

# Designing the Infrastructure for Service-Oriented Architecture and Extensible Markup Language

## Using the Power of Web Services Applications

The shift from traditional application design to service-oriented architecture (SOA) principles using Extensible Markup Language (XML) and Web services promises increased IT agility and reduced technology costs. However, without effective production-caliber infrastructure, the benefits available from SOA and Web services will not be achieved.

Many early adopters that started with a simple Web service and a corresponding consumer group are now implementing composite or sequenced applications where multiple Web services are used to build an application. Others have found that the success of their initial deployment has created a huge demand to address the needs of new partners and consumers of Web services. Still others have found that the myriad identities required by disparate services create significant identity and access enforcement challenges.

As application systems are developed using SOA principles and implemented using XML and Web services, the increasing sophistication of these applications stresses the supporting infrastructure. Network traffic increases to support distributed application components; integration with identity and access management systems is required to identify and secure loosely coupled services and other elements; and monitoring, auditing, and control systems are needed to understand and manage dynamically changing application systems.

This document describes the requirements for an XML infrastructure that is designed to help ensure that the opportunities and benefits of SOA and Web services can be achieved. For the purposes of this document, infrastructure represents the supporting capabilities in the network that allow semi-autonomous, loosely coupled elements to operate together in a meaningful way.

## Standards and XML Infrastructure

Many architectural models and design patterns have been devised to support SOA concepts. A significant body of standards and specifications has been created to address many aspects of Web services development and integration and enable cross-platform, cross-application loose coupling.

However, the standards and specifications used to define even mature architectural models are not sufficient to fully address all the concerns that arise when developing an operational system. The task of addressing interoperability between various implementations is handled by several groups, such as WS-I and the Liberty Alliance. Valuable as these efforts are, the unique aspects of each Web service deployment leaves a significant number of system concerns that are not addressed.

The enormous body of standards and specifications underlying Web services require support from an infrastructure that is designed to address the implementation, operation, integration, and optimization requirements of practical and scalable deployments. These concerns must be addressed before usable and operational application systems can be constructed and deployed.

This is where XML infrastructure plays a vital role. XML infrastructure is the substrate that allows

complex distributed application systems to be built, tested, managed, secured, integrated, deployed, operated, validated, and monitored.

## Crossing Domain Boundaries

The complexity of these new application systems increases dramatically when the Web service components and consumers reside in different domains. Integration of business logic from many parts of an organization may require contribution of service components from different domains including the following:

- Servers
- Data centers
- Geographies
- Trust domains
- Organizational groups

The underlying policies and mechanisms that allow these boundaries to be crossed effectively must be provided as part of the implementation, deployment, and management environment supporting these application systems. XML infrastructure integrates the other systems and infrastructures within an organization that facilitates the crossing of these boundaries.

## The Essential System Problem

Applications constructed from XML and Web service components are rapidly becoming complex systems in their own right. It is necessary to understand the operation of the system and to be able to control the system to help ensure its optimal operation.

## The Blind Men and the Elephant

A traditional story describes an encounter between a number of blind men and an elephant.

While at a fair, four blind men hear that an elephant is on display in one of the tents. As none of them had ever encountered an elephant, they were excited to pay their money and have a chance to get close to this fabulous animal. One by one, each man entered the tent and felt the strange creature (Figure 1).

As the last man emerged, they group began to compare impressions. "My, that elephant was like the side of a barn," said the first man. "What do mean? It was more like a tree trunk," said the next man. "Nonsense. It was broad and flat and flexible," contended the third man. "You are all crazy. It was thin and short like a snake," maintained the last man. The problem is that standing in just one place, each blind man could appreciate only one part of the elephant.

Likewise, standing at an application platform, a person sees only the messages that arrive or are sent from this platform and so cannot understand the application system as a whole. To understand and control the operation of these new application systems, it is necessary to understand all the components and all the message flows through them.

