SYS-CON MEDIA

IS THE WEB SERVICES LOVE AFFAIR ENDING? pg46

March 2004 Volume 4 Issue 3

WebServices JOURNAL

.NET J2EE XML

SECURING YOUR ENTERPRISE WEB SERVICES IN A SUSPICIOUS WORLD

pg20

THE WORLD'S LEADING MAGAZINE DEDICATED TO WEB SERVICES TECHNOLOGIES • WSJ2.COM

Introducing OpenLink Virtuoso

A CROSS-PLATFORM Universal Server that transparently integrates disparate application databases and disparate applications across your enterprise. It provides a Virtual Database, XML Database, SQL Database, Web Application Server, and Web Services Platform as part of a **single-server solution.**

Empower your organization today, download a FREE evaluation copy from www.openlinksw.com/virtuoso or call us on 1 800 495 6322

Database Support
Oracle, SQL Server, DB2, Sybase, Informix, CA-Ingres, Progress, MySQL, PostgreSQL, and other ODBC or JDBC accessible databases.

Programming Languages
Any .NET bound language, Java, C/C++ etc.

Protocol Support
SQL, ODBC, JDBC, XML, XSL-T, XQuery, XPath, HTTP, WebDAV, NNTP and POP3

Web Application Hosting
ASP.NET, Java Server Pages, PHP, Perl, and Python

Platform Support
Windows, UNIX (all major flavors), Linux, Free BSD, and Mac OS X

Runtime Environment Hosting
Microsoft .NET, Mono, Java, Perl, Python, PHP, and others

www.openlinksw.com/virtuoso

OPENLINK®
SOFTWARE

CURSOR:
CIO who discovers that his expensive new integration system needs yet another integration system.

Data has a funny habit of getting itself trapped inside functional silos. But you need the right technology if you want to get it out. Our exteNd™ family of web service solutions lets you liberate information and get it to everyone who needs it. That means Marketing can learn things from Operations. And Sales can share what it knows with R&D. Even when the systems and applications aren't naturally compatible. And the more sharing that goes on, the more productive and profitable you are. To find out how our team of experienced consultants and partners can help improve the flow of information at your company, call us at 1-800-764-3700 or visit www.novell.com/extend. ➔ **WE SPEAK YOUR LANGUAGE.**

Novell.

March 2004 Volume 4 Issue 3

WebServices JOURNAL
.NET J2EE XML

Inside WSJ

WSJ: Features

From the Editor

Industry Commentary

WSJ: Guest Commentary

WSJ: Guest Editorial

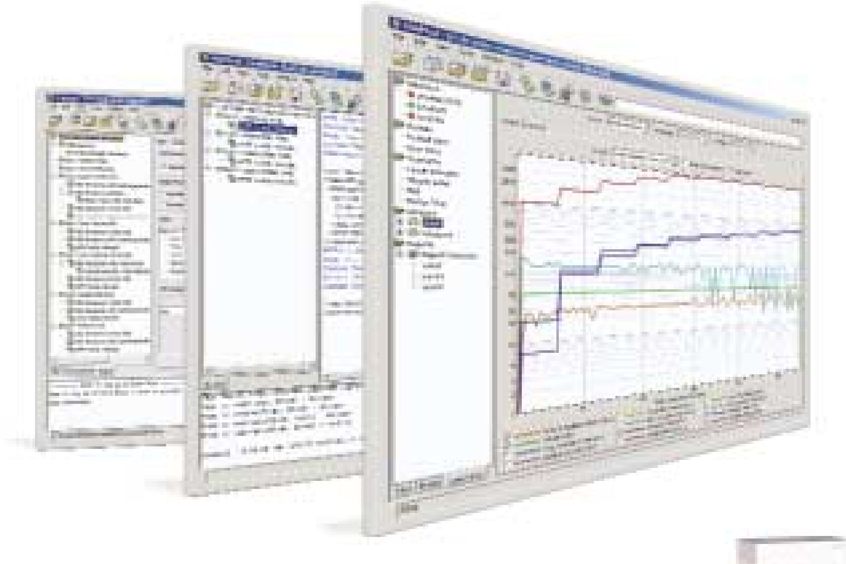
WSJ: Focus Security

WSJ: Standards

WSJ: Guest Editorial

WSJ: Product Reviews

It's automated. It's fast. And it's the most versatile web services testing tool ever.



Parasoft SOAPtest.®

The simple fact is, no other web service testing tool can do what SOAPtest can do. Or do it as fast and reliably. From ensuring functionality and interoperability to telling you how your web service is performing, SOAPtest automates all critical testing processes across the entire lifecycle of your web service.

But don't take our word for it... Go to www.Parasoft.com/SOAPtest and try it for free - no commitment, no obligation. If you like it (and we suspect you will) just let us know. We'll be more than happy to help you and your development team get up and running.

Server Functional Testing

- Automatically reads and tests client WSDL, to generate test cases.
- Records and captures HTTP traffic to build test cases.
- Flexible scripting with Java, JavaScript or Python
- Ability to emulate as server or client.
- Support for HTTP GET/POST.
- MIME attachments (per SOAP with Attachments specs)
- WS-Security, XML Encryption, XMLSignature Support.
- Data-driven testing through data sources.
- Works on new or already existing web services.

Load Testing

- Creates Windows and SNMP monitors to collect network information.
- Generates detailed reports for QA

Regression Testing

- Automatic Regression tests for functional tests suites and WSDL.

Conformance

- Conformance to WS-I Basic Profile 1.0
- WS-I Analyzer tool validates messages sent to and from a web service.

Protocol Support

HTTP 1.0
HTTP 1.1 w/Keep-Alive Connection
HTTPS
TCP/IP
JMS: supports testing of asynchronous Web services

Compatibility

- Works with any SOAP-compliant implementation, including Microsoft & IBM
- Windows 2000/XP
- Linux
- Solaris

Free Download...
Parasoft.com/SOAPtest

PARASOFT®
We make software work™

For Downloads go to www.Parasoft.com/SOAPtest. Or call 888-305-0041.

www.altova.com
xmlspy®
2004
NEW Microsoft® Visual Studio®.NET Integration
Enhanced Database Connectivity
XML Differencing
XPath 2.0 Analyzer
improved XML Schema/WSDL Editor
XSLT Debugger and Code Generator
XML
Schema
XSLT
WSDL
J2EE Station
CORBA
.Net Station
mapforce Terminal
Introducing mapforce™ 2004
New Visual Database-to-XML and XML-to-XML mapper!
Generate Java, C#, C++, and XSLT
mapforce® 2004
FREE 30 day Trial Version

WebServices JOURNAL
.NET J2EE XML

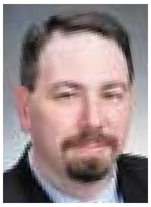
EDITORIAL
EDITOR-IN-CHIEF
Sean Rhody sean@sys-con.com
INDUSTRY EDITOR
Norbert Mikula norbert@sys-con.com
PRODUCT REVIEW EDITOR
Brian Barbash bbarbash@sys-con.com
.NET EDITOR
Dave Rader davidr@fusiontech.com
SECURITY EDITOR
Michael Mosher wsjsecurity@sys-con.com
TECHNICAL EDITORS
Andrew Astor aastor@webmethods.com
David Chappell chappell@sonicsoftware.com
Anne Thomas Manes anne@manes.net
Mike Sick msick@sys-con.com
EXECUTIVE EDITOR
Gail Schultz gail@sys-con.com
EDITOR
Nancy Valentine nancy@sys-con.com
ASSOCIATE EDITORS
Jamie Matusow jamie@sys-con.com
Jean Cassidy jean@sys-con.com
ASSISTANT EDITOR
Jennifer Van Winckel jennifer@sys-con.com

PRODUCTION
PRODUCTION CONSULTANT
Jim Morgan jim@sys-con.com
LEAD DESIGNER
Richard Silverberg richards@sys-con.com
ART DIRECTOR
Alex Botero alex@sys-con.com
ASSOCIATE ART DIRECTORS
Louis F. Cuffari louis@sys-con.com
Tami Beatty tami @sys-con.com

CONTRIBUTORS TO THIS ISSUE
Andy Astor, Brian Barbash, Marc Chanliau, Clayton Donley,
Peter Fingar, Matthew Fuchs, Jeannine Hall Gailey,
Phillip Hallam-Baker, Bret Hartman, Paul Lipton, Prateek Mishra,
Michael Mosher, Sean Rhody, Howard Smith, Prasad Yendluri

EDITORIAL OFFICES
SYS-CON MEDIA
135 CHESTNUT RIDGE ROAD, MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 FAX: 201 782-9637
WEB SERVICES JOURNAL (ISSN# 1535-6906)
Is published monthly (12 times a year)
By SYS-CON Publications, Inc.
Periodicals postage pending
Montvale, NJ 07645 and additional mailing offices
POSTMASTER: Send address changes to:
WEB SERVICES JOURNAL, SYS-CON Publications, Inc.
135 Chestnut Ridge Road, Montvale, NJ 07645

Real Security



WRITTEN BY
SEAN RHODY

There's a joke in the industry that states that the only really secure computer is in a room where no one is allowed to go. It has no Internet connection, and no network connectivity. It has no monitor, so no one can peak over a user's shoulder to see what is on the screen. And it's not plugged in. And of course, it doesn't run Windows. And frighteningly enough, there are people who work in security who truly believe this.

Security is a topic that concerns people at all levels. In Web services, security is viewed by many as particularly important, and the lack of security is often cited as the single biggest barrier to Web services adoption. This is troubling, because much can be done today with Web services and the current state of security.

Security can mean different things to different people. People think about hackers, users, messages, and the like; and depending upon their point of view, and their business needs, the current state of Web services security may or may not fit their needs. In a nutshell, it comes down to four related items: integrity, identity, entitlement, and prevention.

Integrity means ensuring that all messages and conversations carried out via Web services are private, controlled, and unadulterated. No message content should be seen by anyone except the intended participants.

Identity is often needed to establish that type of integrity. A reliable mechanism for ensuring that the parties communicating are in fact who they say they are is at the heart of managing a conversation with integrity. It's also necessary to know who the participant is in order to properly enforce entitlements.

Entitlement is the ability to limit access to Web services based upon the identity of the participant, or in some cases to make decisions based on identity. This can play a role in the development of services, or in the management of services once they are deployed, depending upon how the information is provided. Entitlement should ideally be declarative, but sometimes because of the implementation of the Web service identity information may need to be transferred into the service itself (usually because the service wraps some legacy code).

Finally, security has to be able to prevent tampering. Such prevention includes the ability to determine that an attack is under way, prevent the tampering, and ideally locate the perpetrator. Depending upon your degree of paranoia, this may be as necessary within the organization as without.

Fortunately, some of this is already available, and more is under way in the form of a host of standards, including WS-Trust, WS Secure Conversation, SAML, XACML, and XML Digital Signatures. These and a few other standards form a hierarchy of complementary technologies aimed at providing integrity, identity, entitlement, and prevention.

Even more important, the reality of security is that in many cases what is currently available is sufficient to meet requirements. Just as the computer system I spoke about initially represents security nirvana, a place that is sought after but never achieved, so it is with security and Web services. And just as the cost of operating that ideal system is infinite (pick a number, then divide by zero, because the system is never turned on), so too the cost of securing services has to be figured into the equation of Web services. Certainly many instances of Web services will need highly secure conversations, especially financial transactions where money changes hands and nonrepudiation is essential for the adoption of the service. But there are many other cases where the requirements for identity, or entitlement, or even for integrity or prevention just aren't so paramount that what we have today isn't already capable of meeting those needs. Internal Web services that never touch the Internet, for example, are likely to need much less in the way of security.

So it's important to understand the need for security, as well as the cost of security, combined with the abilities of security to meet the business need. After all, what good is a computer if you can't even turn it on? ⓔ

About the Author

Sean Rhody is the editor-in-chief of *Web Services Journal.* He is a respected industry expert and a consultant with a leading consulting services company.
■■■ Sean@sys-con.com

Mindreef News

Volume 2 March 2004 www.mindreef.com

INTRODUCING

Mindreef, WS-I, and Interoperability

WHITEPAPER

Q: What is Web Services interoperability, and why is it important?

A: Web services exist because there is a need for language and platform independent communication between computers. Ensuring interoperability between Web service components is essential to achieving this vision. The platform-independent specifications that make up the core Web services protocols (SOAP, WSDL, UDDI) are powerful, but expressive to the point of ambiguity. The same service can be represented with WSDL in numerous ways, and toolkits can encode the same message with SOAP in entirely different ways. Without commonly-followed guidelines that limit protocol use to an unambiguous subset, Web services would not interoperate.

Q: Who is the WS-I, and what is the Basic Profile?

A: The Web Services Interoperability Organization (WS-I) is an open industry organization that promotes interoperability among Web services across platforms, applications, and programming languages. Acting as a "standards integrator," the WS-I developed Basic Profile 1.0, a set of guidelines to limit the scope of what is acceptable Web service usage. The Basic Profile 1.0 describes how these core Web service specifications should be used to develop highly interoperable Web services. To make the Basic Profile 1.0 immediately useful, the WS-I recently released the WS-I Testing Tools, which enable Web services developers to generate a Basic Profile compliance report against any number of Web services artifacts.

Q: How is a Basic Profile compliance report used?

A: Generating a Basic Profile compliance report is excellent for pass/fail analysis. If you are developing a Web service, you can deploy a passing compliance report alongside your Web service to publicize its high quality, as well as support your interoperability claim. For clients, performing pass/fail analysis via a Basic Profile compliance report can aid in your choice of toolkit, and ensure that you are invoking Web Services with quality requests.

Q: Where does Mindreef fit in?

A: Mindreef understands the need for interoperable Web services and the Basic Profile. And while the WS-I testing tools may notify you of the existence of a problem, the report can be long and cumbersome, and specific errors give only a broad indication of where the problem might be. In an effort to help you discover and resolve your interoperability problems, Mindreef presents SOAPscope 3.0.

Q: What is SOAPscope and how does it go beyond the WS-I Testing Tools?

A: SOAPscope 3.0 collects messages and WSDLs in a variety of ways, and provides an intuitive framework for performing Web services diagnostics and analysis. SOAPscope is capable of analyzing Web service artifacts against a myriad of specifications and guidelines, including the WS-I Basic Profile, SOAP, and WSDL, as well as Mindreef Best Practices. SOAPscope provides an interactive UI to help solve problems, not just report them. As Web service artifacts are analyzed, SOAPscope pinpoints and highlights the XML fragments causing the problem, and presents it within a larger context, including extensive help, to assist you in solving the problem.

Q: How do I resolve errors as I discover them?

A: SOAPscope 3.0 addresses the resolution of Web services interoperability errors by providing a unified Web service diagnostic system, and supporting a process of iterative improvements. As artifacts are collected and automatically stored, they can later be examined, and analyzed for potential problems. If a problem is found with a SOAP message, you can edit the artifact, resend it, diff the result against previous results, and re-analyze the transaction. WSDLs can be viewed, dynamically invoked without writing code, diffed for changes, and re-analyzed as they evolve.

Q: Isn't analysis something that a tester does at the end of the project?

A: No. Of course you'll want to generate a compliance report at the end of your development to support your interoperability claim, but running analysis throughout the development process can quickly uncover latent bugs and issues at a time closest to when they were introduced. Uncovering those bugs automatically with intermittent analysis saves developers and testers from spending hours hunting down something that can be detected within seconds. SOAPscope 3.0 provides you with life-cycle value, performing analysis at any stage, and even generating the WS-I Basic Profile 1.0 compliance report when you are ready to deploy.

Q: Now that I've done all this, can I guarantee interoperability?

A: You can never guarantee interoperability, but by integrating analysis into your development and testing process, you can be assured that you've done everything possible to be interoperable.

Q: Where can I learn more?

A: Read more about Web services interoperability in our FREE whitepaper at http://www.mindreef.com/interop-wp.

"SOAPscope rocks!"
- Don Box, XML Messaging Architect, Microsoft and co-inventor of SOAP

Assande EAI Messaging Suite features

- XML Schema Repository
- Data Abstraction, Transformation, and Mapping
- Message Assembly and Generation from Components
- Messaging Integration Service Deployment
- Automated Naming Standards, Version Control, and DIFF Functions
- Infrastructure Utilization and Pipeline Reporting
- Conversation Deployment and Message Release Management
- State Management and Integrated Workflow
- Advanced Search Capabilities
- Integration Documentation (MS Word, Excel) Repository

ASSANDÉ™
INTEGRATION MANAGEMENT COMPANY

http://www.assande.com

WebServices JOURNAL
.NET J2EE XML

PRESIDENT AND CEO
Fuat Kircaali fuat@sys-con.com
VP, BUSINES S DEVELOPMENT
Grisha Davida grisha@sys-con.com
GROUP PUBLISHER
Jeremy Geelan jeremy@sys-con.com

ADVERTISING
SENIOR VP, SALES & MARKETING
Carmen Gonzalez carmen@sys-con.com
VP, SALES & MARKETING
Miles Silverman miles@sys-con.com
ADVERTISING DIRECTOR
Robyn Forma robyn@sys-con.com
DIRECTOR, SALES & MARKETING
Megan Mussa megan@sys-con.com
ADVERTISING SALES MANAGERS
Alisa Catalano alisa@sys-con.com
Carrie Gebert carrie@sys-con.com
ASSOCIATE SALES MANAGERS
Kristin Kuhnle kristin@sys-con.com
Beth Jones beth@sys-con.com

SYS-CON EVENTS
PRESIDENT, SYS-CON EVENTS
Grisha Davida grisha@sys-con.com
CONFERENCE MANAGER
Lin Goetz lin@sys-con.com
NATIONAL SALES MANAGER
Sean Raman raman@sys-con.com

CUSTOMER RELATIONS/JDJ STORE
CIRCULATION SERVICE COORDINATORS
Shelia Dickerson shelia@sys-con.com
Edna Earle Russell edna@sys-con.com
Linda Lipton linda@sys-con.com
Merline Noel merline@sys-con.com

JDJ STORE MANAGER
Brunilda Staropoli bruni@sys-con.com

SYS-CON.COM
VP, INFORMATION SYSTEMS
Robert Diamond robert@sys-con.com
WEB DESIGNERS
Stephen Kilmurray stephen@sys-con.com
Christopher Croce chris@sys-con.com
ONLINE EDITOR
Lin Goetz lin@sys-con.com

ACCOUNTING
FINANCIAL ANALYST
Joan LaRose joan@sys-con.com
ACCOUNTS RECEIVABLE
Charlotte Lopez charlotte@sys-con.com
ACCOUNTS PAYABLE
Betty White betty@sys-con.com

SUBSCRIPTIONS
SUBSCRIBE@SYS-CON.COM
1-888-303-5282
For subscriptions and requests for bulk orders,
please send your letters to Subscription Department
Cover Price: $6.99/issue
Domestic: $69.99/yr (12 issues)
Canada/Mexico: $89.99/yr
All other countries: $99.99/yr
(U.S. Banks or Money Orders)

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

For list rental information:
Kevin Collopy: 845 731-2684,
kevin.collopy@edithroman.com;
Frank Cipolla: 845 731-3832,
frank.cipolla@epostdirect.com

SYS-CON Publications, Inc., reserves the right to revise,
republish and authorize its readers to use the articles
submitted for publication.

SYS-CON MEDIA

WS-Policy – Making Web Services Simple

WRITTEN BY
PHILLIP HALLAM-BAKER

Those in the security business, like me, often complain that security is the last thing that people consider when designing a new application. If a little more thought had gone into the security of the e-mail protocols, for example, it would be easier to trace the true origin of an e-mail, which would make tackling the mounting problem of spam much less daunting.

One of the reasons Web services are so important is that they represent the first time security issues were considered at a very early stage in the design of a protocol framework. Now that Web services are being used to solve real-world problems, the issues we are starting to face are the problems of success – how will we manage when we are dealing with hundreds of Web services protocols connecting thousands of partners?

Managing changes to a network protocol is hard. The first lesson taught at network protocol design school is to include a version number so that the machine running version 4.2 or the protocol knows to refuse requests from a machine running the now obsolete version 2.3. At least, that's the theory. The practice tends to be that once a protocol is deployed, you rarely get a second chance. Most of the Internet protocols we use every day, such as e-mail and news, have changed remarkably little in the past 10 years. The Web, only a little over 10 years old, has seen more change but none of major consequence for the past eight. It is one thing to announce a new version of a protocol, quite another to see it deployed.

