Federated Dynamic Security and Process Management Within Service Grids

Martin Milani
Chief Technology Officer
Intersperse Inc.

John Seely Brown
Former Chief Scientist
Xerox Corp.

Abstract: *The latest trend in distributed systems architecture is the development of Service-Oriented Architecture (SOA) where new and existing software assets are broken down into services that are available to multiple applications and processes. The vision is that this new architectural paradigm will lead to a world where organizations will share services composed dynamically from components across many distributed systems and spanning different organizations. But before this vision can become reality, software architects and businesses must find a way to provide consistent real-time and context aware security across distributed processes while meeting the diverse needs of different shareholder organizations, each with its own security mechanisms and policies. This paper proposes a service grid construct supporting federated and dynamic security within the service grids, explaining how the federated security concept meets the needs of diverse systems and organizations and exploring the business and technology implications of the federated security grid concept.*

INTRODUCTION

The hottest topics in IT circles these days are business process management, distributed collaboration, web services, grid computing and security. And then there are, of course, the on-going discussions surrounding what Service-Oriented Architecture (SOA) really is and why it even matters. In this paper, we will discuss how these topics come together from a security perspective, and how they logically sit at the core of a *service grid* construct.

At a conceptual level, one easily can argue that an enterprise consists of a number of employees, groups, departments and divisions with their respective roles, the relationships and constraints among those entities and roles, and a set of rules and constraints or processes that define those relationships within the enterprise. In fact, the entire enterprise is a "collaborative hub", in which groups work together in defined processes, and their output is some unit of work, a product or service or both.

As web services standards and technologies mature, businesses and system architects are increasingly looking to web services based on Service-Oriented Architecture (SOA) as the model that will take the automation of business processes beyond the walls of the enterprise and beyond the static workflows of today's enterprise integration. The vision is that packaging applications and components as web services will enable fully dynamic, fully distributed, collaborative real-time business processes with all the autonomy and fluidity of business interactions in the physical world.

But distributed services systems have no value to business and do not have a place in a mission-critical business environment unless they are as secure, highly available, and reliable as applications within the enterprise firewall. Many current component technologies were designed for interactions within the enterprise where there is (theoretically) a fair degree of control and consistency, and current web services standards lack some of the features needed for a true enterprise-to-enterprise SOA.

This paper focuses on security as a critical requirement for distributed SOAs. Just as businesses are made up of individuals, tasks, and organizations with differing security needs and processes, SOAs will be comprised of organizations, applications and systems with different security requirements. Inter-enterprise processes built on distributed SOA (DSOA) will require new security models that decouple security from applications to provide consistent protection for business information assets, personal information, and business transactions between organizations. We will examine service grids and federated security as a model for implementing security in distributed SOAs, and we will consider the implications of this new security model for networks, future Web service standards, and business models.

THE EVOLUTION OF SERVICE-ORIENTED ARCHITECTURE

The concepts surrounding service-oriented architecture have been around and evolving since the mid nineteen-eighties. But now, with IP-based networks and Internet connectivity virtually everywhere, these concepts are once again in the limelight. In an SOA, different pieces of an application could reside on different systems and communicate with each other in a "loosely coupled" manner over a network. "Loosely coupled", in this case, means that the communication is asynchronous (i.e., non-blocking), and that the interfaces to the applications are published to a directory service and are available dynamically to each other. In fact, web services as defined today, is a primitive and limited case of the Distributed Services Oriented

Architecture (DSOA) (Web services today are thought of as loosely coupled, but in reality as defined today they communicate via RPC, which is a synchronous, tightly coupled protocol).

With the drive to real-time systems and business process integration beyond the walls of the enterprise, business requirements and ecosystems have caught up with the related technologies. Today there are real high-level business requirements and use cases that require the distributed systems and distributed service architectures envisioned years ago by distributed systems architects and technologists.

The next generation of distributed systems will be built on the foundation of component technologies being used inside the enterprise today. CORBA, J2EE, DCE, DCOM, and .NET are all used to create component-based and object-oriented applications. Fine-grained business logic is defined and programmed into objects. A group of objects working together comprises a component that performs a higher-level business function. Groupings of components can be packaged as services callable by multiple applications. All of these objects, components and services interact with each other through well-defined public interfaces and perform the business logic of an enterprise.

All these component technologies were designed to work within the walls of an enterprise. The new world envisioned by web services specifications requires these systems to interact with outside parties such as customers, business service providers, vendors and software services vendors. Objects, components and services will interact with other objects, components and services that live outside the firewall and most likely inside the firewalls of another enterprise. Web services standards such as XML, SOAP, WSDL, UDDI and others were defined to help facilitate these communications and interactions.

THE SECURITY CHALLENGE

Now, if we buy into the Web services vision, the enterprise walls are about to come down and barriers will be eliminated between systems and people in different organizations. Inter-organizational collaboration will flourish. Business processes will cross enterprise boundaries and blend in to external business processes from other organizations. Dynamic, real-time and context aware composition of distributed services into higher-level services will be common.

To technology executives and system architects, this means that we will have to provision intra-enterprise end-to-end process management and orchestration to handle situations not typically found in within the enterprise. In the world of inter-enterprise services, distributed transactions could take weeks rather than seconds to complete. In inter-enterprise processes, even a user’s role and privileges may be determined dynamically, based on the results of a combination of external services or transactions. We will need to manage long-lived and short-lived transactions across multiple resource managers in multiple enterprises using different integration, security and process management technologies. We will need to support users in different roles than they play within more deterministic enterprise processes. Security and security management sits at the core of all of this, so we must also find a way to manage different security policies and infrastructures seamlessly across multiple systems and enterprises. It must be noted that achieving consistent, effective security even within the walls of the enterprise is no simple matter. While each application or organization within a company may have good security mechanisms, intra-application security management and enterprise application integration have already proven challenging.

Naturally managing multiple authentication, authorization, encryption and certificate technologies across multiple companies will be far more difficult than distributed security management within the corporate firewall. Different requirements, different mechanisms, and security teams with different skill sets could leave us with an entangled web of multiple systems and security technologies interconnecting and communicating across possibly hundreds of different applications across numerous organizations.

