XML Networking:
Moving Up The Stack

The Impact of XML On Networking And
The Emergence of Application Data Routers

The HTRC Group
P.O. Box 2087
San Andreas, CA 95249
www.htrcgroup.com

Author:
Greg Howard
Principal Analyst & Founder
greg@htrcgroup.com

About The HTRC Group, LLC
The High-Tech Resource Consulting Group focuses on advanced IP services, providing consulting, custom market research, and market research studies, white papers, and consulting to enterprises, service providers, and product manufacturers.

Table of Contents

Executive Summary

The need to cost-effectively integrate existing and deploy new applications has created a need for Application Data Routers (ADRs).

The abstraction of XML processing, management, and policy enforcement from Web services platforms to ADRs provides a cost-effective means to access, move and track Web services throughout growing enterprises.

The purpose of this paper is to examine the impact of XML on network infrastructure and the emergence of Application Data Router (also referred to as Web Services Management) solutions.

Introduction

The growth of information and applications is staggering. Enterprises across all industries are rapidly deploying Web-based applications as a means to more effectively communicate and distribute information to their customers, partners, and employees. Today's applications are extremely varied. Application types often range from CRM, to ERP, to Enterprise Information Portals, to B2B Exchanges, to Partner/Supplier Extranets, to E-Commerce sites, and to numerous custom applications. Web services are being deployed throughout enterprise networks at a healthy pace.

Unfortunately, the vast majority of today's Web-based applications were designed, built, and deployed at break-neck speed. Fierce competition, coupled with the Internet economy's incredible rate of change, forced companies to make very quick technology decisions. As enterprises focused on simply getting the application up and running, ad-hoc processes for moving data in and out of these new systems were developed quickly.

Looking back, there is no wonder why many companies, both dot.coms and brick and mortar, have engaged in aggressive build-outs of stove-pipe applications, each with a separate function. However, this pressure to remain competitive has engulfed many companies and consumed those that made fatal technology decisions. Homegrown processes and tools are now proving to be extremely rigid and brittle.

Nevertheless, the first wave of innovation gave many a glimpse of how powerful the IP connected world could be—a vision of a future connected world and the opportunity it presents.

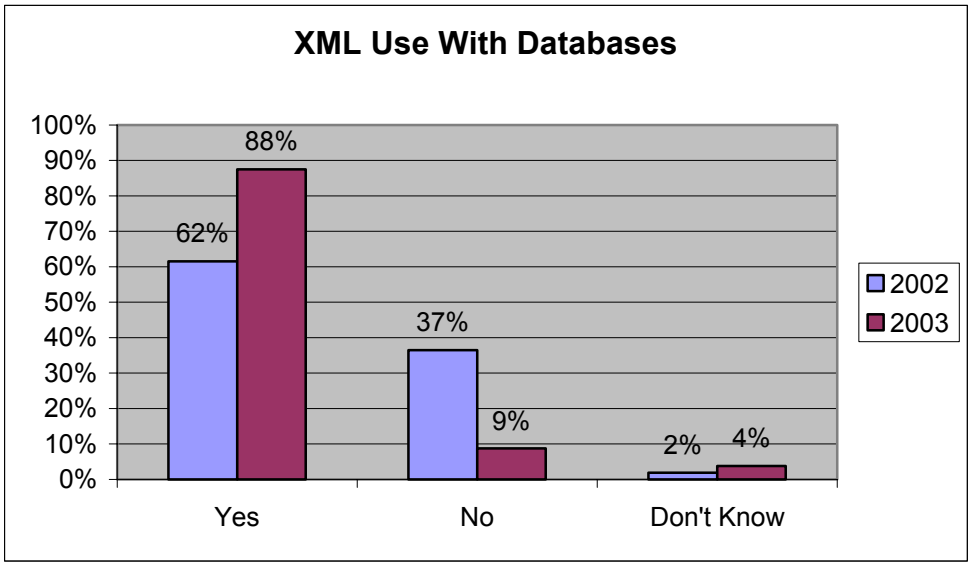
The Growth of XML

Today, XML is rapidly becoming the *lingua franca* for business data.

Web services, based on open-standards like WSDL and UDDI for service definition and discovery, are developing as the dominant trend in enterprise computing in this decade.