Stability has advantages. E-mail could never have become so widely used if the Internet mail protocols had changed each year. But the price of that stability is high. The effect is that the Internet tends to run using lowest common denominator technology. As the number of Internet users approaches a billion, we are using a two-decades-old protocol from the dawn of the Internet designed to serve a user community of thousands. The original design flaw that left security out of the design of the e-mail system would not have mattered so much if it was easier to correct its consequences.

This is why the WS-Policy mechanism currently in development is such an important part of the Web services architecture. Readers familiar with Web services will know that Web Services Description Language (WSDL) provides a description of a Web Service protocol. WS-Policy goes further and allows the configuration of a specific Web service to be described.

It's a bit like going to a hamburger restaurant. You know in advance that they serve hamburgers and fries, but do they serve onion rings or milk shakes? Do they accept credit cards or is it cash only? Knowing that information up-front allows you to choose the right place to eat.

The result is that administrative operations that used to be performed manually today can be automated. Automation may not sound like a big deal today when few enterprises are running Web services that can be seen outside their firewall. Few networks of Web services users have more than 10 members. If you need to do an upgrade you can just pick a public holiday to take down the network, change the software, and restart.

If you are running Web services in a production environment with links to a few hundred e-commerce partners, automated management becomes essential. Even though the protocols you are running may be "standard," there are inevitably configurations and options that have to be set right before your Web services can talk to each other. WS-Policy allows this to be done at the appropriate levels – let the machines do the work.

A similar change took place in the Internet 20 years ago when the Domain Name System (DNS) replaced the list of host names and IP addresses that used to circulate between network administrators. Without the DNS, the Internet could never have grown to a million users, let alone a billion. Yet today we take it for granted that when you type in www.verisign.com your browser will connect to one of the machines currently

–Continued on page 27

Snow White's FIRST Web Services

A cautionary fable for IT management

One day, Snow White decided to deploy a Web service. Her IT dwarves immediately went to work and were pleasantly surprised to find how easy it was to create the Web service using modern development tools. To Snow White's development dwarves, it almost seemed like magic.

WRITTEN BY
PAUL LIPTON

Since Snow White's cottage was a Java shop, they deployed the Web service in their J2EE application server, but they could have just as easily used .NET and it would have seemed just as magical – maybe even more so, given the wealth and power of the Wizard of Seattle.

Since Snow White had lived in a palace with a wicked witch, she was no stranger to corporate culture in general and risk aversion in particular. Snow White also had clear goals. She had wisely eschewed the use of magic mirrors, and tended to favor a few industry analysts along with a handful of software vendors who seemed both willing and able to partner with her for the long haul. She wanted to achieve a more flexible and agile IT infrastructure by gradually moving IT to a service-oriented architecture (SOA). Snow White understood that you can't build a robust SOA for your enterprise based on a foundation of unmanaged and unsecured Web services. She wisely instructed her IT dwarves to make sure that this first production Web service was manageable and secure before they implemented any other Web services.

Chapter One – The Stage Is Set

Security wasn't difficult to enable for their first Web service. Their application server provided a magical runtime environment that allowed developers to specify security declaratively within an XML file or using a pretty GUI. Her staff used this magic to make sure that their Web service, using WS-Security, would only work with client applications that supported XML Encryption and XML Signature. The identity of her customers was wisely required to be passed as a security token within the WS-Security element of the SOAP messages that she received. There was no need for federated identity management at this early stage since the cottage directory server had the IDs of all their customers firmly in hand, but they had a good plan to expand, as needed, toward a wider community of distributed identities in the future.

With their experience in building and securing a Web service behind them, Snow White's development dwarves next recommended the purchase of a Web services management product to monitor the availability of their Web services. As developers, they were particularly pleased that this product could manage a Web service without having to change a line of code. Also, the product could automatically discover and manage new Web services as needed. Automatic discovery was particularly important, since they were concerned about rogue Web services being deployed in the enterprise. Certain office productivity products had made this almost too easy, even for non-programmers. Of course, this Web services management product could also report on important service metrics and help make sure that the service was responsive and reliable.

Everything was tidy and in place, and Snow White felt safe, secure, and highly profitable in her little house in the woods. Everything seemed fine until one day the head IT dwarf (who used to be Sneezy before he found allergy medication) found his boss on the floor weeping. Six important customers had complained in the last hour about poor performance on the Web service. "How could this have happened?" demanded the tearful Snow White, "I thought you said that our Web services management software would warn us of potential problems!"

The world, by way of Memphis.

FedEx, the very model of corporate efficiency, always looks for new ways to improve service. HP helped FedEx IT managers deploy HP OpenView,™ which lets them identify and correct potential issues quickly and simply. The result is a smoothly running operation that produces happy customers from Memphis to Monaco— not to mention Mexico, Morocco and Martinique. www.hp.com/plus_fedex

Chapter Two – What Went Wrong?

In truth, there were a number of IT management, development, and product evaluation issues that had contributed to Snow White's tears. One important issue was the ineffective and superficial integration between their existing enterprise management system and their new Web services management software. The operations staff was running the entire IT infrastructure (a multitude of hardware and software entities such as operating systems, application servers, messaging middleware, routers, networks, databases, networked storage, and so on) using an enterprise management solution from a different vendor than the one who had provided the Web services management software. This decision had unintended consequences.

Their Web services management software had correctly warned them that their Web service was performing poorly. So, from the perspective of the Web services developers, the Web services management software had performed admirably – reporting a wide variety of metrics that are typically of concern to the operations staff. It had even managed to send its messages to the enterprise management system console. But, the Web services management product used different terminology and had a different user interface than the enterprise management system. Despite some efforts to train some operations staff in the particulars of both management systems, in a crisis the staff was confused and frustrated. They found it difficult to work with two different management systems.

In terms of internal Web services expertise, Snow White had been forced to rely almost exclusively on the development organization since they had been the first to work with Web services. In retrospect, Snow White should have driven greater participation from her operations staff in the product evaluation – providing the training and consultative resources that they would need to better manage the issues from their perspective.

Web services management software is quite naturally focused on the higher-level specifics of Web services, such as messages and service descriptions (SOAP and WSDL). While such software can often identify a troublesome Web service even in complex aggregations of cooperating Web services, it quite properly lacks any root cause–analysis capability down to the IT infrastructure level. In other words, it isn't intended to trace the underlying cause of a problem down to a particular IT software or hardware entity, like a database or router. The underlying business logic and the supporting IT infrastructure are invisible to the Web services management software. So, in the case of Snow White's Web service performance problem, the operations staff had tried to correlate warning messages sent by the Web services management software with the large number of warning and error management messages related to underlying IT infrastructure and business logic reported by the enterprise management solution, but the lack of deep integration between the two management systems made such work tedious, time consuming, and error prone.

In retrospect, Snow White's strategy and evaluation team would have benefited from the understanding that management cannot be done piecemeal. As part of a comprehensive plan to properly manage new technology stacks such as Web services, on-demand computing, and Grid, the team should have considered the long-term interoperability, training, overhead, and partnership challenges that derived from the use of multiple management solutions. The IT dwarves had selected new Web services management software that was unlikely to enjoy a more useful level of integration with their enterprise software solution in the future. Were they prepared to deal with the added cost and complexity? Had they investigated Web services management products from their own enterprise management vendor? What was the current level of integration being offered by that vendor and, more importantly, what was the enterprise management software vendor's commitment to deeper, more useful levels of integration in future releases?

Of equal concern, the security officer had been absent from discussions concerning Web services management because of the common, but mistaken, notion that security and management are two entirely different concerns. These days, security management increasingly interacts with traditional areas of management such as systems and lifecycle management. The interoperability, visibility, and exposure provided by existing and emerging Web services standards are creating ever more interdependence between management and security. Consider the simple example of a denial-of-service attack on a Web service. Is this a Web services security issue (the enterprise is clearly under assault) or is this a Web services management issue (the service has experienced a change in utilization and SOAP message traffic)? The answer, ultimately, is both.

Many organizations are still in the early adopter phase of Web services use and might justifiably defer consideration of the inevitable convergence of security with other management concerns in the short term. However, Snow White's admirable commitment to an SOA and the deployment of her first production Web service clearly demonstrate that Snow White's strategy team should have had a long-term partnership and deployment plan in place that would allow them to steadily evolve their management and security operations toward a cohesive whole, as needed.

The absence of proper input by the security officer during the planning and evaluation phase also meant that enterprise-level security policy played a surprisingly small role in the decision by the development dwarves to utilize the Web services security functionality provided by the application server. While it is often true that platform-provided security can provide a relatively quick and inexpensive way to comply with enterprise Web services security and management concerns, this is not always the wisest course of action.

Tying security to the Web services platform can make it difficult to centrally administer and maintain policy in a heterogeneous enterprise. Even if the enterprise has standardized on one application server, there are often many other legacy processes and data sources that are not able to leverage the security and management capabilities provided by the Web services platform. In any heterogeneous SOA, integrated, enterprise-level Web services security and management solutions that are independent of the Web services platform may be the only way to ensure that all Web services, not just those deployed on the application server, are fully compliant with corporate policy and can be centrally monitored.

Conclusion

What conclusions can we draw from this IT management fable? Snow White's problem wasn't a poisoned apple (Snow White was not the kind of CxO to fall for that old trick!). It appears that even well-run IT organizations like Snow White's, with a clear vision of where they want to go, can be surprised by the complexity and challenges of managing and securing Web services as part of an SOA. The moral of the story is simple and of value to IT shops in enterprise cottages everywhere. To be useful in the long term, Web services management needs to be comprehensive and holistic – a carefully mixed potion of true Web services management genuinely integrated with IT infrastructure management. Also, in terms of implementing security for Web services, an important part of the total management equation, IT organizations would do well to look beyond the security needs of any particular Web service. Rather, they should begin to formulate a more comprehensive security and management policy and mechanisms that extend beyond any one Web services–enabled platform to serve the enterprise and the SOA as a whole. With these lessons learned, Snow White and her IT dwarves should live happily ever after.

About the Author

Paul Lipton is a senior architect and technology strategist in the Office of the CTO at Computer Associates (CA). He has been an architect and developer of enterprise systems for more than 20 years, and has worked closely with key CA customers to solve important business challenges through the creation of mission-critical distributed solutions. Paul has represented CA in numerous standards organizations, such as the W3C, OASIS, and the Java Community Process, and is currently serving on standards committees involved in the definition of new Web services standards for management, orchestration, and choreography. He is also a highly sought-after author and conference speaker on a wide range of topics.

■ ■ ■ paul.lipton@ca.com

Rogue Web Services

Risks and success strategies

■ Like the hero of a Greek tragedy, Web services' most compelling advantages are simultaneously its most serious dangers. Web services have passed the initial hype cycle. The convergence of industry support, ease of use, and the desire for cost-effective solutions for integration and services-oriented architectures (SOA) has made it a popular choice for architects, developers, and integration analysts, with numerous projects underway. Web services technologies are making inroads within organizations in much the same way Web site technologies proliferated. However, the benefits of loose coupling, decentralized development, and support for heterogeneity – rapid grassroots development of Web services with flexible, agile architectures – introduce a multitude of new issues organizations must address to prevent the negatives from outweighing the positives. Security, reliability, and performance are all critical issues to be specially managed in a Web services environment. This article looks at "rogue Web services," already a growing concern in IT, particularly for organizations that have not applied top-down governance on usage.

WRITTEN BY
BY MATTHEW FUCHS

Rogue Web Services

A rogue Web service (RWS) is a Web service that's out of control. It might be perfectly benign, but unsanctioned by IT. Or it might be intentionally malicious – either attacking your systems or squatting and consuming your resources. It might even be an officially sanctioned service that unintentionally starts hammering other Web services due to a coding bug. Of course, even the most benign rogue service could turn up in the last category at any time – almost by definition it hasn't gone through the same QA or testing as production code.

Perhaps the most compelling reason RWS threaten to become a significant danger is the ease with which they can be created. Although veterans of earlier large-scale distributed technologies, such as DCOM and CORBA, frequently disparage Web services, those were very complex systems requiring a fair amount of knowledge and programming skill to deliver a functional application. In addition, distributed object technologies were never able to break out of their silos. The prime differentiators between these systems and Web services are the ease with which a Web service can be constructed to perform fairly sophisticated tasks, and the loosely coupled nature of Web services technologies. These significantly lower the barriers to entry for both the technical knowhow for building a Web service and the time required to get a new service initiated or integrated with an existing service. And that significantly increases the number of people capable of building an RWS.

Unsanctioned internal Web services, particularly clients, but servers as well, can arise on any computer accessible through HTTP. It takes relatively little programming skill to execute a Perl or Python script in a Command shell to listen for requests on a particular port, do some additional processing, and return the results. From there, it's also possible to create a Web service client that creates messages for a variety of Web services and coordinates the result. These are often called "composite" Web services, but despite becoming a buzzword they are scarcely more difficult to build than ordinary ones, especially if you don't worry about making them safe.

The primary means of describing a Web service, WSDL, is a fairly easy-to-read interface definition language. Unlike CORBA or ASN.1 stub generators, an astute programmer can easily generate a stub from the description, and there are many easily available generators for common programming languages. Even where that is not available, a message itself is often self-explanatory – a new message can be "cloned" from an old one just by replacing bits and pieces.

The barrier is even lower when Web services are easily integrated into the latest versions of popular desktop software, such as Word macros, Excel spreadsheets, and PowerPoint slides. There is explicit Web service support in MS Office 2003, but it is possible to access Web services through macros and extensions in earlier versions. Given an RPC-style service, a stub only needs a URL, a function name, and a list of parameter names, types, and values, to create a SOAP message. For simple return values, little is needed beyond simple pattern matching to retrieve the answer. A PowerPoint slide set containing a Web services call made publicly available could generate a request every time a particular slide is viewed.

Once a Web services message is prepared, it moves along one of the most ubiquitous and familiar protocols created – HTTP. Many programming languages already have libraries to create HTTP messages, but it is easy to create an HTTP message by hand and send it along a socket. From a programming perspective, it is a simple request/response requiring very little code.

So we see that Web services lower the barriers to entry for the construction of distributed applications for legitimate developers and users as well as for illegitimate ones.

Everything OLD Is NEW Again!

Risks Associated with Rogue Web Services

Rogue Web services traffic is more difficult to protect against than random traffic because much of the danger is in information targeted at the application level that cannot be filtered at the IP level the way traditional firewalls can. It is quite possible for rogue traffic to originate behind the firewall from people in your own IT shop or even from end users. Also, RWS cannot be identified just by source and destination IP – it may be that the message is coming from an RWS at a partner location, so it's important to cut off just the aberrant user, not the entire site. The destination host may contain any number of Web services through information not accessible at the IP or even Web server proxy level. While the server may recognize the URL, the actual identity of the operation being invoked is in the contents of the message, requiring a level of filtering capable of looking at application–level information.

> "A rogue Web service (RWS) is a Web service that's out of control"

RWS, even of the most benign sort, represent a threat to a company's ability to control its own destiny. Even avoiding, for the moment, the worst possible abuses, unknown Web services can create a considerable drain on network resources. Allowing unimpeded grassroots development of Web services without any centralized attempts at standardization can lead to significant duplication as well as many avoidable mismatches among Web services. While the flexibility of the Web services SOA makes it much easier to deal with independently developed Web services, a small investment in shared design can go a long way to avoiding extra work in the long run. Therefore, it is important for an organization to control the set of technologies used.

As many Web services are a thin layer over existing applications, once access to a Web service spreads beyond the approved users, the damage can be as bad as any other kind of intrusion. The intruder can have the same kind of impact as anyone who has logged into your system. As more functionality becomes accessible through Web services, such as management and provisioning, there won't be much that can't be done using Web services. Worse yet, if your security credentials, such as private key, are stolen, then it is not just your internal systems that are compromised, but your expanded Web services environment as well, including fee-based services.

Success Strategies

Every organization is different. The most successful strategies depend not only on the technologies that are being used but also on the people and organizations involved. Organizationally, many IT groups deal with the rogue service issue through top-down governance, usually by an architecture and standards body. These groups define the ground rules for how services are created, what standards should be followed, and the rules that are required for corporate and industry compliance. In other organizations, governance of Web services is enforced by the CISO or associated security group. In still other organizations, it may be defined and enforced by the IT operations group. More often than not, all of these groups are somehow involved in defining the minimum security, monitoring, and management requirements for WS development, deployment, and management.

Many tools are in existence for detection, enforcement, and management of the XML Web service environment. A variety of sniffer tools are available for detecting XML and SOAP traffic, many of them free. Using simple rules, you can determine if the traffic is unsanctioned and fire off the necessary alert. Firewalls and other proxies can also be configured to perform content inspection, although they may lack sophisticated rule sets and the performance for more robust environments. UDDI directories and other service directories can be used to store sanctioned Web services to help ease management. A new class of product called XML Firewalls and Web Services Management (WSM) platforms can be used to address the security, monitoring, and management of services. These products are typically noninvasive and help detect and address RWS while providing a management framework and set of tools to enforce top down governance requirements. Many analysts agree that a fully integrated XML firewall and WSM solution provides, among many other benefits, the best solution for enforcement and ongoing administration for RWS.

Nevertheless, an important part of the value of Web services is lost in a regime that is strictly maintained. Not all Web services are made equal, and infrastructures that don't appropriately distinguish between the varying requirements will veer unacceptably in one direction or another. An effective regime will be able to distinguish between core and periphery, where the core represents the bottom tiers of client/server architecture, and the periphery represents Web service clients. Another important distinction is among services that cause dynamic updates to information or consume significant resources (such as money), and those that don't and may be simply informational. Rather than taking an overly restrictive stance, tools can be used to create policies to adaptively manage Web services traffic so that important systems are only accessible from approved clients, but others can be accessed in a more relaxed fashion with content filters at the periphery to inspect outgoing information.

The proliferation of RWS is not necessarily a bad sign. In fact, it might be said that this is an indication of the benefits that Web services provides organizations today. However, there are associated risks when Web services traffic is not appropriately controlled. A combination of managing the proper procedures and controls mixed with the appropriate technologies can enable any organization to realize the full value of Web services while minimizing the security and cost risks. ⓔ

About the Author

Dr. Matthew Fuchs is a member of the technical staff at Westbridge Technology. Previously, he was chief scientist for XML Technologies at Commerce One, and pioneered the theory and practice of using domain-specific languages in XML and SGML for distributed applications and agent-oriented communication over the Internet. At Commerce One he developed a variety of XML technologies, including SOX, the first implemented, publicly available, object-oriented Schema language and parser for XML.

■■■ mathew@westbridgetech.com

Lowering the cost and complexity of developing secure, manageable enterprise-class Web services with BEA WebLogic Workshop™ and Confluent for BEA

With Confluent for BEA, Web services developed in BEA WebLogic Workshop™ 8.1 are automatically monitored, managed and secured at development time. Confluent's Control is bundled with WebLogic Workshop 8.1.