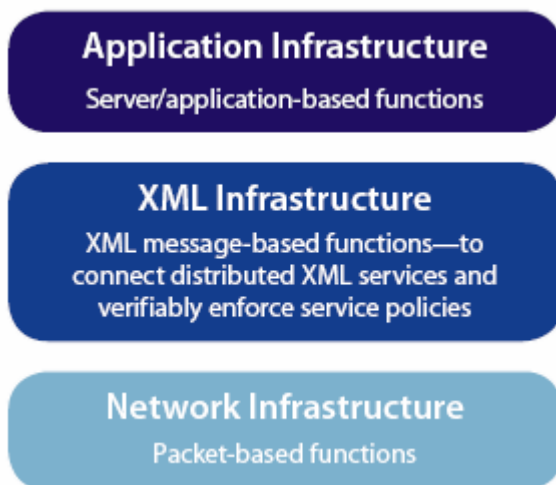
Figure 1 The Blind Men and the Elephant



## The Need for a New Infrastructure

A dedicated XML infrastructure is needed for several reasons. Web services provide a different level of abstraction from either a network infrastructure or an application infrastructure. The resulting XML infrastructure sits between the traditional network and application infrastructures, as shown in Figure 2.

Figure 2 XML Infrastructure



At the network level, the granularity of the application processing resources addressed and the need to understand complex application context are not supported by existing network capabilities. At the application level, existing infrastructures do not support the loosely coupled, highly distributed, and extremely heterogeneous models that Web services entail. Applications are becoming complex distributed systems that must be understood and controlled as a whole.

### *Heterogeneous Environment*

One of the differences between these emerging application systems and previous monolithic architectures is that heterogeneity must be assumed at every level. This assumption differentiates Web services from the previous distributed system models. The distributed business logic segments that are assembled to build an application system may run on different operating

systems, be created in different development environments, and operate across different network transports and many different underlying third-party software systems and vendors. XML infrastructure helps ensure the interoperability of heterogeneous systems.

## ***Operating Environment***

Traditional operating systems, virtual machines, application containers, enterprise service buses (ESBs), and development systems no longer provide a sufficiently safe and effective operating environment. The distributed and heterogeneous nature of Web services means that existing infrastructures and traditional tools are not sufficient to address the security and operational needs of these new application systems.

## ***Cost Reductions and Revenue Opportunities***

The value of Web services includes the reuse of business logic, agility through loose coupling, and independence across platforms, applications, transports, and vendors. These technology benefits provide measurable business benefits such as reduced development time, rapid deployment of new business applications, and less costly maintenance and change processes. However without an XML infrastructure that provides service virtualization, syntactic and semantic transformation, and deployment support, these benefits are dramatically constrained.

## **XML Infrastructure Capabilities**

The complex, system nature of the applications being developed from distributed services requires a radical shift in the way that application designers think about what they are developing and how it will operate. The fundamental capabilities of XML infrastructure are operations that are processing intensive, repeated across services, high risk, or simply too complex for the network or business logic.

### **Safe and Secure: XML Infrastructure for Web Services**

- XML infrastructure provides an efficient and controlled mechanism to help ensure data confidentiality and integrity.
- XML infrastructure delivers versatility and robustness for service access control and provides insight into access activity and patterns.
- XML infrastructure provides a dynamic and evolving threat defense against XML message-based attacks with tuning and graduated alerting.

XML infrastructure services include the following:

- Impedance matching: mediation, transformation, and more
- Service virtualization and versioning
- Performance acceleration and optimization
- Security
  - Identity
  - Access control
  - Threat mitigation
  - Data confidentiality and compliance

- Control
- Monitoring
- Integration

## ***Impedance Matching***

In XML infrastructure, impedance matching is the process of harmonizing the differences between implementations across platforms or applications. The goal is to maximize the flow of information through application systems by eliminating bottlenecks and suboptimal connections. Practically, impedance matching deals with interoperability problems caused by different interpretations of specifications or the selection of different functional subsets within a standard that are allowed for an implementation.

### **What Is Impedance?**

In electrical engineering terms, impedance is the measure of the total opposition to current flow in an alternating current circuit, made up of two components: ohmic resistance and reactance. Resistance is the opposition to current flow in a circuit. Reactance is the opposition to the flow of alternating current caused by the inductance and capacitance in a circuit.