Security is a set of distinct business rules and processes that govern the interactions among all subsets of the enterprise at a micro level and all the interactions between the enterprise and the outside world at the macro level. Security processes are-based on company bylaws, hierarchical structure, business rules and constraints, state and federal laws such as banking regulations and negotiated contracts, which protect the integrity of the enterprise and its business practices, its customers, partners and suppliers.

In the traditional component-based technology world, there are numerous design schemes, architecture and frameworks for distributed systems that offer a very high level of security and would protect applications and systems. We will briefly touch on some of the general security services that should be employed at the application layer without going into detail of security aware distributed systems architecture.

1. Identification and authentication

The users and calling objects/calling programs are authenticated through open systems standards such as Kerberos, X.500 and X.509 (PKIX).

2. Authorization and ACL

ACLs may be used to check a user or an object or a program acting on behalf of a user requesting some service and crosschecking it with user attributes, access controls and security profile policies so as to grant or deny the request.

3. Auditing

Auditing service logs all interactions and communications with the security service/subsystem.

4. Confidentiality

Communication between programs and objects can be made secure through encryption.

5. Delegation

Managing Access controls and authorization levels when a client program requests some service from an object/program which does not process the request to the end itself, but instead calls another object/program. Security polices need to move down the chain and be enforced.

6. Non-repudiation

Non-repudiation service provides evidence, proof of creation of a message, and the proof of receipt of that same message between two systems.

7. Digital signatures

Code, files and XML files may be digitally signed to ensure authenticity and integrity.

8.Security Administration

9. Reliable Message Transport

WEB SERVICES SECURITY

Security is a major component of service-oriented architectures. Unfortunately, security has been an afterthought in web services. It was not one of the foundations of the original web services specifications, and current efforts do not fully address the security needs of SOA.

SOAP, the original object access standard for web services, using http as transport, was envisioned and designed to bypass corporate firewalls and allow application connectivity. We believe this is a fundamental flaw; there are very good reasons why corporations have firewalls to protect their internal systems from access by outside systems and users. Security-related specifications in web services did not start till well after the original standards such as XML, SOAP, UDDI and WSDL were defined and ratified. The two main security specification proposals that have been submitted since are Security Assertion Markup Language (SAML) and WS-Security.

Security Assertion Markup Language (SAML) is an XML-based framework that allows for the exchange of authentication, authorization and user profile information among a group of organizations. In a nutshell, SAML provides a Single Sign On (SSO) solution for applications across multiple organizations in an autonomous distributed model where each organization is in charge of its own user and resources management.

The issue with SAML is that it is just a framework. It relies on the existence of a SAML-based SSO solution within the enterprise, before authentication and authorization information is shared with other partners. This is a big "if." There are very few organizations, even among the Fortune 100, that have such systems in place today, and such centralized authentication and authorization across the entire software infrastructure is almost non-existent in small and medium organizations. It is difficult, costly and cumbersome to roll out and maintain an enterprise-class SSO infrastructure, and in the distributed SOA model, these SSO systems would have to interact with possibly hundreds of other SSO systems point-to-point. Supporting and managing such crucial collaboration among large and small organizations is prohibitive in terms of both cost and complexity.

WS-Security is another web services specification, which has been submitted to OASIS. It proposes a set of new SOAP extensions to attach signatures and encryption headers to SOAP messages. Using XML encryption and XML signatures, WS-Security will support message authentication, integrity and confidentiality. Use of different security token technologies such as X.509 and Kerberos will also be supported. The assumption is that binary security token infrastructure exists and is fully functioning across corresponding applications infrastructures within participating organizations. Like SAML, WS-Security involves authentication, integrity and confidentiality for point-to-point messaging between two applications. And again, the problem is the complexity of point-to-point among different organizations, partners and service providers with different security infrastructures and technologies.

THE HAGEL-BROWN SERVICE GRID

Some of the players in the web services space have argued that web services will conform to peer-to-peer architecture, but our belief is that web services will not evolve to be primarily peer-to-peer, at least not for mission critical business processes. Significant infrastructure will be required to provide truly loosely coupled and highly secure and predictable connections among mission critical applications. Needed enabling services include asynchronous message queues, long and short-lived transaction management, distributed event management, metering, accounting, billing, auditing, non-repudiation, quality of service, traders, brokers, directory services, XML transformations, security and security management and many more**.** If the adoption of web services must depend on individual enterprises to implement all the event management, payment services, security, and other needed infrastructure, web services may never be adopted for widespread inter-enterprise business needs.

John Hagel and John Seely Brown have proposed the model of a "service grid" construct as a way to provide a variety of services needed by different parties in a distributed, inter-enterprise SOA.[1] Not to be confused with grid computing (a form of peer-to-peer computing that harnesses unused CPU cycles for large computing tasks), the Hagel-Brown service grid is comprised of a group of services that represent business functions. They are not computing resources, devices, files systems, or datasets. In the Hagel-Brown scenario, an enterprise could interact with multiple grids that offer different services. These services could range from supplier services to payroll services. The enterprise will use these service grids to interact with other partners, service providers, customers and suppliers' systems, just as businesses use outside services in the physical world. The service grid allows participating companies to act as a loose federation, each with its own internal infrastructure and policies, while each of the services in the grid functions as a trusted third party, defining and enforcing policies common to all the participating organizations. (One can compare this structure to the federal system of the United States, where each state has its own laws and public infrastructure, and the federal government defines and enforces policies common to all the states.)

[1] Hagel, J. and Brown, J. S., "Service Grids: The Missing Link in Web Services." (2002: John Hagel and John Seely Brown)