Real-time monitoring

No additional coding required, turn on instrumentation for a Web service, view all conversations as they execute, and drill down to views of individual operations and controls

Policy specification and enforcement

Define operational policies including security, logging and monitoring for Web services within BEA WebLogic Workshop 8.1

Explicit instrumentation with custom controls

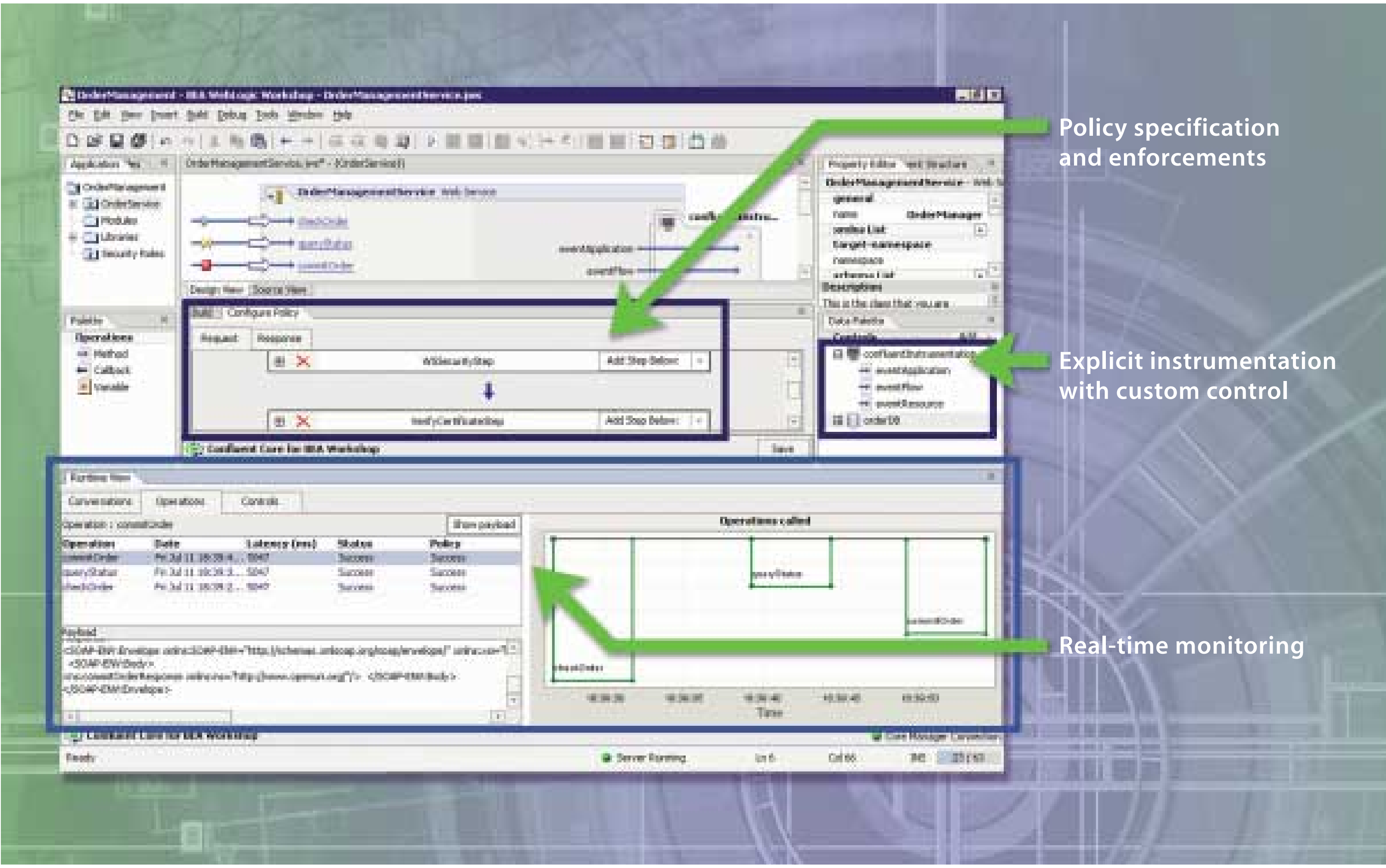
Use Confluent Control to explicitly call operations such as monitoring and security from Web services

BENEFITS:

If you're a Developer—focus on application logic not infrastructure services

If you're an Architect—consistently implement security, change and other operational policies

If you're an Operations Manager—efficiently monitor, manage and evolve increasingly distributed applications


OrderManagement - BEA WebLogic Workshop - OrderManagementService.jws
File Edit View Insert Build Debug Tools Window Help
OrderManagementService
Configure Policy
Request Response
WSSecurityStep Add Step Below
VerifyCertificateStep Add Step Below
Confluent Core for BEA Workshop
Conversations Operations Controls
Show payload
Operations called
Time
Ready
Server Running

Operation	Date	Latency (ms)	Status	Policy
commitOrder	Fri Jul 11 18:39:4...	5047	Success	Success
queryStatus	Fri Jul 11 18:39:3...	5047	Success	Success
checkOrder	Fri Jul 11 18:39:2...	5047	Success	Success

Policy specification and enforcements

Explicit instrumentation with custom control

Real-time monitoring

Confluent Software

Download Confluent for BEA at:
http://www.confluentsoftware.com/solutions/download/beacib.html

For more information on additional Confluent Web services management solutions, visit: **http://www.confluentsoftware.com**

Securing Your Enterprise WEB SERVICES in a Suspicious World

Out of many pieces, a harmonious whole

Deploying XML Web services in the enterprise has many compelling advantages. Web services provide a powerful foundation for building loosely coupled distributed applications and service-oriented architectures (SOAs). Enterprises use Web services to lower the integration cost of business-to-business solutions, allowing partners to share business documents without custom coding.

BY BRET HARTMAN

Web services flexibility comes with risks: Sensitive business data may end up in the wrong hands. Web services providers may be flooded by XML denial-of-service (XDoS) attacks, preventing legitimate users from gaining access. Business documents may be forged or altered, resulting in fraudulent transactions.

In this article, I'll discuss the security considerations for building Web services in a suspicious world – the dangers of working in this world; some old, some new. I then review the technologies that may be applied to address Web services security, along with the significant challenges when using these technologies. I conclude with recommendations on a security architecture that relies on an XML Web services security gateway as the first level of defense for Web services.

The Dangers

Because much of the value of Web services is about connecting sensitive systems together, the underlying risk is quite clear: wide sharing of valuable data leads to more exposure of that data. If we examine this issue more closely, however, we see that data sharing using Web services leads to new risks that are not so obvious.

Where's the Perimeter?

In the world of client/server and operating system security, we have the concept of a *trusted computing base* (TCB). The TCB consists of the hardware and software mechanisms that are responsible for enforcing the security policy, which defines when a user may access a resource. The TCB creates a *security perimeter* – valuable resources are protected within the security perimeter, and users must authenticate and pass authorization checks before they are allowed to access data within the perimeter.

In this model, the security perimeter is easy to understand. Bad guys and good guys are outside of the security perimeter, and the TCB distinguishes between the two parties so that only the good guys can access the data within the perimeter. The whole point of a security perimeter is to make sure the bad guys do not get inside.

Figure 1 shows the typical example of this model when applied to Internet connectivity: the IP firewall.

For Web servers, IP firewalls accept or reject HTTP traffic based on corporate policy. It's common, for example, to establish a security perimeter by constraining HTTP traffic to a demilitarized zone (DMZ) in the enterprise and providing an access-control policy for protected resources. IP firewalls usually prohibit external HTTP traffic from entering the internal corporate network.

When we transmit Web services over the IP firewall the simple model of a security perimeter falls apart. XML and SOAP traffic most commonly travel over HTTP, which can be controlled by the IP firewall. However, unlike browser-based Web server traffic, XML and SOAP messages do need to get through to enterprise servers within the corporate network. If IP firewalls are configured to permit Web services traffic within the corporate network, then we have lost our perimeter. IP firewalls cannot distinguish between the trustworthy and nontrustworthy Web services traffic tunneled via HTTP. As a result, attackers could bypass firewall checks

and gain access to sensitive enterprise data.

Web services don't have a clear definition of a security perimeter. Because Web services architectures are built from many different service providers distributed across different enterprises, there is no simple way to distinguish the good guys from the bad guys. It's difficult to tell who is trustworthy and who is not.

Consequently, Web services should be created as *mutually suspicious islands*. Every time a client uses a Web service, the client and Web service establish a trust relationship. Security is enforced at each link of the Web services chain rather than at a single security perimeter. The resulting architecture defines a layered security model where each Web service has its own layer of protection.

Next, we'll examine the security facets that are the basis of trust in the Web services world.

Security Requirements and Related Risks

Web services applications have the same basic information security requirements as any other computer system:

- ***Confidentiality:*** Safeguard user privacy and prevent the theft of enterprise information, both stored and in transit
- ***Integrity:*** Ensure that electronic transactions and data resources are not tampered with at any point, either accidentally or maliciously
- ***Accountability:*** Detect attacks in progress or trace any damage from successful attacks (security auditing and intrusion detection). Prevent system users from later denying completed transactions (nonrepudiation)
- ***Availability:*** Ensure uninterrupted service to authorized users. Service interruptions can be either accidental or maliciously caused by DoS attacks

When deploying any Web services application, you need to consider how to address each of these information security requirements. As you identify security mechanisms that will address the requirements for your Web services, you will encounter common areas of security risk, shown below. Since a Web services architecture defines service interfaces and associated messages sent to those services, the risk areas naturally fall into message, services, and interoperability-related risks:

- ***Message-related risks:*** Caused by potential compromises of the message content transmitted by a Web service. The risks include damaging ill-formed messages, message modification and eavesdropping in transit, and messages sent from nontrustworthy sources.
- ***Services-related risks:*** Caused by potential compromises of the Web service interface. The risks include access to the service by unauthenticated users, access of the service by unauthorized users, and unaudited use of services.
- ***Interoperability risks:*** Caused by potential mismatches of Web services implementations. The risks include vulnerabilities from lack of interoperability with currently deployed security products, and insecure application-to-application message exchange.

As you can see, there are plenty of risks to worry about when deploying Web services

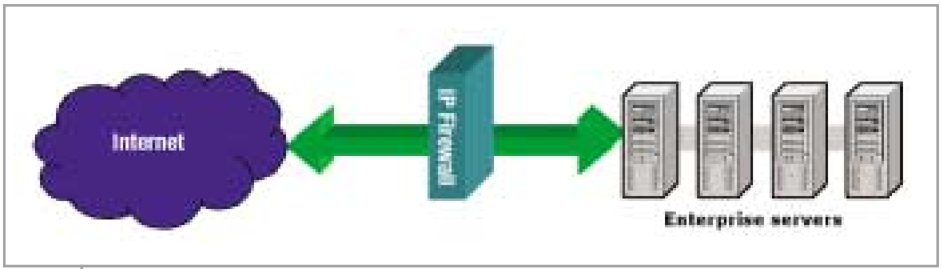


FIGURE 1 **Security perimeter applied to IP firewall**

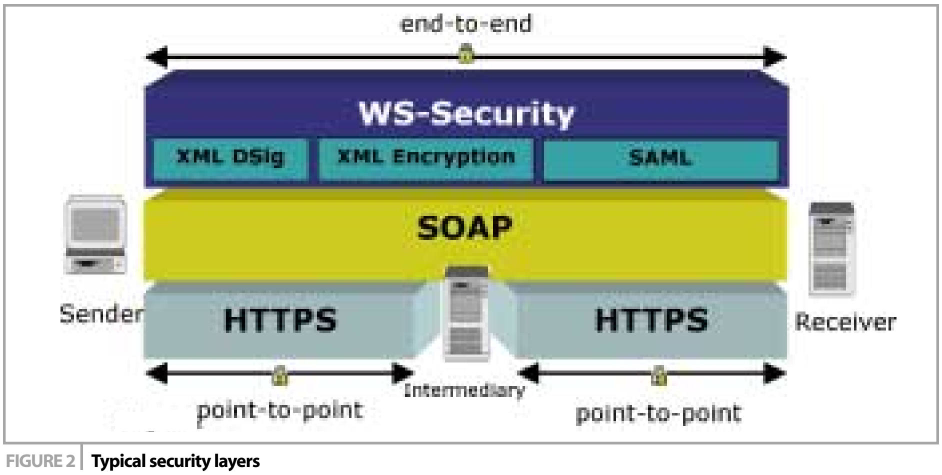


FIGURE 2 **Typical security layers**

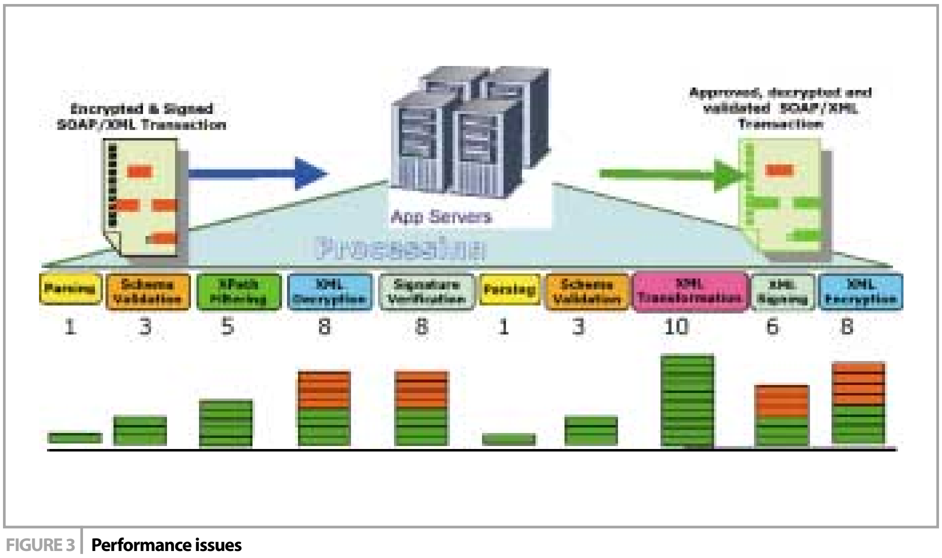


FIGURE 3 **Performance issues**

applications. Although the underlying security requirements are not new, the lack of a distinct security perimeter and the list of related security risks have resulted in the creation of new technologies that address the specific needs of Web services security. We'll discuss these technologies next.

The Technologies

Standards groups, particularly the Organization for the Advancement of Structured Information Standards (OASIS), the World Wide Web Consortium (W3C), and the Internet Engineering Task Force (IETF) have been very active over the past few years on security-related topics. Several specifications have been defined that together serve as the basis for Web services security solutions.

Security Building Blocks

The security of Web services relies on three key standards: Secure Sockets Layer (SSL)/Transport Layer Security (TLS), XML Digital Signature, and XML Encryption. SSL/TLS provides public key–based transport layer security that enforces confidentiality and integrity of message content. SSL/TLS is especially pervasive for Internet-based HTTP traffic, including Web services messages. XML Digital Signature specifies how to sign XML document elements to prevent tampering, while XML Encryption specifies how to encrypt elements to prevent disclosure.

The combination of SSL/TLS, XML Digital Signature, and XML Encryption go a long way toward addressing many Web services security requirements. Why aren't they enough? Although these standards are flexible and powerful, they are often too low level for convenient use. In particular, there are so many ways to enforce security with these standards that different uses may not easily interoperate without additional guidelines.

Upper-Layer Security

Built on the security building blocks we just described, the upper-layer security standards provide Web services security support in a more convenient package:

- ***WS-Security:*** Provides packaging for SOAP messages – it defines how to attach signature, encryption, and security tokens to SOAP messages.
- ***Security Assertion Markup Language (SAML):*** Defines the XML format for exchanging authentication, authorization, and attribute assertions; and includes the protocol for requesting this information from security authorities.
- ***Extensible Access Control Markup Language (XACML):*** XML specification for expressing access control policies.
- ***XML Key Management Specification (XKMS):*** Defines the interface for accessing keys and PKI servers using XML.

The Challenges

There are several technical challenges when assembling these technologies together to secure a set of Web services. In this section, we focus on three of the most significant: interoperability, performance, and preventing XML denial of service (XDoS).

Putting Together an Interoperable Security Solution

At this point you may have the impression that there are a lot of standards and technologies involved in Web services security, and that is certainly true. The main technical problem is not whether the appropriate Web services security mechanism is available, because it probably already is. Instead, the issue facing both vendors and customers is how to assemble an effective, simple, and interoperable solution out of the many available building blocks.

> "The traditional concept of a single security perimeter does not apply to the loosely coupled concept of Web services"

Figure 2 shows a typical layering of security mechanisms and protocols in support of a Web services deployment. As shown in the figure, the Web services sender and receiver are connected via an intermediary server, which could be providing infrastructure services such as message routing, or business-level services such as an e-commerce marketplace. We can see that HTTPS (HTTP over SSL/TLS) secures the point-to-point connections from sender to intermediary, and intermediary to receiver. Because SSL/TLS is transport-level security, it cannot by itself provide end-to-end security between sender and receiver. For example, SSL/TLS is inadequate if the sender needs to send encrypted data (say, a credit card number) that is not exposed to the intermediary. In this case the sender could use XML encryption and WS-Security to tunnel encrypted data through the intermediary to reach the receiver.

The Web Services Interoperability Organization (WS-I) is chartered to promote interoperability, and is working on providing guidelines for interoperability of Web services security. In particular, the WS-I Basic Security Profile Working Group is defining security scenarios and a security profile document to help enterprises piece together many of these standards to ensure interoperability.

Performance

Performance is often the most significant challenge when deploying Web services security solutions. We'll use Figure 3 to illustrate some of the performance issues you are likely to encounter. This figure offers an example of a typical set of processing steps that might be required to secure a SOAP/XML transaction. As the transaction is processed, it is parsed, its schema is validated, XPath filtering is applied, it is decrypted, and so on, until the transaction processing is completed.

The first thing to note about the example is the sheer number of security processing steps required. For Web services with security requirements that span the

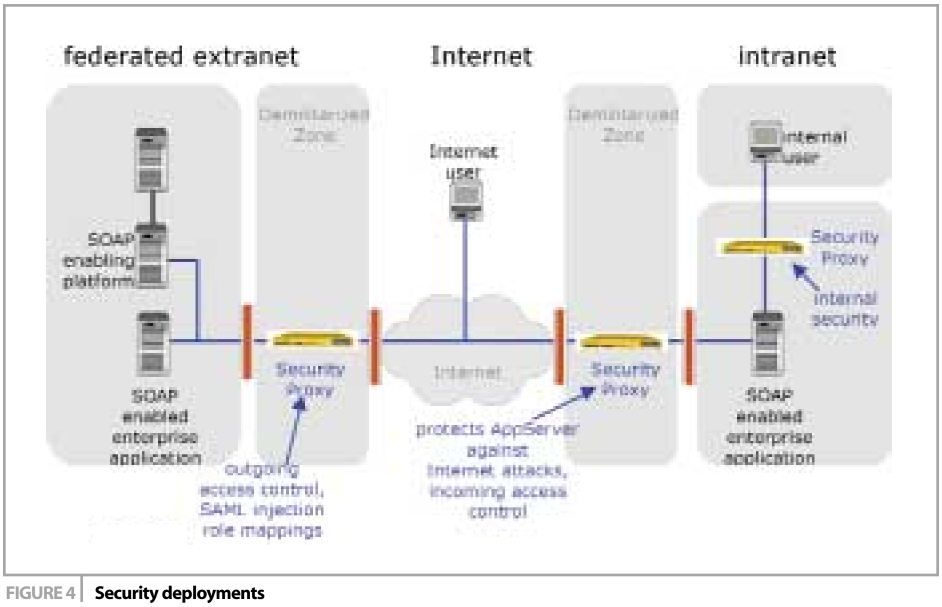

federated extranet
Internet
intranet
Demilitarized Zone
Demilitarized Zone
SOAP enabling platform
SOAP enabled enterprise application
Security Proxy
outgoing access control, SAML injection role mappings
Internet user
Internet
protects AppServer against Internet attacks, incoming access control
Security Proxy
internal user
Security Proxy
internal security
SOAP enabled enterprise application

FIGURE 4 **Security deployments**

areas of confidentiality, integrity, accountability, and availability, this number of steps is not unusual.

In Figure 3 the numbers and colored bars show the relative computational cost of each processing step. The green bars represent the time spent in XML-related computations, such as parsing, schema validation, and transformation. The red bars represent the time spent in cryptographic-related computations, such as public key encryption and decryption. To give you an idea of the latency involved in one of these steps, an XML signature of a modest-size message using typical software-based implementations can take well over 100 milliseconds.

Although you might expect cryptographic processing to dominate Web services security processing, the example shows this is typically not true. Because XML is so expensive to analyze and transform, Web services security enforcement generally spends much more time performing XML processing than cryptographic processing.

As a consequence, it's important to realize that performance optimization for Web services security requires addressing both cryptography and XML acceleration. SSL and cryptographic hardware accelerators help speed up Web services security processing, but unless you also consider acceleration of XML processing you are unlikely to see significant performance improvement.