Impedance matching may mean the following:

- Matching different transports used by a Web service consumer on one side and any selection made by a Web service provider on the other, including the following:
  - HTTP
  - HTTPS
  - Simple Mail Transfer Protocol (SMTP)
  - IBM Websphere MQ
  - TCP
  - Java Messaging Service (JMS)
  - FTP
- Selection among the many standard and domain-specific methods of authentication such as the following:
  - Username or password
  - X.509 certificates
  - Security Assertion Markup Language (SAML) assertions
  - Kerberos tickets
  - eXtensible rights Markup Language (XrML) licenses [[OKAY? OR DO YOU MEAN Extensible Rule Markup Language (XRML)?]]
  - Role identifiers
  - Identity attributes
- The locations and format of authentication credentials (in addition to the standard locations defined in Web Services Security (WSS)).
- Transformation of message syntax or data semantics to accommodate versioning of service providers and migration of service consumers

- Anything that requires mixing and matching of technologies or implementations

An XML infrastructure must deal with impedance matching through governable configurations across semantics, transports, trust boundaries, standards evolution and adoption, and credentials.

## ***Service Virtualization and Versioning***

One of the most lauded propositions of SOA is loose coupling. Its value is usually described as the capability of a service interface definition to be independent of a changing service implementation. However, the other side of loose coupling is seldom discussed. In an operational system, how is the versioning of the services defined by the service interface handled? How can multiple versions of a service interface be tied to a manageable set of service implementations? How are services retired or deprecated in a controlled way?

Application systems need an effective mechanism to coordinate the operation, versioning, introduction, and retirement of services. These can be built into the application itself, but this approach significantly reduces the value of the loose coupling and reusability characteristics of Web services and SOAs.

Service virtualization allows a deployed service interface to be described to its consumers independent of the service interface that the service provider implements. This approach supports the movement, replication, deployment, versioning, and retirement of services in a controlled way. Service virtualization may be supported by the use of a registry, but additional capabilities such as transformation and publication control are needed to make virtualization effective.

XML infrastructure addresses the lifecycle and coordination needs of services in a policy-controlled and verifiable way.

## ***Performance***

Performance is one of the most significant considerations when working with XML, as the overhead associated with processing XML structures places considerable load on the application servers. Performance includes the point-to-point acceleration of XML processing or compression and the optimization of the application system as a whole.

## ***Acceleration***

Acceleration speeds up the point-to-point interactions between a Web service and its consumer. It generally includes the fast processing that gateways can apply to messages to perform functions such as validation and message transformation. Reduction in bandwidth consumption by XML traffic can be achieved by the use of compression techniques on XML messages or attachments. In the future, reduction in application processing overhead may result through the use of binary encoding techniques on XML messages or the provision of preparsed structures.

One of the proven mechanisms for optimizing application systems is the caching of data, such as security credentials, that are reused frequently or are elements in common operations. Integration with other infrastructures, such as Identity and access management systems, significantly reduces the amount of traffic exchanged.

## ***Optimization***

Optimization of application systems is essential to reduce the overhead that results from building generalized, reusable services. The size of the XML messages may cause bandwidth concerns,

particularly with low-speed connections. The term *acceleration* is often used to describe the techniques used to address this overhead. These techniques may include high-performance parsing and validation of XML messages against schemas, high-performance semantic transformation using Extensible Style Language Transformation (XSLT), provision of preprepared XML or tagged structures to an application, compression of XML messages or attachments, and conversion to binary formats.

Optimization uses a set of techniques to improve the operation of the application system as a whole, including reducing network traffic, through integration with other infrastructures such as identity and access management systems, caching of information, and application of heuristics to speed up common operations.

## **Security**

From the very first service to a robust SOA, helping ensure security while maintaining utility and availability is crucial. The growing necessity for auditable confidential data to comply with the Sarbanes-Oxley Act, Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA), and other regulations creates tremendous pressure for the system to protect private and confidential data. The performance implications of integration with existing infrastructure, as well as the array of identity credentials that may be presented to a service, are significant and require thoughtful planning.

XML infrastructure enables the enterprise to comply with corporate and governmental security and confidentiality requirements while minimizing the performance cost, errors, and interoperability hurdles introduced by encryption technologies. By enabling scalable integration and the use of existing infrastructure, XML infrastructure enables the optimization of credential authentication and mapping.

## **Identity**

Application systems composed of Web services create additional requirements for the use of identities that can be correlated and verified to support access control, auditing, and trust. The identities of principals initiating a transaction, the Web service components involved in processing a message, and the roles or other attribute-based authorization and entitlement systems all become important in various Web service implementations. Additional replay attacks and opportunities for insertion of invalid transactions or consumption of information by unauthorized users all become more significant concerns in an application system.