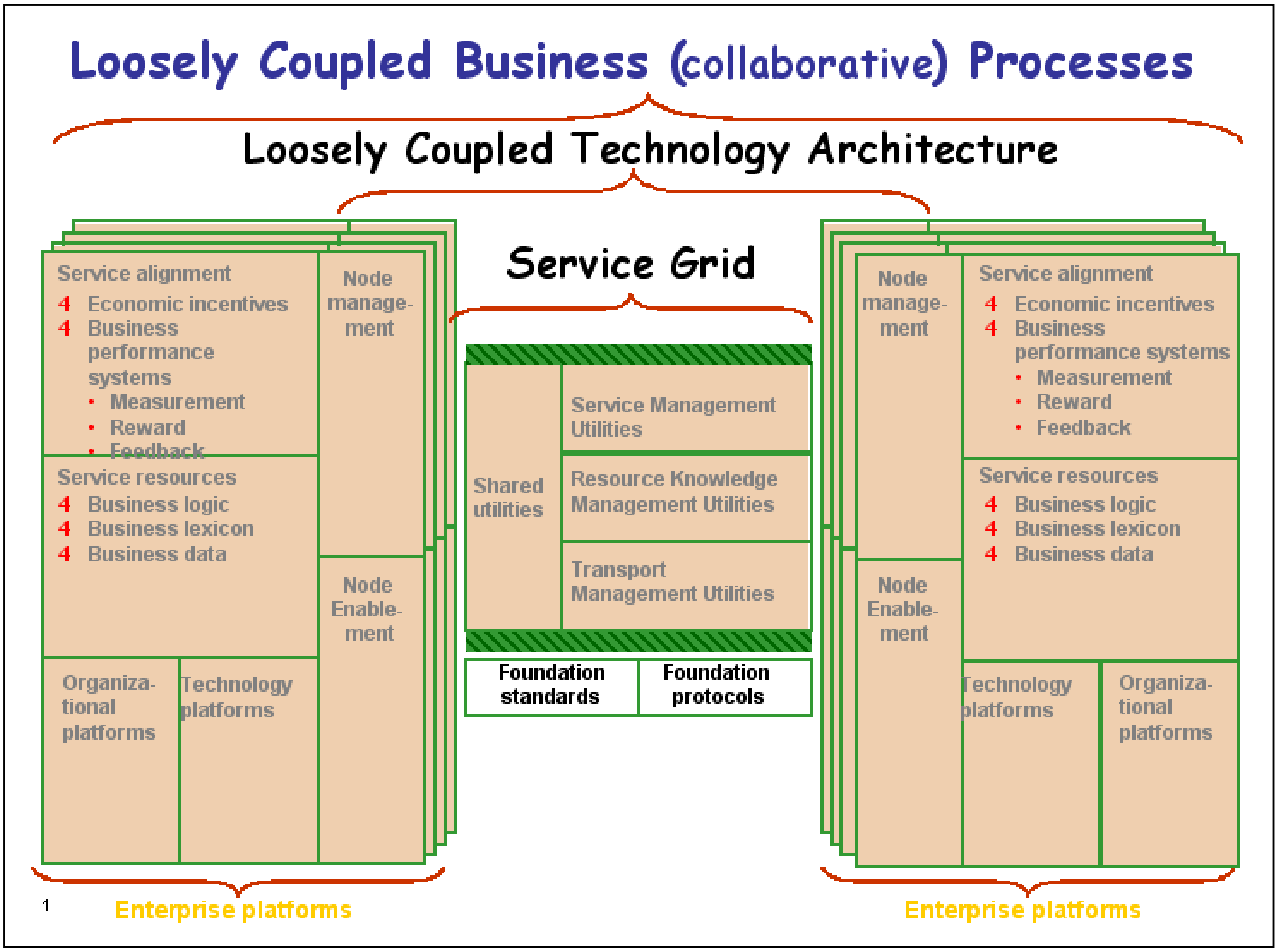

Figure 1: A Service-Oriented Architecture for Order Processing

Figure 1 shows a conceptual/architectural model of the Hagel-Brown service grid. Each enterprise has its own platforms, business applications, and organizational structure and policies. The service grid provides four broad categories of managed services that facilitate connectivity between enterprises:

- **Shared utilities** include security services such as authentication, authorization, and auditing; performance auditing and assessment utilities to assure agreed-upon performance levels between participants in the grid; and billing and payment utilities to aggregate charges and ensure payment for the use of Web services.
- **Transport management utilities** are messaging services for reliable communication and orchestration facilities that help companies assemble sets of services from different providers.
- **Resource knowledge management utilities** include Web services directories, brokers and registries to help participants find and use available Web services, plus specialized data conversion utilities.
- **Service management utilities** ensure reliable provisioning of Web services, manage sessions, and monitor conformance with service-level agreements (SLAs).

Logically, and we emphasize, NOT physically, the service grid could be imagined as a directory consisting of all the users, systems, resources and services. This directory would also include all of the business constraints, security constraints, policies, procedures and possible relationships among them. All interactions are mediated, orchestrated and managed by the service grid. From an integration perspective, if the Enterprise Service Bus (ESB) analogy were to be used to interconnect silos within an enterprise, then the service grid could be imagined as an inter-enterprise Super Bus in which ESBs from participating organizations would plug into. From a transport perspective it can be viewed as an advanced and intelligent message broker and router interconnecting different organizations. From a security perspective, it could be viewed as a security broker that is aware of all the security constraints, policies, procedures and possible relationships among them. A Security service grid construct sitting at the core of the service grid, acts as a security broker, brokering security services and managing security orchestration and mediation between those organizations.

In the Hagel-Brown scenario, the world could include many different service grids by which different verticals and eco-systems are represented and serviced. So in fact, the service grid itself could be a federation of smaller service grids that represent services such as payroll, sales, or inventory management, or services provided by the ecosystem of business service providers surrounding larger service grids. The anatomy of the service grid would resemble the structure of snowflakes in which all the smaller parts look just like the overall structure. In fact, our discussion of the service grid in this paper presents architectural design patterns that will serve equally well for the software and network design of the overall service grid or any portion thereof.

SECURITY IN THE HAGEL-BROWN SERVICE GRID

We have discussed how utilizing the service grid simplifies the complexity involved in connecting partner enterprises together. Suffice it to say in this paper that managing and maintaining the connectivity to partners and service providers is far simpler and cost-effective by connecting to a service grid rather than by directly connecting point-to-point to all these partners and service providers.

Today, to integrate two different security architectures, they must be integrated at all levels, the authentication information must be shared, the mechanism must be integrated, the data encryption policy must be shared and there must be shared ACL. Due to this complexity, most point-to-point integration efforts run into trouble either because they are very costly and un-manageable, or because they simplify the integration in a way that reduces security. The barrier to secure integration is very high in a multi partner scenario, as the overhead of managing multiple connections and multiple security policies are expensive, cumbersome and can quickly become overwhelming.