Finally, you should consider the additional security performance overhead due to the stateless nature of Web services. Browser-based Web server security can take advantage of a security session to optimize security. When accessing a Web server, the initial authentication step (for example, using a password or digital certificate) may be slow, but it happens only once per session, so the user experiences minimal delay. In contrast, every Web service transaction must be authenticated, so this performance penalty is paid on every transaction. The end result is a potentially lengthy fixed authentication delay built into the response time of every Web service transaction.

XML Denial of Service (XDoS)

One challenge that enterprises are only beginning to consider is XDoS. This concept may be new to most people, but experts agree that protecting against XDoS attacks will be a common issue as more Web services are exposed on the Internet.

Years of experience have shown that Web servers must be protected against DoS attacks. There are many variants on these attacks, such as the virus infections of huge numbers of consumer PCs with Trojan horse programs, which are then launched remotely at a later time to flood a target Web server host and bring it down. IP firewalls commonly contain countermeasures against these attacks by limiting the rate of traffic from an IP address and detecting hostile patterns of incoming messages.

The analogous XDoS attacks in the Web services scenario will be more serious. Because of the resources required to process XML, it is much easier for the attacker to create and transmit malicious XML than it is for the defender to process and reject the XML. XML supports rich and complex document structures, including recursion, which can potentially cause infinite loops during input processing. While IP-based DoS attacks usually require large numbers of messages, an XDoS attack can be launched on a Web service with a single 2KB malformed XML message.

Countermeasures for XDoS attacks are straightforward to define. For example, incoming XML should be schema validated to ensure that the XML conforms to a supported Web service interface. Message monitors may be installed to check thresholds on message rates, and content-based filters can be used to detect recursion depth and other complexity measures of the XML document.

Although XDoS countermeasures may be clear, their implementation is definitely a challenge. The only way to effectively deal with XDoS is to have a high-speed XML engine that can detect and dispose of these attacks before the server is overwhelmed. From our previous discussion, we have already seen that performance is a major challenge when deploying Web service security; the burden of XDoS detection compounds this problem. As in the case of IP firewalls that handle DoS attacks, it is sensible to consider an engine that offloads the Web service server from needing to handle this processing.

The Solutions

Web services security can be enforced in a variety of places in the architecture. In this section, I make some recommendations on the best approach to use. I discuss using an XML security gateway as the first level of defense, and then adding Web services application–based security as a second level of defense as needed.

First Level of Defense: XML Security Gateway

Based on the challenges I described previously, I recommend starting with a hardware-based XML security gateway as the simplest and most effective way to enforce Web services security. An XML security gateway is typically deployed behind an existing IP firewall, and secures all XML traffic before it reaches the Web service on the application server. A hardware-based XML security gateway has many advantages, including:

- ***Performance:*** An optimized hardware solution that addresses both XML acceleration and cryptographic processing will improve latency and throughput of XML security processing by a factor of at least 10 over software implementations. In many cases the performance improvements are considerably larger.
- ***Scalability:*** By deploying a high-capacity XML security gateway, the number of application server platforms may be significantly reduced. An XML security gateway can handle an increased Web services transaction load without needing to add additional application servers.
- ***Manageability:*** By channeling all Web services traffic through a small number of high-capacity gateways, the number of security enforcement points is reduced. This simplifies the security configuration and makes changes easier to manage.
- ***Simplicity:*** An XML security gateway can enforce the majority of Web services security requirements, thus avoiding the need to write security code within the Web service applications.
- ***Security:*** Removing security from applications is a best practice, and improves the security assurance of the architecture. As in the case of an IP firewall, an XML security gateway is a hardened security platform that protects potentially vulnerable application servers.
- ***Availability:*** XDoS is a significant threat to Web service availability. An XML security gateway provides high-performance XDoS checking to protect Web services applications.
- ***Interoperability:*** Web services security standards and technologies are a moving target, and will continue to evolve. An XML security gateway is a natural place in the architecture to translate across multiple transports and security standards.
- ***Monitoring:*** Because Web services traffic passes through the gateway, it provides an effective central enforcement point for audit logging and accountability.

Second Level of Defense: Web Services Application

Although the first level of defense for Web services belongs on an XML security gateway, there are important cases where it makes sense to have a second level of defense on the Web services application platform.

Both J2EE and .NET application server platforms have their own container-based security models. In existing component deployments, the application server security policy may be an important part of protecting the application. Legacy applications may also have business-specific security embedded with in. In these cases, it is possible to integrate the security enforced at the XML security gateway with application server security. In particular, security context information that is authenticated at the XML security gateway and based on Web services standards such as WS-Security and SAML may be used to enforce authorization and audit policies on the application server.

Deployment

The most common deployment of an XML security gateway is as a proxy within the enterprise DMZ (see Figure 4). In this configuration, the XML security gateway protects the application server against Internet-based XDoS attacks and enforces incoming access control, including authentication and authorization. The XML security gateway may also be deployed as a proxy to protect access within the corporate intranet.

For a more advanced deployment, the XML security gateway may be installed on the Web services client side. In this scenario, the gateway provides outgoing access control, limiting the transmission of sensitive data to the Internet. The gateway can also be used to secure a federated extranet, where the Web services client and server environments do not share common security policies and mechanisms. To address federated extranet security, the XML security gateways can use SAML as a common standardized security token to map client-side security policy to server-side security policy.

Conclusion

This article described many of the issues that need to be considered when deploying an enterprise Web services architecture. The traditional concept of a single security perimeter does not apply to the loosely coupled concept of Web services. Instead, I advocate viewing Web service applications as mutually suspicious islands that need to establish trust before communicating to a partner application. Mutual suspicion means that there is no central point of trust in the architecture. XML security gateways protect each Web service application and establish trust; in this manner, a security architecture is defined by the network of XML security gateways and application servers.

I described a number of security challenges when assembling a Web services security solution. The first is interoperability; the existing standards are complex and still evolving, so it's difficult to ensure that your Web services security implementation will interoperate with your partner's. Future guidelines from groups like the WS-I Basic Security Profile Working Group will help you through this process. In the meantime, an XML security gateway can serve as a translation point between incompatible security technologies, and evolve as your requirements change.

The other major challenge of Web services security is performance. I described the close relationship between security and performance, which is due to the large processing burden of XML as well as the additional processing load caused by XDoS attacks. An XML security gateway has the processing capacity to handle XML security traffic quickly and efficiently. ⓔ

About the Author

Bret Hartman is a nationally recognized expert on distributed systems security; and he is a regular speaker and panelist on a variety of secure distributed system topics. He is a co-author of Mastering Web Services Security (Wiley). Early in his career, Bret was an officer in the USAF and worked at the National Security Agency on the creation of the "DoD Trusted Computer System Evaluation Criteria" (Orange Book), the original standard for building secure information systems.

■■■ bhartman@datapower.com

AmberPoint Express from AmberPoint, Inc.

Something to keep on hand

Not surprisingly, Web services management tools are quickly appearing to assist developers and system administrators alike with the maintenance of service-based applications. One such product is AmberPoint Express, a free Web services management and monitoring tool whose mission is to provide developers with the ability to "…incrementally measure, debug and fine-tune the performance and functionality of their Web services…" The product is currently available in three flavors: .NET, WebSphere, and Apache Tomcat with Axis. Versions tailored to other major application servers will be released in the future.

WRITTEN BY
BRIAN BARBASH

AmberPoint Express is targeted to developers and provides the functionality for monitoring and debugging Web services. However, all of AmberPoint's products are based on a distributed architecture geared to managing and monitoring Web services applications. The key components of their architecture are:

- ***Agents:*** Deployed at the interface between Web services and client applications. Serve as the tools of instrumentation for applications.
- ***Analytical servers:*** Serve as data aggregators across multiple agents while maintaining historical data.
- ***User interface/desktop console:*** A set of desktop and Web-based user interfaces that provide views to the information collected and aggregated by the analytical servers.

How It Works

When AmberPoint Express is installed, instrumentation agents that monitor traffic against the deployed services are plugged into the application server. For the .NET release, agents are installed directly into IIS, while the Tomcat version is shipped with a preconfigured release of Tomcat 4.0.6 (Express may also be installed on an existing instance of Tomcat). An additional feature of the .NET release is direct integration within Microsoft's Visual Studio .NET IDE, providing access to the debugging functionality.

Using AmberPoint Express

The main user interface to AmberPoint Express is the Web-based console seen in Figure 1. The left side of the console shows the Web services that are deployed on the current server, each of which is expandable to show its operations. Services that are currently being monitored show a small bar-graph icon that when hovered over shows the total number of messages and faults consumed by that service. The right-hand portion of the console contains a very nice Flash MX application that provides real-time status information about the currently selected service. When the mouse is moved over the main graph, a callout shows the details of the service events within the slice of time under the mouse pointer. The information includes the total messages, the number of faults, and the total processing time.

The lower section of the information area shows summary statistics for the currently selected Web service within the context of the overall time interval. Figure 1 is an example of the five-minute time interval. Summary information is presented for the entire five minutes, including the number of successful messages, the number of faults, average response time, the average number of successful messages transferred, and the maximum number of messages transferred. Additionally, summary information is presented for the five-second time slice locked on the graph. A time slice may be locked by clicking the mouse within the graph at any data point. If viewing the graph using the one-hour interval, the same information is presented; however, each individual time slice is one minute in duration.

Data from this view may also be exported to Microsoft Excel for further analysis. By clicking the Export Data link, an Excel spreadsheet is generated on demand and downloaded in a separate browser window. The data exported includes the start and

Company Info

AmberPoint, Inc
155 Grand Avenue, Suite 404
Oakland, CA 94612
Phone: 510-663-6300
E-mail: info@amberpoint.com
Sales: sales@amberpoint.com

Download Information

http://express.amberpoint.com/?wsj=free

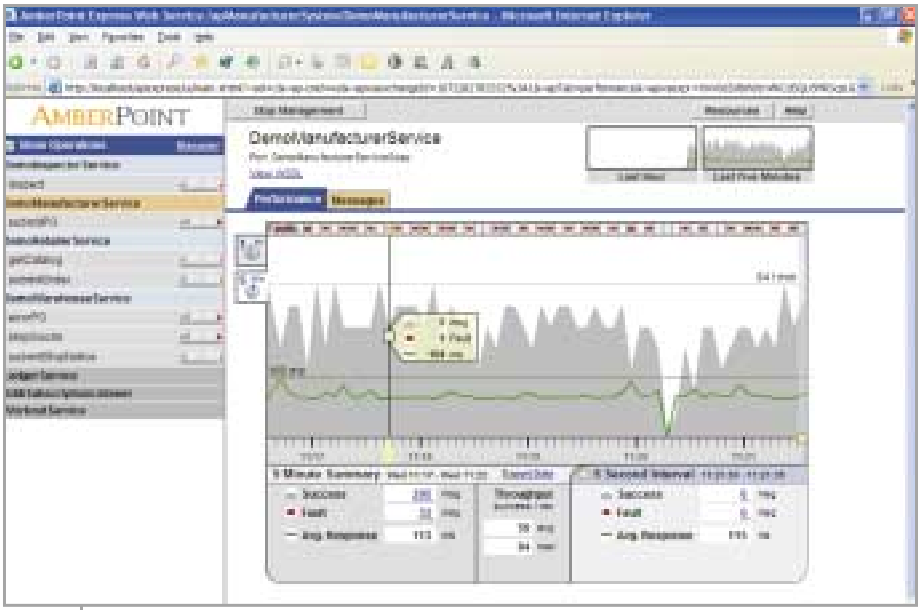


FIGURE 1 AmberPoint Express Web interface

end times of each time slice in the graph, the number of messages and faults processed, and the average and maximum response times for the operation or service.

Debugging Tools

AmberPoint Express provides several tools to assist developers with debugging their Web services. The Messages tab provides access to the collection of SOAP messages transferred to and from a particular service. This listing includes a rolling collection of 20MB worth of request and response service data that may be viewed in a hierarchical structure or its raw XML form. The Messages tab also provides a powerful query mechanism to easily filter out unwanted information. The query language itself presented in plain English, is intuitive, and is easy to apply.

Developers may also send test messages to any monitored Web service using the AmberPoint Express GUI. By selecting an individual operation on a service and invoking the Send Message operation, a new message is created and presented to the developer. At this point, the data to be sent may be edited, along with the number of messages to send and the address to send the message to. For example, if a production message fails, the console can be used to create a test message from the production service, which can in turn be directed at a test server, eliminating the risk of modifying production data.

Working in Visual Studio

As I mentioned earlier, AmberPoint Express is integrated into Visual Studio (the Whidbey release of Visual Studio will include AmberPoint Express). Projects may be added or removed from the set of managed applications through a new item on the project's Properties page. Once a project is built, its status is updated in the main AmberPoint Web console.

In addition to the managed property, a new item is placed in the Tools menu of Visual Studio, providing quick access to the View Messages and Send Test Message areas of the AmberPoint console. These items are available for any item in a project that is being managed by AmberPoint.

Summary

AmberPoint Express is an easy-to-use, well-designed management and debugging system for Web services that keeps the needs of the developer in mind. Its Web console is thoughtfully designed and intuitive, and the tool is architected in such a way that it may be added to a project at any phase. Overall, this is an excellent utility for Web services developers to have on hand for their projects.

About the Author

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a consultant for the Consulting Group of Computer Sciences Corporation, where he specializes in application architecture and development, business and technical analysis, and Web design.

■■■ bbarbash@sys-con.com

–Continued from page 11

responsible for serving the VeriSign Web site even though a different machine may have been in use a few days or even a few hours earlier.

Arthur C. Clarke once wrote that any technology that is sufficiently advanced should be indistinguishable from magic. This same rule applies to the Internet and Web services. Ten years ago the magical feature of the Web was the fact that you didn't need to think about how you were getting the information you wanted from the Internet, you just pointed, clicked, and let the machine work out the details. WS-Policy allows that same principle to be applied to management of Web services.

About the Author

Dr. Phillip Hallam-Baker is principal scientist and Web services architect for VeriSign, Inc. In those capacities, he is responsible for driving and delivering key security specifications and technologies through industry recognized standards bodies and other organizations. Phillip has co-authored numerous security specifications that have been widely embraced in high technology.

■■■ pbaker@verisign.com

Enterprise Web Services Security: A Reference Architecture, part II

Focus on design and functionality

Last month (*WSJ,* Vol. 4, issue 2), we looked at how Web services should not depend on specific security environments and rules but should be managed as part of all of an enterprise's corporate data assets such as Web applications, ERP systems, and in-house applications.

WRITTEN BY
PRATEEK MISHRA &

MARC CHANLIAU

We recommended that Web services security be integrated with the overall enterprise security infrastructure at the very beginning of the Web services deployment phase. This month, we'll look at some of those possible deployment models.

Deployment Models

There are four deployment models based on the guidelines presented in our earlier article.

Terminology

The terms used in the deployment models are defined as follows:

- ***Reverse proxy server:*** Intermediary server (e.g., a Web server) configured to filter requests coming from Internet users into the enterprise, providing security, management, and caching capabilities.
- ***User repository (or user store):*** A persistent data store that maintains user information. A user repository can be implemented in LDAP, RDBMS, Microsoft's Active Directory, and mainframe applications. Information related to a single user may be maintained in multiple, separate user stores, each of which needs to be queried for authentication and authorization purposes.
- ***Web Services Management Point (WSMP) :*** Enforcement point for implementing a Web service management policy
- ***Web Services Security Enforcement (WSSE):*** Enforcement point for implementing a Web service security policy

Simple Proxy Deployment

Access to enterprise resources is achieved via a reverse proxy server. The first line of defense is the network firewall, which filters requests to the reverse proxy server (see Figure 1).

The request for a Web service is submitted using a SOAP message that can be sent over a variety of transport protocols (HyperText Transport Protocol [HTTP], Simple Mail Transport Protocol [SMTP], File Transfer Protocol [FTP], Java Message Service [JMS], and other message queue [MQ] services). The sender's identity is expressed in transport protocol headers or in the SOAP document submitted for the request. WSSE authenticates against the submitted credentials, binds the message to an identity, and, if authorized, grants access to the Web service.

The outer network firewall ensures that resources cannot be accessed externally. The inner network firewall ensures that the enterprise resources (including the IAM security policies and the user repositories) are protected against internal attacks.

Low-level access control methods are used to ensure that only the reverse proxy server can forward requests to back-end resources. This means that back-end resources (.NET, J2EE, and legacy) require additional container-level security configuration outside the IAM policy model.

WSSE enforces security across all types of transport. It supports both inbound and outbound flows (i.e., requests received by the enterprise from a third party and requests sent by the enterprise to a third-party). WSSE communicates with the IAM security policy server for security policy decisions.

This deployment template does not include many moving parts and does not require a complex and costly implementation. On the other hand, it does not scale very well because the container of each back-end resource needs its own access management layer.

Full IAM Deployment

All of the enterprise resources are protected by a single IAM system. WSSE points for J2EE and .NET containers may be deployed in two ways:

- ***Interceptors (or "agents"):*** The agent can be integrated with a variety of Web services containers. The agent interacts with the IAM policies to provide security services.
- ***SOAP message handlers:*** Use either the Java API for XML-based remote procedure calls (JAX-RPC) or .NET pipelines. SOAP message handlers have the benefit of being transport independent.

Legacy applications continue to use proprietary security but may synchronize with the enterprise user repositories. SAML may be used as a means of communicating with "opaque" containers that cannot accommodate WSSE points, such as legacy applications or more proprietary application servers (see Figure 2).

Once a Web service requester has successfully been identified, authenticated, and authorized, the IAM platform provides the ability to leverage the user identity to personalize the behavior of the Web service.

The Web service can be bound to aspects of an identity through user entitlement information passed to the Web service by the IAM platform. For example, when a user has successfully been authenticated and authorized to access a Web service, the IAM platform can identify which entitlements should be obtained about that user, retrieve them from the user store(s), and associate them with the Web service request by binding them to the XML message.

This eliminates the need for a Web service to keep its own entitlement database and handle the retrieval of entitlements in the application logic. Thanks to the IAM platform, these entitlements can be centrally and securely managed and associated directly with a user's identity.

In some variations of this deployment model, the reverse proxy is optional.

IAM + WSM Deployment

The SOAP message is first received by the WSM enforcement point (WSMP). The WSMP makes a call to the WSSE point to ensure that the SOAP flow is secured.

WSSE authenticates the requester against IAM policies and returns security information to the WSMP. The WSMP can then enforce the WSM policy once the SOAP message is bound to a Web service consumer's identity.

The WSM platform can integrate with the IAM platform in two ways:

- The WSM platform can explicitly invoke the WSSE through the IAM application programming interface (API)
- The WSM platform can use the WSSE agent or the message handler model described above.

In this deployment model, the WSM platform takes advantage of the IAM platform, which links it to the enterprise-wide IAM infrastructure (see Figure 3).

This deployment model allows the enterprise to leverage a corporate infrastructure for security (the IAM platform) and Web services management (the WSM platform). Both provide a layer of abstraction that relieves Web services developers from security and management tasks so that they can concentrate on the design of the Web service business functions.

Network Appliance Deployment

This is the full-blown integration including a network appliance. The network appliance is added to provide wire-speed XML processing (see Figure 4).

In this deployment model, the network appliance delivers network security services combined with an authentication service, as described last month (see *Protection and Threat Prevention Layer*).

Typically, the network appliance interacts with Web services flows by intercepting the incoming request as soon as it passes through the network firewall. The network appliance decrypts the request, parses the XML document, validates the document against an XML Schema, and applies transformations if required.