Identity information is often expensive to use and requires careful application of policy to help ensure consistent use of resources. An XML infrastructure must integrate effectively and efficiently with the existing identity and access management infrastructures of an organization. Under policy control, new methods of authentication and access control and new types of authentication credentials must be transparently integrated as additional customers and partners with different requirements are integrated into the application system.

## **Access Control**

At a minimum, every service needs to help ensure that only appropriate messages are accepted and processed. Depending on the service, access-control decisions can be very complex policies to enforce. For example, a particular service may be accessible only to messages sent from a particular group of machines and domains. The target service may also expect all identity credentials to be delivered to it as SAML credentials even though its consumers may or may not

support SAML. Finally, policies and decisions about access permission are often controlled by identity management systems or directories, requiring the service to integrate with that infrastructure and articulate the decisions that the service is expecting.

These growing degrees of complexity require the simplification that is delivered by XML infrastructure. The XML infrastructure does the following:

- Integrates and uses all existing identity management and directory infrastructure
- Enforces access-control policies articulated in existing infrastructure or within the XML infrastructure
- Transforms and maps appropriate credentials to help ensure proper access and extensive interoperability

## Threat Mitigation

Because of the nature of XML Web services, XML traffic can circumvent existing threat defense mechanisms such as antivirus and intrusion-detection systems and firewalls. Because the vast majority of XML traffic either transits port 80 or port 443 or transits using a bus, Web services are open to a number of threats that must be addressed consistently across the enterprise to help ensure application and system availability and data confidentiality. Attacks such as malicious content and denial of service (DoS) cannot be addressed by standards and require deep understanding of XML message content, context, origin, and destination. Existing infrastructure cannot deliver this level of understanding, yet can measurably affect the performance of existing infrastructures and their capability to meet their purpose-built requirements.

The XML infrastructure must deliver a flexible mechanism to detect and defend against XML-based attacks. These mechanisms should include the following:

- Configurable thresholds for alerts that integrate with existing network security management systems
- Graduated and automated response for alerting and blocking
- XML message throttling to reduce load on applications until the threat has subsided or been eliminated
- Secure update mechanism to protect the enterprise against emerging threats
- Operational security to minimize the possibility of administrative error or internal malicious attacks

## Data Confidentiality and Compliance

XML messages are self-describing and human readable. These attributes enable the very broad interoperability that makes XML attractive for extensive system-to-system integration for new applications. These attributes also provide opportunities for confidentiality and integrity violations that can undermine the credibility of those applications.

The mechanisms to help ensure the confidentiality and integrity of XML messages are well-understood and supported for both XML and Web services (through the WS-Security specification). For example, confidentiality can be provided through the use of XML encryption. This mechanism ensures confidentiality of the message regardless of transport. But this confidentiality may come at a high performance cost and increase the latency of the system beyond acceptable levels. However, integrity can be provided through XML digital signatures. In this case, an entire message or key elements can be signed to help ensure that the content is tamper-proof. Again, these

operations come with a high performance cost, and both XML encryption and XML digital signatures are complex standards with numerous implementation options, creating opportunities for errors and lack of interoperability.

XML infrastructure offloads and optimizes these expensive operations while delivering a robust record of the confidentiality and integrity of the XML messages. That record can be used to prove compliance and to repudiate a transaction's processing. By enabling service and message content-specific policy, XML infrastructure can enforce granular confidentiality and integrity policies at the same level of detail as the destination service.

XML infrastructure also helps ensure that XML Web services comply with Sarbanes-Oxley Act Section 404, HIPAA, GLBA, California Senate Bill 1386 (SB1386), the European Union (EU) Privacy Directive, and corporate compliance and privacy standards by facilitating tracking of critical Web services traffic and identification of abuses. It enables inspection of and reporting on data such as the following:

- User credentials and identities
- Company credentials and identities
- Services and applications accessed
- Requesting parties, servers, and applications

## Monitoring and Control

Understanding what is happening in the new application system is vital. An XML infrastructure must effectively provide both real-time and historical views into the dynamic operation, message flows, and transformations occurring within an application system. In addition, the late binding and reusability aspects of Web services require that the message flows between Web services be monitored, logged, and audited to understand how the application system is functioning over time.