Furthermore, mediation services are provided by the service grid. Service grids, through their ability to natively integrate with each partner and provide mediation between partners at a business level, can significantly improve security and perhaps, more significantly, improve visibility of security. The decoupling of security through the service grid provides the ability of individual enterprises to constantly improve and upgrade their security without being constrained by their connections to the other partners. This also allows each organization to be moderately autonomous in terms of the use of preferred security technologies and products as well as enforcing and maintaining local security policies, procedures and policy hierarchies, thereby allowing a federated security ecosystem among service providers and consumers.

In this case, each enterprise manages and maintains only one connection to a service grid instead of having to deal with all the overhead of managing multiple connections and security policies. If point-2-point application connections were to be created and maintained, N*(N-1)/2 different pieces of connections would have to be supported and maintained. In medium and large organizations (N*(N-1)/2) could mean hundreds to possibly thousands of different application connections. In addition, maintaining and enforcing consistent security policies are far simpler and more flexible through the use of the service grid.

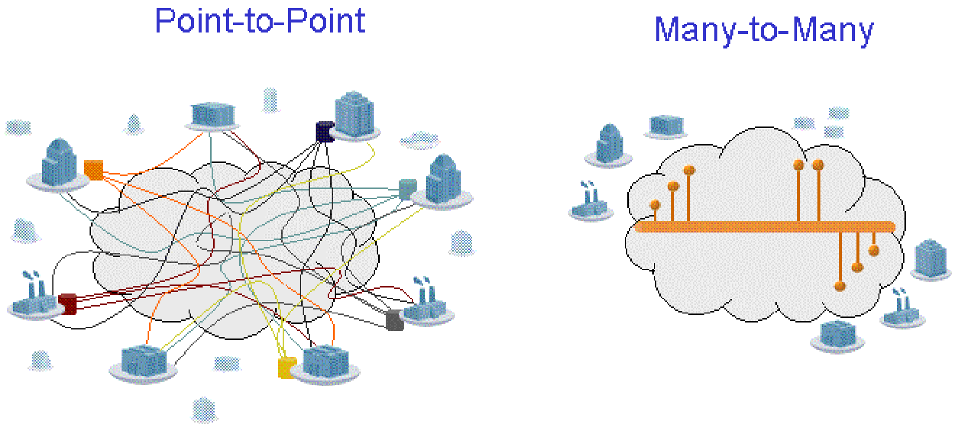

Figure 3. Point-to-point versus grid security model

The Hagel-Brown model calls for a federal entity that mediates interactions across of participating organizations and acts as a buffer to protect all the organizations from each other. The security service grid acting as the federal trusted entity can warrant the security practices of all the IT shops involved and certify their conformance to security-related SLAs, service-level contracts (SLCs), and agreements.

The Hagel-Brown federated security model assures the consistent security that organizations need to take part in multi-participant web services. Whenever there are more than two participants in a process in a point-to-point world it is very difficult to assure the protection of data and information. As data is passed across multiple organizations, it is hard to ensure that critical information about users and resources as well as other confidential information is well guarded by all organizations in the chain. Participants cannot assume that every organization in a delegation chain is equally secure.

We assume that the users and services are large in numbers and dynamic. Users, attributes, authorization levels, services, functionality and security policies are changing all the time. Security association is maintained and managed by the service grid between initiators (users and client programs) and responders. Security brokers negotiate on behalf of these systems and employ the appropriate security mechanisms and products based on security policies, profiles and degrees of trust.

Creating a standard set of security practices and policies is especially crucial when distributed services will be composed dynamically into higher-level services. The high-level service one requests may have certain policies, but what about the other services being called in the background by that service? In the distributed service-oriented software world, we can no longer afford to have security policies and logic embedded within the application logic of each application or each service layer.

If distributed service-oriented architectures are to become a reality, we need security mechanisms that can protect information and guarantee transactions as effectively as security technology within the enterprise firewall, while functioning as a trusted mediator between the security policies of possibly hundreds of different applications and organizations. To make inter-enterprise web services a reality, security must be enforced, managed and audited end-to-end by employing a multi-layered and multi-tired approach with an overall view and context management. By introducing a comprehensive dynamic, adaptive, context aware, multi-layered and multi-tired security framework, standard security practices and policies can be governed, managed and maintained across all parties.

The security service grid lies at the core of the Hagel-Brown service grid as a managed service. The decoupling of security from business applications and processes through the service grid frees individual enterprises to constantly improve and upgrade their security without being constrained by their connections to the other partners.

We believe that the service grid concept and construct would enable rapid adoption of web services and reduce the cost and complexity of SOA for the enterprise. Service grids would provide a far simpler and more cost-effective way to manage and maintain connectivity between partners and service providers than direct point-to-point connections. For inter-enterprise security, the service grid concept would allow the use of sophisticated security technologies and managed security services without putting the complexity and the burden of management on the edge (enterprise), instead pushing the complexity to the center (the service grid).

TECHNOLOGY FOUNDATIONS OF THE SECURITY GRID

Let us now turn to some high-level architectural aspects of the service grid and security management within the service grid. In doing so, we will give an overview of ideas and concepts that should be considered in constructing security frameworks within the service grid.
Our first proposition is that neither the service grid itself nor the security service grid at it's core need be-based on web services architecture as defined by web services specifications today. Web services specifications are often moving targets, with multiple competing submissions to the standards body by different groups and alliances (e.g., WSFL, XLANG, and WSCI). Such fragmentation will slow or even derail the adoption of standards for web services security and other shared services. However, the well-defined, proven security services in traditional distributed and services-based architectures would serve today to build a grid of security services within a general service grid. The security grid would expose these security services to the enterprise and to service consumers. This approach preserves existing security infrastructures and architectures already employed by the enterprise. It also speeds up the adoption of web services by organizations, as they do not need to wait for standards and or go through the expensive and cumbersome task of replacing already existing technologies and infrastructures.

THE NODE-ENABLED ENTERPRISE

At the edge of the service grid sits the "node-enabled enterprise." In order to integrate their processes with outside parties such as business partners, third party service providers, customer's enterprises must expose some of their applications, systems and processes as services. Internal components also need to be exposed when enterprises buy or rent web services from outside software vendors to enable their processes running over the Internet. To facilitate the connection between internal and external components, enterprises need to go though a metamorphosis, discussed in the Hagel-Brown article, which we call "node-enablement."

