (J2EE), Tivoli Access Manager for e-business helps you secure access to business-critical applications and data spread across the extended enterprise.

Web-based single signoff (SSO) can span multiple sites or domains by exploiting Access Manager cross-domain SSO technology or by using *Security Assurance Markup Language* (SAML) and other token-passing protocols.

IBM Tivoli Federated Identity Manager

Although not generally available at the time of this writing, IBM has developed a new product which extends identity management capabilities beyond the enterprise in order to support the new Web SSO, cross-enterprise provisioning and Web services security standards. IBM Tivoli Federated Identity Manager was announced for limited availability as a PRPQ in August 2004. More information about this product can be found in the IBM Redbook *Federated Identity Management with IBM Tivoli Security Solutions*, SG24-6394.

Competitive positioning

In the first half of 2002, the identity management marketplace was crowded with a number of competing vendor offerings. Over the course of the next 18 months, a number of key mergers and acquisitions resulted in significant consolidation of former niche players. As a result, some key competitors such as Netegrity, Sun, RSA, Microsoft and Oblix have emerged. IBM, however, has some significant advantages over the competition in this space.

Several analysts reported that IBM makes a strong case for one-stop solutions with offerings in almost every category of identity management.

- IBM Tivoli Access Manager is in the Leadership Quadrant of Gartner's Enterprise Access Management Magic Quadrant.
- IBM Tivoli Directory Server is in the Leadership Quadrant of Gartner's Directory Servers Magic Quadrant.
- IBM Tivoli Access Manager is the first Web access management product to achieve Common Criteria EAL3 (ISO 15408) certification.
- IBM Tivoli Directory Server is certified to Common Criteria EAL3 (ISO 15408).

IBM's strong portfolio of enterprise-class identity management products along with its contributions to industry standards (see Figure 9 on page 14) puts IBM in an excellent

position to lead as the technology advances to federated identity management. The following time line shows IBM adopting this technology almost a full year before its competitors.

June 2002	September 2002	February 2003	November 2003	December 2003
► IBM acquires Metamerge	► IBM acquired Access360	► Oblix partners with Microsoft	► Sun acquires Wave Set ► BMC partners with Business Layers	► Netegrity acquires Business Layers

Federated Identity Management standards

Interoperability is essential if identity management and Web services are to succeed in cross-enterprise environments. Federations of trusted partners need the ability to communicate a wide variety of transactions including provisioning, SSO, single signoff (SSOff), deprovisioning and purchasing in a secure manner. In order to achieve this level of interoperable security, a number of standards efforts have been introduced into the marketplace.

One of the first such efforts was by Microsoft with its *Passport* initiative. Passport was intended to serve as an SSO solution across Web sites on the Internet. This approach had the advantage of having the necessary client code embedded directly into the most commonly deployed desktop operating systems (Windows), giving Passport a considerable potential install base.

Passport was widely rejected, however, by security and privacy groups who pointed out problems that could arise from having a single broker of all trusted credentials. Few could agree on any single organization which would have the universal, world-wide appeal to serve as the keeper of the credentials for the global Internet. Concerns of this sort were exacerbated by the fact that a private company known for heavy-handed, monopolistic practices was appointing itself to such an important role.

Sun Microsystems led a counter movement called the *Liberty Alliance* which intended to quash Microsoft's efforts by using a more open, distributed approach. Instead of a central credential store controlled by a single organization, Liberty proposed a federated approach which spread such power among a variety of business partners.

Rather than getting caught up in what had become a rancorous debate being waged in the trade press, IBM chose to work, instead, with some of the key players such as Microsoft and VeriSign to build a bridge that would span the chasm of competing interests with a far more inclusive, industry standards-based approach. The result of that effort was the *Web Services Security* specifications which are the subject of much of this paper.

In addition, IBM announced in July 2004 that it had signed a deal with France Telecom for the largest implementation (50 million users) of identity and account management based on the Liberty Alliance specification to date. This deal, along with the completion of Liberty conformance testing a month earlier, underscored IBM's commitment to openness and interoperability, allowing clients to choose the standards they feel best suit their requirements.

The following is a more in-depth description of some of these and other related federated identity management and Web services security initiatives. Figure 9 depicts an overview of the federation standards in progress.

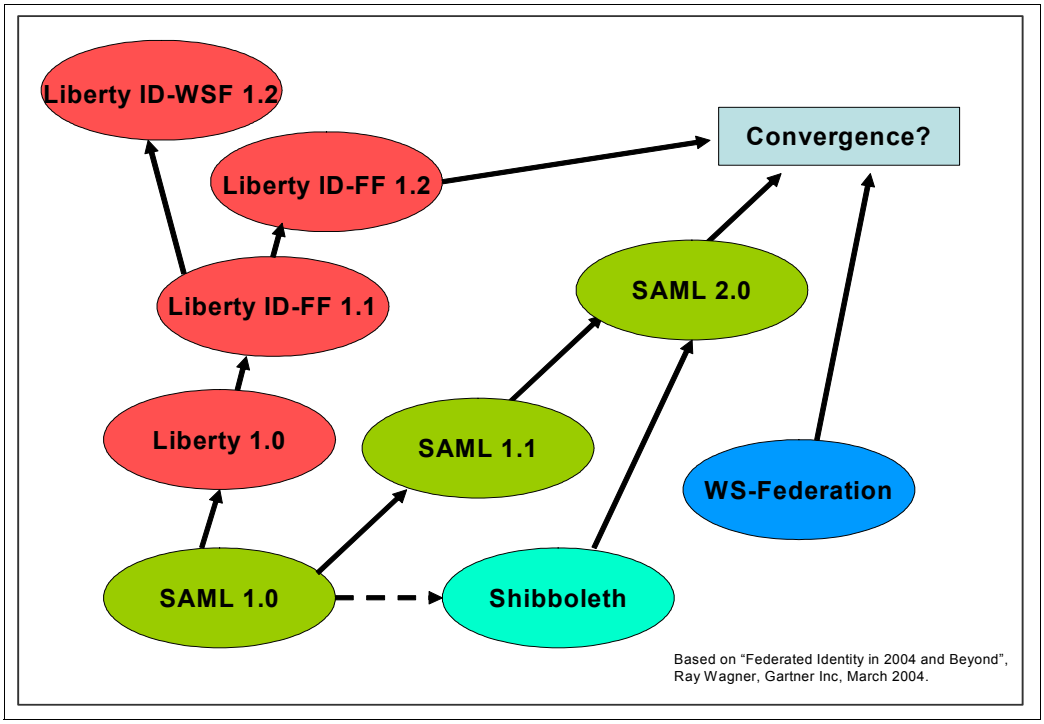

Figure 9 Federated Identity Management standards

SSL/TLS

Secure Sockets Layer (SSL), standardized as Transport Layer Security (TLS), provides session-level security through the use of encryption. While not often thought of as an identity-management protocol, SSL can be used to authenticate senders and receivers through digital certificates, verify data integrity and ensure confidentiality. As such, SSL is often the first and only option considered in securing transactions over the Internet. SSL can be used in both browser-to-Web server and server-to-server communications.

Despite its popularity, SSL has some shortcomings in the following areas:

- Granularity

  Either all the data over the session is encrypted or none is. This can impact throughput in cases where large amounts of data are exchanged, but only small portions of data actually need to be encrypted and decrypted.

- End-to-end

  SSL protection ends if intermediate components need to examine transactions. No provision is made for encrypting end-to-end across intervening components.

WSS overcomes these issues.

SAML

Security Assertions Markup Language (SAML) is a specification designed to provide cross-vendor SSO interoperability. SAML was developed by a consortium of vendors,

including IBM, under the auspices of OASIS through the OASIS SSTC (Security Services Technical Council). SAML has two major components:

- SAML assertions are used to transfer information within an SSO protocol.

  A SAML assertion is an XML-formatted token that is used to transfer user identity and attribute information from a user's IdP to trusted Service Providers (SPs) as part of the completion of an SSO request. A SAML assertion provides a vendor-neutral means of transferring information between federation partners. As such, SAML assertions have a lot of traction in the overall federation space.

- SAML protocols are specified as bindings and profiles.

  SAML also defines protocols for implementing SSO. These protocols are HTTP-redirect-based and involve the user's browser. SAML defines two HTTP-based profiles for these SSO protocols:

  – Browser/Artifact

    With the Browser/Artifact profile, information is transferred both through HTTP redirection and through a direct, SOAP-over-HTTP back channel. With this profile, a SAML assertion is not transferred within the GET-based redirection flows. Instead, a SAML artifact is sent and used as a pointer to the SAML assertion. The assertion is then retrieved using a back-channel XML-over-HTTP request.

  – Browser /POST

    With the Browser/POST profile, the SAML assertion is included in an HTML form as part of the browser-based HTTP POST flows, thus providing a front channel means of transferring the SAML assertion.

Despite marketing claims by competitors to the contrary, SAML is not rich enough to support federations by itself. SAML does not address such issues as trust, user management, and privacy. SAML assertions are, however, rich enough to provide a *vouch-for* token for use within cross-domain SSO scenarios and across federation. SAML also can be leveraged within a larger WSS context to address federation requirements. In fact, the OASIS SSTC, together with the WSS effort, has defined a Web Services Security SAML Token Profile which describes how to bind a SAML assertion in the context of WSS:SOAP Message Security for securing SOAP message exchanges.

Shibboleth

Shibboleth, a project of the Internet2/MACE initiative, is developing architectures, policy structures, practical technologies, and an open-source implementation to support inter-institutional sharing of Web resources subject to access controls. Shibboleth, which is based upon SAML, also introduces the notion of *Where are you from?* processing allowing a service provider to implement both push and pull-based SSO protocols. In addition, Shibboleth will develop a policy framework that will allow interoperation within the higher education community, which is important because there are very strict rules on the release of information about an institution's students even to other higher-education institutions. IBM is a main member of the Shibboleth project. For more information, refer to

http://shibboleth.internet2.edu/

Liberty

The Liberty Alliance Project is a consortium formed to deliver and support a federated network identity solution for the Internet that enables SSO for consumers and business users in an open, federated way.

The Liberty Identify Framework (ID-FF) describes federation functionality that goes beyond SSO. Liberty ID-FF profiles also include: Single Logout (SLO), Register Name Identifier (RNI), Federation Termination Notification (FTN), and Identity Provider Introduction (IPI). The Liberty specified common user identifier (CUID) is referred to as a *NameIdentifier*. It is an opaque reference to an account that acts as an alias, meaning that it cannot be used to infer information about the user such as their identity. A Liberty NameIdentifier is used to establish and maintain the account linking between an IdP and an SP. The RNI profile allows a reset of a user's NameIdentifier, replacing a current value with a new NameIdentifier value. The FTN process removes all references to a NameIdentifier, achieving account delinking. Taken together, these profiles are intended to provide richer user management functionality within a federation, rather than a simple SSO.

The Liberty approach is based on business affiliates forming *circles of trust*. This term has two main, but different uses. First, a circle of trust refers to a group of partners (both IdP and SP) that have a trust relationship. In general this does not imply a pairwise trust relationship across all federation partners, but simply a pairwise trust relationship between all SPs and a given IdP. Within a circle of trust, a user is able to SSO from an IdP to any participating circle of trust SPs. The other definition of circle of trust refers to a common DNS alias that is shared by all circle of trust participants. This common DNS alias implements a form of Where are you from? functionality.

For more information about the Liberty approach, see

http://www.projectliberty.org

Web Services Security road map

On 11 April 2002, IBM, Microsoft, and VeriSign cooperated on a project announced in the press release titled *IBM, Microsoft and VeriSign Announce New Security Specification to Advance Web Services.* This press release announced the publication of a new Web services security specification. This specification is intended to *help organizations build secure, broadly interoperable Web services applications.* The announcement revealed a road map which included a set of seven specifications designed to build upon the existing SOAP standard, *an XML-based industry protocol for accessing Web services in a platform- and language-independent manner*. For more information, refer to the IBM Press Room at

http://www-1.ibm.com/press/PressServletForm.wss

Click **Press Releases** on the left.

The complete family of WSS specifications provides companies with a standards-based interoperable secure digital identity and trust platform for Web Services-based architecture. Further, these specifications promote reusability of existing IT security investments, enabling companies to work with multiple security token types and multiple scenarios including vanilla browsers, enhanced browsers, active clients and application-to-application connectivity.

The components of this road map of specifications (see Figure 10 on page 17) can be described briefly as follows:

- WS-Security: How to attach signature encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets (an encryption system developed at MIT), to messages.
- WS-Policy: The capabilities and constraints of security and other business policies on intermediaries and endpoints (for example, required security tokens, supported encryption algorithms and privacy rules).

- WS-Trust: A framework for trust models that enables Web services to securely interoperate.
- WS-Privacy: A model for how Web services and requesters state subject privacy preferences and organizational privacy practice statements.
- WS-Secure Conversation: How to manage and authenticate message exchanges between parties including security context exchange and establishing and deriving session keys.
- WS-Federation: How to manage and broker the trust relationships in a heterogeneous, federated environment including support for federated identities.
- WS-Authorization: How to manage authorization data and authorization policies.

For more detail see:

http://www.ibm.com/developerworks/library/ws-secmap/

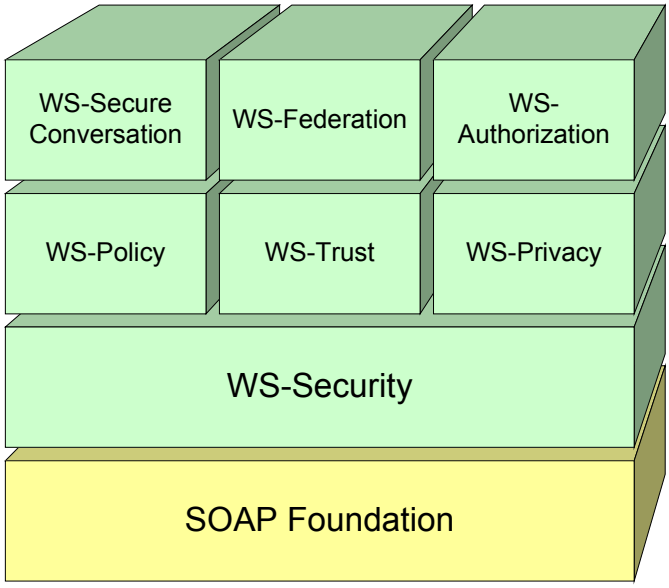

Figure 10 Web services security specifications

WS-Federation

Of particular note among the Web Services Security road map specifications is WS-Federation. The interoperability of this specification between IBM and Microsoft has been demonstrated several times, including by Bill Gates and Steve Mills in New York City in September of 2003. Subsequent to that, a public interoperability exercise was held on March 29-30, 2004 between IBM, Microsoft and other third-party vendors.

WS-Federation describes how to use the existing Web services security building blocks to provide federation functionality including trust, SSO and SSOff as well as attribute management across a federation. WS-Federation is really a family of three specifications, WS-Federation (WS-FED), WS-Federation Passive Client (WS-FEDPASS) and WS-Federation Active Client (WS-FEDACT).

WS-Federation Active Client describes how to implement federation functionality in the active client environment. Active clients are those that are Web services-enabled, able to issue Web services requests and react to a Web services response. Leveraging the Web services

security stack, WS-Fed Active describes how to implement the advantages of a federation relationship, including SSO, in an active client environment.

WS-Federation Passive describes how to implement federation functionality in a passive client environment. A passive client is one that is not Web services enabled. The most commonly encountered example of a passive client is a vanilla HTTP browser. WS-Federation Passive describes how to leverage the advantages of a federation relationship such as single signoff in a passive client environment. Because this solution leverages the WS-Security foundation of the infrastructure support, the same components used to provide a passive client solution may be leveraged for an active client solution.

The three specifications that make up WS-Federation are available for download from IBM DeveloperWorks at:

- WS-FED

  http://www-106.ibm.com/developerworks/webservices/library/ws-fed/
- WS-FEDACT

  http://www-106.ibm.com/developerworks/webservices/library/ws-fedact/
- WS-FEDPASS

  http://www-106.ibm.com/developerworks/webservices/library/ws-fedpass/

Web Services Security architecture

WSS provides a flexible mechanism for secure process-to-process communications such as those used in complex, extended enterprise applications. It provides a basis for building secure, interoperable solutions on heterogeneous systems across enterprises using standards-based Web services, as well as device independence by isolating security mechanisms from function security and business logic.