After deployment, a collection of loosely coupled Web services can soon appear unwieldy, making construction of a usable application system difficult. An enterprise must be able to apply policy consistently across all platforms, development environments, third-party applications, trust domains, and political and geographical domains that contribute Web service components to the application system.

XML infrastructure performs valuable processing, from security to interoperability to routing for XML Web services SOAs. Consequently, XML infrastructure captures considerable data about XML message traffic, such as the number and size of messages from a specific consumer or to a specific service, the relationship of identities to messages and services, and the number of failed schemas and access attempts from a consumer or specific service. All this data has control thresholds that trigger responses, from throttling traffic, to alerting administrators, to logging specific messages and events. As a result, XML infrastructure dynamically controls the XML message flow to optimize performance and minimize exceptions.

## Integration

### When Is an XML Infrastructure Needed?

- An XML infrastructure enables efficient service development and central and reliable service operation. Planning and implementing a scalable XML

infrastructure at the start of SOA or XML Web services initiatives offers a number of immediate and essential benefits:

- Reduced complexity and increased simplification
- Reuse of common capabilities
- Coordination of components
- Performance and optimization
- Access control and secure messages

Enterprise customers seeking the operational, cost savings, and performance benefits of Web services should consider deploying an XML infrastructure.

Traditional application development environments hide many elements related to integration of supporting infrastructures, such as attribute repositories and identity and access management. Web service implementations, in contrast, potentially expose business logic programmers to the many other supporting infrastructures deployed within an organization, such as the following:

- Authentication systems
- Federated identity systems
- Information repositories
- Authorization systems
- Entitlements systems
- Network, application, and Web services management systems
- Aggregated logging systems

In addition, XML infrastructure enables the use of a wide array of transports and message handling, ranging from HTTP to MQSeries to ESB. The asynchronous message transports such as MQSeries, JMS, and buses provide reliable messaging, publish and subscribe messages, and event message that are not yet widely standardized for support within the network. The XML infrastructure must enable the enterprise to use these transports as easily as Internet transports.

## Infrastructure Implementations

A variety of XML infrastructure implementation options are available, and several are described here.

### Toolkits

Software development toolkits are the most common implementation of any emerging technology. They provide tight integration of XML processing capabilities within the business logic that is being developed for the application. However, toolkits place a significant burden on developers to understand the technology they are using and to integrate it with other systems and infrastructures. For technologies as complex as Web services, this approach is not scalable from a resource perspective. In addition, the high overhead of implementation, testing, and maintenance make toolkits a costly option.

### Platforms and ESBs

Many application platforms<sup>1</sup> and ESBs provide implementation support for Web service developers,

allowing them to write code that can address many of the underlying XML, Web services, and SOA concerns. A platform-specific implementation can be a useful and valuable early step for those starting with very simple Web services. The problems raised by such implementations are fairly extensive, however.

Web services by their nature will be implemented across many different platforms. Each of these platforms will have implemented XML standards slightly differently, have a different version of a particular specification in use<sup>2</sup>, or have selected a different subset of the available options<sup>3</sup> within the standards. Impedance matching to align the many differences in platform capabilities is important to any operational deployment.

The more significant challenge is that platform-based or endpoint-dependent implementations of XML infrastructure are particularly susceptible to problems because platforms see only the messages sent and received directly to them. The only way that platforms can effectively gain an understanding of the application system built using Web services is to engage in a vast amount of information distribution. To effectively control the application system, a significant investment in agreed-upon standards would have to take place to support a highly distributed and heterogeneous solution. These standards are nowhere in sight today, and the necessary agreements necessary between vendors and the interoperable implementations are even further away.

ESBs sometimes form an application platform within an organization. Many ESBs now support Web service interfaces that allow them to expose entry points to a Web service-based application system. In many cases, however, this approach hides many of the business services with which a composite application would want to integrate. Over time, ESBs may become open Web service-based technologies, but they currently represent an older system that needs to be integrated with a Web service-based application system.

As with toolkits, platform implementations of XML and Web services support are valuable and will help in the development of many Web services. However, application systems built with SOA and Web services cannot be effectively monitored and controlled or understood by taking a platform view.