In order for the enterprise to expose itself to the world of partners, customers and third party service providers in an efficient and meaningful manner, two major steps must be taken:

1. Create an object-based abstraction layer that sits on top of internal systems. The abstraction layer provides connectivity between various applications that are employed within the enterprise and act as a message broker. These applications communicate with each other only through this abstraction layer and will no longer communicate directly through dedicated point–to-point connections. The abstraction layer is a message broker and acts as the enterprise message bus. It also acts as the security broker and manages all the aspects of security across internal applications.

   This abstraction layer also includes a management and monitoring layer. Applications and services need to have a standard way of discovering the status and attributes of each other in terms of instrumentation, performance, availability, security logs and events, configuration, run-time errors and exceptions to name a few. The management and monitoring abstraction layer would provide the facilities for all of such services to discover and perform transformation on all required management information. Services would subscribe to events of interest instead of directly querying each other about management and security events in a point-2-point manner, and they should do so without having any knowledge of internal components and technologies that other applications/services are composed of. This management and monitoring abstraction layer would provide a unified access to network, systems and application management information, hence marrying network, system and application layers. This would enable organizations to develop micro-level management applications with real-time access to all management events including security events across all tiers and layers. The collection of these micro-level management applications would enable the construction of an enterprise level federated management and monitoring infrastructure that is required to effectively "real-time" manage the "real-time" enterprise.

2. Once the internal abstraction layer is defined, constructed and operational, some services and systems can be exposed to the external world. These interfaces should not be exposed directly. They are only exposed to an external object-based abstraction layer, which also acts as a message broker and is maintained by the enterprise. The external abstraction layer sits on a DMZ on the edge of the enterprise. It acts as the externally focused enterprise message bus, message broker, and security broker. The external abstraction layer communicates directly to the service grid. This is similar to how internal and external DNS is managed by organizations today.

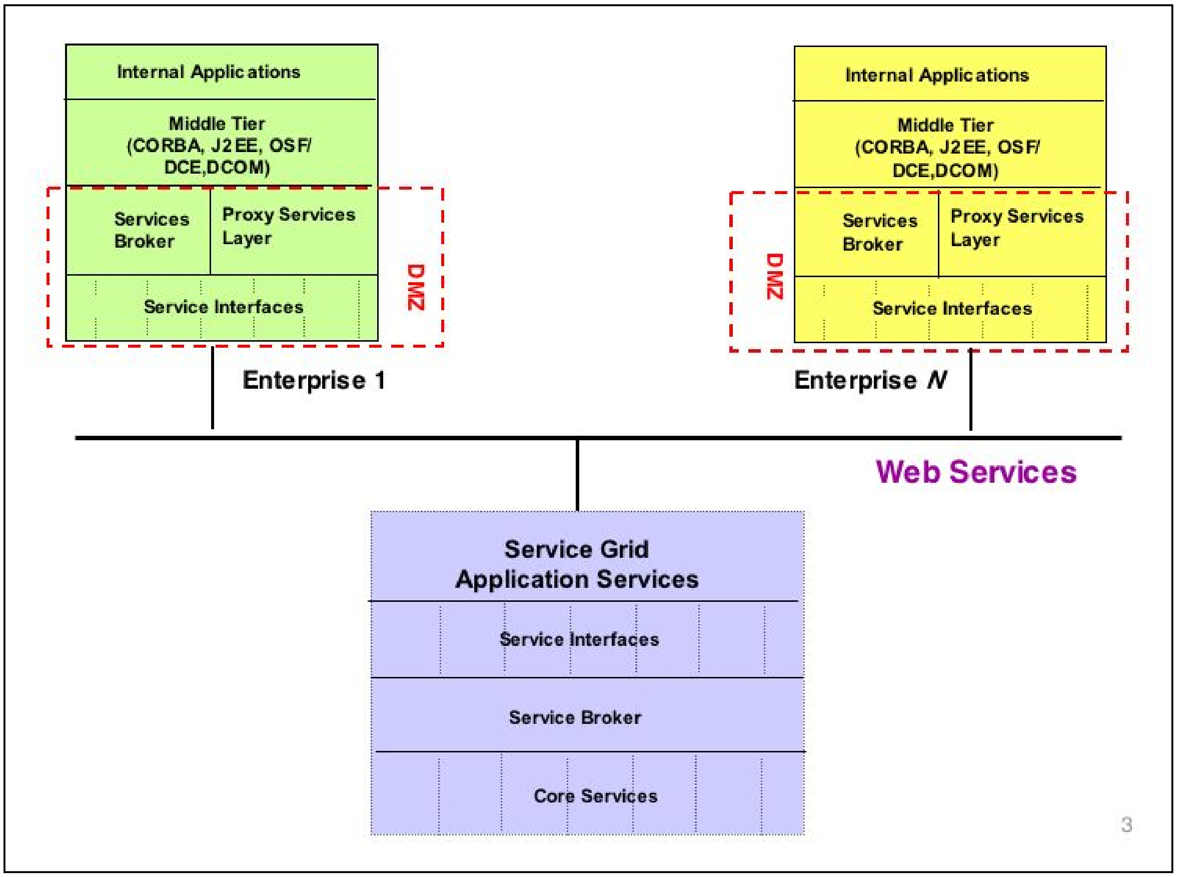

Figure 4: The Node-Enabled Enterprise

The external hub, residing on a DMZ of an enterprise, acts as a proxy and a message broker that communicates with components that are exposed to it by the internal hub residing inside the firewall of the enterprise. The internal hub communicates with the middle-tier of the enterprise that communicates with enterprise applications APIs directly.

In this scenario the enterprise systems are three steps removed from the actual service consumers in the outside world. The service brokers, acting as traders, broker the communications and access between service consumers, the service providers and the service grid (and the security service grid). This is done-based on identity, identity context, authorization level, security clearance, degree of trust, quality of security service, type of service, price, bandwidth, protocol desired, etc. dynamically on a case by case basis.

QUALITY OF SECURITY SERVICE (QoSS)

Even within the enterprise, security policies and profiles are changed and revised all the time. In an inter-enterprise distributed SOA, the policy management overhead could grow exponentially as thousands of policies and rules need to be created, maintained and updated all the time between hundreds of participating organizations and service providers. Static, linear and predetermined policy and processes management can't handle this scenario.