The network appliance may provide authentication against a user store config-

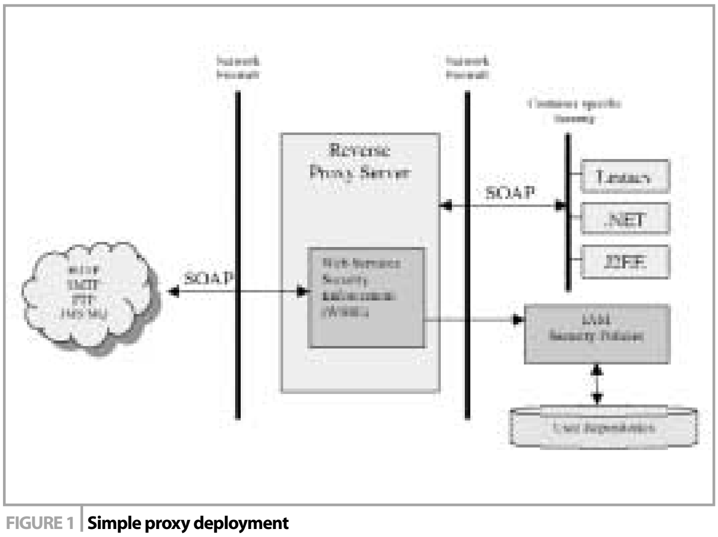


FIGURE 1 **Simple proxy deployment**

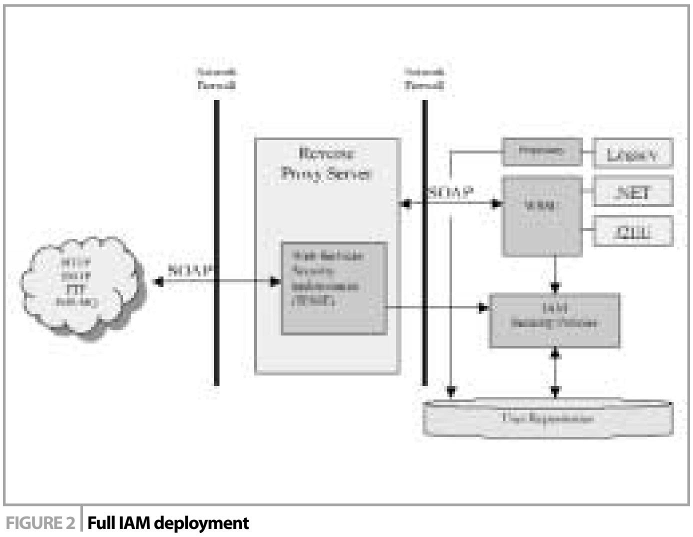


FIGURE 2 **Full IAM deployment**

ured with the IAM platform. The result of authentication is then communicated downstream, using SAML for example. The WSSE and WSM enforcement points can also benefit from authentication information provided by the network appliance.

Once the Web service requester is authenticated, the IAM platform can move to grant access to the Web service, and the WSM system can apply business-level management policies.

Conclusion

With a focus on security and management, a Web Services Reference Architecture can help the prevention (network security), enablement (identity and access management), and enforcement (Web services management) layers of a Web services architecture fit together.

Web services providers can focus on the design and functionality of the Web services they expose to their employees, customers, or partners, while relying on enterprise-wide security and management services that increase overall availability, scalability, interoperability, and manageability.

References

- www.w3.org/XML: The W3C's XML 1.0 Second Edition Recommendation describes the Extensible Markup Language (XML), "the universal format for structured documents and data on the Web."
- www.w3.org/Security: The W3C's security-resources home page includes many links to various aspects of Web and Internet security (cryptography, authentication, authorization, etc.).
- http://searchwebservices.techtarget.com: Web services–specific information resource for enterprise IT professionals Includes useful articles and technical notes covering all aspects of Web services, in particular security.
- www.w3.org/DSig/Overview.html: The W3C's XML Signature (XML-DSIG) Recommendation describes digital signatures as applied to XML documents.
- www.w3.org/Encryption/2001: The W3C's XML Encryption Recommendation defines a process for encrypting and decrypting XML documents.
- www.w3.org/TR/xkms: The W3C's XML Key Information Service Specification (XKMS) Recommendation defines protocols for distributing and registering public keys, used together with XML Signature and XML Encryption.
- http://xml.coverpages.org/xrml.html: The Extensible Rights Markup Language (XrML) home page.
- www.oasis-open.org/committees/security: The OASIS specification for the Security Assertion Markup Language (SAML) defines an XML-based security framework for exchanging authentication and authorization information.
- www.w3.org/TR/SOAP: The W3C's SOAP Recommendation (v1.1) describes the Simple Object Access Protocol, designed as a messaging framework for exchanging XML documents between peers in a platform-neutral environment.
- www.oasis-open.org/committees/ wss/documents/WSS-Core-08-1212-merged.pdf: Working draft of the Web Services Security core specification.
- www.w3.org/TR/wsdl: The W3C's WSDL submission specifies the Web Services Description Language, a framework providing definitions for network services and the automation of application communication through XML documents.
- www.uddi.org/about.html: The Universal Description, Discovery, and Integration (UDDI) project is an industry initiative designed to create an XML framework for describing Web services providers and a description of the services they provide.

About the Authors

Prateek Mishra, PhD, has more than 10 years experience with enterprise-class distributed systems. He is director of technology at Netegrity and works on strategy and standards. He was an editor of the SAML 1.0 specification and is co-chair of the SSTC (SAML) Committee and participates in the WSS (WS-Security) Committee.

■■■ pmishra@netegrity.com

Marc Chanliau has been in the software industry for more than 20 years and is currently the product manager for Netegrity where he is responsible for the company's XML technologies. Chanliau is heavily involved in security and XML standards groups including serving as the first chair person of the OASIS Security Services Technical Committee (SSTC), which culminated in the adoption of SAML as an official OASIS standard, participating on the WS-Security Technical Committee, helping to define the Liberty Alliance 2.0 specifications, and participating in the Java Specification Request (JSR) committee.

■■■ mchanliau@netegrity.com

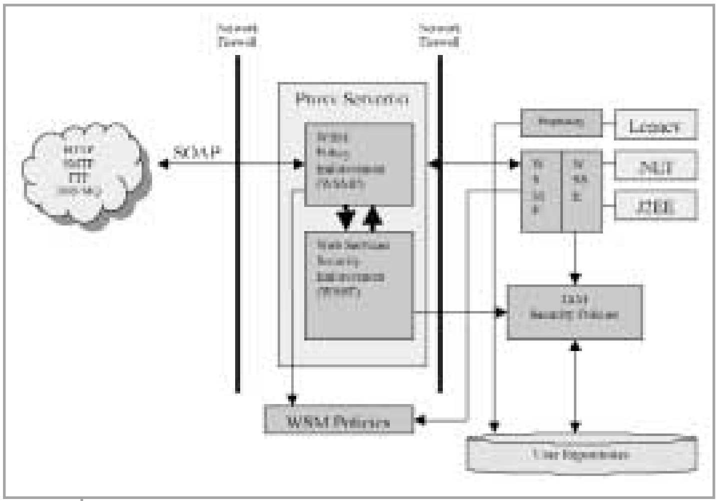


FIGURE 3 IAM + WSM deployment

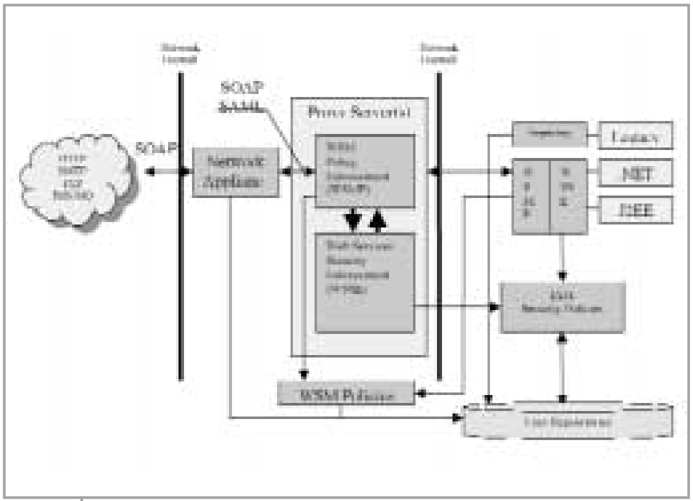


FIGURE 4 Security appliance as firewall extension

A Virtual Solution to Real Identity Issues

Repairing the fractures – virtually

To quote the Scarecrow from the Wizard of Oz, "There are pieces of me here. There are pieces of me there."

WRITTEN BY
CLAYTON DONLEY

Thanks to years of independent evolution, user identity information also exists with bits and pieces in different places. This presents a challenge to application developers responsible for writing software that needs to take into account potential access from people across the enterprise who may be in any number of separate identity sources. It also presents a security challenge as allowing access to one application may open doors to others that are best kept shut.

Metadirectories like IBM's Directory Integrator (IDI) and Microsoft's Identity Information Server (MIIS) solve some identity problems by consolidating data from these multiple repositories into a new repository that contains the full picture. Consolidation is important because it reduces the management effort to maintain and improve the quality of attributes that exists for the same individuals across multiple enterprise data sources. Consolidation through the use of a metadirectory can be extremely powerful, but as those who have walked the yellow brick road to metadirectory know, consolidation brings new challenges.

One is data latency. Because they're drawing from other sources, metadirectories need to receive updates from the source directories on a regular basis. Often some of that data can be very old when dealing with batch export jobs that run at night. In some cases that may be acceptable. But what if you're looking at access rights to the network? A terminated employee may have his/her identity removed from the source databases. If it's left in the metadirectory until the batch run, however, that employee could have access to the network the entire day. That's a huge security risk, especially if the employee was terminated suddenly or under very negative circumstances.

Another concern is data ownership. Many large organizations use Web services to create portals for suppliers or employees. Those portals may pull data from a variety of sources. Suppose a portion of the data comes from HR, giving employees the ability to check on their 401(k), number of vacation days left, health benefits, and so on. If it's sitting in a metadirectory controlled by IT, the HR department loses a portion of its control over the data, and the organization is vulnerable to potential liabilities. Should a problem arise, such as confidential information about salary structures leaking out, it could spell disaster and/or lawsuits.

Another example would be regulated industries such as health care service providers, where a given user may be both an employee and a subscriber to the benefits. Both populations could have access to the same application(s); however, strict guidelines or laws mandate that subscriber data is contained in separate physical data stores.

Rather than being the wizard behind the curtain, virtual directories work to present data to applications directly. They are designed as middleware that takes requests using standard protocols like LDAP. They then rewrite and route the request in real time to one or more directories, databases, or other sources that contains the information necessary to fulfill the request. Once the operation is fulfilled, they simply dissolve like the Wicked Witch of the West when the water is thrown on her.

As middleware, rather than behind-the-scenes infrastructure, virtual directories eliminate the need to synchronize identity information to a central place. The application always works with the most current information because it's drawing from the source directory and not a copy of the information. Eliminating the need for replication and hard storage also assures that the data remains under the control of the original owners and that it complies with regulations that ensure data privacy. In the previous example, when the employee accesses the HR portal, the data is drawn and presented to that employee. When the employee is finished, the access point is closed and the data is again protected by HR until the next authorized query.

Another advantage is that virtual directories have the ability to present the same source information differently to different applications in much the same way that a database administrator can create multiple views of the same database tables. As a result, drawing and routing the information for new applications is greatly simplified. Finally, rather than a nine-month infrastructure project that could delay production rollout of portals and other key applications, virtual directories tend to have deployment cycles measured in days due to their non-invasive nature.

While fast, non-invasive deployment is usually great, there are places where metadirectories are still the right choice. For example, they are great for keeping key infrastructures such as NOS and e-mail in synch. These are special-purpose enterprise directories that need to be kept up-to-date with their own proprietary and application-specific data. The key is to determine the requirements of the job and its limitations, and then select the directory option that best fits the parameters.

The fractured nature of user identity information is a fact of life. Yet it doesn't have to be a barrier to accomplishing what needs to be done in the enterprise. Virtual directories provide Web services developers with the ability to take all the individual pieces of straw and rebuild the Scarecrow in new, more interesting, and more secure ways – all while speeding the development cycle. That alone makes them worth a look.

About the Author

Clayton Donley is founder and chief technical officer of OctetString, whose Virtual Directory Engine and other products allow organizations to manage user identification quickly and seamlessly. He is an internationally recognized authority on identity management, and has served as a consultant on numerous high-visibility projects and as an author on the topic.

■■■ clayton.donley@octetstring.com.

For more detailed information, please visit:
www.lighthouseseminars.com or call **Joe Richard** at **781.821.6734**

Learn about successful implementation of content technology through the unique combination of

- Case Studies
- Project Management Techniques
- New Technology Trends

Gold Sponsors

GMC · context media · VIGNETTE

Media Sponsors

cms WATCH · EContent · The Gilbane Report · InfoWorld GET TECHNOLOGY RIGHT · WebServices · TRANSFORM · XML JOURNAL

Analyst Sponsor

IDC *Analyze the Future*

Association Sponsors

IDEAlliance · G-SAM · OASIS · OSCOM OPEN SOURCE CONTENT MANAGEMENT · WoW · XML

Register NOW!

Register now for the **CONFERENCE PLUS PACKAGE** and receive a **FREE iPod!**

LIMITED QUANTITY, REGISTER EARLY!

Get more information on this offer and the conference at **www.lighthouseseminars.com**

The Gilbane Conference

ON CONTENT MANAGEMENT

March 24-26 2004
Los Angeles, California
The Westin Bonaventure Hotel & Suites

Lighthouse Seminars

SOAPScope 3.0 from Mindreef

An already solid tool gets better

Since *WSJ* last looked at Mindreef's SOAPScope back in July '03 (Vol. 3, issue 7), much has been added in functionality and features to benefit the package. New items include integration with Visual Studio .NET, integration with the WS-I testing tools, a new Graph View for looking at historical data, and a differencing engine, among others. Overall, this is an excellent update; what follows are some of its highlights.

WRITTEN BY
BRIAN BARBASH

Visual Studio

Mindreef has nicely integrated access to the SOAPscope tool into Visual Studio .NET. (at the time of this writing, a plug-in for the Eclipse platform was also under development, and will be available shortly) As seen in Figure 1, developers have complete access to all SOAPscope functions from the workspace. One of the nice advantages of this is the ability to launch a Web service in the debugger, and then use SOAPscope's Invoke functionality to test the service without using the separate instance of IE. Seemingly a small detail, but these details are what this tool excels at.

Another nice feature of the Visual Studio integration is the ability to use SOAPscope to test and validate Web References before adding them to the project. By pointing the Visual Studio Add Web Reference tool to the SOAPScope repository, all services registered in the repository become available to the developer. From here, each available WSDL may be viewed, tested using WS-I tools, analyzed, compared against other versions for changes, and tested using the Invoke functionality. We've been told by Mindreef that by the time this article has been printed, an Eclipse version will be available.

Interoperability Testing

SOAPScope provides a set of tools for testing the interoperability of Web services. The Message Analysis tool submits logged messages to a series of compatability tests including conformance to SOAP 1.1, conformance to WSDL 1.1, XML parsing and schema validation errors, as well as against a set of best practices. Additionally, SoapScope has added SOAPscope has added the ability to use WS-I testing tools to produce a WS-I Basic Profile Conformance Report. It supports both the Java and C# versions of the toolset. The only requirements from the developer's perspective are downloading and installing either the Java or C# test kit from the WS-I Web site, and pointing SOAPscope to the installation directory. Once set up, executing a test is as simple as specifying the WSDL document and the service port to use.

Graph View

Figure 1 also shows the new Graph View that has been included in this version of SOAPscope. Messages archived in the local database may be viewed in three different graphs: Response Time, Message Frequency, and Transaction Size. The graph itself is also interactive. By clicking on a data point within the graph, the view is zoomed to include the relevant information captured by that point. This utility helps developers visualize the performance and characteristics of the Web services they are using.

Differencing Engine

Another new addition since we last looked at SOAPscope is the differencing engine, which supports both WSDL documents and SOAP messages. Its output is a clear and concise summary of the changes within a document. When comparing WSDL documents, developers may compare the WSDL stored in SOAPscope with the server version, a previously cached version, or a version stored at an arbitrary URL. This makes it easier to spot out-of-date WSDL references.

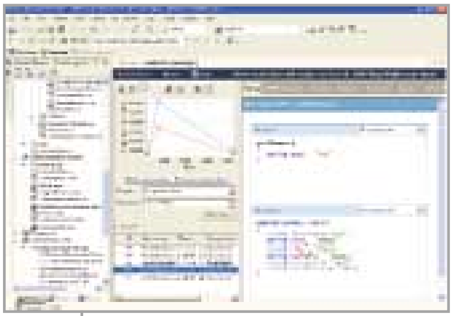


FIGURE 1 | SOAPscope workspace

Summary

Mindreef's new version of SOAPscope features a set of very nice enhancements that add to an already solid product. Its Visual Studio integration will improve the productivity of its user; it has made WS-I testing tools easy to use; and the additional information available from the Graph View, enhanced analysis module, and the differencing engine make development and testing Web services more efficient. As Joe Mitchko said in the original review, "Like a Swiss Army Knife, it is a trusted tool that you will use again and again."

About the Author

Brian R. Barbash is the product review editor for *Web Services Journal*. He is a consultant for the Consulting Group of Computer Sciences Corporation, where he specializes in application architecture and development, business and technical analysis, and Web design.

bbarbash@sys-con.com

Company Information

Mindreef, Inc.
22 Proctor Hill Road
Hollis, NH 03049
Tel: 603-465-2204
Fax: 603.465.6583
Web: www.mindreef.com
E-mail: mrsoapscope@mindreef.com

Download Information

www.mindreef.com

Price: $99

See for yourself. Upgrading to ColdFusion MX gives you amazing new features and a powerful Java™ architecture. Incorporate XML and web services with ease. Build flexible and maintainable applications with ColdFusion Components. Integrate with J2EE™ and .NET to deliver rich user interfaces with native connectivity to Macromedia Flash.™ Easily migrate existing applications and save time with the powerful new Macromedia MX development tools. Try ColdFusion MX today and see what you can do now.

Download the free 30-day trial at
www.macromedia.com/go/cfmxad

Advanced Web Services Security and Microsoft WSE

Use WS-Security support in Microsoft WSE to secure your .NET Web services.

As we move from the "Hello World" days of Web services toward development that can truly support the enterprise, there are some advanced functional requirements for Web services, including secure messaging, reliable messaging, and Web service policies. Since interoperability is the "Holy Grail" of XML and Web services, we must maintain this interoperability while supporting such advanced Web service functionalities. We can do this by developing and promoting the widespread industry adoption of so-called "advanced" Web service specifications. While you might envision creating a single monolithic specification that defines all such advanced Web service behaviors, this goal is better accomplished as a set of finely scoped, modular, and orthogonal Web services specifications.

WRITTEN BY
JEANNINE HALL GAILEY

Advanced Web Services Specifications

This modular approach to the development of our next generation of Web services specifications has been advanced, in particular by Microsoft, IBM, and a host of other companies. One benefit of using such an architecture of advanced Web services specifications is that you need only implement the particular specifications that provide the functionality required by your application without having to implement the other specifications. The proposed advanced Web services specifications, to varying degrees, leverage the foundational standards of XML, SOAP, and WSDL as well as the underlying transport protocols of TCP and HTTP. Figure 1 shows how advanced Web services specifications provide a modular platform for the development of interoperable Web services to support the enterprise.

Note that in Figure 1 the WS-Security specification is also leveraged by other security specifications to provide even more specific security functionalities.

Securing Web Services

In most ways, you secure a Web service in the same way that you would secure any other Web-based application. Since I don't have the space here to discuss all aspects of implementing end-to-end Web services security, this discussion will focus on the facilities provided by WS-Security for securing SOAP messages. (For an overall discussion of Web services security, pick up a copy of my book, *Understanding Web Services Specifications and the WSE)*. If you have ever written a secure application for the Web, you probably implemented secure socket layer (SSL) security. The "silver bullet" that has driven e-commerce and the dot-com boom, SSL provides security between two parties in an HTTP-based exchange by encrypting messages sent between the two. While SSL is secure, reasonably efficient, and can be used to secure SOAP messaging, it does have a major limitation in an enterprise Web service topology, namely that SSL limits message exchanges to only two parties. For example, a Web service would have to completely decrypt an SSL-secured SOAP message to read the SOAP message headers to determine what to do with the message, even if it is not the final message recipient. In complex Web service topologies where the service has to make such routing and forwarding decisions, WS-Security provides a much better solution as it enables you to secure the

A LIMITED TIME SAVINGS OFFER FROM SYS-CON MEDIA

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY
WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

Pick a 3-Pack, a 6-Pack or a 9-Pack

□ 3-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 6-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 9-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

□ MX Developer's Journal

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $167 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $29.99 /	Save: $60
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $137 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $127 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $30
Digital Edition - One Year (12)	You Pay: $19.99	

□ Linux World Magazine

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ Java Developer's Journal

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ Web Services Journal

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ .NET Developer's Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ XML-Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ WebLogic Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ ColdFusion Developer's Journal

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ Wireless Business & Technology

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ WebSphere Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ PowerBuilder Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

3-Pack

Pick any 3 of our magazines and save up to $\$275^{00}$
Pay only $175 for a 1 year subscription plus a **FREE CD**

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $\$350^{00}$
Pay only $395 for a 1 year subscription plus **2 FREE CDs**

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

Pick 9 of our magazines and save up to $\$400^{00}$
Pay only $495 for a 1 year subscription plus **3 FREE CDs**

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA

critical contents of the message while leaving the SOAP header unencrypted for more efficient routing.