## Agents

In this discussion, agents are software components that are implemented on each of the application platforms that support a Web services component. As a whole, if implemented correctly, they form a network of distributed management components.

Agent-based architectures may seem to provide an attractive and intellectually pleasing approach to XML infrastructure, offering a distributed management approach to Web services, themselves a distributed framework. However, to stay synchronized and to share enough useful information about the operational state, command, control, and monitoring systems create a significant amount of administrative traffic.

The other problems with agents are traditional concerns: Can you get sufficient coverage for the multitude of application platforms? How do you manage the versioning of the many interdependent agents and applications running on a platform? Are new versions of agents available in a timely manner to integrate with the required changes in operating systems, applications, and operating environments? How are new versions quality assured, distributed, and updated in a complex environment? If an agent from a different vendor is implemented by some other part of the organization, how will the systems interoperate?

For practical purposes, the use of agents has not proven to be an effective solution.

## Gateways

The standardized and self-describing nature of XML interfaces and messages can handle most infrastructure requirements from a networked perspective. The Simple Object Access Protocol (SOAP) was designed with a structure that supports processing by intermediaries, independent of the payloads and application message structures. XML gateways offload processing from application platforms, allowing them to dedicate their resources to running the business logic implemented for a SOA by Web services, rather than performing low-level XML processing.

Gateways may represent the only effective mechanism for monitoring and understanding the operation of a Web services application system. They allow caching and optimization of the application system. They also provide an effective mechanism for logging, aggregating, and auditing message flows and events through the system. XML gateways allow platform-independent, consistent policy definition and enforcement. With the addition of transformation and mediation capabilities for transports, XML syntax and semantic mapping provide active support to achieve the loose coupling values of SOAs.

## Who Benefits from an XML Infrastructure?

XML infrastructure provides value and opportunity to many functions throughout an organization. Among those who benefit the most are system architects, designers, programmers, operations staff, auditors, and security staff.

## System Architects

System architects need to define where the functions required of a technical solution will reside. Any XML or SOA application system includes a collection of common functions that need to be performed. This definition is analogous to the definition of a network infrastructure. The system architect knows that naming services or routing will be handled by the network and considers how to use those capabilities rather than creating them.

When the system architect develops the application system, the XML infrastructure defines standard functions and capabilities that the architect can assume, allowing the architect to turn to the other aspects of the application system that need to be defined. Capabilities such as message transformation, service virtualization and publication, identity integration and validation, monitoring, and failover are supplied by the XML infrastructure for the system architect to use rather than define.

## Designers

Designers need to know what interface definitions they have available to them and what the operational characteristics are. If each service has slightly different characteristics based on the platform, vendor, version, or implementation of standards, designing a reliable system becomes difficult.

XML infrastructure provides the impedance matching capabilities that allow designers to focus on the Web service interface definitions for their applications; the underlying technology concerns are taken care of for them.

## Programmers

Programmers writing business applications should be primarily concerned with implementing applications using their specialized domain knowledge to quickly and effectively enable the business to meet its goals. Programmers should not need to be concerned about integration of Web services with identity and access management systems, or with how to handle DoS attacks, or with working around the different and often incompatible versions of underlying technical standards.

Application programmers should be able to create code that runs in a safe environment. In the world of highly distributed, heterogeneous Web service implementations, it is the XML infrastructure that creates this safe and feature-rich development environment.

## Operations Staff

Web service deployments by their nature and value proposition consolidate and integrate business logic across many disparate platforms and environments. Operations staff members need tools that allow them to understand the operating status of the application system, perform capacity planning, achieve effective failover and load balancing, bring new instances of services or new consumers of existing services online, and maintain predictable and reliable operation of the system.

XML infrastructure provides these tools independent of where the services are implemented.

## Auditors

Increasingly, enterprises must demonstrate compliance with regulatory or corporate governance requirements. The use of highly distributed application systems based on Web services makes auditing extremely difficult. Examining the static business logic is not sufficient for auditing in a loosely coupled application. The only way to understand these new application systems is to track and examine the message flows and transformations that take place during the operation of the application.

XML infrastructure supports aggregation of logging information, monitoring capabilities, and compliance validation mechanisms to help auditors and quality assurance (QA) staff quickly identify compliance with policies.