Web services applications communicate using SOAP messages which are defined in XML. These messages are most often transported over HTTP, but, in principle, can be carried over any underlying protocol. WSS provides a framework for encrypting, signing and authenticating SOAP messages. A high level depiction is shown in Figure 11.

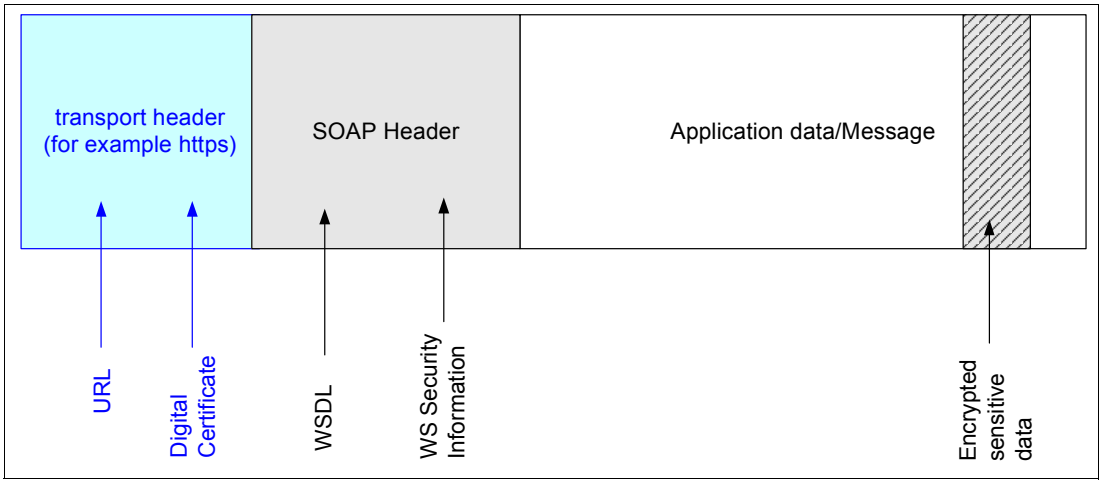

Figure 11 Web Services Security packet layout

WSS defines XML elements that can be used to provide confidentiality, integrity and authentication. It does this by using other specifications while adding some key elements of

its own. For example, WSS implements digital signatures and encryption by referencing the XML digital signature and XML encryption recommendations developed by the World Wide Web Consortium (W3C).

A security element for each node that receives the message appears in the SOAP header. The information in this element lets a sender tell a receiver what data has been signed, what has been encrypted, the order in which to perform the operations and what keys to use. WSS also lets senders specify the creation and expiration date and time of security information.

WSS does not specify a particular set of message exchanges and cryptographic operations the way Kerberos or SSL/TLS do. It is expected that other specifications will describe detailed usage patterns and even add elements to implement and optimize complex capabilities.

WSS can be used directly to meet simple needs, such as signing or encrypting portions of a single message such as that shown below in Figure 12.

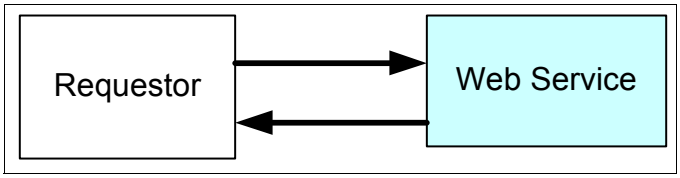

Figure 12 Simple Web Services flow

Alternative mechanisms such as SSL/TLS can be used to protect Web services messages. For many applications they might prove satisfactory, but there could be limitations. For example, even in this simple scenario, use of SSL could result in a security exposure on redirection, such as that encountered on proxies because data may be decrypted by the proxy before reaching the Web service provider.

WSS extends the security provided by such traditional transport level security mechanisms. It allows for signing and encrypting of sensitive parts of a message so that only the endpoint may decrypt it and verify its integrity as shown in Figure 13.

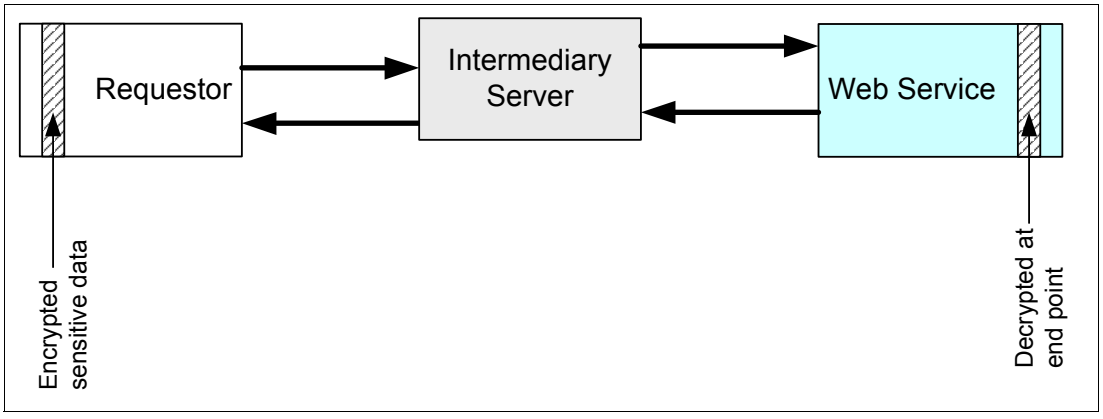

Figure 13 Web Services flow with intermediary

It is possible for a message to be signed at one node, additional data to be added and signed by a second node, and finally verified by the receiver. An intermediate node can process portions of the message, even though other portions have been encrypted. As Web transactions become more complex, and intermediaries of varying levels of trust become involved, the need for selective encryption and signing becomes more important both for security and performance reasons.

For instance, if only one KB of a 50 MB message needs to be encrypted, SSL session-layer security has no means to economize, as all traffic over the session is equal and, therefore, all incur the overhead necessary to do the encryption. WSS provides better granularity by encrypting only those parts of the message that need that level of protection, thereby reducing overhead substantially. However, these savings might not be realized if a large number of messages are exchanged because a new public key operation must be performed for each message. With SSL, a single public key operation is done at setup and a session key is used. At the time of this writing, WebSphere Application Server does not yet support the sort of element-level encryption described here, but plans call for this to be included in the 6.0 release.

Because Web services will be used in many environments, WSS makes it possible to use various systems to distribute keys and other authentication information such as security tokens, X.509 certificates and Kerberos tickets, which are carried in binary tokens, while SAML assertions and extensible rights markup language, (XrML) licenses are represented as XML tokens. WSS also defines a user-name token, which might be used in conjunction with a password.

WSS specifications allow for access to a token service, allowing additional flexibility in the types of tokens that can be used in a Web services interaction. Depending upon the level of trust, WSS also allows for a validation service to be used by the Web Service provider. These are shown in Figure 14.

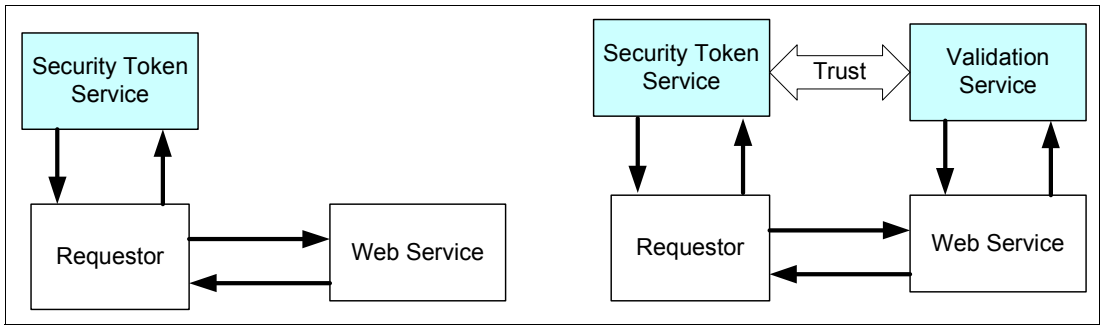

Figure 14 Security token and validation services

WSS specifications also allow for federation of identities, as shown in Figure 15.

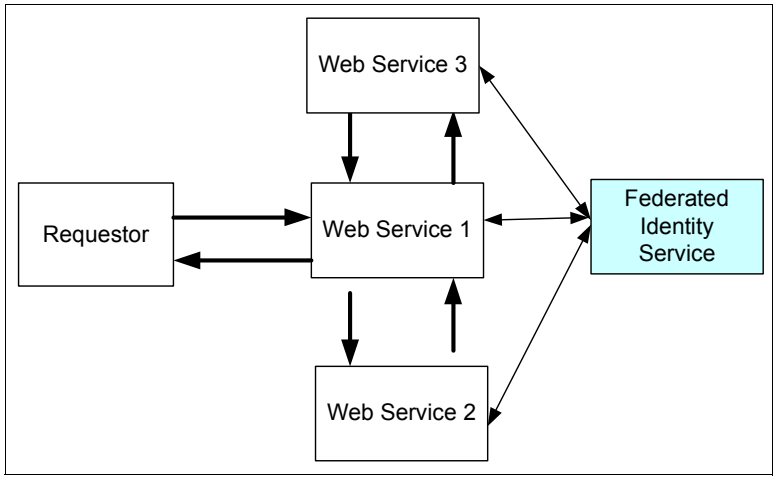

Figure 15 Federated identity service

The benefit of the architecture in Figure 15 on page 20 is complex transactions between multiple Web services and potential service providers operate in a transparent security context, even though they may have different security definitions and requirements.

Current status

As shown in Figure 16, the WSS capabilities build on a foundation of SOAP and WS Security. Currently the specifications for SOAP Message Security, Username Token Profile and X.509 token profile have been approved by OASIS.

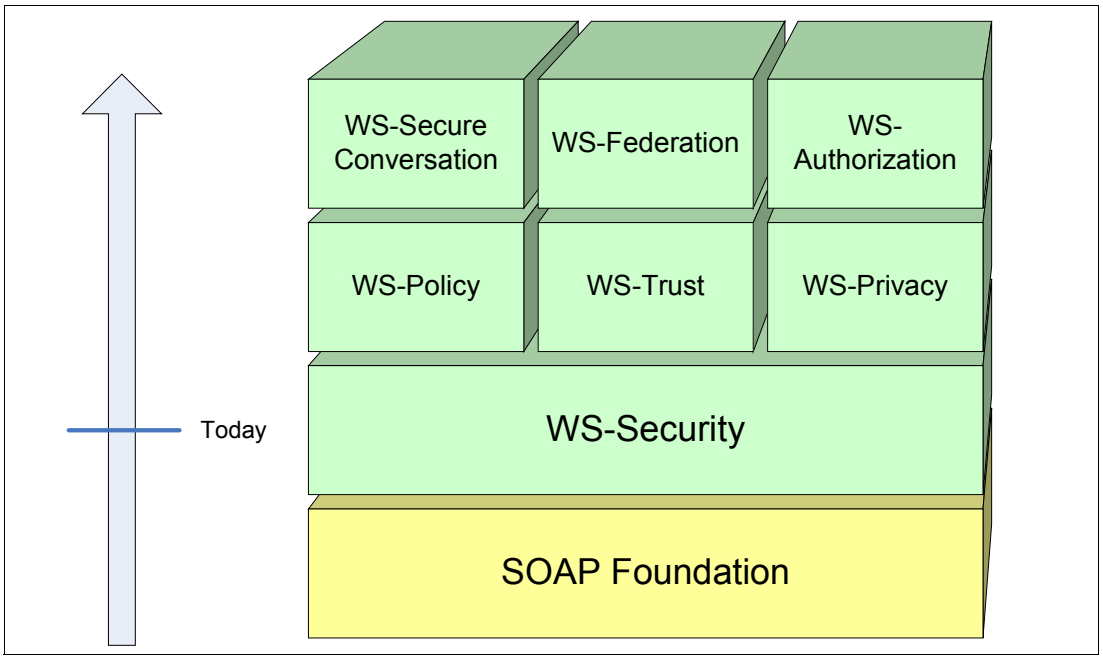

Figure 16 WS-Security road map

From an adoption standpoint, leading-edge customers are starting to work with WS-Security and develop best practices. Current implementations generally involve some amount of static definitions and negotiations with partners, as well as relatively homogeneous tokens. More dynamic identity environments and heterogeneous tokens are expected to evolve as industry professionals gain more practical experience with these technologies.

Product capabilities

WebSphere Application Server Network Deployment V5.1 provides a feature known as the *Web Services Gateway*. The WebSphere Web Services Gateway allows an enterprise to safely expose internal business processes as Web services to external users, or to safely use external Web services from internal business processes.

The latest release of the Gateway, available in WebSphere Application Server Network Deployment Version 5.1, adds new features that support improved performance, standard mechanisms for mediation of Web Services messages and improved flexibility. The Web Services Gateway currently provides basic Web Services Security as discussed earlier in this section. It supports password and digital certification tokens. However it does not currently support credential mapping or identity federation. In the future product packaging might be different. For instance, this function could be included in TFIM rather than the Web Services Gateway,

In addition, clients that have Tivoli Access Manager installed can map URLs to Web Services and use Access Manager to protect these Web Services at a URL level. Access Manager and WebSphere Web Services Gateway can be used either separately or together. The latter option involves having Access Manager use the HTTP header while leaving SOAP header or envelope processing to the Web Services Gateway. Variations on these alternatives are discussed in “Scenario one: B2B Web Services security” on page 25.

FIM architecture

Federation: A group of two or more organizations that have agreed to allow a user or application from one federation member to access internal resources from another member in a secure and trustworthy manner.

A major inhibitor of Federated Identity Management (FIM) is the lack of a trust mechanism that allows a view of the federated space as a single entity. Without this technology, federation is ineffective. Authentication is still required by each partner and each partner must manage identities in its own identity life cycle process. A new FIM solution provided by IBM under the Tivoli brand fulfills the promise of FIM by delivering this mechanism.

Basic FIM architecture pattern for browser-based activity

FIM solutions will not be adopted if they impose onerous architectural constraints and prerequisites on an existing environment. The IBM Tivoli FIM solution is designed to be modular and easily built, minimizing the effort required to integrate the solution into an existing environment. The basic pattern for a FIM architecture is shown in Figure 17.

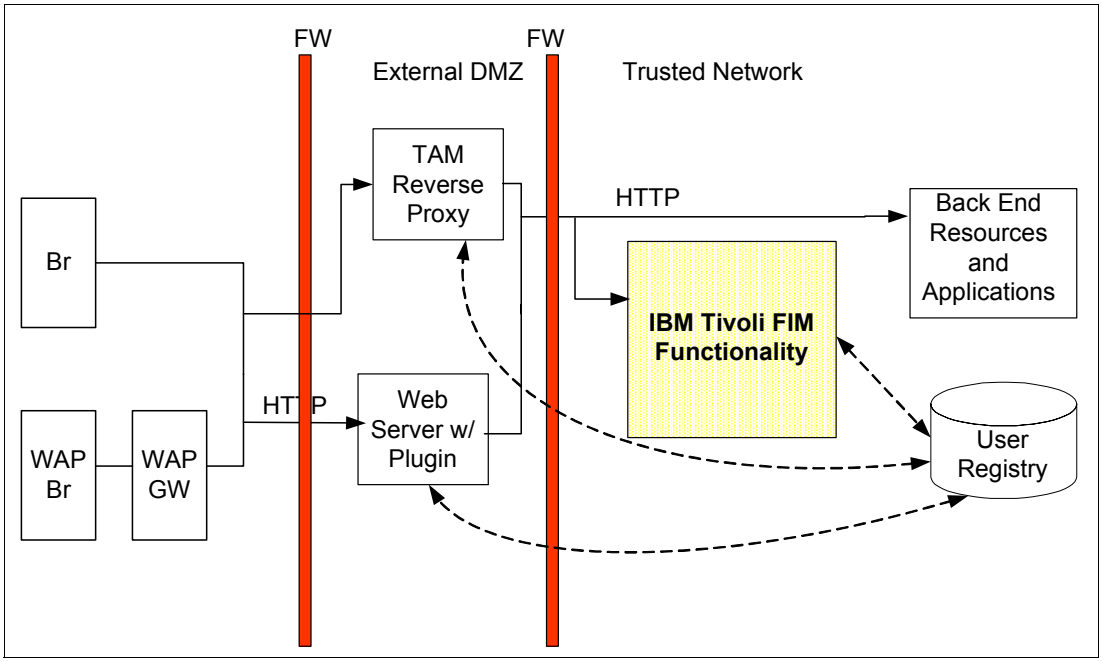

Figure 17 Generic FIM architectural pattern

In this figure the IBM Tivoli FIM functionality is required to integrate with the HTTP Point of Contact (PoC), for example the Access Manager reverse proxy, as well as a local user registry. This architecture has been designed to minimize the impact and integration requirements on an existing environment. The components in this figure, their roles, and the implications of a FIM solution on that component's functionality are as follows:

General browser requirements

A browser provides an interface between the user and the infrastructure. In order to participate in FIM protocols, the browser must be capable of supporting at least one of redirects and automatic forwarding (POST) of requests by JavaScript or ActiveX.