According to The World Wide Web Consortium (W3C), "The Extensible Markup Language (XML) is the universal format for structured documents and data on the Web." According to The HTRC Group's "The 2002 Application Data Router Study: The Evolution of Content Networking," the use of XML with database driven applications is increasing significantly over the next year. Respondents in the study were asked if they use XML with database driven applications in 2002 and 2003. Enterprise XML use with database driven applications grows from 62% in 2002 to 88% in 2003. XML is rapidly becoming the protocol of choice for enterprise application integration. XML enables the flexible transmission and interpretation of data between applications. As enterprises increase their exchange of data/information to create new Web-based applications and services, XML offers a new level of data interoperability.

Chart 1: XML Use With Databases (n=104) Q6, 7 ADR Study



XML and Web services are focused on enabling the next wave of Internet computing often referred to as "the semantic Web" where applications are able to easily share information and work together. These technologies, however, are not solutions in and of themselves.

The Real-time Enterprise Goal

The real-time enterprise is a business strategy of having real-time (to the minute) information by automating the processes among business systems used by customers, suppliers, partners, and employees. The result is a more competitive organization that can make decisions with the most up to date information. However, making the real-time enterprise a reality is challenging. Only 9% of respondents from our ADR Study said they have no problems making real-time data available, confirming our belief that enabling the real-time enterprise is a daunting task.

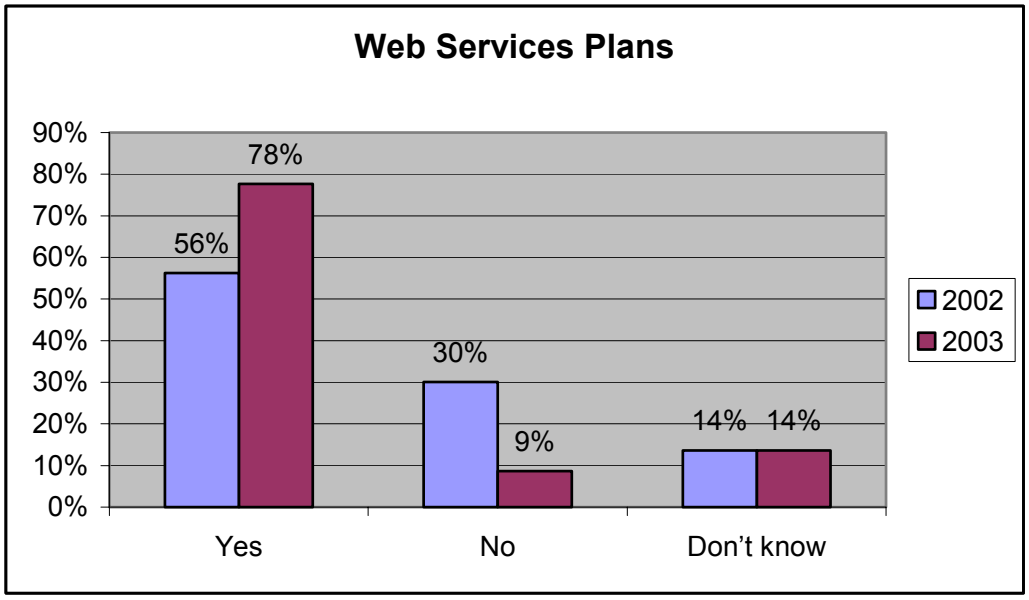
The concept of the real-time enterprise continues to gain momentum, fueled by advancements in technology such as ADRs. To this end, many organizations are looking to automate business processes, which require application integration or interconnectivity. ADRs will play a critical role in automating business processes. The case for creating the real-time enterprise is compelling where potentially huge returns on investments (ROIs) are associated with the automation of business processes.

Web Services

Web services represent a significant class of technologies that have emerged to provide greater means to integrate disparate applications and services. Web services provide a common framework for finding, invoking, and integrating distributed components/services over the Internet. What makes Web services particularly unique is that their framework utilizes the Internet's current open data standards and protocols, such as HTTP and XML, unlike earlier component models such as DCOM and CORBA that relied on proprietary object models and protocols. With Web services, components that are implemented in a multitude of different languages and platforms, and that reside in different locations in the network, can now be loosely coupled to form distributed applications. To this end, Web services provide a standards-based mechanism for applications to programmatically request information from other applications over the Internet. Every major platform and application vendor has embraced the core Web services standards, pressing the need for the emergence of a new class of solutions called ADRs.

There are very real plans for Web services deployments, according to HTRC Group research, respondents were read a definition of Web services and asked if they deployed or plan to deploy Web services in 2002 and 2003. Respondent plans for Web services are surprisingly high, increasing from 56% in 2002 to 78% in 2003.