To handle changing security requirements in the distributed SOA, we recommend a new concept, Quality of Security Service (QoSS). QoSS rules would govern security technologies and protocols to be used as requested by the application and governed by existing policies, rules, history and trust. Different systems and communications require different levels of security applied to them. . This means different security services, policies and processes should be able to dynamically discovered and used. Dynamic composition of these security services to create higher-level composite services should be possible.

First, based on a predefined request for a certain security level from the application service layer, certain services to be rendered to the requesting service. One can imagine this as a security stack that is composed dynamically based on certain requirements and requests from the application. We strongly believe that both fine and coarse grain security services and every thing in between should be available to the application layer. In order to pick and choose what components and protocols to be used the application needs to be able to discover what services are available to it at a given time. It also needs to be aware of the network connectivity and network layer security services that are available to it for a certain request. QoSS agreements would be negotiated dynamically. An association control service needs to provide the service elements for establishing, handshaking and agreement on security level and termination of such association. The association could be long lived or short lived depending on the nature of the request.

New world of process management

There are different groups of processes within an enterprise such as internal business processes that mainly are involved with internal operations much like internal organs in a human body. There are external processes such as customer and partner/service provider interaction processes. We will call these grouping process spaces. In the inter-enterprise SOA   world, these external processes living on the external process space take the center stage as enterprises start connecting and integrating their processes with customers', suppliers', vendors', service providers' and partners' processes.

The buzz of Web Services is painting the vision of a new era of facile process and data integration that moves from the data center and into the mainstream of day-to-day business activity. Though these are important new developments, they are targeted at automating the flow of data across applications and organizations. Their success is limited to environments that are predictable, based on largely static relationships. However, the nature of interaction, collaboration and resource sharing across different groups and divisions are highly dynamic and constraint based and the business process can go different paths depending on the context. This is a departure from finite state based process flow and data centric nature and design of the systems that run businesses and enterprises. The path is going from A to B. A is the start and B is the finish but there are many different ways to get from A to B and the different business context and constraints that are dynamic will drive the decision path. There are too many exceptions and permutations of possibilities that it is impossible to program and script all these cases in a traditional process flow mechanism. The dynamic nature of sharing relationships means that we require mechanisms for discovering, matching, and characterizing the nature of the relationships, requested services and security policies that exist at a particular point in time based on constraints and conditions at that point in time (context).

Invoking of Web service is conditional and contextual. In the real world of the enterprise there are often numerous service providers that offer a certain service. There are business and security constraints which are derived by many different factors such as location, availability, cost, level of security, existing contracts, trust, laws, regulations, financial consideration, etc which govern the decision which service provider is to be used for a given service. In the real business world, higher level processes (services) are really made of a number of lower level processes (services) that are usually dynamically composed based on conditions and context. The process of dynamic composition of services, there will be a number of different possibilities for each service to be chosen at each composition step. The coordination, mediation, orchestration and management of these complex, dynamic and state-full higher level processes (services) which interconnect enterprises from a process centric perspective and not a data level perspective, is extremely hard to achieve in a point-2-point scenario. The inter-enterprise process space sits at the core of the Hagel-Brown service grid concept. Again as processes are federated, the Hagel-Brown service grid acts as the federal government dealing with federal level process flow and orchestration.

THE NETWORK PERSPECTIVE ON SERVICE GRIDS

From a network perspective, service grids can be thought of as a form of private Internet. Like the Internet, service grids deliver resources at the time they are required from an undetermined source. Since resources are allocated on demand, an extension of IP addressing is used to make sure consuming tasks are mapped to appropriate resources to get the work done.

Because the resources and infrastructure that are required to deliver these services is beyond the skills of most current Internet providers, a new form of service grid operator will evolve. The typical service grid must provide a very high level of security, performance monitoring and usage tracking within their grid and its connections to other networks. The demands of the grid and its network interconnections will mean that a new layer of network services must be built within the current network infrastructure. Services which track, monitor and route network traffic will be imbedded within the IP layer, delivered by a new set of "power tools" managed by service grid operators. These "Intelligent Network Routing Tools" (INRT) will face the challenge of delivering dynamic services, efficiently using compute resources, providing security, monitoring performance, and tracking usage for billing.

Since a high degree of security expertise and new network and systems management tools are required, current network security experts will most likely evolve to become the new service operators. Security experts will closely monitor network performance at all times, using the new layer of intelligent network routing technology (INRT), built upon the current network technology stack. Like the switchboard operators of early telephone companies, these grid operators will work closely with multiple providers of "connections," including backbone providers and ISPs to deliver intelligent "value added" packet routing.

It is easy to see that operating a grid of resources will require a form of expertise that is scarce in most enterprises. For this reason, most enterprises will connect their current Intranet infrastructure directly to a service grid operator's network, moving resource intensive processes to the grid over time. Enterprises will connect at a very high level. Through the use of choke and edge routers running IPSec/IPV6 (both sides) the traffic flow between the service grid and the enterprise will be protected, with security brokers and proxy services to enforce existing security policies and invoke appropriate security services such as authentication, authorization and encryption.

SSL/TSL and/or IPSec-based schemes will protect the communication channels. Service levels will be-based on the identity of the requestor, type of request and authorization access policies. Once services are assigned, access is granted to the appropriate ORBs or objects/methods/APIs that offer the services. All communications between objects and ORBs/programs within the service grid itself and the external enterprise abstraction layer (node enabled piece of the enterprise) are authenticated, authorized and encrypted. PKI and directory services such as X.509, X.500 (LDAP) and SSL could be used to achieve this.

By using a PKI infrastructure, authenticated traffic will be routed intelligently across the service grid between different organizations-based on policies and use of policy-based routing (PBR). State-full firewalls check all packets and their contents flowing between different organizations. Transport services, such as asynchronous message queues with persistent queues can be used to provide high availability as well as guaranteed delivery of messages between clients and the service grid when needed. In case of network or systems failure, messages are guaranteed to reach their destination in order and will not be lost. Digitally signed code, files and XML working in conjunction with PKI technologies are used to further protect against malicious and unauthorized code.