WAP browser requirements

A WAP browser, a browser used by a mobile device, might have limitations, such as query-string size, not found with Internet-based browsers. Specific techniques to address WAP use cases exist for FIM functionality and are described in more detail in the following sections.

HTTP point of contact

The Point of Contact (PoC) is a generic component normally located in the *demilitarized zone* (DMZ), a neutral server or network between a trusted network and an untrusted one. PoCs will generally provide authentication, authorization and session management services. With the introduction of FIM based solutions, including federated SSO, a PoC will continue to provide direct authentication services, meaning direct evaluation of user presented credentials such as userid and password. The PoC will, in general, delegate the runtime implementation of federated SSO protocols to a separate FIM component. The PoC will continue to provide authorization and session management services for all users, whether they are directly authenticated or leverage federated SSO solutions.

Tivoli Access Manager, for example, is an HTTP PoC that implements Authentication Services, Session Management Services, and Authorization Services. It is also able to invoke (when required) SSO services as part of the implementation of federation identity management functionality.

FIM functionality

From a high-level point of view, FIM functionality can be viewed as a single black box component. It contains several logical components that are discussed in detail later in this paper, including the protocol runtimes and trust infrastructure support. This component provides the FIM protocol implementations required for SSO, account linking and so on. This component must communicate with the HTTP PoC for the purposes of completing SSO and SSOff functionality. It must also integrate with a data store user registry for management of the federation common user identity (CUID) used in all of the federation tasks. Note that the FIM component must also be treated as a stand-alone application (endpoint) for some federation tasks that do not require interaction with the PoC, such as account delinking.

The IBM Tivoli FIM functionality implements the full set of federated user life cycle functionality, addressing the user life cycle from account linking to session creation and account delinking. This is logically handled through two main service components; a *single signoff protocol service* (SPS) and a *trust service* (TS). The trust service, in turn includes, a logical security token service (STS).

The SPS provides support for the protocol runtimes required for federated functionality, such as account linking, single signoff, single sign-out, and account delinking.

The TS and STS provide support for the evaluation of assertions exchanged within a federated protocol runtime. This evaluation includes:

- Validation of trust relationships
- Validation of assertion formats
- Identity and attribute mapping to translate a received token into a locally valid token with locally valid attributes

User registry

IBM Tivoli FIM functionality requires some form of user registry or data store for maintaining the CUID-local identity mapping. This user registry may be implemented as an internal, private FIM registry that is used by FIM only (minimizing integration requirements with local user registries). IBM Tivoli FIM functionality may also leverage existing user registries. Leveraging an existing user registry allows the FIM components to access additional, partner-maintained user attributes that may be required as part of the user's CUID-local identity mapping.

Note: FIM can also integrate with non HTTP points of contact when brokered by entities such as WebSphere Application Server. For a more detailed discussion of the Tivoli FIM solution and FIM generally, see *IBM Federated Identity Management*, a white paper by Hinton, Nadalin, Nagaratnam, and Raghavan.

Part three: Scenarios

In this section, two scenarios are presented involving a fictitious company called JK Toy Distributors. These scenarios illustrate some of the key security issues and solutions in an on demand environment.

- The first scenario highlights security aspects of transactions among business partners over the Internet using WSS standards.
- The second scenario involves managing employee access to out sourced financial and human resources functions and focuses on how identities can be managed in a federated environment.

Scenario one: B2B Web Services security

This scenario provides a high-level discussion of the security considerations involving Web services in a business-to-business, supply-chain management environment. It addresses securing the exchange of critical business information - a significant security issue with customers.

The customer pain points on FIM and Web Services security from *Federated Identity Management and Secure Web Services*, REDP-3678 address the following:

- Inability to drive revenue from new business models by leveraging tighter relationships with customers, suppliers, business partners, and so on.
- Poor ROI resulting from high transaction costs of interconnecting enterprise systems with partner systems using the Web.
- Lack of a security framework for conducting cross-enterprise business.

The intent of the following set of alternatives is to provide the architect with some decision making information. The discussion is at an architectural and positioning level. It does not include implementation level details. The implementation details for individual products are available from other sources.

Business context

JK Toy Distributors is a mid-sized distribution company as shown in Figure 18 on page 26. Their customers are primarily other businesses. These include large retail enterprises such as RBG Stores as well as smaller family businesses such as G&G Toys. JK Toy Distributors currently has no direct retail business. JK Toy Distributors' current Web presence is primarily informational.

JK Toy Distributors has long term, on-going relationships with its business customers. These customers are segmented according to the level of business conducted. Prices, terms and conditions, and so on, are customized for each customer based on this segmentation model. Customers place orders with JK Toy Distributors when their inventory levels fall below a predetermined threshold. Payments are made through purchase orders, purchase cards, and so on, with terms based again on the segmentation model.

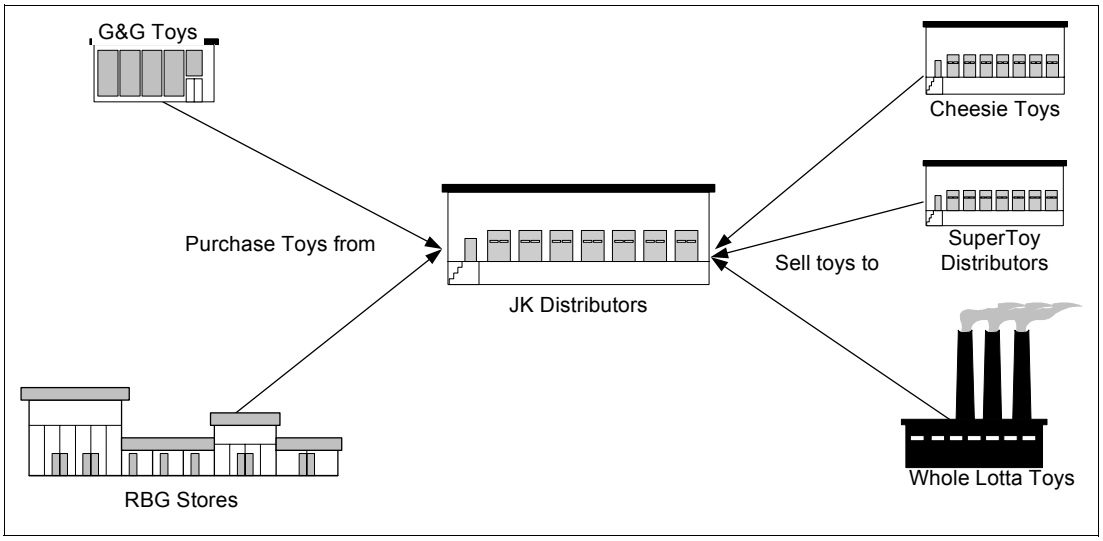

Figure 18 JK Toy Distributors business context diagram

Upon receiving an order, JK Toy Distributors fulfils the order through its inventory maintained in its warehouses or through its relationships with multiple suppliers.

Internally, JK Toy Distributors operations are segmented primarily into the following units and lines of business:

- Customer Relations, Sales and Marketing
- Procurement and Supplier Relations
- Warehousing and Inventory Management
- Support Operations - Finance, HR, IT, and so on

Currently there is minimal integration across the lines of business. Communication with partners (customers and suppliers) is primarily through manual channels such as phone, fax and so on. An ERP system is used for inventory, warehousing, accounting and financial reporting.

There are several factors driving change in JK Toy Distributors' business environment:

- JK Toy Distributors' customers are attempting to reduce inventory costs.
- JK Toy Distributors needs faster response to orders from their suppliers.
- JK Toy Distributors wants to reduce the costs of ordering through more automation.
- Larger customers such as RBG Stores had initially looked at asking distributors like JK Toy Distributors to integrate into their legacy purchasing systems. However given the diversity of suppliers and systems, most of JK Toy Distributors' customers are looking at more open methods to integrate their purchasing with partners.

In order to stay competitive. JK Toy Distributors is, in turn, considering a similar approach with their suppliers.

Any new system adopted by JK Toy Distributors must:

- Integrate with the new channel with minimal change to existing business processes and systems within JK Toy Distributors.
- Ensure sensitive customer information is protected.
- Provide maximum flexibility for the interaction so that this model is extensible and scalable to new partners coming online.

- Provide a strategic growth path as the business need requires.

As a result, a service-oriented architecture (SOA) approach based upon open Web services standards was chosen.

Web services

The Web services used in this scenario are located at both JK Toy Distributors and each of JK Toy Distributors' suppliers. For simplicity each supplier will name their own order entry Web service orderEntryService.

JK Toy Distributors will publish the following services, which will be used by its business partners:

- **AvailabilityQuery**: Query the JK Toy Distributors product catalog for availability on items of interest.
  - Sends: queryNum, itemNum, itemName, qtyRequired, and dateRequired.
  - Returns: queryNum, itemNum, itemName, qtyOnHand, dateAvailable and price.
- **OrderCancellation**: Cancel a catalogQuery request in the event the dateAvailable is not adequate for the customer.
  - Sends: queryNum.
- **SubmitPurchaseOrder**: Submit a purchase order to the JK Toy Distributors Inventory Control (IC) system. Once the items and quantities requested have been committed, this Web service returns an order confirmation document.
  - Sends: purchaseOrderNum, itemNum, itemName, qtyRequired, and dateRequired.
  - Returns: purchaseOrderNum, orderStatus (rejected OR confirmed).
- **SupplierQuery**: Query the product catalog for availability on items of interest.
  - Sends: queryNum, itemNum, itemName, qtyRequired, and dateRequired.
  - Returns: queryNum, itemNum, itemName, qtyOnHand, dateAvailable and price.

  Note: Each supplier may have its own unique form of a SupplierQuery Web service.

- **SupplierPurchaseOrder**: Submit a purchase order to JK Toy Distributors suppliers' systems. Once the items and quantities requested have been encumbered this Web service returns an order confirmation document.
  - Sends: purchaseOrderNum, itemNum, itemName, qtyRequired, and dateRequired.
  - Returns: purchaseOrderNum, rejected OR confirmed.

  Note: Each supplier may have its unique form of a SupplierPurchaseOrder Web service.

Business considerations

There are several business situations to consider:

- Customers (for example, RBG Stores & G&G Toys) receive different pricing due to volume discounting. As a particularly large retailer, RBG Stores' pricing is specially negotiated and not on the standard volume discount methodology.
- RBG Stores and G&G Toys are offered a different set of toys to choose from. Some toys are offered to both and some are unique.

- Some of RBG Stores' prices are negotiated directly with the manufacturer and are not exposed to JK Toy Distributors. JK Toy Distributors is allowed a pallet handling fee, but the product cost is not exposed to JK Toy Distributors.

Basic use case

The system involves interactions between the *partner* (customer and supplier) systems and those at JK Toy Distributors. These interactions are based on Web Services standards in order to provide maximum flexibility and loose coupling between the partner systems and those at JK Toy Distributors.

The interface specifications are negotiated with each partner based on the use cases described above. There are no third-party trust relationships or intermediaries involved in processing for the use cases discussed above. X.509 digital certificates are used as tokens with all partners for authentication and XML signatures.

JK Toy Distributors presents a Web Services interface to its partners through a *point of presence*. This point of presence provides the following functions:

- Performs security functions
- Abstracts JK Toy Distributors internal processes and systems from partners
- Off-loads security processing from Web Services providers within JK Toy Distributors' infrastructure

From a security perspective, the functions JK Toy Distributors provides are:

- Processing of X.509 certificates as tokens, authentication and mapping to appropriate Web Services
- Encryption and decryption of sensitive information at the end-points as provided for in the use cases

The following use case, depicted in Figure 19 on page 29, illustrates the fundamental supply chain process flow model for the JK Toy Distributors environment.

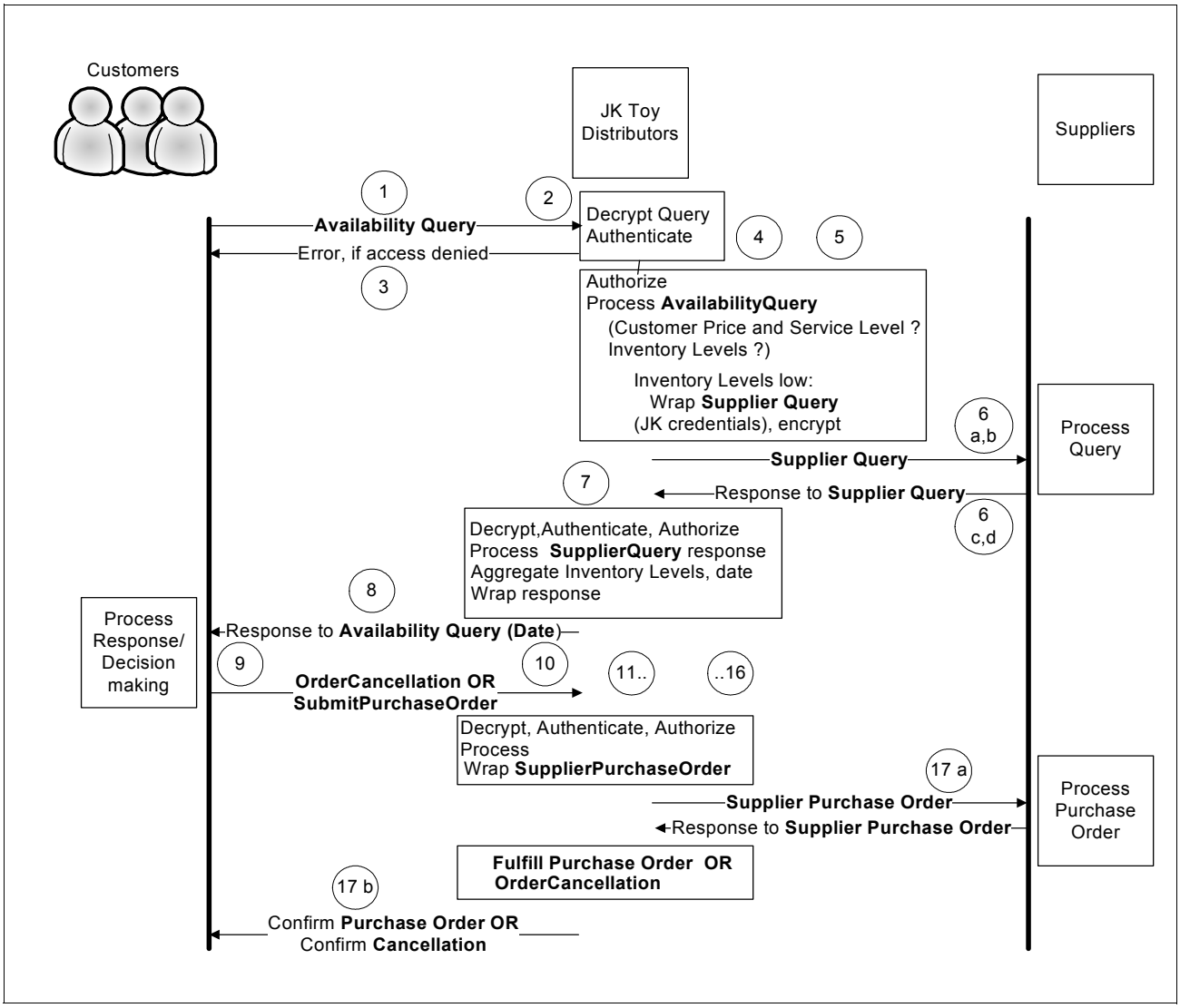

Figure 19 Purchasing flow diagram

All the architectural alternatives discussed later use the same basic process flow. Each alternative uses a different point of presence in the DMZ and analyses the security implications arising from them.