Chart 2: Plans for Web Services (n=103) Q12-13 ADR Study



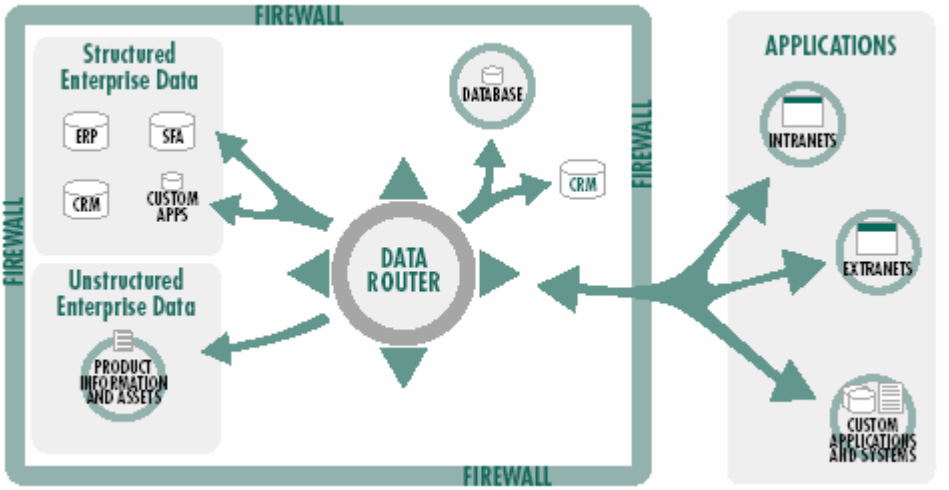
Application Data Routers

What is an Application Data Router?

Application Data Routers (ADRs) have emerged to perform XML processing functions, policy management, additional security, and raw performance for scaling Web services deployments. Application Data Routers (Web Services Management) represents a new class of solutions that leverage XML and Web Services to help automate the process of sharing data/information between systems. While XML and Web Services provide the standards and protocols necessary to more easily exchange data, ADRs will ensure that the flow of data between systems is handled in a reliable, secure, timely, and scalable fashion.

ADRs are systems that automate delivery of data from multiple sources to multiple destinations. They facilitate the movement of multiple data types whether they reside in databases, file systems, or enterprise applications. ADRs allow companies to index relevant data, facilitate data transfer over open standards protocols, and programmatically access the router via Web Services.

Figure 1: What is an ADR



Another core function of ADRs is the ability to configure and manage the movement of SOAP messages. In addition to enabling the specification of data sources and targets, ADRs also allow users to build complex routes that can include multiple transforms, sources, and destinations. ADRs can be triggered by specific events, calendar-based, or one-off requests for data. ADRs provide additional controls for encryption, authentication, caching, and other options. Moreover, data routers provide queues, which provide for policy-based traffic prioritization and active queue management. This load balancing function is needed in order to scale in- production Web services environments.

ADR services are beginning to emerge from forward thinking providers such as British Telecom (BT). Organizations have the option of implementing ADR solutions themselves or outsourcing ADR functions to services providers (similar to other managed services such as firewalls and VPNs) to connect partners, customers, and suppliers to Web services.

The Need For Application Data Routers

The wide adoption of Web services standards by platform and application vendors is driving multi-vendor platform deployments that create an environment of distributed Web services enabled applications with many federated management points. ADRs provide critical functions necessary for real world Web services deployments, including the following:

- Management over federated Web services control points
- A needed layer of security that enforces access control policies
- A wide variety of means by which to request and receive data

- The creation of a standardized set of solutions that use Web services technology to share data between systems
- Standardized control point for data flows from internal and external Web services

Because of their strategic location and function, ADRs provide a level of management and access control policies in addition to existing security policies. Privileges for users and applications are checked with the ADR to ensure only authorized usage. Additionally, data can be routed over open standards based security protocols. Secure Application Data Routing enables multiple Intranet, Extranet, and Internet applications to securely use multiple services throughout the public and private enterprise network.

ADRs provide a wide variety of means by which to request and receive data. Requests for data that are issued via protocols such as SOAP, XML-RPC, and JMS, are interpreted via standards-based interfaces. Because ADRs have Web Services interfaces, requesting applications can communicate with the router over HTTP (port 80) and avoid any conflicts with firewalls. Moreover, data routers' scriptable interfaces provide for extensive flexibility and automation.

Applications will increasingly require data flows from internal and external Web services. The need to automate the process of moving data between multiple sources and destinations will increase with time. As such, the ability to support a growing number of protocols and standards that are fundamental to this process, is very analogous to the industry evolution of packet-level routing/networking.