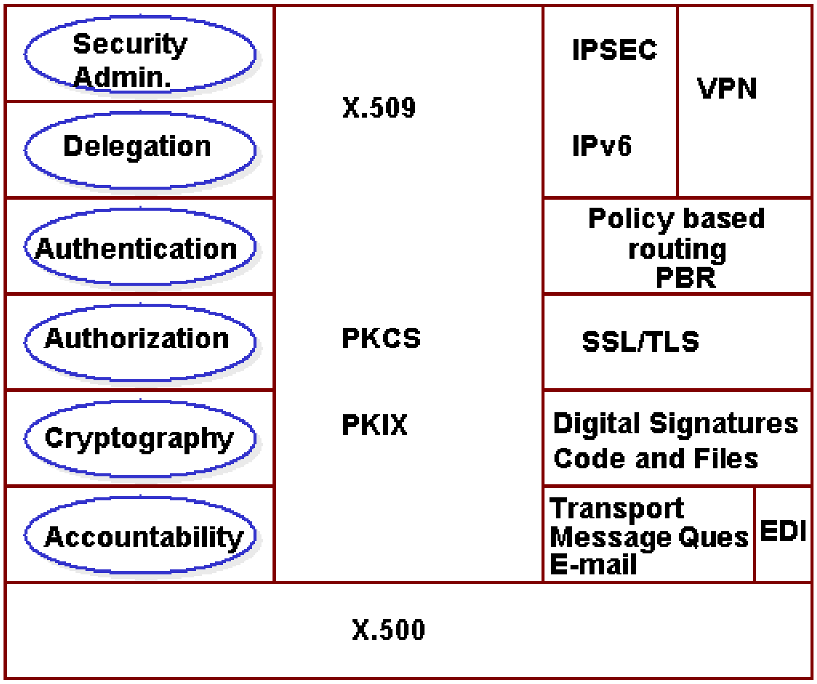

Figure 3. Network Architecture for the Security Grid

In one sense, networks will become an application resource delivered via a new abstraction layer that marries the network and application layers. This would allow for the creation of new “network aware” services [applications] and distributed composite applications. 'Object Routing' technology, will be applied to locate software components, software services, content, data and other network resident objects. This technology will create cost-performance advantage for distributed applications and distributed content, by allowing the existing server infrastructure to scale and dynamically support client connections. Routing agents will match client request with the closest, best available instance of host server and specific object requested.

One approach that has already shown promise is marrying network and application service layers. The core of their network-application layer is an extension of IP addressing that assigns specific MD5 "fingerprints" to each unique software object. By mapping these fingerprints to a routable address space, the new layer of services applies the usual routing algorithms such as

OSPF, BGP to deliver packets dynamically to the right resource. The address space is within current IP-v6 address space, allowing object routers to create VPN-like connectivity, to compute network latency, and to compute a forwarding path. The application of these technologies within the object router allows the entire service grid to use the optimal location for content/service delivery that is customized to the client locale.

In the simplest implementation, server resources (e.g., EJBs, servlets, URLs) are mapped by published hash algorithms; mapped to IP-v6 address space and then routed throughout the object router network. Clients can either send requests as URLs or computed hash values, using either HTTP or a published API to retrieve the best location where the objects exist. This is a platform neutral technique, which can be easily adopted for a large number of networks or platforms. The solution is ideal for any software platform/technology that requires redundancy; fail over; load balancing on a global scale and delivery from the edge (closest to the end user). The solution also solves a need to control latency and data-path of delivery with an easy, transparent “built-in” solution.

By interjecting security at different layers such as IP, TCP, Data and application and providing the framework in which each layer is conscious of all other layers (hence marrying the network and application layers), it is possible to achieve a very high degree of security. One can even imagine using XML encryption and digital signature along side with SSL/TLS, and IPSec/IPV6 in transport or tunnel mode at the same time. Of course this might have a high toll at the expense of performance but for a very high degree of security. The information would be protected not only while in transport across different organizations but in fact protected end to end against intrusion and eavesdropping inside the organization’s firewall (inside hacking).

This level of completeness, complexity and cohesion is an important necessity for the future of a meaningful web services world. However, through the use of the service grid, this complexity is transparent to the end users and enterprises. This highly complex and complete security framework is moved to the center from the edge, thereby keeping the edge simple! It also allows much innovation on the edge without compromising security. At first, the communication from the edge to the service grid, in fact, could be using very simple technologies such as SOAP/XML/HTML without the need for supporting and understanding the complex security models and techniques that reside within the service grid. There are many new initiatives by organizations such as OMG under way to define the next generation of SOAP and XML technologies as well as to better integrate them with already established and mature distributed computing technologies such as OSI, CORBA, JAVA/EJB/JMS and IIOP. SOAP and XML will evolve as time passes and become much more complete and security conscious.

With rapid growth in bandwidth, computing power and memory bandwidth on new servers and systems combined with the decrease in the cost of bandwidth, one can imagine that in a few short years, stateless types of protocols such as HTTP and SOAP, will be replaced by much more complete and rich set of stateful, asynchronous, synchronous and reliable protocols which will reshape the nature of distributed computing and mission critical web services over the Internet.

In closing, we envision the world of web services that is not limited to simple application integration or one in which one can use a stock quote providers piece of code to present a stock quote on a corner of a web page. We believe ultimately, the world would comprise of many sophisticated distributed software ecosystems that will run and facilitate mission critical applications and business processes and transactions among their constituents. We foresee a multi-protocol and dynamic world in companies will choose the best technology address different needs for a very long time. The RPC and SOAP/http protocols of today’s web services stack will only be used for certain areas of software architecture where they fit specific needs. There are many areas where RPC is not a well-suited architecture for design and other paradigms would be much better suited. In fact the first of the web-based applications such as SMTP/mail, ftp, news and DNS did not use RPC like communication. So in this chaos we call the technology world today, in order to evolve, we believe the service grid concept is an important first step.

We admit that the above is but a sketch of a vision of providing a level of security for a distributed service architecture that is mediated by a managed service grid construct. Crucial to our thinking is to find a framework that enables consumers (enterprises) to keep their own node enablement for web services simple yet to have access to whatever level of security is deemed appropriate for however they are using a shared web service within their own business processes. The service grid provides the mediation and visibility of security technology allowing secure integration between enterprises to be a policy decision. The decoupling of security technology by the service grid allows the rapid evolution of secure communication between enterprises and will result in a higher level of security rather than the opposite. We also do not claim that the current protocols in use today for invoking web services (e.g., SOAP) have all the appropriate security layers, but rather that these protocols can be progressively refined along with the emerging capabilities of service grids to provide whatever level of security is needed.