Introducing the WS-Security Specification

WS-Security, proposed by IBM, Microsoft, and VeriSign, introduces the concept of a security token that is transported in the SOAP message header and can be used to authenticate the sender as well as to secure the message itself. In WS-Security, security tokens can be either plain-text or binary. Binary security tokens, which include X.509-based and Kerberos-based tokens, are encoded as base-64 binary XML for transport in the message header. The benefits of using binary security tokens are that they can be used to both digitally sign and encrypt the message, and they themselves can contain digital signatures made by root authorities. Tokens that contain signatures from security authorities are known as signed security tokens. WS-Security leverages existing XML security elements that are defined in other XML security specifications (namely the XML Signature and XML Encryption standards) and defines how they are used to secure SOAP messages.

Listing 1 shows an example of a request message containing an X.509-based security token (the BinarySecurityToken element), where this token is used to digitally sign the message. (For readability, I truncated the token's base-64 encoded data.) Information about how the message was digitally signed, including the data that was signed, the token used for signing, and the signature itself, are included in a Signature element, an XML security element defined in the XML Signature specification. Also, in this listing the body of the message is encrypted using the recipient's public key and included in the message body within the EncryptedData element, an XML security element defined in the XML Encryption specification. Upon receipt, the recipient verifies the signature using the sender's public key in the attached token and decrypts the message body using its private key.

Microsoft Web Services Enhancements

While you could follow the WS-Security specification to write a secure implementation for your Web service, you can avoid this extra work by leveraging Microsoft's implementation provided in their Web Services Enhancements (WSE) product offering. In the rest of this article, I will focus on WSE (I used the 2.0 Beta release for the code shown here) support for WS-Security and show how to use WSE to secure requests to a .NET Web service. While more complete support for advanced Web service standards will be included in the next version of Windows (now code-named Longhorn), WSE can be used today to implement advanced Web services functionalities, including security, reliable messaging, and Web service policies.

WSE is essentially a .NET assembly that supports both TCP and HTTP transports, and implements a set of input and output filters as well as a rich API. The WSE runtime, which hosts these filters and API, is implemented in Microsoft.Web.Services.dll. Input filters read incoming SOAP messages and translate known SOAP header elements into WSE programming objects, which you can access in your code using the SoapContext object, a collection that abstracts the message headers. Figure 2 shows how a SOAP message is passed through the series of input filters, called the input filter "pipeline."

In a similar fashion, WSE output filters construct SOAP headers based on the properties of the SoapContext object for the outgoing message. By properly constructing the SoapContext object, we can secure both request and response messages at both points of a Web service exchange. Later, I will show how to use the Soap-Context to secure a Web service request.

Getting Started with WSE

WSE provides full X.509 certificate support for authenticating and securing messages, and this is the type of authentication that I will use in my example. When using X.509 certificates, you first need to obtain and install valid certificates for both the client application and the server hosting the Web service. The server certificate, which must support both encryption and digital signatures, should be installed in the Local Machine store of the server hosting the service. The public key portion of this certificate should also be installed in the Current User store for the computer where any client applications will be run to

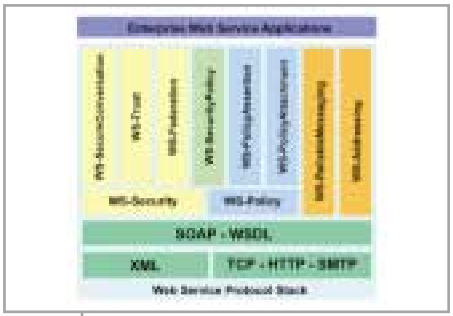

Enterprise Web Service Applications
WS-Trust
WS-Security
WS-Policy
SOAP - WSDL
XML
TCP - HTTP - SMTP
Web Service Protocol Stack

FIGURE 1 Advanced Web services specifications

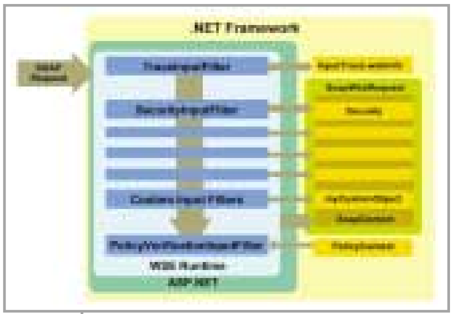


FIGURE 2 Representation of the WSE's Input Filter Pipeline

FIGURE 3 WSE Settings tool

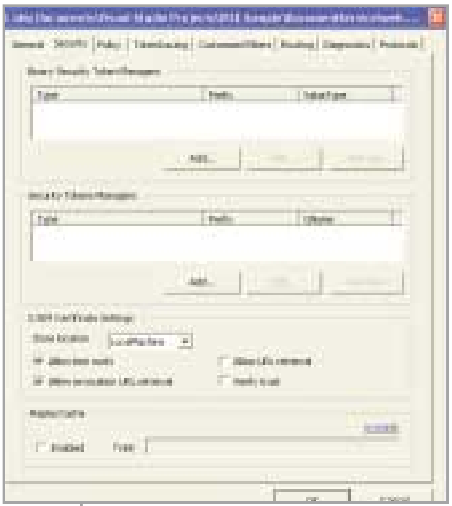


FIGURE 4 WSE security settings

access the service. These client computers also need to have their own client certificates installed in the Current User store, and these client certificates must also support encryption and digital signatures. Remember, the ASP.NET process that hosts the Web service must be able to access the key file for the X.509 certificates that it needs to use. See the WSE SDK documentation provided with the product for more information on granting WSE the required access to key files.

To help you get started using X.509 certificates, WSE provides you with a set of makecert.exe-generated client and service test certificates. However, for better performance you should obtain your certificates from a certificate authority, even a test one. When you install WSE with the Visual Studio Tools option, you get a handy X.509 Certificate Tool that makes it easier to manage certificates. You also get the WSE Settings Tool, which is a Visual Studio add-in that automatically makes the necessary updates to your application's XML configuration file.

Configuring WSE for Your Project

Since WSE can be employed by both a Web service and the client applications that consume that service, it really makes sense to simplify the discussion into a requesting application (requestor) and request-receiving application (recipient). Since in two-way messaging each request generates a response, this process is reversed when the response message is sent. However, WSE only sees an incoming message as a request and an outgoing message as a response. Except where the X.509 certificates are concerned, configuring WSE for a client application is just about the same as for a Web service.

To configure WSE for your project:

1. Right-click the project in the Visual Studio "Solution Explorer" window and select "WSE Settings 2.0…" This displays the WSE Settings Tool (see Figure 3).
2. In the "General" tab, check both checkboxes to enable WSE for the Web service. (When enabling WSE for a client application, the ASP.NET checkbox will be grayed out.)
3. In the "Security" tab (see Figure 4), set "Store Location" to "localMachine" if the project is a Web service and to currentUser if the project is a client application.
4. To make X.509-based authentication work using test certificates, check "Allow Test Roots" and uncheck "Verify Trust" checkboxes in the "Security" tab (see Figure 4). (Note that you would never do this outside of development and test environments.)
5. Click "OK" to close the WSE Settings tool. WSE automatically modifies the appropriate configuration file for the application and adds Microsoft.Web.Services.dll to the list project references.

Once you have enabled WSE for both the Web service and client application, you are ready to program WSE to secure your request and response messages.

Programming WSE

With WSE configured, you can program the WSE runtime from your application. Just remember that with WSE enabled, all request messages must contain a security token that can be positively authenticated by WSE or else they will be immediately rejected. Also, when an incoming message is digitally signed or encrypted, if WSE cannot verify the signature or decrypt the message using either the attached security token or one that it has access to, it will reject the message. WSE handles both X.509-based security tokens and Username tokens out of the box. For other types of security tokens or to do your own custom authentication you will need to implement your own security token handler.

To make your code a bit easier to read, you should add the following C# using directives to your code:

- using Microsoft.Web.Services
- using Microsoft.Web.Services.Security
- using Microsoft.Web.Services.Security. Tokens

When programming a client application using the Visual Studio Add Web Reference tool to generate a proxy programming object that abstracts access to a Web service, WSE automatically generates a second proxy class for the referenced Web service. This class, appended with "Wse," inherits from the Microsoft.Web.Services.Web ServicesClientProtocol class that implements the RequestSoapContext and ResponseSoapContext properties, which are used to access the SoapContext objects for the incoming and outgoing SOAP messages. Make sure that you use the WSE-generated proxy class in your code or WSE will be bypassed.

Retrieving X.509 Certificates

When using X.509 certificates, you must tell WSE which certificates you want to use when it creates X.509-based security tokens. This is done using key identifiers, which are strings that uniquely identify X.509 certificates. You can use the X.509 Certificate Tool at design time to determine key identifiers, and while WSE can also do this at run time, it is much more complicated. I recommend storing the key identifier values for certificates required by your application in the configuration file using appSettings.add elements, like

```
<add key="clientPrivateKey"
value="LptOGEbzc2HXS67ZSMsiNnrfDA0=" />
```

adding one for each certificate you need to use. This enables you to change certificates on the fly without having to recompile the application. Using key identifiers, WSE can easily create an X509Security Token object based on a specific X.509 certificate (see Listing 2).

Building the SoapContext for a Request Message

To secure a request message to a Web service using X.509 certificates, you must do the following:

1. Instantiate the Web service proxy object, which should be prefixed by "Wse" and inherits from Microsoft.Web.Services.WebServicesClientProtocol.
2. Get the RequestSoapContext property of the Web service proxy object.
3. Add an X509SecurityToken object (created in Listing 2) to the Security.Tokens collection of the request SoapContext.
4. To digitally sign the request, add a new Signature object to the Security.Elements collection of the request SoapContext. This Signature object is created using the same X509SecurityToken object added in step 3, assuming that this token supports digital signatures. You can determine if a token can be used for encryption by checking the object's SupportsDataEncryption property. WSE will generate a digital signature over the message body using the X.509 certificate, and it will add the signature information to a new Signature element in the Security header of the request message (Listing 1).
5. To encrypt the request, add a new EncryptedData object to the Security.Elements collection of the request SoapContext. This EncryptedData object is created using an X509SecurityToken object that instead represents the service's public key, which is different from the one added in step 3. You can still use the code in Listing 2 to create this token, but you will need to use the service certificate's key identifier instead of the one for the client certificate. Of course, the token must also support encryption, and you can determine if a token can be used for encryption by checking the object's SupportsDataEncryption property. WSE will encrypt the message body using the public key in the service's X.509 certificate and place the encrypted data inside an EncryptedData element in the body of the request message (Listing 1).

Listing 3 shows how these security objects are in turn added to the SoapContext for a request message to a Web service. In this example, myTokens[0] represents the client's token and myTokens[1] represents the service's token. Based on the code in this listing, WSE generates the SOAP request message in Listing 1.

Handing Incoming Requests

While X.509-based authentication can be tedious to configure (see the WSE 2.0 SDK documentation for troubleshooting potential error messages), it does make things easier to handle on the receiver-side of the exchange. For example, when an incoming message contains an X.509-based security token, WSE automatically attempts to authenticate the requestor by validating the certificate chain and looking for a corresponding certificate in the server's Machine Store. If it finds the certificate there, it passes the request on to the requested resource. Likewise, WSE automatically uses the attached certificate to validate any attached digital signatures, and it will also try to decrypt messages using the private key portion of the certificate used to encrypt the message, if it can find it. If you plan to use a non-X.509 authentication scheme, you will need to write a custom security token manager to authenticate other security tokens.

Conclusion

WS-Security provides the facilities necessary for securing SOAP messaging. However, other specifications like WS-Trust, WS-SecureConversation, WS-SecurityPolicy, and WS-Federation have been proposed to build even more robust and functional security scenarios. Out-of-the-box, WSE provides an immediate security benefit by blocking all request messages that cannot be authenticated and that contain invalid digital signatures or encryption. With a little effort, WSE can be configured to provide much-needed security for .NET Web services, and this is particularly true when using X.509 certificates. Note: The sample code shown in this article is from my book; you should be able to download it from the Microsoft Press Web site (www. microsoft.com/mspress/books/ 6708.asp).

About the Author

Jeannine Hall Gailey is a former Microsoft manager who has published numerous technical magazine articles. Her recent book, *Understanding Web Services Specifications and the WSE*, was published in February 2004 by Microsoft Press.

■■■ author@jeanninegailey.com

Listing 1

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

xmlns:wsse="http://schemas.xmlsoap.org/ws/2003/06/secext">
  <soap:Header>
    ...
    <wsse:Security soap:mustUnderstand="1">
      <wsu:Timestamp
        wsu:Id="Timestamp-c600bbb2-7d35-441a-ad89-
6b356777c2da"

xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility">
        <wsu:Created
          wsu:Id="Id-69e5ed3d-ef43-4995-af05-1eb146d96f80"
          >2004-01-06T21:50:20Z</wsu:Created>
        <wsu:Expires
          wsu:Id="Id-09185624-4f41-44eb-8dd6-4bed57f53c54"
          >2004-01-06T21:55:20Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken ValueType="wsse:X509v3"
        EncodingType="wsse:Base64Binary"

xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/utility"
        wsu:Id="SecurityToken-69b4cdaa-4cf4-4e88-a591-
78d0c73ba61d"
         >MIIFIDCCBAigAwIBMRM
...=</wsse:BinarySecurityToken>
      <xenc:EncryptedKey
        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
        <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-
1_5" />
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmld-
sig#">
          <wsse:SecurityTokenReference>
            <wsse:KeyIdentifier
              ValueType="wsse:X509v3"

>F5XpYpi3n00/mqB8/W8tWIBF4TA=</wsse:KeyIdentifier>
          </wsse:SecurityTokenReference>
        </KeyInfo>
        <xenc:CipherData>
          <xenc:CipherValue
            >GSwglkSTqNM5h5nyzeZSFNTWMpQ
...=</xenc:CipherValue>
        </xenc:CipherData>
        <xenc:ReferenceList>
          <xenc:DataReference
            URI="#EncryptedContent-48d1ac67-0bab-4e8e-
99d3-b12c45ebebbb" />
        </xenc:ReferenceList>
      </xenc:EncryptedKey>
      <Signature xmlns="http://www.w3.org/2000/09/xmld-
sig#">
        <SignedInfo>
          <CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#" />
          <SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmld-
sig#rsa-sha1" />
          <Reference URI="#Id-8b4fd84b-44a5-41c7-8458-
0f11eb9c2883">
            <Transforms>
              <Transform
                Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#" />
            </Transforms>
            <DigestMethod
              Algorithm="http://www.w3.org/2000/09/xmld-
sig#sha1" />

<DigestValue>cEKveGWL2UBX5TRrF4yyqtyxKg0=</DigestValue>
          </Reference>
            ...
        </SignedInfo>

<SignatureValue>P3Ah7ZhCZucoEz20y2BFsJ...=</SignatureValue>
        <KeyInfo>
          <wsse:SecurityTokenReference>
            <wsse:Reference
            URI="#SecurityToken-69b4cdaa-4cf4-4e88-a591-
78d0c73ba61d"
            ValueType="wsse:X509v3" />
          </wsse:SecurityTokenReference>
        </KeyInfo>
      </Signature>
    </wsse:Security>
  </soap:Header>
  <soap:Body wsu:Id="Id-8b4fd84b-44a5-41c7-8458-
0f11eb9c2883"
    xmlns:wsu="http://schemas.xmlsoap.org/ws/2003/06/util-
ity">
    <xenc:EncryptedData
      Id="EncryptedContent-48d1ac67-0bab-4e8e-99d3-
b12c45ebebbb"
      Type="http://www.w3.org/2001/04/xmlenc#Content"
      xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
      <xenc:EncryptionMethod

Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
      <xenc:CipherData>

<xenc:CipherValue>s1SNQenKOIFQQxF...=</xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedData>
  </soap:Body>
</soap:Envelope>
```

Listing 2

```
// Instantiate a new binary security token for the
// X.509 certificate used to sign the message
X509SecurityToken myToken;

// Set the key bytes based on the supplied key string
byte[] keyIdentifer;
keyIdentifer = Convert.FromBase64String(keyString);

// Open and read the current user certificate store
```

```
X509CertificateStore myStore;
myStore  = X509CertificateStore.CurrentUserStore(
  X509CertificateStore.MyStore);
myStore.OpenRead();

// Get the certificate that matches the supplied key
X509CertificateCollection myCerts;
myCerts = myStore.FindCertificateByKeyIdentifier(
  keyIdentifer);

// Instantiate a new certificate object
X509Certificate myCert = null;

// If the collection is not empty, get the first
// certificate in the collection
if (myCerts.Count == 1)
{
  // Use the returned certificate
  myCert = myCerts[0];

  // Create the security token
  // based on the certificate
  myToken = new X509SecurityToken(myCert);

  // Return the token
  return myToken;
}
else if(myCerts.Count > 1)
{
  // Multiple certificates exists
  // with the same key
  MessageBox.Show("There are more than one "
    + "certificates corresponding to the key "
    + keyString + ". \n"
    + "Please resolve this issue.");
}
else
{
  // The certificate could not be found
  MessageBox.Show("The certificate corresponding "
    + "to the key " + keyString
    + " could not be found. \n"
    + " Please verify that this certificate is "
    + "installed properly.");
}
return null;
```

Listing 3

```
// Instantiate the Web service
DocumentServiceWse myService;
myService = new DocumentServiceWse();

// Create a new SoapContext for the request message
SoapContext myReqContext;
myReqContext = myService.RequestSoapContext;

if (myTokens[0] != null)
{
  // Add the new token to the Security.Tokens
  // collection in the SoapContext of the
  // request message
  myReqContext.Security.Tokens.Add(myTokens[0]);

  // Verify that the token can be used for signing
  if (myTokens[0].SupportsDigitalSignature)
  {
    // Create a Signature using the token
    Signature mySig = new Signature(myTokens[0]);

    // Add the Signature to the SoapContext
    myReqContext.Security.Elements.Add(mySig);

    // If we have a second token, verify that
    // it is a X509-based token that
    // supports encryption
    if (myTokens[1] != null &&
myTokens[1].TokenType == TokenType.X509v3
&& myTokens[1].SupportsDataEncryption)
    {
// Create a new EncryptedData object
// that tells WSE to encrypt the
// message body using the provided
// security token
EncryptedData myEncData;
myEncData = new EncryptedData(myTokens[1]);

// Add the EncryptedData to the SoapContext
myReqContext.Security.Elements.Add(myEncData);
    }
  }
  else
  {
    throw new ApplicationException("You cannot use "
+ "this token to access the service.");
  }

  try
  {
    // call the GetDocument method on the Web service
    docNames = myService.GetDocument(docNames);


    // get the context from the response message
    // that contains XML documents as attachments
    SoapContext myRespContext;
    myRespContext = myService.ResponseSoapContext;

    // get the XML documents from the attachments
    string[] myDocs;
    myDocs = GetAttachments(myRespContext, docNames);