The increasing need for interoperable applications coupled with the growing adoption of open standards has created a need for ADRs. ADRs provide the opportunity to create a standardized set of solutions that use Web services technology to reliably and securely share data between systems. Application layer data routing is fundamental to supporting business processes within an organization - whether it is moving unstructured document files between file systems in order to support interdepartmental processes, populating new Web databases to support a new portal initiative or providing new Web services to offer request-based transaction log access to trading partners.

In fact, the overall vision of collaborative computing/commerce is based on the premise that applications should be able to share information freely across all boundaries whether organizational, geographical, or technical. Over time, as corporations continue to deploy Web-based applications that rely on a composite of data/information aggregated from both internal and external systems, the complexity and costs associated with the sharing of data between these coordinating applications is destined to increase exponentially without standardization.

Why ADR versus Traditional Middleware

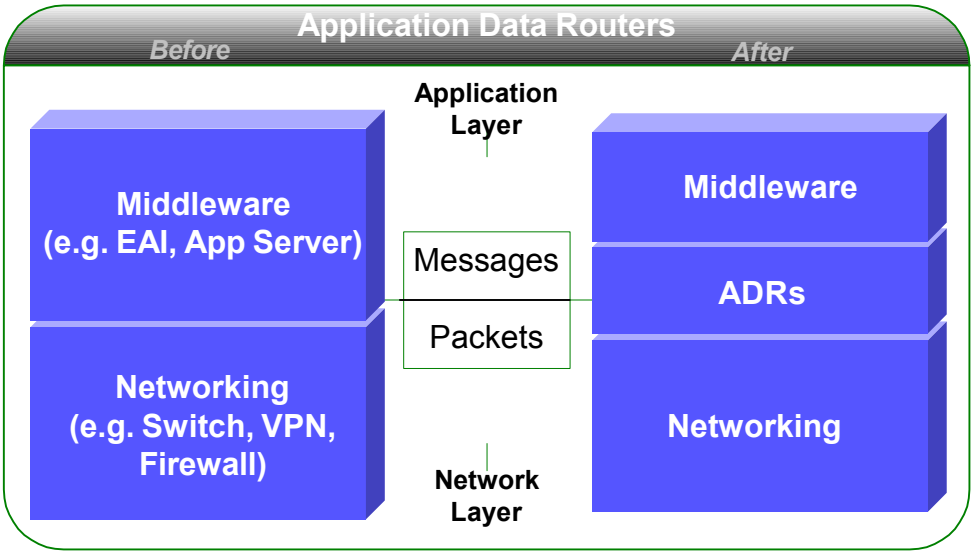
ADRs

The processing of XML for Web services can be done within the framework of the current Web services platforms; however there are several critical factors that should be taken into consideration when developing a Web services strategy. Production deployments of Web services will likely have many federated control points requiring a need for centralized management in a heterogeneous environment. The key enabler to Web services deployments is standardization; however, enforcing policies centrally, such as SOAP and security, over an eclectic mix of Web services application and platform vendors is required in order to realistically manage a production Web services environment.

High volume Web services will require load balancing across multiple Web services servers. When demand increases because of volume or length of XML messages, application servers will bear the brunt of processing requirements. Web services architects are faced with a decision to add more application servers or add an ADR in order to accommodate and balance the increased loads. ADRs are specifically designed to handle heavy XML loads, centralized management and policy enforcement, making it the ideal platform to offload the heavy lifting.

The deployment and adoption of Web services is forcing badly needed standardized protocols to the application layer. Standardization of protocols played a significant role in the development network layer products—from the first versions of software based routers to the purpose built router hardware we know today. Similar to network layer hardware, ADRs reside above the network layer and provide a centralized point for management and policy enforcement for Web services. ADR software can be installed on an existing server or purchased in an appliance model. The figure below depicts where ADRs function in XML networking.

Figure 2: How ADRs Fit



Large-scale deployments of Web services have many distributed points to be managed. Each Web services connection represents an important enterprise application that may interface with employees, customers, suppliers, and business partners. Current Web services platforms do not have a mechanism for the management of federated control points across disparate Web services platforms and applications. ADRs provide a centralized way to manage organic growth of Web services control points, which would otherwise require manually managing each Web services connection.

Pioneers such as National Student Clearinghouse (NSC), have a Web services environment with many disparate systems communicating and exchanging data. NSC has found that the many-to-many application messages require distributed management control. When the volume of Web services systems increases, so does the need for central control policy for security. Current Web services platforms do not provide centralized control for security and management.