1. When the customer's inventory falls below preset threshold levels for the products, the customer's inventory system issues a request for product availability to JK Toy Distributors. The query calls JK Toy Distributors' AvailabilityQuery Web service. The message payload is encrypted for processing at the end-point server in JK Toy Distributors. The customer's digital certificate is included in the Web services SOAP header for authentication. The header also includes a digital signature to ensure integrity of the message.
2. JK Toy Distributors receives the request, authenticates the certificate included in the AvailabilityQuery SOAP header, and grants or denies access for the customer to the Web service.
3. If access is denied, an error document is sent back to the customer.
4. If access is granted, the request is processed and passed to the internal AvailabilityQuery Web service.

5. The AvailabilityQuery service in turn calls the appropriate transactions on the legacy system to check for inventory. The following conditions are checked:
   - Price and service levels for the requesting customer
   - Inventory levels for the requested item
6. If inventory levels are not sufficient, a request is formatted for JK Toy Distributors suppliers.
   a. The message from the legacy system is wrapped in the SupplierQuery Web service using a similar process as the AvailabilityQuery service described earlier. Similar trust and authentication mechanisms are used.
   b. These requests are sent out to the suppliers. If an acknowledgement is not received within the predetermined time out period, the request is retried a specified number of times, after which it is cancelled.
   c. The suppliers receive the SupplierQuery request and provide the requested information as a response in a similar manner using a predefined Web services interface consistent with access management policy.
   d. The information from the SupplierQuery response messages is aggregated in JK Toy Distributors' legacy system using the appropriate policies and a similar process as that described earlier.
7. An aggregate AvailabilityQuery response is composed by the legacy system using the policies for the customer as discussed earlier.
8. A message is sent back to the customer with credentials of JK Toy Distributors as the AvailabilityQuery response which contains prices, terms and conditions, and so on, using flows similar to that described for the suppliers.
9. The customer's system processes the response document and accepts or rejects the dateAvailable.
10. If the customer rejects the dates, they cancel the order by invoking JK Toy Distributors OrderCancellation Web service.
11. If the customer accepts the dateAvailable, they prepare a SubmitPurchaseOrder Web service request to send to JK Toy Distributors.
12. The SubmitPurchaseOrder Web services request is signed by the customer and sent to JK Toy Distributors.
13. The SubmitPurchaseOrder Web services request is received at JK Toy Distributors and is validated.
14. The JK Toy Distributors point of presence validates the signatures on the SubmitPurchaseOrder. If the signatures are valid then the request is forwarded to the JK Toy Distributors internal systems.
15. If the signatures are *not* valid, the JK Toy Distributors point of presence generates an error message and returns the request to the customer.
16. The JK Toy Distributors ERP system validates the SubmitPurchaseOrder (for example, there is still sufficient stock to fill the order; the requested purchase price is acceptable, and so on). If the purchase order is rejected, JK Toy Distributors sends the SubmitPurchaseOrder rejected response to their customer.
17. If the purchase order is accepted, the JK Toy Distributors Inventory Control system commits the stock based upon the priority in customer profile.
    a. If all required stock is not available at JK Toy Distributors, then JK Toy Distributors generates a Web services request to each of its suppliers to determine who has the required items, when they can be delivered and at what price. Each supplier may have its unique form of a SubmitPurchaseOrder Web service to be invoked.

b. When JK Toy Distributors receives the necessary SupplierPurchaseOrder confirmed responses, the SubmitPurchaseOrder confirmed response is sent to their customer.

Figure 20 summarizes the overall system context for our business scenario.

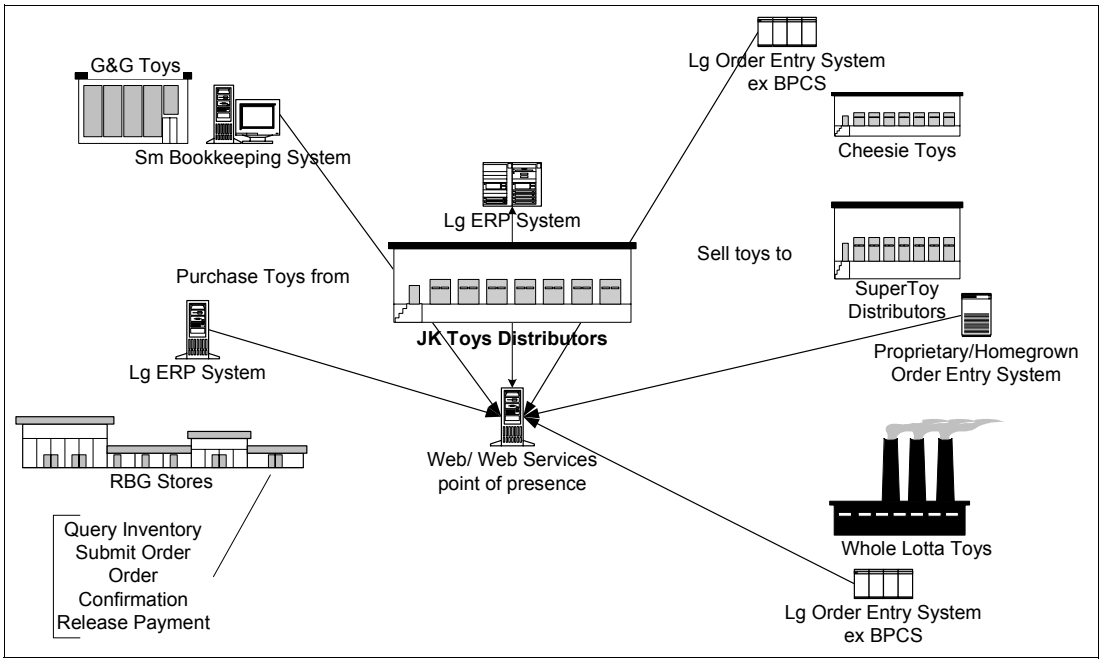

Figure 20 JK Toy Distributors system context diagram

Architectural alternatives

There are any number of ways to implement this use case. Three different architectural alternatives are presented in this section in order to illustrate some of these possible approaches.

Alternative one: Small enterprise

The small enterprise alternative, depicted in Figure 21 on page 32, demonstrates the minimalist approach to securing Web services.

In this alternative, JK Toy Distributors has implemented the Web Services Gateway included with WebSphere Application Server Network Deployment. The customer authenticates by sending a certificate to JK Toy Distributors.

JK Toy Distributors will authenticate the certificate at the channel of the Web Services Gateway and grant or deny access to the customer.

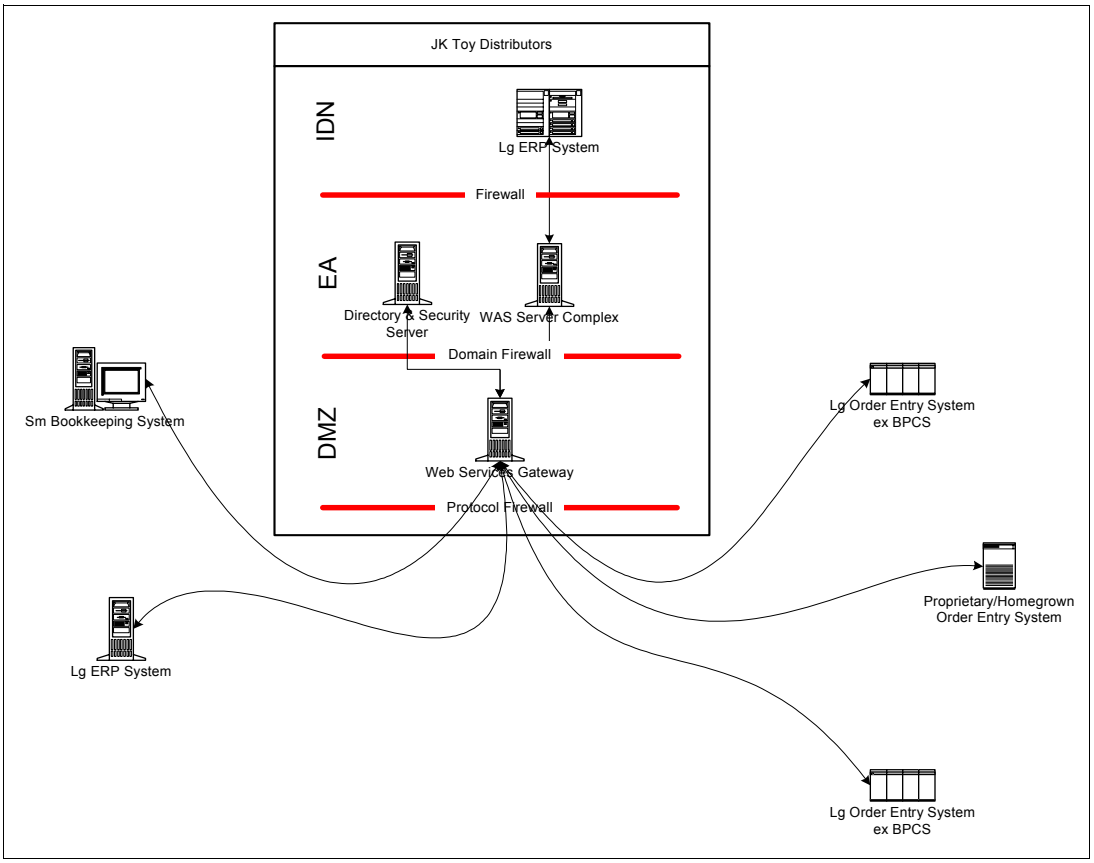

Figure 21 Small enterprise alternative

Assumptions

JK Toy Distributors will issue the Web Services Description Language (WSDL) interface specification to the customer and the customer must develop a Web service interface with the WSDL. A private Universal Description, Discovery, and Integration (UDDI) standard can be used. However, the most common method of publication today is a specification document negotiated and exchanged directly between two enterprises.

Since this application is directly exposed across organizational boundaries, it must implement or exploit necessary security features such as authentication, authorization confidentiality, integrity, and logging for non-repudiation purposes.

JK Toy Distributors', the SP's, X.509 digital certificate must be exchanged with G&G Toys and configured into G&G Toys' runtime prior to invoking JK Toy Distributors' services. The root certificate of the certifying authority (such as VeriSign) that issued the parties' certificates will need to be imported into the local key stores for both JK Toy Distributors and G&G Toys. This validates the digital signatures of the individual certificates passed as binary security tokens in the SOAP messages.

The G&G Toys' application (the consuming application) will have a service proxy or a JAX-RPC stub component that has been generated from an Integrated Development Environment (IDE) such as WebSphere Studio Application Developer using JK Toy Distributors' WSDL. When a Web services invocation is made, the proxy or SOAP runtime on the client system performs the WS-Security functions prior to sending the request.

The corporate security policies of the businesses state that all confidential information exchanged between the parties will be encrypted between the boundary nodes at each enterprise.

One of the signed message parts includes G&G Toys' business identity and secret password. The password provides a second authentication factor to be used by security logic within the legacy systems where finer grained access control is performed. JK Toy Distributors uses the credentials to perform authorization for controlling access to the back-end business functions. This solution provides only coarse-grained authorization by the Web Services Gateway. JK Toy Distributors uses this method because fine-grained authorization logic resides in the back-end legacy systems.

Web services request messages might have the following elements signed:

- Security Token (UsernameToken, XML-based token) in the SOAP header
- TimeStamp in the SOAP header
- Body in the SOAP envelope

Web services response messages might have the following elements signed:

- TimeStamp in the SOAP header
- Body in the SOAP envelope

Web services request messages might have the following elements encrypted:

- Security Token (UsernameToken only) in the SOAP header
- Body in the SOAP envelope

Web services response messages might have the following elements encrypted:

- Body in the SOAP envelope

Process flow

Let us take a closer look at the individual steps of the process flow.

1. The SOAP message is digitally signed. The SOAP runtime will access a key store to retrieve security keys and certificates as needed. Depending on the WS-Security support G&G's environment provides, they might be able to sign just the SOAP body, or they might be able to sign individual elements within the body. In addition, SOAP header blocks might be signed. The signature is performed using G&G Toys' private key. Once the message has been signed, the X.509 certificate itself is included in the SOAP header as a binary security token.
2. The message is encrypted using a symmetric algorithm with a shared key. The key used for the data encryption is encrypted itself using an asymmetric algorithm with the public key associated with JK Toy Distributors' X.509 certificate. Once the message and shared key have been encrypted, a reference to JK Toy Distributors' X.509 certificate is included in the SOAP header because JK Toy Distributors might be using multiple certificates.
3. When JK Toy Distributors receives a Web services request, the request is directed to the SOAP processing engine (SOAP runtime) based on the request's URL (published access point for the service). The message data and shared key passed in the request are encrypted, so the first step is to identify the X.509 certificate referenced in the SOAP header and retrieve JK Toy Distributors's associated private key from a key store. Once the private key is obtained, the shared key can be decrypted using an asymmetric algorithm.
4. With the shared key in the open, the message data can be decrypted using a symmetric algorithm.

5. With the entire message now in the open, the X.509 certificate passed in the SOAP header can be accessed to retrieve its public key.
6. The message's digital signature is validated with G&G Toys' public key.
7. As a result of the signature's successful validation, JK Toy Distributors' SOAP runtime not only validates the message integrity but also is ensured that G&G Toys actually signed the message. This process also authenticates the message's original sender because only the sender who owns the certificate has access to the private key used to sign the message.
8. Once the message has been decrypted and the signature validates, the SOAP runtime calls the Web services implementation.
9. Once JK Toy Distributors has processed the request and a response is available, the same WS-Security operations take place for the Web services response message. However, the roles of X.509 key pairs are reversed for digital signature and encryption.
10. Once the message has been signed and encrypted, JK Toy Distributors' SOAP runtime sends the response to G&G Toys.
11. G&G Toy's processing of the Web services response is very similar to that of JK Toy Distributors for the request performed. Again, the roles of X.509 key pairs are reversed.

Rationale

The decision criteria for selecting this alternative include:

- Central access point for all services inside the enterprise

  The gateway provides a single, well-known access point to Web services within the enterprise.

- Decoupling the deployment of Web services from clients

  The gateway isolates any changes in the deployment of services within the enterprise from consumers of the services. The location of services also becomes transparent to clients of the service. This provides the flexibility to change the deployment infrastructure without notifying all the service requestors. For example, the hosting server may fail or be taken down for maintenance or may be replaced with a newer server. Therefore, there needs to be a process to route the invocations to an alternate service.

- Central security control point

  Access control can be applied to Web services so only authorized clients are allowed to access services. The gateway allows setting access control on each service, so that not every client can access every service.

- Protocol conversion between Web service requesters and providers

  Clients might need access to applications that use protocols other than HTTP. Using the Web Services Gateway, it is possible to trap a request from a client and transform it to another messaging protocol. For example, JK Toy Distributors might have implemented a service using SOAP/JMS because of the existing infrastructure, but they should also offer the service to clients that only understand SOAP/HTTP.

- Return on investment

  Processes that have already been developed as Web services can be easily reused by partners. Using the gateway, JK Toy Distributors can take advantage of the existing messaging infrastructure to make Web services requests and use the existing Web services for external process integration.

- Simplicity and product purchase cost

  This alternative does not require the purchase of additional software beyond WebSphere Application Server Network Deployment.

The decision criteria for not selecting this alternative include:

- Security token conversion

  This alternative does not provide the ability to receive a security token in one format such as SAML and exchange it for a security token in another format, Kerberos for example.

- ID translation

  This alternative does not provide ID translation between enterprises as might be desirable in a federated trust configuration between enterprises.

Alternative two: Large enterprise

The large enterprise alternative, shown in Figure 22, demonstrates the full suite of application software appropriate to securing Web services.