    // return the XML documents
    return myDocs;
  }
  catch(Exception ex)
  {
    throw new ApplicationException(ex.Message);
  }
}
return null;
```

vb

www.mxdevelopersjournal.com
0.9
MX
developer's journal
Toppling the...
Tower of Babel
with MX
How to produce the internet application
BabelChat

The WS* Standards – A Primer

What's coming — and what might be gone

Over the past couple of years, several technology vendors have defined a comprehensive set of specifications that, when complete, will provide an infrastructure for enterprise-class Web services interoperability. The names of these specifications generally begin with "WS-", so the group of them is sometimes referred to as WS* (pronounced "WS Splat").

WRITTEN BY
ANDY ASTOR &

PRASAD YENDLURI

This article identifies the important WS* standards, briefly defines those that have not yet achieved mass-market acceptance, and describes the current state of development for each. At the end, we offer our view of each specification's relative market importance.

We will use Figure 1 to structure the discussion. *Note:* "Composable" means that items are independent, and can be plugged together (or not) with relative ease. "Composable Service Elements" means that developers can add security, reliable messaging, and transactionality to their Web services in any combination.

1. Transport Level

- ***HTTP/HTTPS:*** Currently at version 1.1.
- ***SMTP:*** Far less prevalent for Web services usage than HTTP. Worth considering for specific business applications.

2. Messaging Level

- ***XML (including XML, XSL, XPath, etc.):*** Currently at version 1.0, although v1.1 is a Proposed Recommendation.
- ***SOAP:*** Currently version 1.1 is most widely deployed, although v1.2 has recently become a W3C Recommendation.
- ***Attachments for SOAP:*** There are two important specifications around SOAP Attachments: SOAP with Attachments (SwA) and Message Transmission Optimization Mechanism (MTOM). A third specification, DIME, has recently lost its momentum and is therefore no longer particularly significant.
  - *SwA*: Currently available for both SOAP 1.1 and 1.2, SwA is by far the leading mechanism for handling attachments in Web services. However, it will likely be supplanted by MTOM.
  - *MTOM:* Some vendors are going straight to MTOM without supporting SwA. MTOM appears to be the important Attachment specification for the future (late 2004 and beyond).
- ***WS-Addressing:*** This specification abstracts WS endpoint references away from the transport and messaging infrastructure, allowing them to be specified independently of the transport system. It is currently in version 1.1, but is still private and has not yet caught on significantly in the marketplace.

3. Description Level

- ***WSDL:*** Currently at version 1.1, with v2.0 in progress at W3C.
- ***UDDI:*** Very important in the marketplace, although relatively few companies have actually deployed it. This spec is currently at version 2.0. However, v3.0 is already complete at the technical committee level and should supplant v2.0 shortly.
- ***WS-Inspection:*** Once a potential complement and/or competitor for UDDI, this private specification appears to have lost steam.
- ***WS-PolicyFramework (includes WS-PolicyAssertions and WS-PolicyAttachments)***: Currently a private specification, WS-Policy allows a Web service to specify exactly how it wants to be called. For example, a service might allow either Kerberos or X.509 authentication, but prefer Kerberos. We believe that this specification will become important and widely used over time.
- ***WS-MetadataExchange:*** A new, private specification that allows a Web service client to easily get the policy, schema, and WSDL information about another Web service. Likely to grow in importance.

4. Reliability

There are two competing efforts to ensure reliable transmission among Web services: The first is private, and is called WS-ReliableMessaging. The second is WS-Reliability, which is currently in OASIS. The jury is still out on which specification will achieve market prominence, or how the two might cooperate.

5. Security

- ***WS-Security:*** Already at an advanced stage within OASIS, WS-Security will continue to mature in 2004.
- ***Other important security specifications***: All of these are currently private. All are likely to be important, probably by late 2004.
  - *WS-Trust:* Using Tokens to establish trust in a conversation
  - *WS-SecureConversation:* Gives the same security to a long-running conversation that WS-Security gives to a single message
  - *WS-Federation:* Decentralizes control over trust, authentication, and authorization
  - *WS-SecurityPolicy:* Allows specification of specific security policies within the WS-Policy framework.

6. Transactionality

- ***Multiple competing frameworks:*** None of these have real market traction yet. One major private initiative is called WS-Transaction. A competing effort is WS-Composite Application Framework, currently in OASIS.
- ***WS-Transaction:*** Describes coordination among distributed application components. It is made up of three sub-specifications: WS-Coordination, WS-AtomicTransaction (WS-AT), and WS-BusinessActivity (WS-BA).
  - *WS–Coordination:* Extensible framework for coordinating actions of distributed applications. This is the basis for transaction management (see next 2 sections).

WS–AtomicTransaction: For managing tightly linked distributed transaction components using classic two-phase commit protocol.
- *WS–BusinessActivity:* For managing transaction coordination between looser-knit components that may not be completely under the coordinator's control.

- ***WS-Composite Application Framework (WS-CAF):*** An OASIS-sponsored effort. Its stated purpose is "to propose standard, interoperable mechanisms for managing shared context and ensuring [that] business processes achieve predictable results and recovery from failure." It includes the following:
  - *Web Service Context (WS-CTX):* Lightweight framework for simple context management

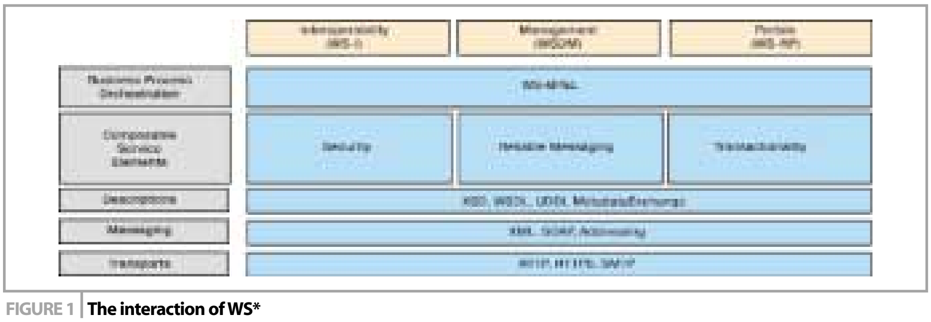


FIGURE 1 **The interaction of WS***

  - *Web Service Coordination Framework (WS-CF):* Sharable mechanism to manage the life cycle of composite application messages
  - *Web Services Transaction Management (WS-TXM):* Supports three protocols across transaction managers (two-phase commit, long-running actions, and business process flows)

7. Service Composition

- ***WS-BPEL:*** The de facto standard and state of the art for specifying Web service–based business process orchestration. Currently in OASIS, this specification is likely to emerge as a standard in late 2004.
- ***Additional work*** is ongoing in W3C in the Web Services Choreography Working Group, which is drafting a more abstract definition of Web services–based business process components. A requirements document is complete, but this will take a long time to develop into something meaningful in the market.

8. WS-I

- The WS-I Basic Profile 1.0 (BP1.0) defines interoperability guidelines for XML, XML Schema, SOAP, WSDL, and UDDI. This standard is critical to the industry. Both Gartner and Forrester/Giga recently began recommending BP1.0 conformance to most customers seeking Web services guidance.
- BP1.0 is undergoing amendments for attachment processing (both SwA and MTOM), to be called Attachment Profile 1.0 (AP1.0), and will soon be important in the marketplace.
- WS-I expects to release the Basic Security Profile 1.0 (BSP1.0) in July 2004, which will define proper usage of WS-Security. This will also be an important conformance milestone.
- Additional future WS-I profiles are not yet defined, but they are likely to follow the same prioritization as that described by this article.

9. Management

OASIS is developing WSDM, the only major WS management standard candidate in the marketplace. Version 0.5 is expected in March, and v1.0 in June. Initially, WSDM will include two specifications. The first is MUWS (Management Using Web Services), which defines a generalized model of how to manage any resource with Web services. The second is MOWS (Management of Web Services), which adds to MUWS the specifics required to manage resources that are Web services.

10. Portal

WSRP (Web Services for Remote Portlets) is a recently approved OASIS standard. It defines a framework for portlets based on standard Web services interfaces. The specification seems to have picked up market traction.

WS* Prioritizations

Following is our view of the relative market importance of each of the specifications mentioned here. If you are involved with Web services, we believe that Priority One items are absolutely required now. You should be working on Priority Two implementations, and planning for Priority Three. Priority Four items should be watched with interest.

Priority One

1. HTTP v1.0 and/or v1.1
2. XML (including XML Schema, XSL, XPath, XQuery)
3. SOAP v1.1
4. WSDL v1.1
5. UDDI v2.0
6. WS-I Basic Profile 1.0
7. WS-BPEL

Priority Two

1. SOAP v1.2
2. SOAP Attachments: SwA and/or MTOM
3. WS-I Attachment Profile 1.0
4. WSDL v2.0
5. WS-Security
6. SMTP (as required)

Priority Three

1. UDDI v3.0
2. WS-I Basic Security Profile 1.0
3. WS-Policy
4. WS-ReliableMessaging or WS-Reliability
5. WSDM
6. WSRP

Priority Four

1. WS-Addressing
2. WS-MetadataExchange
3. WS-Trust, WS-SecureConversation, WS-Federation, WS-SecurityPolicy
4. WS-Transaction OR WS-Composite Application Framework
5. WS-Choreography

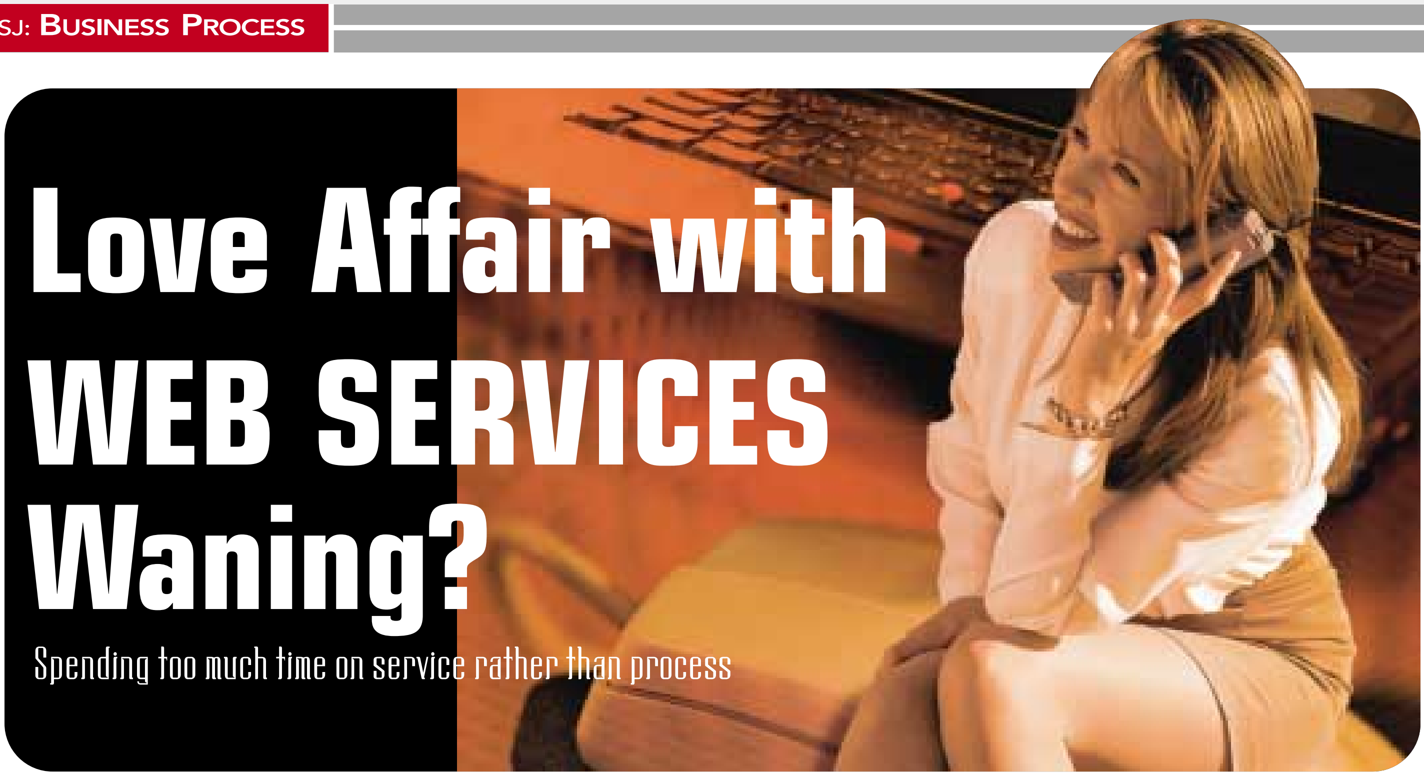
About the Authors

Andy Astor is a vice president at webMethods, responsible for driving the company's strategy and execution in key areas, such as Web services. He also serves on the Board of Directors of WS-I. Prior to joining webMethods, Mr. Astor was vice president at D&B, where he led the development of customer-facing products worldwide, including all Web and Internet-based applications. His work at D&B included the development and launch of one of the earliest commercial web services. Andy is on the International Advisory Board for *Web Services Journal.*

■■■ andy.astor@webmethods.com

Prasad Yendluri is principal architect at webMethods with a focus on Web services and the related standardization efforts. Prasad represents webMethods in W3C, OASIS, and WS-I and is currently serving as editor of WS-I Basic Profile and WS-BPEL specifications. He was also lead architect for and coauthor of the RosettaNet Implementation Framework 2.0 and coauthor and contributor to the ebXML Messaging Specification. Prasad has published several papers on Web services, recently on Web services reliability and Web services choreography.

■■■ pyendluri@webmethods.com



Love Affair with WEB SERVICES Waning?

Spending too much time on service rather than process

■ The Component Based Development Forum, an analyst firm and think tank covering business software creation, reuse, and management, recently wrote, “Service Orientation – So What?” They went on to explain,

“First we had Web Services. Then we realized that Web Services were simply a technology, and we needed architecture to manage the loose coupling of pretty much everything. For a time SOA was top of the toy box. Then we started to see Service-Oriented Programming and Service- Oriented Design. All very sensible, even if it is a bottom-up process of discovery. Of course the logical conclusion to this is that we are progressively establishing a comprehensive approach to architecture, modeling, design, programming, deployment, interoperability etc., that is service oriented.”

WRITTEN BY

HOWARD SMITH &

PETER FINGAR

Really?

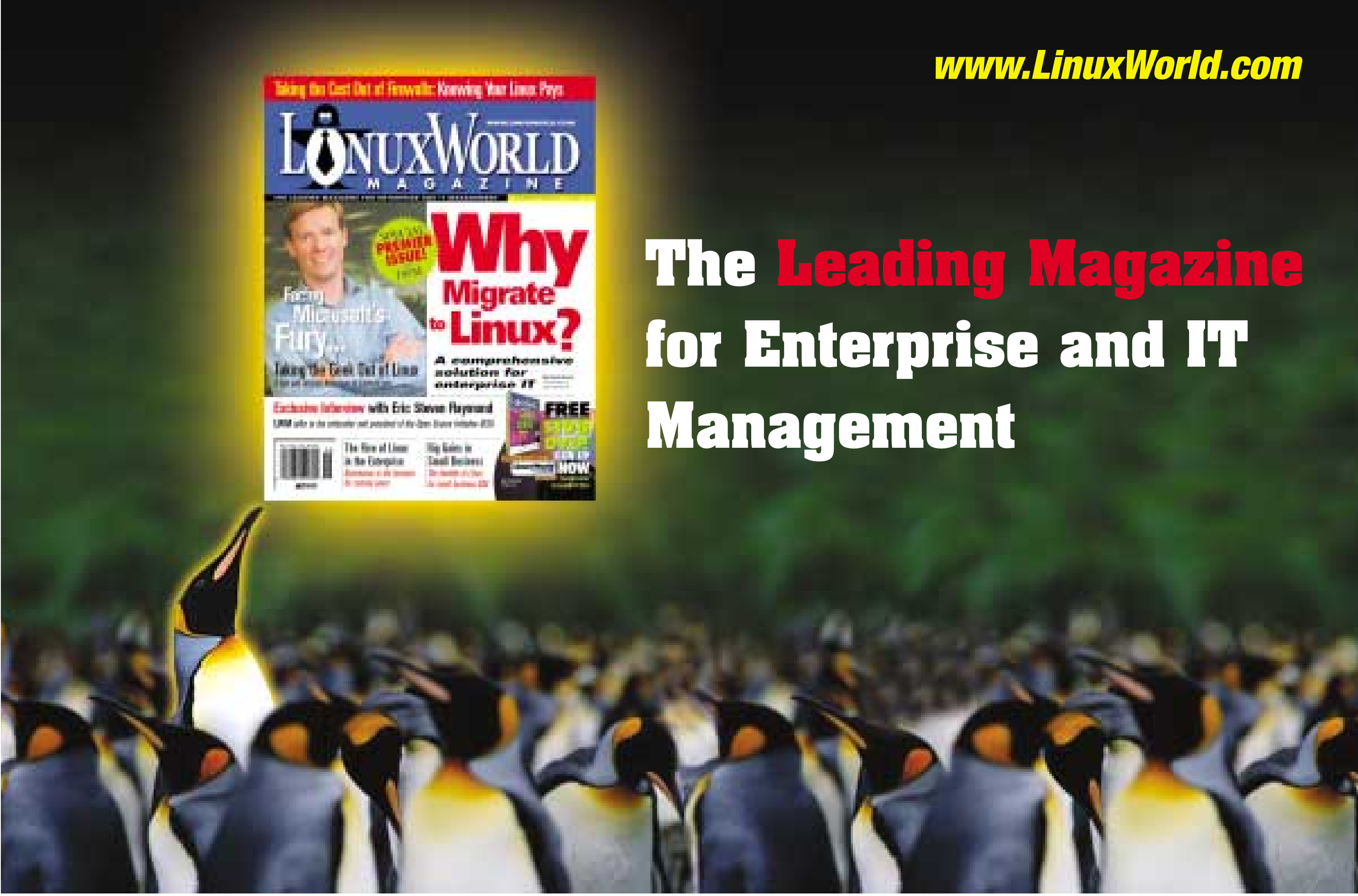
We see something else. The industry is progressively establishing a comprehensive approach to business process discovery, design, deployment, execution, operations, analysis, and optimization. Web services technology–inspired terminology is clouding the fact. Take one prominent example:

There’s a wonderful new book by Dr. Ravi Kalakota and Marcia Robinson entitled *Services Blueprint: Roadmap for Execution*. Despite the title and a chapter entitled “Services is the Mega-Trend,” the majority of the book is concerned with business process management (BPM). Indeed, the book is replete with the word “process.” Much of the book concerns itself with discussions of process configuration, process flexibility, process trends, process digitization, the composite process layer, the need for business process management, translating services into processes, linking processes via integration layers, from a technology focus to a process focus, better process automation, enabling composite processes, mapping processing into applications, process outsourcing, creating a process strategy with Six Sigma, moving from strategy to process design, picking a process improvement method, and customer-centric process transformation. The list goes on and on. And all this is taken directly from the table of contents!

If we venture inside the book we find that it too is largely concerned with processes. Explicit statements are made as to the significance of processes to the so-called “Services Blueprint.” In Table 1.1, entitled “Historical Perspective: Changing Process Priorities,” the year 2003 is tagged as “Services-Centric,” but is then defined as “Digitization of Cross Enterprise Processes; End to End Supply Chain Enablement, Business Process Outsourcing and Business Process Management.” Numerous examples of end-user companies that have executed on “process improvement” are given throughout the text and the authors correctly point to the logical next step in Six Sigma, “process digitization.” Kalakota and Robinson point to the reality that

“there is a clear pattern that can be gleaned from the turmoil: non-stop digitization of business processes … Digitization is the outcome of the non-stop business need to be more customer-driven and process-centric … A successful digitization effort is one in which the company treats technology not as a sole solution, but as an enabler for innovating, improving and integrating business processes.”

The dichotomy between Kalakota and


www.LinuxWorld.com
LINUXWORLD MAGAZINE
Taking the Cost Out of Firewalls: Knowing Your Linux Pays
Premier Issue!
Why Migrate to Linux?
A comprehensive solution for enterprise IT
Facing Microsoft's Fury...
Taking the Geek Out of Linux
Exclusive Interview with Eric Steven Raymond
The Rise of Linux in the Enterprise
Big Gains in Small Business
FREE
NOW
The Leading Magazine for Enterprise and IT Management

LinuxWorld Magazine

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *LinuxWorld Magazine* is aimed squarely at providing this group with the knowledge and background necessary to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *LinuxWorld Magazine* does not feature low-level code snippets but focuses instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month presents a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

Regular features include:

Advice on Linux Infrastructure

Detailed Software Reviews

Migration Advice

Hardware Advice

CEO Guest Editorials

Recruiting/Certification Advice

Latest News That Matters

Case Studies

SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE $71.76

YOU PAY ONLY

$49.99

12 ISSUES/YR

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE*

SUBSCRIBE TODAY!

WWW.SYS-CON.COM
OR CALL
1-888-303-5282

FOR ADVERTISING INFORMATION:
CALL 201 802.3020 OR
VISIT WWW.SYS-CON.COM

IDG INTERNATIONAL DATA GROUP
LINUXWORLD® IS THE REGISTERED TRADEMARK OF INTERNATIONAL DATA GROUP, INC.

SYS-CON MEDIA
SYS-CON IS USING THE MARK PURSUANT TO A LICENSE AGREEMENT FROM IDG

The World's Leading i-Technology Publisher

Robinson's use of the title "services" in a book about "processes" is difficult to fathom when one considers other bold statements made in the text, such as, "Service platforms are the emerging foundation for integrating and digitizing end-to-end processes," "The emerging service platform is the quintessential process environment," "Translating customer's need into business objectives, business objectives into processes, and processes into interactions is the role of service platforms," "As a result, we need a much better understanding of how to create, deploy, maintain, and enhance cross-enterprise processes," and "For most companies, the next step in the digitization journey is setting these composite process in motion."

Kalakota and Robinson even point out the weakness of a services-based approach, stating that, "It is not enough to have multiple Web services scattered all over the place. You need a framework that can pull everything together and make everything talk to each other." This sounds like a process to us. They go on to state that, "Services are built from composite processes," and "In our opinion, the design, management, and integration of business processes are increasingly what separates the good companies from the rest of the pack." They give examples of emerging services platforms, describing them as "an approach for designing and developing cross-application business processes," and that this entails "A great deal of subprocess design," and "The services perspective helps to eliminate functional stovepipes and replace them with processes that focus on creating value for customers." Finally, they provide a message for business people and technologists when they correctly point out that, "People tend to trivialize the need for process management. This is where initiatives often fall into problems. Actually, process management is becoming a field unto itself."

Should the book have been called "Processes Blueprint: Roadmap For Execution"?

Can't We Just Call a Business Process a Business Process?

Where are the "services" in Kalakota and Robinson's book? What's in a name anyway? As they say on page 244, "The different operational lines of business speak the process languages of Lean Enterprise or Six Sigma." So why confuse readers by introducing terminology (services) inspired from Web services?

The notion of "service" in Kalakota and Robinson's book is really a synonym for three things.

First, they equate the need for services (a.k.a., processes) with the notion of a "focal point," another new term they introduce. They define "focal points" as business drivers such as "easy to do business with," "single voice of the customer," "low daily prices," "fast and responsive service," etc. But why introduce the term focal point? We think business people already understand the concept of a business driver.

Second, they equate service with process outcomes.