The deployment of Web services and standardization of XML is opening up potential security threats. For most IT groups, maintaining security in a Web environment is challenging. According to our ADR study, security is the top concern of 55% of respondents for rolling out Web services. IT groups need a solution that sits in the network and serve as the first line of defense for XML traffic. ADRs can perform this critical security function by figuring out what traffic is real and authenticated verses malicious. ADRs can apply on the fly security schemes, certificate management, user level authentication and security mediation, effectively accelerating security. For Web services, organizations should be able to identify the requesting system, authenticate,

and authorize transactions. Organizations must also be able to ensure the integrity of data to make sure it has not been compromised. Finally, organizations need to be able to audit transactions for reporting and more effective management.

The capacity to scale to unpredictable levels in a production Web services deployment is a requirement for organizations that are planning to connect enterprise applications, customers, suppliers, and potentially business partners. Without the ability to scale quickly to meet the requests in a growing Web services environment, large-scale Web services deployments are doomed. Current Web service platforms rely on the application server to process XML messaging - offloading to an ADR enables cost effective scaling.

High volume Web services face similar demand and performance problems that high volume Web servers face—scalability. To solve the Web site problem, Web site architects place a local load balancer in front of multiple Web servers in order to balance the traffic among available server resources. Similarly, architects of high volume Web services can place an ADR in front of multiple Web services servers to balance messaging traffic. Using a proven methodology in load balancing from the lower network layers, the high volume Web site architecture supplies a model for ADRs to balance application layer traffic to accommodate high volume production Web services environments.

Current Web services platforms do not have a centralized policy engine. It is our understanding that the major Web services platforms have plans to include a centralized SOAP policy engine in the future; however, any production Web services deployments will require central policy enforcement now. Without centralized management, SOAP policies would have to be managed in each platform for each services connection. Currently, most ADRs include a centralized SOAP policy engine and management point that enables policy enforcement in a heterogeneous Web services environment.

Middleware

Just because you can implement XML processing in current Web services platforms doesn't mean you should. Platforms and ADRs should be used when it makes sense. Production deployments of Web services will include a myriad of platform and application vendors, increased complexity, and an increased number of federated management points. For most production Web services environments it makes sense to use an ADR, however, there are a number of instances where ADRs are not necessary, including:

- Small deployments
- Pilot implementations
- Lab environments

If Web services requirements are small and simple, keep it small and simple. At some point in the future the gambit of organizations that range from small to multinational will likely benefit from a Web services deployment. The companies that develop Web services platforms are actively targeting companies of all sizes. If a Web service is small scale with no significant growth plans, XML processing is likely best kept within the Web services platform. If in an enterprise, small inter-group deployments of Web services are planned with no significant volume requirements in the future, it is likely unnecessary to deploy an ADR solution.

Organizations deploying pilot implementations of Web services likely do not have the immediate need to scale Web services across a vast array of enterprise applications, customers, suppliers, and business partners. If it is a simple deployment, use a simple architecture. However, should the pilot include the requirement to scale to a production Web services environment, architects should evaluate ADRs assuming a management solution will be required as the project scales.

Organizations that are fortunate to have a lab environment to test possible Web services solutions likely will not generate enough XML processing demand that warrants offloading to an ADR solution.

Finally, there are some functions that, even with emergence of ADR, will still likely best be done by traditional middleware, such as business process orchestration, design-time change management and versioning, and XML-to/from-other format transformation.

Conclusion

Over the last few years the proliferation of Web-applications to support networked business processes has led to the vast need for integration between Web applications and data systems. Data integration solutions evolved from one-off in-house implementations to simple ETL tools, replication, federated database solutions, and highly complex EAI systems.

With the standardization of semantic data representation it becomes possible to also standardize on data services. Web services, based on open-standards like WSDL and UDDI for service definition and discovery, are developing as the dominant trend in enterprise computing in this decade.

The increasing need for interoperable Web applications coupled with the growing adoption of open data standards has created a need for Application Data Routing. Application Data Routing provides the opportunity to create a standardized set of solutions that use Web services technologies to reliably and securely share data between systems. Application Data Routing is fundamental to supporting production Web services deployments - whether it is moving unstructured document files between file systems in order to support interdepartmental processes, populating new Web databases to

support a new portal initiative or providing new Web services to offer request-based transaction log access to trading partners.

Application Data Routing will be as fundamental to enterprise business processes as packet routing is to networking. ADRs are the equivalent of packet routers for semantic data. ADRs enable the secure, scalable and optimized routing of semantic data at the right place and time, not only for immediate cost optimization but also for the long term robust build-out of new interoperable Web services in today's and tomorrow's real-time enterprise.