## Security Staff

The new security challenges raised by Web services have been discussed for several years now. The revenue generation and operational value of Web services often results in business pressures to deploy an application before the security challenges of the deployment have been met. The high overhead in development time and application platform resource consumption becomes an impediment to rapid deployment.

The XML infrastructure supports secure deployments by directly taking responsibility for the whole range of security and policy requirements, from threat mitigation and validation, to privacy and integrity, to identity and access control processing. Security policies are defined and enforced independent of the capabilities of the application platforms, and monitoring mechanisms support detection of security problems in real time.

## Cisco XML Infrastructure Solutions

Cisco® ([www.cisco.com](http://www.cisco.com)) provides infrastructure solutions that enable enterprises and government

agencies to transparently deploy and expand secure XML-based Web services and SOA. The Cisco ACE XML Gateway provides XML security, integration, management, and acceleration, enabling rapid SOA adoption. The versatility of the Cisco solution is demonstrated by many successful enterprise and government implementations.

## Impedance Matching

Flexible Cisco pipeline technologies decouple message transport, semantics, security, and credentials so that requests and responses can be generated on available systems and processed by the Cisco ACE XML Gateway to match the expectations of the service on the other side of the request response. The Cisco architecture makes it easy to add new transport support, integrate adapters to older non-XML systems, and transform data semantics using a variety of methods, including Extensible Stylesheet Language (XSL) and XML Path Language (XPath). Cisco delivers over 250 non-XML system integrations spanning both semantic and protocol transformations. Extending mature systems delivers the best value when deploying XML and SOA.

## Integration: Reuse and Service Coordination

The Cisco ACE XML Gateway supports multiple instances and versions of a service, enabling real-time service load balancing to maximize quality of service as well as controlled version migration to account for the systemwide testing required when a service consumer migrates from one version of a service to another. In addition, the Cisco ACE XML Gateway enforces routing policies that are context, content, and table dependent. As a network infrastructure providing service-level routing, Cisco provides insight into route performance and service performance for further optimization.

## Security

Cisco ACE XML Gateway provides a comprehensive and robust array of XML security services to help ensure that the network can provide security to the XML services and XML messages transiting it.

## Access Control and Identity Compliance Solutions

Cisco provides multilevel access control so that the network can authenticate the system, service, and user-sender of an XML message and pass the necessary authentication and authorization information in a consumable format to the destination services. Cisco natively uses existing identity management infrastructure so that entitlement policies are reused and control of entitlements is centralized. The Cisco Software Development Kit (SDK) enables native integration with custom identity management systems as well.

Cisco identity compliance solutions enable the gateway to capture metadata, log policy changes, generate test messages to prove Sarbanes-Oxley compliance, and report on Sarbanes-Oxley 404 information and events in XML Web services. Cisco identity compliance technology provides advanced authentication and identity enforcement functions to map and record different representations of identity for secure integration and to generate reports on identity accesses of services and correlate access between multiple identities representing the same principals. In addition, Cisco identity compliance technology includes preconfigured content screening for privacy-sensitive consumer, medical, and financial information. Allowing a variety of actions upon discovery of private information, ranging from masking and obfuscation to programmatic actions, Cisco enables XML Web services to comply with HIPAA, GLBA, SB1386, the EU Privacy Directive,

and corporate standards.

Through central role processing of XML traffic and flexible pipeline technology, Cisco delivers value-added identity processing and reporting. Enterprises and government agencies can determine how identities are used by services and provide irrefutable evidence of policy enforcement to demonstrate privacy, Sarbanes-Oxley, and other regulatory compliance. Cisco identity insight technology provides the reports and data to manage the use of identities and identity federation.

## **XML Threat Defense**

Cisco delivers a complete transactional and operational threat defense and an exceptionally secure architecture to defend against emergent zero-day threats. For detailed descriptions of these threats and appropriate tests to help ensure that services and infrastructure are secure, see the Cisco white paper “Security in the XML infrastructure.” [\[\[PROVIDE LINK?\]\]](#)

Transactional threats include identity exploits, transport exploits, DoS, and content-based exploits. Operating exploits include malicious or inadvertent access, leakage of sensitive information, and policy modification. The Cisco ACE XML Gateway includes the Cisco ACE XML Manager, Rivest, Shamir, and Adelman (RSA) SecurID, Federal Information Processing Standards (FIPS) 140-2 Level 3 signed events, message and administrative logs, and role-based administration that delivers policy and reporting isolation.