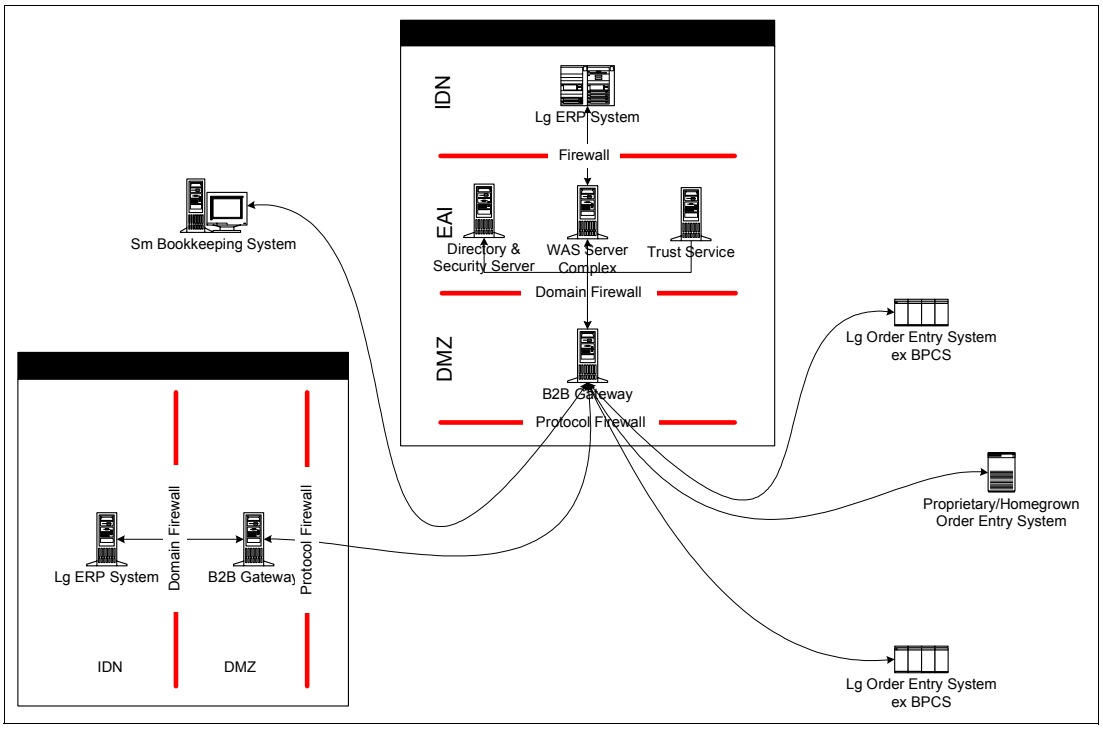

Figure 22 Large enterprise alternative

In this alternative, JK Toy Distributors has implemented the B2B Web Services Gateway and the Tivoli Federated Identity Manager trust service. The customer has their own B2B gateway so communication is accomplished by the SOAP WS-Security extensions.

Assumptions

The following assumptions were made during the creation of this alternative.

1. All customers and suppliers of JK Toy Distributors have existing purchasing agreements in place.
2. Security is required for orders coming into and going out of all participants. Not all security levels will be the same but a minimum of SOAP, WS-Security, is required.
3. No human intervention is required except in error processing.

Rationale

The decision points to select or not select this alternative would be:

- Pros
  - The JK Toy Distributors ERP system does not need the know how authentication is performed.
  - Off loads management of security communication to trust service.
  - Removes security from the application.
  - Provides non-repudiation capabilities.

    Note: Because non-repudiation is a legal, not technical, concept, it is subject to judicial interpretation. Security technologies of the sort described in this Redpaper may be used to form the basis for non-repudiatable transactions or they may not, in accordance with the laws governing the transactions in question.

  - Solid trust relationship between trading partners.
- Cons
  - Higher cost to implement.
  - Additional complexity in configuration and maintenance.

Process flow

The following sections describe how the WebSphere Web Services Gateway processes requests to and from JK Toys Distributors and its partners.

1. The B2B gateway using WebSphere Web Services Gateway receives each outbound request and attempts to secure each request based on the supplier's policy.
2. Cheesie Toys is a small supplier and relies on mutually authenticated SSL for all security requirements. The gateway will simply forward this request and WebSphere Application Server will negotiate the SSL session with Cheesie Toys.
3. Super Toy Distributors is a slightly larger supplier and requires a signature on the Web services request (SOAP body) to ensure that the request is not tampered with en route. The WS-Security handlers at the gateway are invoked, so that the SOAP body is signed and the corresponding signature and certificate are added to the SOAP <Security> header.

   Note: SSL also provides integrity checking, therefore WS-Security is not required to get this sort of protection.

4. Whole Lotta Toys is a very large supplier and requires not only signatures on the Web services request, but that additional information, such as correlation id and other transactional data be included for authentication purposes. Authentication based on the signing certificate is not sufficient.
5. A trust client at the B2B gateway will invoke the TFIM trust service to request the appropriate security token for Whole Lotta Toys (note that neither the trust client nor the B2B gateway know the required token type for Whole Lotta Toys. This information is maintained by the trust service.
6. Based on the intended destination of this request, the TFIM trust service will determine that a SAML assertion is required. The trust service will build a Whole Lotta Toys-specific SAML assertion and return this credential in a WS-Security compatible format to the trust client. This might include specific identity transformation, including attribute retrieval, to build the SAML assertion with the appropriate information.
7. The trust client will insert the SAML assertion into the SOAP <Security> header.

8. The WS-Security handlers will be invoked to sign the required parts of the message (for example, SOAP body).

Futures

An interesting development currently underway is in the area of secure conversations. WS-Secure Conversation will provide state-like binding to Web services conversations. This will be very valuable when there is a requirement for a more *stateful*, monitored, conversation between trading partners. However, due to the increased overhead of conversation setup, a strong requirement for this capability is needed to justify including it.

Alternative three: Enterprise with Tivoli Access Manager

In this alternative, JK Toy Distributors considers using a *reverse proxy*, such as the one provided by Tivoli Access Manager, for handling access management and security in business to business interactions with their partners.

The operational layout for this approach is shown in Figure 23. Access Manager would provide the Web Services point of presence.

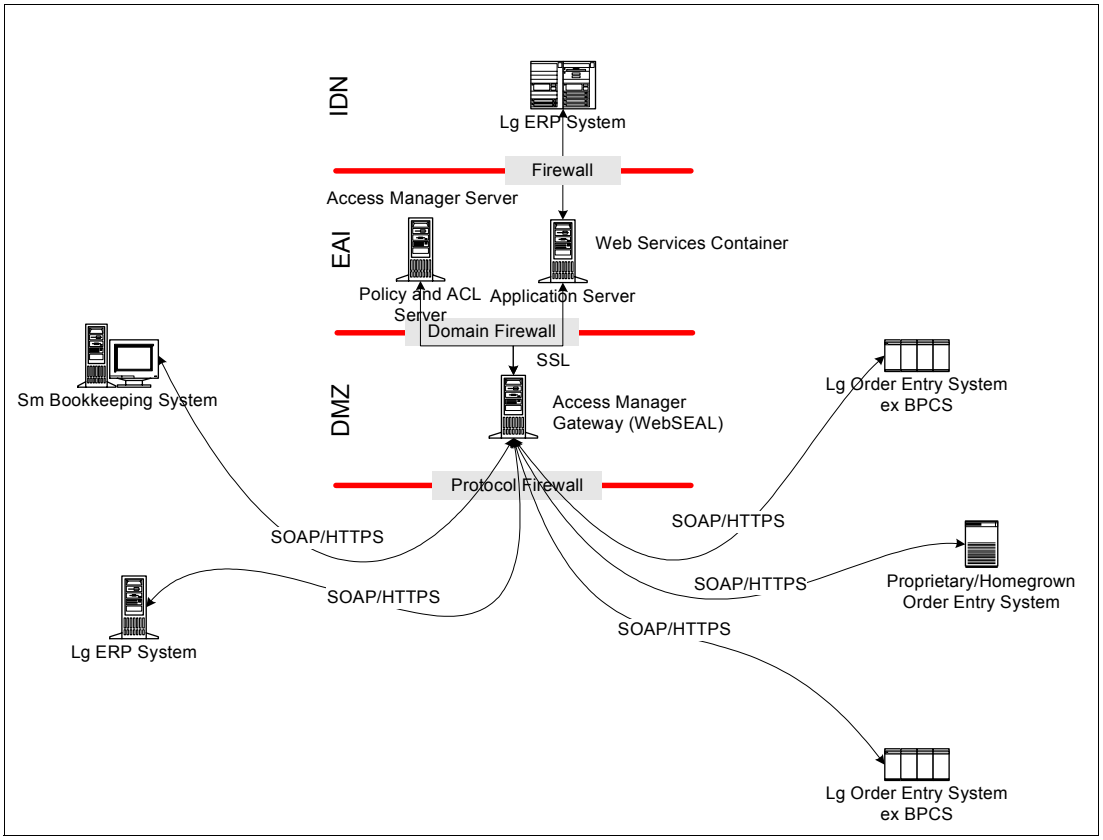

Figure 23 Operational layout: enterprise with existing Access Manager

Environmental assumptions

We made following assumptions about the operating environment:

- The basic Access Manager infrastructure is already in place as JK Toy Distributors is using it to manage access to its Web presence by individual users.
- JK Toy Distributors applications have been Web Services enabled. Each Web Service is represented as a URL. Some form of mapping between URL parameters and WSDL is recommended in order to isolate the external interfaces.

- JK Toy Distributors is able to negotiate the appropriate interface specifications with their business partners and these are relatively uniform across the spectrum. They can be handled through SOAP/HTTPS.
- Trust relationships are established through digital certificates. These credentials provide authentication at a partner process level and not at an individual level. In addition, these provide the basis for digital signatures on the messages in order to ensure data integrity.
- Digital certificate and key distribution to partners is through an out-of-band process.
- Actual message elements are encrypted for transmission. These are decrypted at the endpoints to enhance security. In addition, SSL channel is provided for confidentiality of information flow between partners if additional protection is desired.

Process flow and security considerations

We made the following decisions about process flow and security:

- Interface to customer/supplier systems.

  The process and technology flows at the customer system for queries and order services are similar to the previous two alternatives. The Web Services envelope is similar in structure.

  However, an HTTP header must be added to the envelope, establishing a SOAP/HTTP binding so Access Manager can process it, as shown in Figure 24.

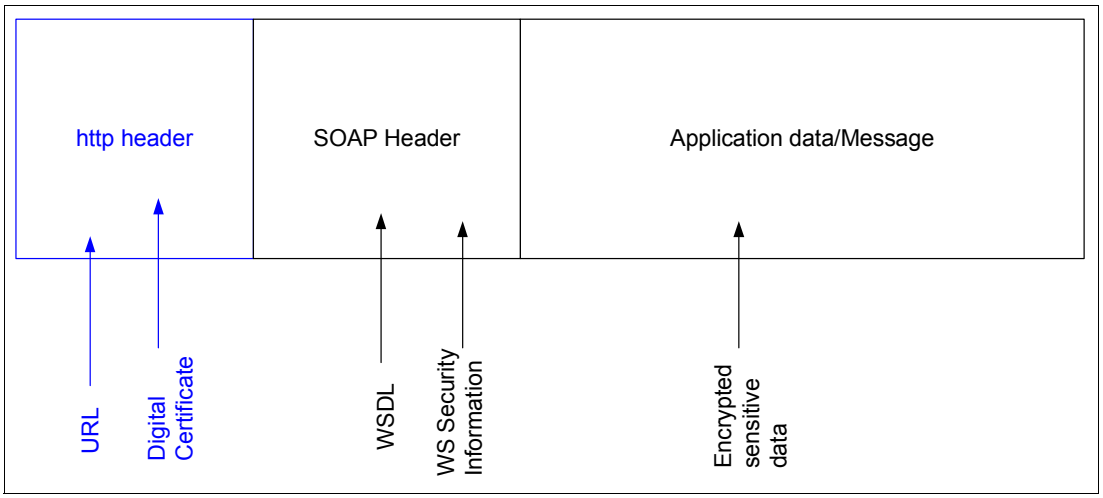

Figure 24 SOAP/HTTP binding

  The HTTP header includes a digital certificate which is used for authentication of the buyer and the supplier on all in-bound Web services messages transported over SSL. Additionally, in this case all Web services are represented by a URL. At a transport level the communication channel is further protected by using SSL for all communications.

- Processing at JK Toy Distributors

  The Access Manager Proxy (WebSEAL) intercepts the HTTPS message and acts as a gateway in the JK Toy Distributors DMZ. The Access Manager Proxy goes through its normal processing to see if the requested URL in the header is a protected resource or not. For a protected resource, Access Manager will inspect the credentials provided, in this case a digital certificate, and validate them to see if the requested URL may be accessed by the requestor. If the credentials are not valid for the URL, the message is returned with an error to the requestor.

If the credentials are valid, the Access Manager Proxy will forward the request to the application server in the secure zone over a separate SSL session. Since the Web service message payload is encrypted, security exposures are further minimized during this processing. The forwarded request also contains the validated Access Manager credential as well as other attributes of the requestor.

The application server receives the forwarded HTTP request. The requestor's identity is available to the application server through the header of the HTTP message. The existing trust relationship between the Access Manager Proxy and the application server makes validation of credentials unnecessary, enhancing throughput and reducing resource requirements.

The application server maps the URL to a Web service and ensures that the requestor has authorization to access it. From this point, the processing of the Web service is similar to the descriptions of earlier alternatives, including sequencing of the messages.

Outgoing messages are treated in a similar manner as described above. The application server creates the HTTP header with a similar structure as that described earlier. In this case, JK Toy Distributors's digital certificate and the URL of the partner target system are inserted into the header.

Decision points

The following are significant considerations in this alternative:

- Is there an existing Access Manager infrastructure or would it require additional investment?
- How diverse are the access requirements and trust mechanisms across the partners, since this alternative will support only SOAP/HTTP(S)?
- Are the interface and trust specifications relatively static and can they be negotiated with partners prior to establishing the service?
- How much of the security processing is off-loaded to the DMZ? (In this case, authentication is performed only in the DMZ and from then on the Access Manager gateway is treated as a trusted resource. This enhances throughput and reduce resource requirements.)
- What is the granularity of URL-to-Web service mapping for access control? (In this case each Web service is mapped to a URL to provide finer granularity for access control in the DMZ using the Access Manager Proxy.)
- A simple, SSL-based authentication approach is likely to be used by relatively small partners. In such cases, the overhead incurred by a Web Services Gateway infrastructure may be overkill.

The following are pros and cons of this alternative:

- Pros
  - It provides rapid response to emerging business requirements such as enabling a new channel with minimal new integration, which is a key of the ODOE driver and part of the framework.
  - It leverages existing investment in Access Manager to allow quick and easy exposure of Web services.
  - The Web services presence is instantaneous with proven technology which is well understood and provides a high level of security. The Access Manager Proxy in the DMZ is a hardened resource.
  - Allows for a single point of management for security/authorization policy.

  - Provides inheritance of access control rights which minimizes errors in policy enforcement.

- Cons
  - It does not provide a true Web service proxy which can be used to abstract out the WSDL for internal JK Toy Distributors services. This abstraction layer has several advantages including isolating changes in internal business processes and applications from partners. However, abstraction is provided at a URL level to address resiliency and flexibility.
  - It provides URL level mapping and routing, which is relatively inflexible for protocol translation and mapping.
  - UDDI changes require an additional administrative task within Access Manager to reflect URL mappings.

Conclusion

In this scenario we discussed how a distribution company could react rapidly to changes in the business environment and interact with their partners in security and with minimal change to existing systems and processes.

Three potential alternatives were discussed to make this scenario as widely adaptable as possible, with a focus on security issues in each. The rationale, applicable environment and relative positioning (pros and cons) of each alternative were also discussed

An on demand operating environment provides customers an infrastructure that is flexible, resilient and dynamic. The alternatives in this paper illustration these characteristics are as follows:

- Flexible: Each alternative used Web services to integrate JK Toy Distributors' business processes and systems with their partners' processes and systems. This provides them the flexibility to integrate with diverse other standards-based systems.
- Resilient**:** Each alternative demonstrates resiliency by providing a level of abstraction of the internal operations and systems through the Web services point of presence. This means JK Toy Distributors can change the infrastructure to re-organize services, enhance infrastructure and service delivery without impacting interactions with their partners. The level of abstraction varies with Alternatives one and two, providing a higher level than in "Alternative three: Enterprise with Tivoli Access Manager" on page 37 which leverages existing Access Manager technology.
- Dynamic: As compared to a B2C/B2E scenario, the JK Toy Distributors of alternative three has a limited number of partners. Therefore, dynamic provisioning was not a key aspect of the solution alternatives discussed and they do not lend themselves to it. This is not a technology issue, but rather a consequence of the business environment of the scenario.