Service Grids and Computing Grids

There has been much interest in the subject of “grid computing” lately. The subject is at times confusing as grid computing often means different things to different people. It is important to draw distinctions between the Hagel-Brown service grid and some other grid concepts.

Most of the grid-computing efforts to date have come out of the “computing grid” concept. This concept dates back to the days of time-share computing, which was about having access to computing power (super computer access for number crunching for academia and research) without really owning the computers. In this computing grid model, transparent access to computing resources such as MPP systems and high performance data storage is the core fundamental concept. IBM, a pioneer in this area has introduced “on-demand computing” as a natural extension of the concept. Based on this concept, high performance computing becomes a utility much like electricity or water, where you pay only for what you use.

A few other grid projects are similar in concept. IBM, in partnership with Globus, has submitted OSGA (Open Grid Services Architecture), in which web services specifications and grid computing concepts and protocols are married together. This introduces much needed mission-critical services in to the world of web services. This move also allows IBM to marry the on-demand utility computing and web services concepts, and offer managed-services of combined software and hardware.

The Legion grid project from University of Virginia is an object-oriented grid construct that also originated out of the computing grid concept. Legion is a distributed computing platform that combines very large collections of independently administered machines into unified, coherent environments. Like traditional operating systems, it builds on a diverse set of lower-level resources and provides convenient user abstractions, services, and policy enforcement mechanisms. The difference is that Legion's lower-level resources can include thousands of heterogeneous processors, storage systems, databases, legacy codes, and user objects, all distributed over wide-area networks spanning multiple administrative domains. Legion provides the means to pull these scattered components together into a single, object-based meta-computer that accommodates high degrees of flexibility and site autonomy.

Though service grids and computing grids are fundamentally different, the lines are getting blurry. In fact, the Globus project is steaming full speed towards the service grid model from the technology perspective. However, unlike Globus and Legion, the Hagel-Brown service grid concept was conceived from the start as a “managed service grid. The Hagel-Brown service grid not only offers the fine-grain and coarse-grain service elements that are offered by Globus and Legion, but it would also offer much higher level coarse-grain services that represent high level business processes and activities together with corresponding business and security constraints between enterprises as defined by contracts, laws, SLAs, and SLCs.

For more information on the Globus and Legion projects, see the Bibliography.

Acknowledgements

We thank John Hagel, Bill Coleman and Martin Brodbeck for their valuable input.

Glossary

ACL	Access Control List
ATM	Asynchronous Transfer Mode
CA	Certificate of Authority
CORBA	Common Object Request Broker
DES	Data Encryption Standard
DoS	Denial of Service
DDoS	Distributed Denial of Service
EDI	Electronic Data Interchange
EJB	Enterprise Java Beans
ESB	Enterprise Service BUS
ICMP	Internet Control Message Protocol
IIOP	Internet Inter-ORB Protocol
IP	Internet Protocol
JMS	Java Messaging Service
LDAP	Light weight Directory Access Protocol
MD5	Message Digest algorithm MD5
OMG	Open Management Group
ORB	Object Request Broker
OSGA	Open Services Grid Architecture
OSI	Open Systems Interconnect
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509
RAD	Resource Access Decision
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UDDI	Universal Description, Discovery and Integration
VLAN	Virtual Local Area Network

WSCI	Web Services Choreography Interface
WSCL	Work Spaces Coordination Language
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
XML	Extensible Markup Language
X.500	ISO 9594-1, CCITT X.500 Directory Services Standard
X.509	ISO 9594.8 CCITT X.509 Authentication Framework Standard

Bibliography

Hagel, J. and Brown, J. S., "Service Grids: The Missing Link in Web Services." (2002: John Hagel and John Seely Brown). http://www.johnhagel.com/paper_servicegrid.pdf

Hagel, J. and Brown, J. S. "Orchestrating Business Processes — Harnessing the Value of Web Services Technology." (2002: John Hagel and John Seely Brown). http://www.johnhagel.com/paper_orchestratingwebservices.pdf

Hagel, J. and Brown, J. S., "Orchestrating Loosely Coupled Business Processes: The Secret to Successful Collaboration." (2002: John Hagel and John Seely Brown). http://www.johnhagel.com/paper_orchestratingcollaboration.pdf

Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S. "The Physiology of the Grid : An Open Grid Services Architecture for Distributed Systems Integration." June 2002. http://www.ibm.com/grid/pdf/it_exec_brief.pdf

Foster, I. and Kesselman, C. "Globus: A Metacomputing Infrastructure Toolkit." *International Journal of Supercomputer Applications*, 11(2):115-128, 1997. Provides an overview of the Globus project and toolkit. Available at http://www.globus.org/research/papers.html - anatomy

Hao He. "What is Service-Oriented Architecture?" Sept. 30, 2003. Published on XML.com. http://www.xml/com/pub/a/ws/2003/09/30/soa.html

Humphrey, M. and Thompson, M. "Security Implications of Typical Grid Computing Usage Scenarios." In *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC)*. San Francisco, California, August 7-9, 2001. http://www.cs.virginia.edu/~humphrey/papers/humphrey_security.pdf

Natrajan, A., Anh Nguyen-Tuong, Humphrey, M., and Grimshaw, A. "The Legion Grid Portal." Submitted to *Grid Computing Environments 2001, Concurrency and Computation: Practice and Experience. http://legion.virginia.edu/papers/HPCS01.pdf*

Natrajan, A., Humphrey, M., and Grimshaw, A. "Grids: Harnessing Geographically-Separated Resources in a Multi-Organisational Context." In *Proceedings of the 15th Annual Symposium on High Performance Computing Systems and Applications(HPCS 2001)*. Ontario, Canada, June 18-20, 2001. http://legion.virginia.edu/papers/HPCS01.pdf

Tan, Y., Topol, B., Vivekanand, V. and Xing, J. "Manage Web Service and Grid Services with Service Domain Technology." February 2002. First article in the Business Service Grid series from IBM. http://www-106.ibm.com/developerworks/library/gr-servicegrcol.html