## **Data Confidentiality and Integrity**

The Cisco ACE XML Gateway offers policy-declared coarse and granular encryption and digital signatures to provide confidentiality and help ensure the integrity of messages. Extensive Cisco performance reporting and logging capabilities enable administrators and security professionals to assess the performance effects of these security measures to balance risk with responsiveness.

## **Acceleration and Optimization**

The Cisco solution provides a central router for XML messages in a simple request and response and in a composite service-based application system. Cisco solutions are positioned to minimize network traffic and maximize XML message processing. Flexible Cisco pipeline technology continuously monitors transaction rates, message sizes, and operations being performed. This architecture uses application-specific integrated circuit (ASIC) based acceleration, caching, memory utilization, and spooling to achieve maximum performance under all conditions. In addition to accelerating the specific operations performed on the gateway, this architecture provides front-end and back-end connections that can be optimized so that the requesting application always receives the fastest response possible.

The experienced Cisco technical team provides patented, optimized software functions that dramatically increase performance of critical operations such as authentication, signature validation, and transformation. Schema pipelining reduces the overhead from schema validation by more than 90 percent, enabling enterprises to proactively use schemas for both interoperability and security purposes. In addition, Cisco credential caching technologies help ensure that a credential is validated once and used repeatedly for a configured period of time, and that all services and messages accessed with the credential are securely recorded for additional alerts and analyses.

The flexible Cisco pipeline technology easily handles extremely large messages and attachments at wire speed, allowing the same technology infrastructure to be used across a variety of XML-

centric deployments.

## Control and Monitoring

The Cisco ACE XML Gateway enables real-time control and insight with secure historical monitoring and debugging. Cisco multilevel logging captures real-time event information, message metadata, and complete messages (if required) and correlates that information with policies and configurable thresholds. Graduated responses provide the combined benefit of automated response and containment with human-determined actions and modifications. In addition to the run-time monitoring and control provided by the Cisco ACE XML Gateway, the Cisco solution provides application developers with a logging infrastructure that is invaluable for debugging the distributed system being created because Cisco pinpoints the source of XML message errors and exceptions; this information can be portioned and securely shared for collaborative problem solving and rapid resolution.

## Rapid, Manageable Deployment

The Cisco solution can be deployed quickly in a modular, noninvasive manner; it is normally up and running within an hour and in production within a month. The Cisco enterprise, service-specific policy articulation engine allows fine-grained control over XML message handling. Cisco reports identify all remediation needed to deliver effective performance and policy compliance for every message.

## For More Information

To learn more about the Cisco ACE XML Gateway and determine how they can accelerate, secure, and simplify your XML Web services and SOA, visit <http://www.cisco.com/go/ace>.

- 1 In this context, application platforms includes operating systems, application development environments, application integration environments, and third-party application systems.
- 2 This problem of variation occurs in many instances of well-defined and ratified standards. For example, currently three versions of SAML (1.0, 1.1, and 2.0) are in deployment; however, the specifications are not backward compatible.
- 3 For example, any of five authentication credential types can be implemented in conjunction with the Web services security standard.



**Americas Headquarters**  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
[www.cisco.com](http://www.cisco.com)  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

**Asia Pacific Headquarters**  
Cisco Systems, Inc.  
168 Robinson Road  
#28-01 Capital Tower  
Singapore 068912  
[www.cisco.com](http://www.cisco.com)  
Tel: +65 6317 7777  
Fax: +65 6317 7799

**Europe Headquarters**  
Cisco Systems International BV  
Haarlerbergpark  
Haarlerbergweg 13-19  
1101 CH Amsterdam  
The Netherlands  
[www-europe.cisco.com](http://www-europe.cisco.com)  
Tel: +31 0 800 020 0791  
Fax: +31 0 20 357 1100

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

©2007 Cisco Systems, Inc. All rights reserved. CCVP, the Cisco logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, GigaStack, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, Packet, PIX, ProConnect, RateMUX, ScriptShare, SlideCast, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0701R)