The infrastructure that would be required to support the dynamic aspect of an on demand operating environment would not require the capability for dynamic provisioning of partners. Further dynamic capability could be provided through UDDI directory and administration services.

The first and second alternatives offer a more robust, protocol-independent approach to Web services exposure. In addition to the technical issues, appropriate inter-enterprise organizational and contractual issues would need to be addressed at a business level. This is simply to state that the architectures outlined in this section are extensible.

Scenario two: B2E or B2C Federated Identity Management

The managers of JK Toy Distributors are continually seeking ways to drive costs out of company business processes. They grasped early the potential of an employee portal. They understood how it provides a simple and effective way for workers to access the relevant information and business processes needed to carry out their day-to-day activities. As JK Toy Distributors out sourced more of its benefits processes, it quickly became clear that the employee portal could also provide a central point of access to the various institutions delivering benefits to JK Toy Distributors workers. Employees appreciated, for example, the convenience of clicking on a link within the portal that would redirect them to their various plan providers of retirement, health care or stock purchase. Figure 25 shows a business context diagram of this scenario.

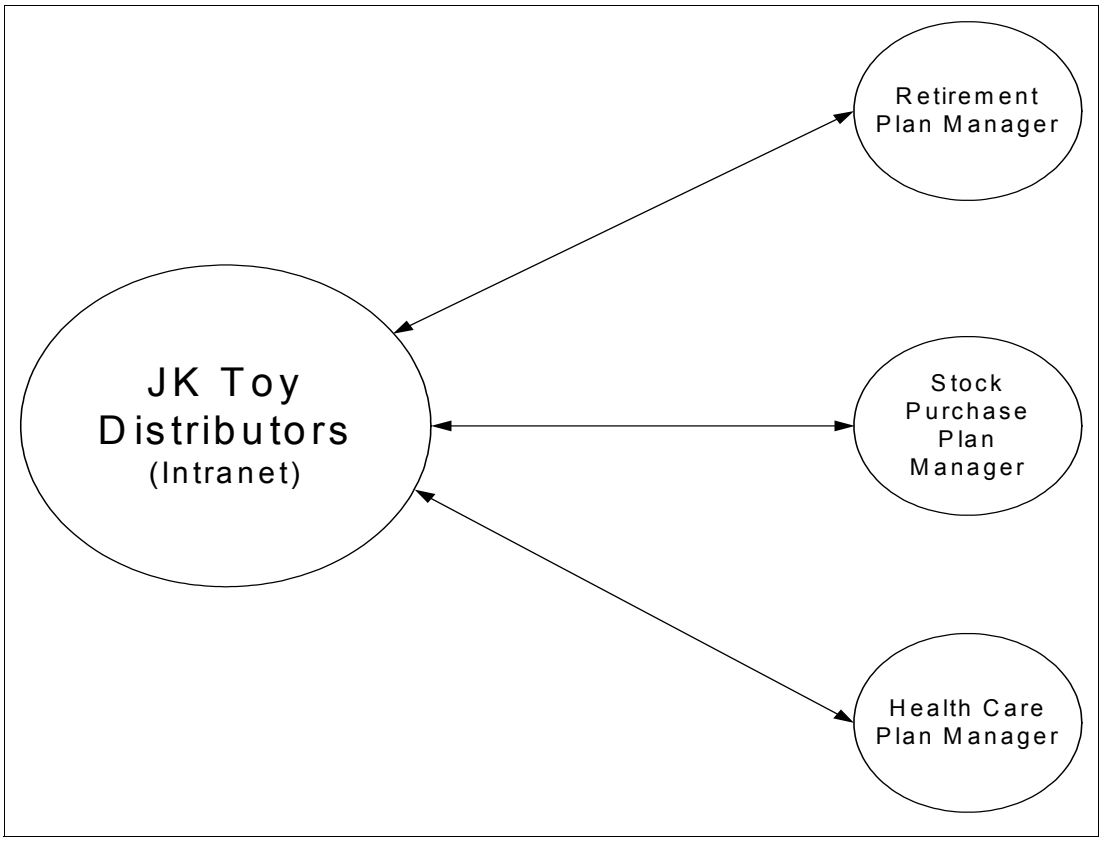

Figure 25 Business context diagram

However, the need to sign on to each of these systems each time the employee needed to connect was a continual annoyance. Moreover, both JK Toy Distributors and its service providers recognized that the cost of registering and maintaining separate user accounts and access rights for each employee was substantial.

The JK Toy Distributors architecture team analyzed the effect of establishing a federated identity management (FIM) environment between themselves and their service providers.The as is and potential to be scenarios they developed are shown in Figure 26 and Figure 27 on page 43.

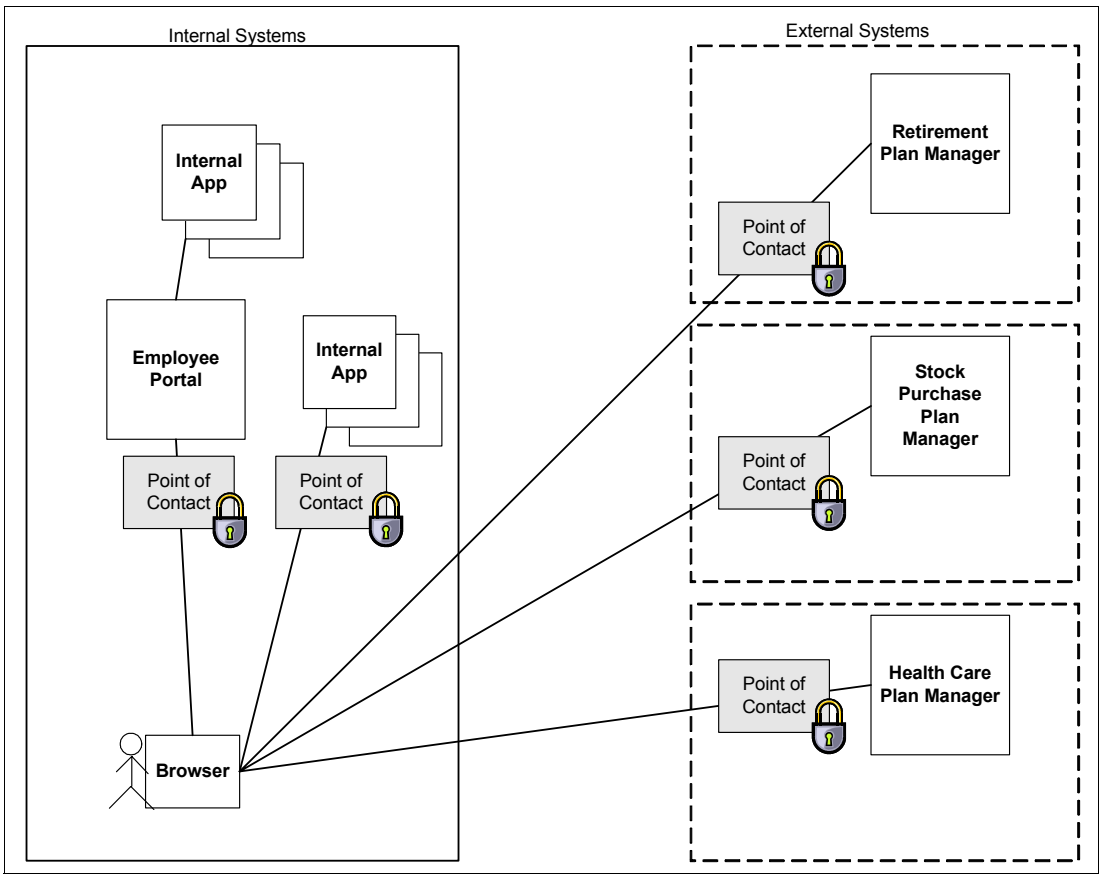

Figure 26 System context diagram before Federated Identity Management

The point of contact in Figure 26 and after in Figure 27 system context diagrams performs security functions such as authentication and authorization.

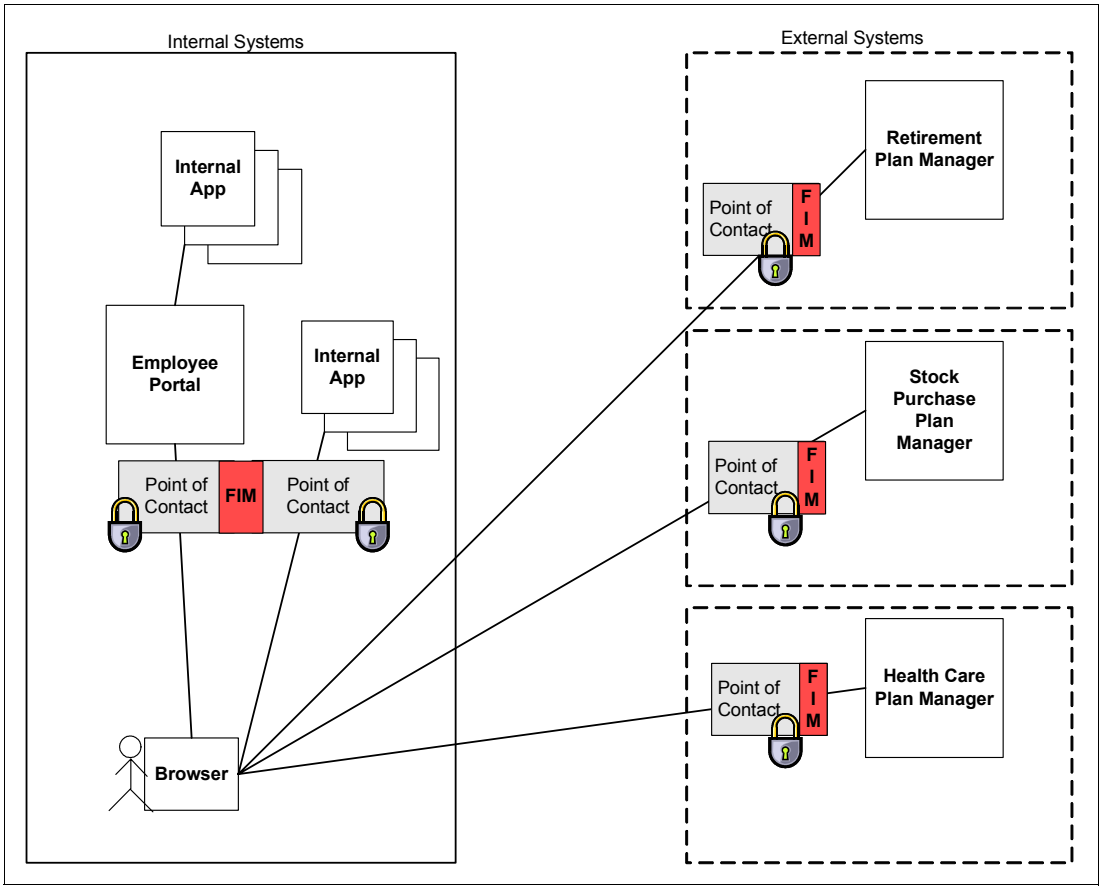

Figure 27 System context diagram after Federated Identity Management

JK Toy Distributors then approached their providers about partnering to establish an FIM environment. MoreStock.com, a visionary in the field of servicing employee stock purchase plans, had already recognized the potential value of an improved experience for its user community, as well as the reduced operating costs for itself. They had a FIM feasibility assessment already underway. The two companies agreed to combine resources to investigate further the establishment of an FIM environment between themselves.

The combined committee established the following project requirements:

- Reduce the overall cost of registering and maintaining users' access rights across the two companies. Cost savings will be shared by a to-be-determined formula.
- Improve the employee experience by providing single sign on and single sign off for JK Toy Distributors employees through their employee portal.

Based on these requirements, the architecture team decided to focus first on developing use cases that would demonstrate in detail how they could be fulfilled. The following use cases were identified:

1. Federated single signoff and signoff
2. Dynamic employee provisioning
3. Dynamic de-provisioning and privilege management

The use cases were constructed with the desire to keep the companies' options open as much as possible. This was fairly simple with the signoff/signoff case. All versions of the Liberty standard as well as a WS-Security standard-based implementation could be supported from the pull model use case that the team created for Use case one: Federated

single signon and single signoff. The team found, however, that the push capabilities of the WS-Security standard were better suited to the delivery of the dynamic employee provisioning use case. For the deprovisioning and privilege management use case, the team found that some of the steps involved in the business process actually were outside of the scope of either standard, but that each could be used to implement the business process.

The architectural team developed the complete use cases in the following sections.

Use case one: Federated single signon and single signoff

This use case addresses the requirement to improve the employee's experience. It describes the steps necessary for an employee to move seamlessly between the JK Toy Distributors Distributors' portal and the Morestock.com Web site without reauthentication during sign on or multiple sign offs during the exit procedure. Flow diagrams of the use case are provided in Figure 28 on page 45 and Figure 29 on page 46.

1. Employee signs on to JK Toy Distributors' portal.
2. Employee clicks on MoreStock.com link within the employee portal.
3. Employee is redirected to MoreStock.com.
4. Employee attempts to access MoreStock.com, but does not have a valid session with MoreStock.com.
5. FIM at MoreStock.com determines the employee's identity provider is JK Toy Distributors Distributors because the employee accessed a JK Toy Distributors specific link.
6. FIM at MoreStock.com builds an SSO request to be sent to JK Toy Distributors, asking for SSO of employee.
7. MoreStock.com redirects the SSO request to JK Toy Distributors.
8. Employee's browser automatically redirects the SSO request to the JK Toy Distributors FIM.
9. FIM at JK Toy Distributors receives the SSO request from MoreStock.com.
10. FIM at JK Toy Distributors builds an assertion about the authenticated employee and builds an SSO response for MoreStock.com.
11. The JK Toy Distributors FIM redirects the SSO response back to the employee's browser.
12. The employee's browser forwards the response back to MoreStock.com's FIM.
13. The MoreStock.com FIM checks to:
    a. Determine if the credential is from a trusted organization.
    b. Determine if the credential is valid.
    c. Determine the employee's local identity.
    d. Build a session credential for the employee.

    The employee has a valid session at MoreStock.com

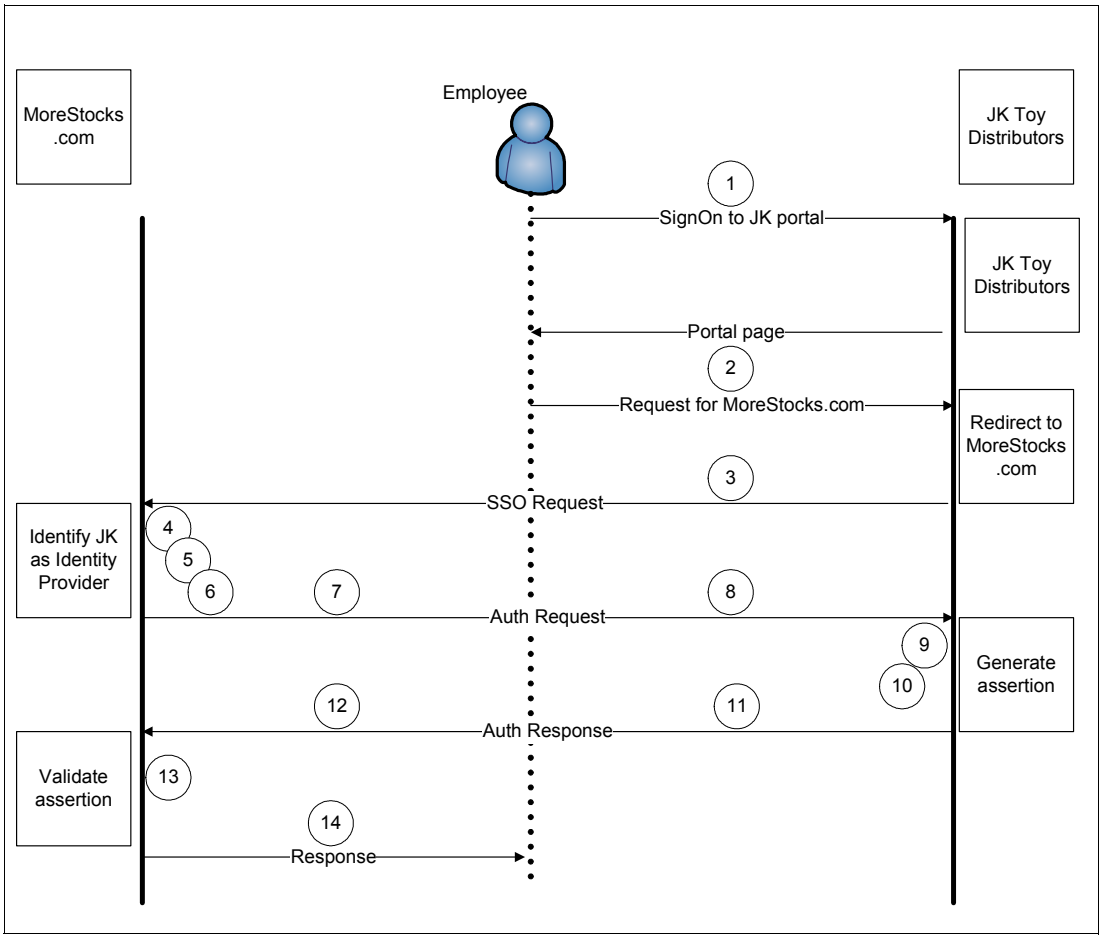

Figure 28 Use case scenario: single signoff using FIM

After some time has passed, the employee needs to leave his desk to attend a meeting. The following steps along with Figure 29 on page 46 illustrate the processing involved.

1. The employee signs off by clicking the Signoff button on the portal.

   Note: Many business relationships will *require* that sign off from the portal be a global signoff to avoid the risk of leaving other sessions open to attack.

2. The JK Toy Distributors FIM builds a list of places the employee has signed onto using federated single signoff and sends a page containing the list to the browser.
3. Embedded code on this page issues a sign off to each of the associated service providers.
4. The browser sends a sign off request to the FIM at MoreStock.com.
5. The FIM at MoreStock.com sends an acknowledgement to FIM at JK Toy Distributors Distributors.
6. Sign off pages from each service provider are updated and presented to the employee.
7. Once all service providers sign off, the FIM at JK Toy Distributors triggers a logout so that no valid sessions exist.

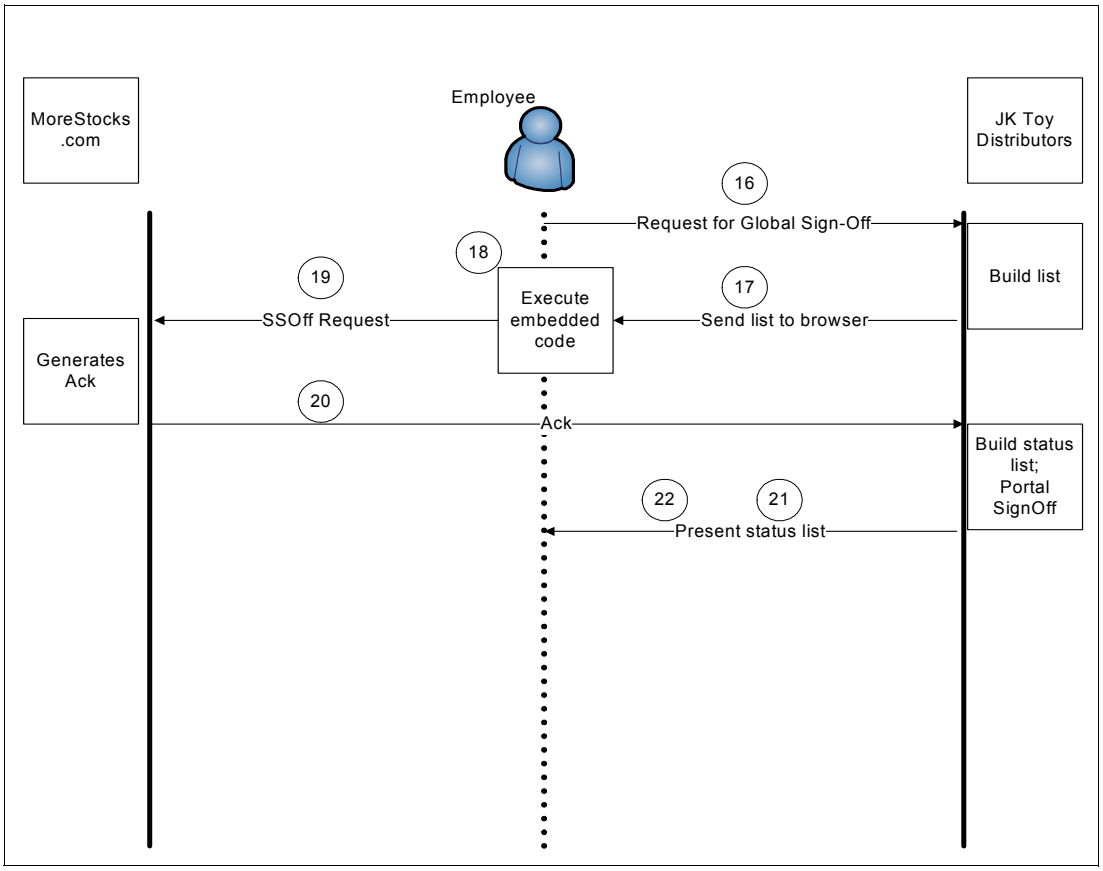

Figure 29 Use case scenario: single signoff using FIM

Push and pull capabilities

As shown in the SSO use case, the employee who logs into the JK Toy Distributors' portal and clicks the MoreStocks.com link gets redirected to the MoreStocks.com site using a *pull*-based SSO approach. That is, after the redirection, the service provider (MoreStock.com) must send a request back to the identity provider. Only then will the identity provider (JK Toy Distributors) deliver the identity assertion. The pull model is supported by both the Liberty Alliance and WS-Federation standards.

In a *push* model, the employee goes to JK Toy Distributors first to obtain a security token which would then be presented to MoreStock.com. This approach is more efficient, because an intermediate request and response sequence is avoided. The Liberty Alliance did not support the push model of SSO in versions 1.0 and 1.1 of the specifications. However, Liberty ID-FF 1.2 makes allowances for a push-based SSO. WS-Federation has always supported the push model. Tivoli Federated Identity Manager supports both the Liberty and WS-Federation approaches.

Mobile environment considerations

Although this use case is based on a regular browser environment, FIM tasks also can be used and deployed in an extended scenario using mobile devices. In an environment using mobile devices such as Subscriber Identity Module (SIM) -based cell phones, these credentials on the SIM card can be used to achieve SSO as well. Additional security can be achieved using step-up authentication, forcing the user to use a Transaction Number (TAN) or an additional password to access higher classified data, for example. Usually Mobile

Operators (MO) already have established trust relationships on both sides (for example, customers and service providers) so an MO can act as an identity provider.

Today's mobile devices are continually improving—providing ever increasing computing power and memory capacity. Even fully functional browsers are nothing new for these devices. Nevertheless, some restrictions still apply for some devices: header, certificate and cookie handling as well as presentation issues. On the infrastructure side, restrictions include session loss, and partially online scenarios. This is not, however, a part of the FIM system itself and must be taken care of by other middle ware.

Use case two: Dynamic employee provisioning

This use case is an extension of the SSO case. It addresses both the requirement to reduce overall costs through an automatic creation of the employee user account at Morestock.com and the need to improve the user experience through the creation of a user account at MoreStock.com on demand - at the time that the employee selects MoreStock.com. The steps required to accomplish this provisioning on demand are described below. Figure 30 on page 48 and Figure 31 on page 49 depict the flow diagrams shown these steps.

1. The employee signs on to JK Toy Distributors' portal.
2. The employee clicks MoreStock.com within the employee portal.
3. The employee is redirected to MoreStock.com.
4. The employee tries to access MoreStock.com, but does not have a valid session with MoreStock.com.
5. FIM service at MoreStock.com determines that employee's identity provider is JK Toy Distributors. This could be accomplished through MoreStock.com logic that identifies the incoming request as coming from a link on the JK Toy Distributors site.
6. FIM at MoreStock.com builds an SSO request to be sent to JK Toy Distributors, asking for the SSO of the employee.
7. MoreStock.com redirects the SSO request to JK Toy Distributors.
8. Employee's browser automatically redirects the SSO request to the JK Toy Distributors FIM.
9. FIM at JK Toy Distributors receives the SSO request from MoreStock.com.
10. FIM at JK Toy Distributors builds assertion about the now authenticated employee and builds a SSO response for MoreStock.com.
11. The JK Toy Distributors FIM redirects the SSO response back to the employee's browser.
12. The employee's browser forwards the response back to MoreStock.com's FIM.
13. The MoreStock.com FIM checks to determine:
    a. If the assertion is valid
    b. If the assertion is from a trusted organization
    c. The employee's local identity
14. Local identity mapping fails because the employee has not yet been provisioned.
15. A new FIM account is built for the employee. In Tivoli FIM, this can be accomplished through XSLT style sheets. Improvements to this capability are planned for future releases.
16. A Request is sent to JK Toy Distributors FIM for additional attributes. These attributes could be roles, privileges, or any additional user-related data that MoreStock.com needs to have resident in its own system.

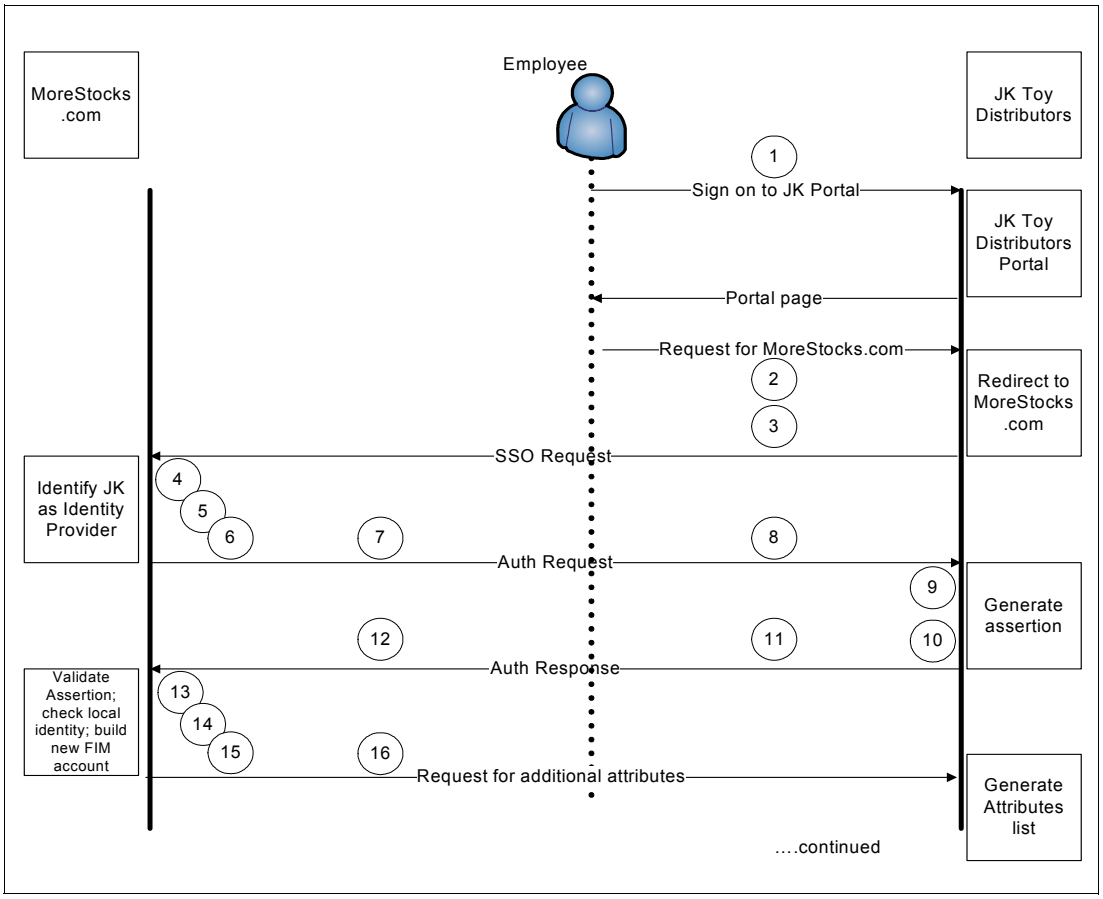

Figure 30 Dynamic employee provisioning (part 1)

17. JK Toy Distributors FIM sends additional attributes to FIM at MoreStock.com.
18. FIM at Morestock.com updates employee attributes.
19. FIM at Morestock.com builds a session credential for the employee.

    The employee now has valid session at MoreStock.com.

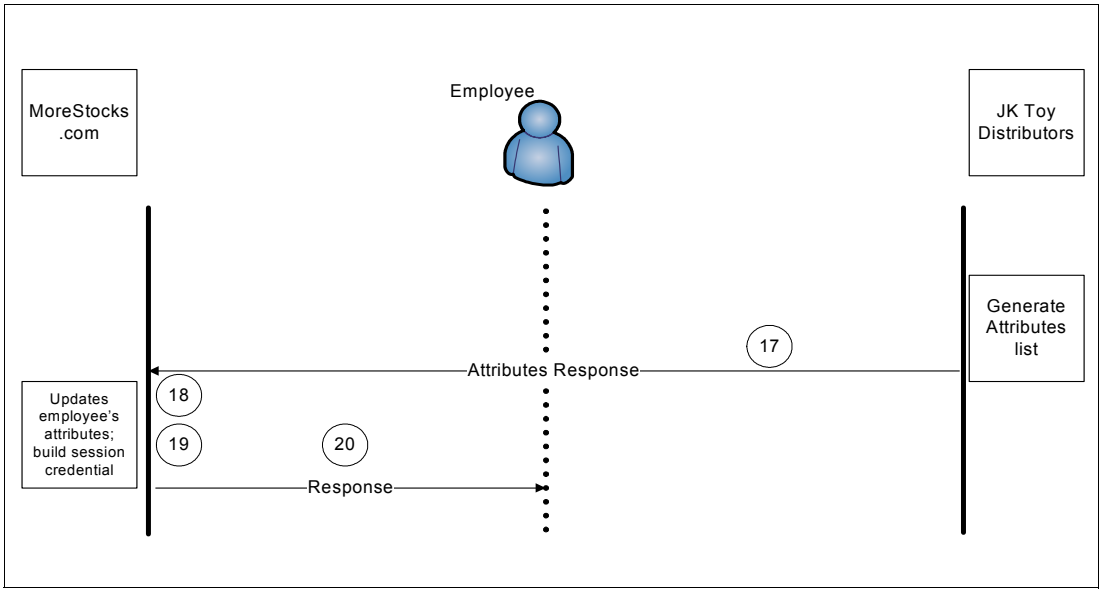

Figure 31 Dynamic employee provisioning (part 2)

Use case three: Dynamic privilege management

Dynamic privilege management is a subset of dynamic attribute management. This use case describes the systems involved when an employee retires from JK Toy Distributors Distributors. A retired employee no longer has the right to purchase any more stock through the company plan. However, the employee still needs to be able to manipulate the stock that is already owned. As part of the retirement process, the employee's permissions at the MoreStock.com site must be updated.

The case addresses requirement to reduce overall costs. The steps in the process are cataloged below. A flow diagram is presented in Figure 32 on page 50.

1. The human resources administrator changes the employee's status in the database to `retired`.
2. The human resources application requests an update to the internal identity management system to reflect the status change for this employee.
3. JK Toy Distributors identity management system revokes the employee's access to internal systems.
4. As part of the revocation process, JK Toy Distributors identity management system sends a request to JK Toy Distributors FIM to notify federation partners of changes to this employee privileges.
5. The request is accepted by the MoreStock.com FIM.
6. MoreStock.com FIM initiates a deprovisioning request to the MoreStock.com identity management system, based on previously agreed upon policy.
7. MoreStock.com sends an e-mail message to the employee containing a link to a self registration form where he can change his contact information. The JK Toy Distributors retirement policy allows employees to retain an e-mail account for two weeks after retirement.
8. The employee opens the e-mail message and clicks the link to the MoreStock.com Registration page.

9. The employee authenticates by answering challenge questions because there is now no access through federated sign on.
10. The employee registers a new password and contact information, such as a new e-mail address, at MoreStock.com. This will provide him access in the future.
11. Morestock.com sends a snail mail to the employee confirming the new information.

From this point forward, the employee accesses the Morestock.com account by signing on through the MoreStock.com system.

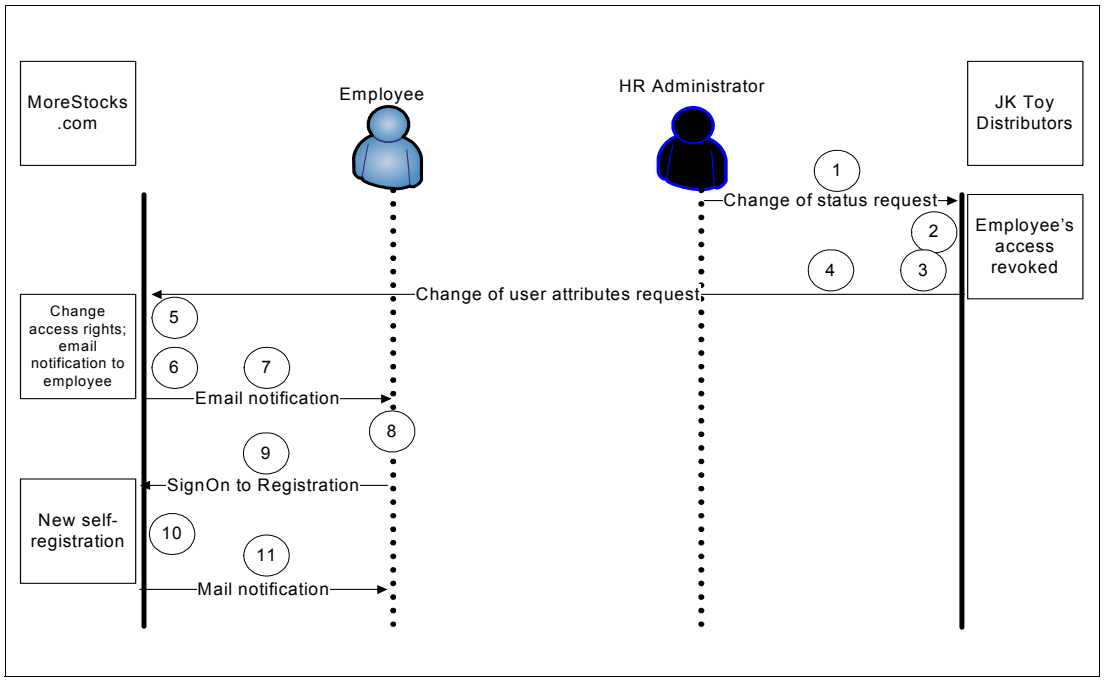

Figure 32 Dynamic privilege management

Conclusion

The joint architecture committee has explained in detail for its management team examples of the kind of logic that will be needed in the construction of a federated identity management system. The committee has also shown the kind of benefit that both the employee and the companies will experience by undertaking this effort. It is clear that the employee benefits through a better experience. Almost all of the advantages of cost savings on user account maintenance occurs at MoreStock.com. In order for JK Toy Distributors to gain some dollar benefit from an FIM system, it appears that MoreStock.com and JK Toy Distributors must come to some agreement on how MoreStock.com will pass some of those savings back to JK Toy Distributors.

The committee has shown that the project requirements can be satisfied.

Part four: Appendixes

Appendix A: On demand logical security architecture

The following sections provide further details for the components identified in “On demand security reference architecture” on page 10.

Anti-virus management

Anti-virus clients run on host platforms. Anti-virus management includes the following:

- AV client distribution and updates to authorized platforms

  Platforms may be initially loaded with anti-virus clients or fetch anti-virus clients from the anti-virus manager
- Notification that updates are available
- Making AV clients and updates available for automatic download when the host platform connects to the manager
- Receiving host AV log files and host AV configuration data
- Providing summary AV event information and alerts
- Providing reports

Assurance

Assurance is the determination that host platforms, end-user platforms, applications, and network component configurations and operation are in accordance with security policy. Entities are monitored to ensure policies have been implemented and used. Detected noncompliance with policies is recorded and reported. Remediation of policy noncompliance is based on the remediation policy.

Audit and non-repudiation

An *audit* is the recording of security events in a log. To ensure future claims that the security events as recorded are accurate and have not been altered (that is, are non-reputable), audit records are collected and secured. Audit records may be used for:

- Internal problem analysis.
- Evidence for a potential breach of contract, breach of regulatory requirement or civil and criminal proceedings, for example under computer misuse or data protection legislation.
- Negotiation for compensation from software and service suppliers.

Audit logs are created by system components, including operating systems, applications, and network devices.

Authorization

Authorization, also called *rights and permissions*, means allowing only users that are approved to access and receive the benefit of systems, data, applications, and networks (public and private). Authorization management is a life-cycle process for authorization data.

Binding security and secure conversation

Security binding is the protocol that ties security attributes together, such as an identity and the authorizations for the identity. Examples of security bindings are:

- Secure Sockets Layer and Transport Layer Security protocols provide for the secure authentication of servers and clients.
- X.509 certificates bind an identity to a public key.
- A Web cookie binds an identity to a service.

Security conversion securely maps information from one form to another form. For example, a password and ID may be converted to a common format for an authenticated identity. Confidentiality may convert plain text information into cipher text using an encryption key or keys.

Credential exchange

The purpose of a credential subsystem in an IT solution is to generate, distribute, and manage the data objects that convey identity and permissions across networks and among the platforms, processes and security subsystems within a computing solution. Credentials are created as a result of a successful authentication. Some common types of credentials are:

- X.509 public key identity certificates that bind an identity to a public key
- X.509 attribute certificates that bind an identity or a public key with some attribute
- Kerberos tickets that are encrypted messages binding the holder with some attribute or privilege
- Encrypted cookies

Credential exchange is the process of passing a credential from one entity to another using a protocol trusted by both entities, or a protocol in which both parties can establish mutual trust.

Identity federation

Identity federation is the life-cycle management of cross-enterprise identities. Such identities may be centrally managed, or may rely on trusted third parties. Trust federation includes:

- Trust management
- Trust brokering
- Single signoff
- Cross-enterprise identity mapping
- Cross-enterprise identity provisioning

Identity management

In accordance with document security policy, *identity management* includes the following:

- Identity proofing, identity approval, and identity rights authorization
- Identity token creation and token distribution to the user
- Provisioning of user identity, rights and profile to relying parties (operating systems and applications). Provisioning can be dynamic
- User profile management
- Enabling user self-care

- Delegate administrative responsibility for approval and authorization as needed
- Processes for token changes to IAW policy as well as revoking and approving issue of new or changed tokens
- Performing identity management in accordance with security policy

There are two facets to Identity Management:

- **Token:** Refers to an object that an entity possesses and controls, typically a key or password, used to authenticate the entity's identity. The token is provided to the entity as a result of successfully completing the identity proofing and registration processes.
- **Credentials:** Refers to objects used in authentication that bind an identity or an attribute to a subscriber's token.

Intrusion defense

Intrusion defense provides defense against attackers attempting to gain access to a network, device or host. Intrusion detection and response capabilities monitor network segments and hosts within a centralized operational and management framework. Responses to detected intrusion attempts include inputs to event management systems, paging and trouble ticket systems.

Intrusion defense is installed on hosts, desktops, notebook computers and on network devices. Intrusion defense management includes the life-cycle management of intrusion detection mechanisms such as the following: hosts, desktops, notebook computers and on network devices:

- ID application distribution and updates to authorized platforms. Host platforms may be initially loaded with ID clients, or fetch ID clients from the ID manager.
- Notification that ID updates are available.
- Making ID clients and updates available for automatic download when the host platform connects to the manager.
- Receiving host ID security event logs, performance log files, and host ID configuration data.
- Providing summary ID event information and alerts.
- Providing reports.

Key management

In accordance with document policy, *key management* provides life-cycle management for public-private key pairs using a trusted public key infrastructure (enterprise or outsourced) operating in accordance with a documented certificate policy. Private keys and X.509 certificates can be used to provide authentication, confidentiality, data integrity, and non-repudiation for transactions and other data.

Mapping rules

Mapping rules are used to convert a security item from a form understood by an origin process to a form understood by a destination process. For example, an application can authenticate a user via any mechanism it chooses (ID and password, certificate, and so on), and then based on the mapping rules, convert the authenticated identity to an identity format defined for a directory.

Network security solutions

Network security solutions for on demand provide secure connectivity and access control to and for the enterprise network. Remote connections to the enterprise network can use a variety of technologies such as dialup and Virtual Private Network (SSL and IPSEC). Network firewalls permit only connections that are specified, in directions that are specified, and using protocols that are specified. Network security solutions feature centralized log and security event audit trail generation and collection, and report generation.

Privacy policies

Privacy policies are security policies for managing access to and use of sensitive personal information, referred to as privacy-sensitive information.

Policy management

Policy management in the on demand security infrastructure is the consistent application of enterprise security policy to on demand infrastructure components, services, and applications, as well as network security solutions. Policy management is applied independent of application logic and operating system platform, and it includes trusted identity and token life-cycle management identity, access control and authorization life-cycle management, federated identity life-cycle, privacy, single signoff, compliance determination and remediation, security event auditing and processing, and failure situations.

Security policy expression

Security policy expression is the means by which security policy is applied to or implemented for specific IT system components and applications (for example, firewall filtering rules in a file, hardware settings, and network configurations).

Secure networks and operating systems

Secure networks are networks that have implemented logical and physical access controls, and may have implemented confidentiality, data integrity, and non-repudiation security services to restrict data access and network management to authorized personnel or entities. *Secure operating systems* are operating systems that have implemented logical and physical access controls and may have implemented confidentiality, data integrity, and non-repudiation security services to restrict data access and network management to authorized personnel or entities. Secure networks and operating systems generate security event audit records and are securely managed.

Secure logging

Secure logging is the means of recording security events and the protection provided to such logs to ensure their non-repudiation. Secure logging also includes a means for processing logs and generating reporting.

Service and end-point policy

Service and end-point policy refers to corporate security policy applied to or developed for services and information technology end points including response to legal, regulatory, and legislative requirements. *Service policy* states the specific security requirements for a service that generally is provided by a configuration of hosts, networks components, and applications. *End-point policy* states the specific security configuration to be implemented an

individual host, network component, or application, and the protocols used to implement the service policy.

Trust modeling

According to the ITU-T X.509, Section 3.3.54, trust is defined as follows:

G*enerally an entity can be said to "trust" a second entity when the first entity makes the assumption that the second entity will behave exactly as the first entity expects.* A *trust model* is a description or definition of how trust is established or conveyed between two entities or among multiple entities that operate under a common set of security policies.

Virtual organization polices

Virtual organization policies involve a statement of security policies for an IT system supporting the business needs of a specific subset of an enterprise or an IT system supporting cross-enterprise business needs operating under a common objective.

Appendix B: Abbreviations

CUID	Common User ID
FIM	Federated Identity Management
F-SSO	Federated single signoff
F-SSOff	Federated single signoff
ID-FF	Liberty Identity Federation Framework
IdP	Identity Provider
IDS	see ITDS
ITDI	IBM Tivoli Directory Integrator, also called IDI
ITDS	IBM Tivoli Directory Server
ITAM	IBM Tivoli Access Manager, also called TAM
ITFIM	IBM Tivoli Federated Identity Manager, also called TFIM
ITIM	IBM Tivoli Identity Manager, also called TIM
PoC	Point of Contact
OASIS	Organization for the Advancement of Structured Information Standards
ODOE	On Demand Operating Environment
SAML	Security Assertion Markup Language
SP	Service Provider
SSO	single signon, also called SSOn
SSOff	single signoff
TAN	Transaction Number
WSS	Web Services Security
XML	Extensible Markup Language
XrML	Extensible rights Markup Language

Appendix C: References

The following resources provide additional information about the topics discussed in this paper:

- *IBM Federated Identity Management*, white paper, Heather Hinton, Anthony Nadalin, Nataraj Nagaratnam, Venkat Raghavan, June 20, 2004.
- *Federated Identity Management and Secure Web Services,* REDP-3678, Axel Buecker, Heather Hinton, Venkat Raghavan, 2003.
- *Security in a Web Services World: A Proposed Architecture and Roadmap*, IBM and Microsoft, April 2002, http://www.ibm.com/developerworks/library/ws-secmap/
- *The Secure Enterprise: Charting IBM's Security & Privacy Strategy to Support On Demand*, Chris O'Connor, IBM Security Strategy Team, August 2003.
- *On Demand Functional Security Architecture*, White paper, Sridhar Muppidi.
- *Federated Identity Management with IBM Tivoli Security Solutions*, SG24-6394, Axel Buecker, Werner Filip, Richard Becke, Tony Cowan, Subodh Godbole, Sampath Kariyawasam, Harri Stranden, September 2004.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Austin Center.

Jeff Crume is an Executive IT Security Architect in IBM Advanced Architecture Support group. He is the author of a book entitled *Inside Internet Security: What Hackers Don't Want You To Know* and has written articles on cryptography and virtual private networking. He holds a Certified Information Systems Security Professional (CISSP) industry certification and is a frequent speaker at international industry conferences. Jeff has a degree in computer science from North Carolina State University and has over 20 years' experience in the area of systems management and IT security. He also serves on the editorial board of the *Information Management & Computer Security* journal published out of the UK.

Axel Buecker is a Certified Consulting Software IT Specialist at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on areas of software security architecture and network computing technologies. He holds a degree in computer science from the University of Bremen, Germany. He has 18 years of experience in a variety of areas related to workstation and systems management, network computing, and on demand (e-business) solutions. Before joining the ITSO in March 2000, Axel worked for IBM in Germany as a Senior IT Specialist in software security architecture.

Keith Gordon is a Senior Software IT Architect with commercial customers in Bethesda, Maryland. He holds degrees in mathematics and civil engineering from Carnegie Mellon University. He has 15 years of experience with IBM with six years as an IT Architect developing solutions for clients across IBM's software portfolio.

Jim Heid is a Software IT Architect with over 20 years experience in IT as a programmer, system administrator, project manager, and software architect. He currently works with small and medium-sized business clients in the Pacific Northwest. Jim holds an masters in business administration (MBA) with a concentration in quantitative methods from the University of Washington. Before joining IBM he worked in the Computing Division of Los Alamos National Laboratory.

Jatinder Pannu is a Senior Certified Client IT Architect in the Public Sector, IBM Americas. He has over twenty years of IBM experience in customer facing roles in building solutions to address business problems. He has presented extensively on e-business topics at IBM, customer and industry forums. Most recently his focus area has been the on demand operating environment. He has an MBA from Duke University and a masters degree in computer science from Syracuse University.

John Sanders is an Executive Architect in the Sales and Distribution organization. He works mainly with the telecommunications industry on SOA and VoIP projects. He has more than 20 years experience in designing and implementing systems. He is an IBM Certified IT Architect in Systems Integration.

Andreas J. Schmengler is a Certified Executive IT Architect in IBM Germany. He holds a degree in Computer Science from the University of Bonn, Germany. He has more than 20 years of experience, mainly in the disciplines of networking, IT security and pervasive wireless solutions and has led design and architecture engagements for complex IT infrastructures. He is a member of the IBM Technical Expert Council (TEC) and lectures at the University of Applied Sciences, Cologne, Germany.

Thanks to the following people for their contributions to this project:

Denice Sharpe
ITSO, Raleigh Center

Mike Campbell, Donald Cronin, Jon Harry, Heather Hinton, Paul Landsberg, Srihdar Muppidi, Nataraj Nagaratnam, Chris O'Connor
IBM US

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on October 26, 2004.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. JN9B Building 003 Internal Zip 2834
  11400 Burnet Road
  Austin, Texas 78758-3493 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	Access360®	Redbooks™
@server®	Domino®	Redbooks (logo)™
Redbooks (logo) ™	IBM®	Tivoli®
e-business on demand™	Lotus®	Wave®
ibm.com®	Metamerge®	WebSphere®

The following terms are trademarks of other companies:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.