Third, they use the term "service platform" to refer to a new class of mission-critical enterprise software. But while some vendors who are developing infrastructure software may use the term "services platform" to refer to their software, we don't think what they are developing is what Kalakota and Robinson had in mind when they used the term. From diagrams in the book, we think Kalakota and Robinson are referring to business process management systems (BPMSs).

While vendors are free to describe their products using their own terminology, many, including those that used to call themselves EAI or workflow, are today using the term BPMS. Some have even adopted the standard symbol for a BPMS, a cube. *The BPMS is a business application that allows the management of business processes without dipping into the technical plumbing*, including Web services plumbing. A BPMS frees business people from infrastructure, integration, and other complexities, just as the relational database (RDBMS) freed ERP users from equivalent file system and data plumbing. In this sense, BPMS products are the combination of a service-oriented architecture (SOA) and a process-oriented architecture (POA). The former is a technical infrastructure, the other a breakthrough business application. SOA and POA enjoy a symbiotic relationship. This pattern of symbiotic relationship between a standards-based commodity platform and a new innovation has been repeated time and again in the IT industry.

One common example of an innovation and a paradigm shift built on a standards-based commodity is the simple, yet eloquent, spreadsheet. The convenience and low cost of the breakthrough was so striking that it led to the PC revolution in business. The spreadsheet could not have been successful had it not been for the fact that personal computers – a standards-based commodity – were spreading like wildfire elsewhere in society. To the business, the PC loaded with a spreadsheet meant a radical simplification of routine calculations, transferring to the average businessperson a function that had once required special programming skills. Quite literally, business people could not understand the value of a PC until they saw someone working with a spreadsheet.

Unix and the RDBMS is another example. No one in business knew what to do with mid-range Unix computers until they saw the value of departmental business data management using new-fangled mid-range relational databases. Today, few in business know what to do with Web services, until they see a team of business and technical architects working with a BPMS. While incumbent platform vendors continue to entice end users to invest in IT on a promise of Web services interoperability and reuse, CIOs are becoming increasingly aware that the ROI in Web services lies not in the infrastructure, but in the BPMS. The value of Web services lies in the technical infrastructure, the standards-based environment in which the BPMS, and other business applications, can thrive.

There are striking correspondences between diagrams in Kalakota and Robinson's book and the architecture of BPMS. But they continue to stress the word *service*. Toward the end of the book they claim that, "Clearly, we are in between eras. Everywhere, strategists, senior and mid-level managers are caught between process-centric models (current state) and service-centric models (future state)," and they urge businesses to bring "service thinkers" into process teams. But they then go on to list "seven points to ponder." The list includes the statements, "Organization outputs are produced through processes" and "A process

“Perhaps it’s time for IT to master their language, the language of process”

improvement methodology is critical.” What all this seems to boil down to is an assertion by Kalakota and Robinson that “services” and their corresponding “focal points” are useful ways to determine requirements for new end-to-end processes. Perhaps they are. If so, let’s celebrate. But one cannot help feeling that this book, published in June 2003, was actually written during the peak of the IT industry’s Web services hype curve… which was precisely the same time that the Web services technical community was waking up to the significance of business processes, culminating with the submission to OASIS by IBM and Microsoft of BPEL4WS (Business Process Execution Language for Web Services).

Was the book caught in the services-processes time warp of the past year and, as a result, unclear as to the terminology it should use? Did the editors wonder how to reconcile the difference in terminology between the Web services and BPM communities?

Kalakota and Robinson’s enumeration of numerous business processes in each chapter is another indicator of the book’s focus on processes, rather than services. Indeed, we hope that business people won’t be put off the book based on its title, because the process content is quite relevant to them. Kalakota and Robinson give these processes catchy titles such as Order To Cash, Engage To Close, Transact to Fulfil, Build To Order, Plan To Produce, Resume To Work, Goal To Reward, and many others. But in the figures, where they place those end-to-end processes they are labeled the “Services Layer,” even though they sound, and look uncannily like, business processes. Why not call a business process a business process? If we do that, business people will immediately understand what we are talking about, and that’s a very good thing.

But perhaps the most obvious element missing in Kalakota and Robinson’s otherwise excellent book is the process life cycle itself. For while attributing to services (and service thinking) many wonderful attributes, the book omits to say how services, or for that matter processes, are actually improved in line with business objectives. There is an assumption at the heart of the book that the business objective, the focal point, is somehow inviolate and never changing. Kalakota and Robinson’s focal points are high-level aspirations, such as “easy to do business with.” What they do not show is how a business changes from not being easy to do business with toward being easy to do business with, and how it maintains a constant improvement in being easier and easier to do business with. After all, a business is not going to wake up one morning, become “easy to do business with,” and then forget about that topic thereafter.

So whether one calls things “services” or “processes,” we ask how are they taken through the life cycle of process improvement, from discovery, to design, to deployment, to execution, to operations, to analysis and optimization? In short, where is business process improvement in Kalakota and Robinson’s service model? Life-cycle management is itself a business process – and how does that process manifest itself in a technology-inspired service stack? We wonder how many business people have looked at technical architecture stack diagrams and wondered about this? It’s not enough to create an end-to-end process at a moment in time, for it will surely change the minute it is created – that’s just the way of business. Time, change, and improvement are the watchwords of BPM, not Web services.

Where Kalakota and Robinson explicitly refer to “BPM” they mostly assign it to a mid-tier of “integration software.” They then align that to EAI technology, about which they say,

“Managers must understand process integration issues before they invest millions of dollars. Many find themselves frustrated when they discover that multi-million dollar investment they made in Enterprise Application Integration (EAI) software is not delivering the ROI the salesperson promised.”

They then go on to say, “To lower the cost of integration, the newer-generation XML-based EAI is being engineered under the banner of business process management (BPM) tool kits.” Such a perspective misses the essence of BPM, management of the complete life cycle of change within the business process.

WSJ ADVERTISER INDEX

ADVERTISER	URL	PHONE	PAGE
Active Endpoints	www.active-endpoints.com	203-949-9400	17
Altova	www.altova.com	978-816-1600	6
Assande	www.assande.com	212-401-2988	10
Confluent Software	www.confluentsoftware.com	866-428-8242	19
Hewlett-Packard	www.hp.com	650-857-1501	13
IBM	www.ibm.com/websphere/middleware.com	1-800-IBM-4YOU	Cover IV
Lighthouse Seminars	www.lighthouseseminars.com	781-821-6734	33
Macromedia	www.macromedia.com/go/cfmxad	415-252-2000	35
Mindreef	www.mindreef.com	603-465-2204	8-9
Networld+Interop	www.interop.com	415-905-2300	21
Novell	www.novell.com	781-464-8000	3
OpenLink Software	www.openlinksw.com	781-273-0900	Cover II
Parasoft	www.Parasoft.com/SOAPtest	888-305-0041	5
WebAppCabaret	www.webappcabaret.com	866-256-7973	Cover III

Advertiser is fully responsible for all financial liability and terms of the contract executed by their agents or agencies who are acting on behalf of the advertiser. This index is provided as an additional service to our readers. The publisher does not assume any liability for errors or omissions.

Our conclusion: Despite having the wrong title, it's a great book. But if business people have to bow to IT industry gorillas for the terminology they use, perhaps there is a compromise to be made? Could G2000 firms persuade IBM and Microsoft to rename Web services, Web processes? Could BPEL4WS just be called BPEL and can it lose its 4 Web services tail? Could vendors stop attributing wonders to Web services technology (SOAP, UDDI, and WSDL) that are more rightly attributed to the ability of a BPMS to manage the life cycle of business process improvement?

To illustrate the difference in perspective between services and business processes, we have included here Figure 6.2 "Supply Chain Blueprint," from Kalakota and Robinson's book, and re-rendered it as conceived from a BPM viewpoint (see Figures 1 and 2).

Can't We Just Speak the Language of Process?

Over the first 50 years of commercial IT, the level of abstraction used in developing business information systems has continually moved higher and higher, from wiring plug boards, to assembly languages, to COBOL, to objects, to components, and finally to a business paradigm – the business process. Each new paradigm required its own terminology, and now it's imperative that we address the terminology business people use.

After 50 years of IT automating the business, the BPMS is the business analyst's CAD/CAM system that allows the business to automate IT. The significance of the BPMS is that BPM can now be of the business process, for the business process, and by the business process. This is the message of BPM and the message obfuscated by Kalakota and Robinson's book and an IT industry enamored with Web services technology and terminology. While the *Harvard Business Review* is telling business leaders that "IT doesn't matter," the IT industry continues to baffle those same business leaders with technology jargon. It's time to call a business process a business process. BPM isn't some technical integration layer buried in a Web services technology stack; it's the king of the hill, for what businesses really want is direct control of business process life cycle improvement, not by IT proxy.

For the business people who control IT investments and spending, Web services terminology is yet another foreign language they don't care to master – to them Web services don't matter. They do indeed understand business processes, more than many IT people care to admit, so perhaps it's time for IT to master their language, the language of process. To do so will require refactoring technical architectures with a process-oriented architecture.

A future article will describe the process-oriented architecture of BPMSs in more detail. The article will explain how the BPMS leverages past investments in IT and the way in which it interacts with its operating system, just as RDBMS was built on Unix. The operating system for the BPMS is IT's new commodity, the standards-based, networked operating system of the Internet era – Web services.

References

- Kalakota, R.; and Robinson, M. (2003). *Services Blueprint: Roadmap for Execution.* Addison-Wesley.
- Smith, H.; and Fingar, P. (November 2003). "Digital Six Sigma" BPTrends.com.

About the Authors

Howard Smith is CTO (Europe) of Computer Sciences Corporation and cochair of the Business Process Management Initiative (BPMI.org). Computer Sciences Corporation has been applying BPM thinking to the design of enterprise architectures since 1998. He is the coauthor of *Business Process Management: The Third Wave* (www.bpm3.com).

Peter Fingar is an executive partner with the digital strategy firm, the Greystone Group. He delivers keynotes worldwide and is the author of *The Death of* "e" and *The Birth of the Real New Economy* and *Enterprise E-Commerce*. He is the coauthor of *Business Process Management*: The Third Wave (www.bpm3.com).

■■■ authors@bpm3.com

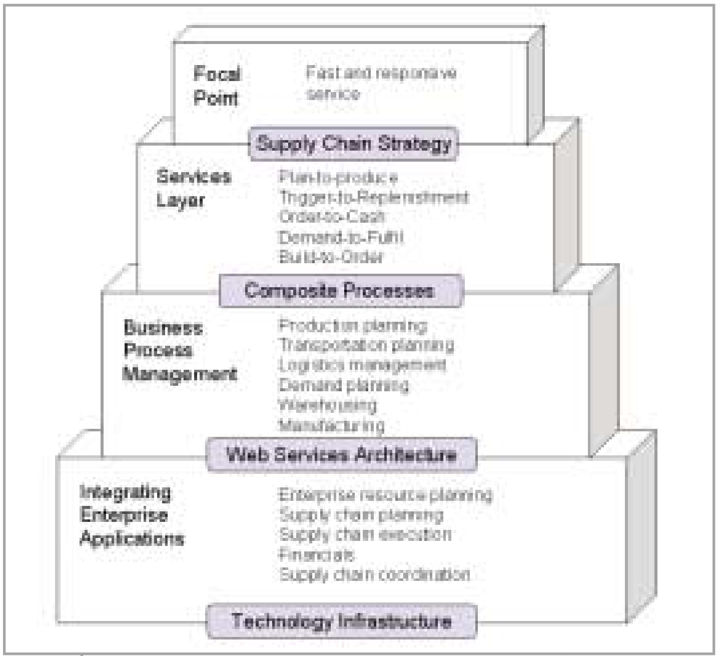


FIGURE 1 Kalakota and Robinson's Supply Chain Blueprint

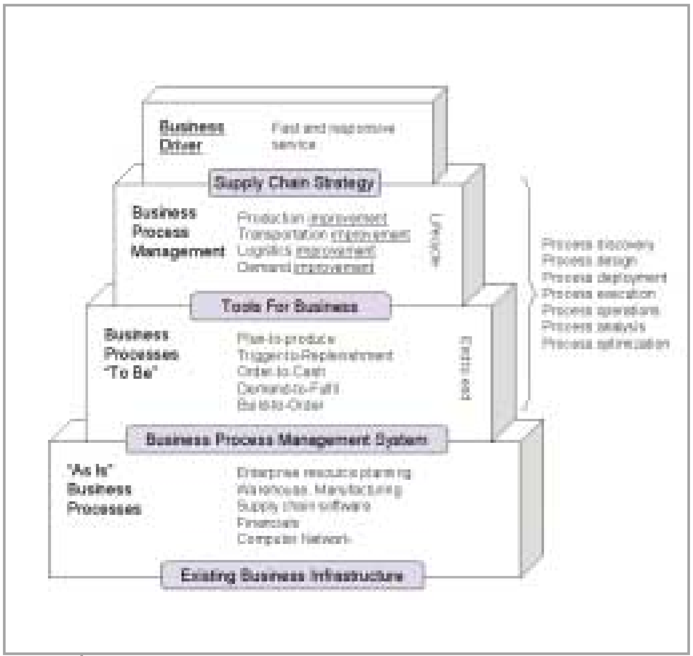


FIGURE 2 Smith and Fingar's Supply Chain BPM

An Accidental Web Services Tourist

Moving past the tripwires

My involvement in Web services was a mistake.

WRITTEN BY
MIKE MOSHER

I don't mean that I regret it, just that I got involved in Web services because of a mistake I made. It actually started when I was preparing an executive presentation on the current trends in security. I came across one of those juicy statistics that security people (like me) love to use. It said a survey found that security was the number one obstacle to companies implementing "Web services." This is just the kind of direct correlation between security and the deployment of a valuable technology that tends to make people sit up and take notice. Unfortunately, this was prior to the time that Web services received the buzz that it has today, and I mistakenly assumed that "Web services" (the continuing confusion over how to capitalize it did not help then or now) meant services on the Web – e-commerce Web sites, transactional Web sites, and so on. Once I discovered my error (after giving the presentation a couple of times – luckily this statistic was not the centerpiece of the message), I dedicated myself to learning much more about security and this new thing called "Web services."

So, what's changed since then? Well, my company provided me with a research grant to learn about Web services security, and I wrote a lengthy report on the subject. I have spoken at conferences, presentations, and webinars, but I'm far from being able to say that I know all there is to know on the subject. Why? Because, like so much of the Web services universe, the security space is in an almost constant state of flux. There are competing standards, proprietary solutions, and almost as many opinions as there are "experts" on the topic. Gartner recently recommended that companies should go ahead with using proprietary security protocols in their Web services development rather than waiting indefinitely for the standards to sort themselves out and gain sufficient acceptance.

> "My deep fear is that unless security in Web services can be made more transparent (and less expensive), the concerns over risk and the desire to do the right thing will give way to other priorities…"

What hasn't changed? Security remains the number one concern in developing and deploying Web services. I have to say that this is pretty comforting for someone who often has to fight to get people to pay attention to security as an important factor in systems development. Yet, although I would like to see security remain a prime focus when considering Web services, I see the need for this aspect to become easier. Gartner has stated that companies should be prepared to spend 50% of their Web services budget on security and, unfortunately, my experience has shown this to be true. My deep fear is that unless security in Web services can be made more transparent (and less expensive), the concerns over risk and the desire to do the right thing will give way to other priorities, like speed to market and lower cost, leaving security "undone."

Unfortunately, another thing that has not really changed is that many people are still not quite sure what Web services are. I continue to encounter a lot of confusion, very much along the lines of the confusion I had when I first started my journey into the world of Web services. This may slow adoption somewhat, but I think that ultimately the value of Web services will win over the uninitiated – like it did for me.

I was honored to be invited to become the security editor for ***Web Services Journal***. This month's issue is a recognition of security's role. My goal will be to keep the focus on security as a key component of Web services all year long. I'll be sharing my own thoughts from time to time, and I want to encourage those who have new ideas or creative thoughts that they would like to share around Web services security to look at using this forum to get the word out. It will take new ideas to turn security from an obstacle into something that enables Web services to take us as far as they can. I look forward to the journey.

About the Author

Michael Mosher is the technology director of the CSC Consulting Business and Technology Risk Management practice. He specializes in security architecture and security strategy, and has designed security solutions for Fortune 500 clients in financial services, manufacturing, energy, and health care. Michael has a broad background in government and commercial security, including six years as a special agent with the U.S. government investigating computer and white collar crimes.

wsjsecurity@sys-con.com

SYS-CON MEDIA

304,187 of the World's Foremost IT Professionals

DIRECT MAIL, EMAIL OR MULTI-CHANNEL

Target CTOs, CIOs and CXO-level IT professionals and developers who subscribe to SYS-CON Media's industry leading publications

Java Developer's Journal... The leading publication aimed specifically at corporate and independent java development professionals

LinuxWorld Magazine... The premier monthly resource of Linux news for executives with key buying influences

Web Services Journal... The only Web Services magazine for CIOs, tech, marketing & product managers, VARs/ISVs, enterprise/app architects & developers

XML-Journal... The world's #1 leading XML resource for CEOs, CTOs, technology solution architects, product managers, programmers and developers

PowerBuilder Developer's Journal... The only PowerBuilder resource for corporate and independent enterprise client/server and web developers

WebSphere Developer's Journal... The premier publication for those who design, build, customize, deploy, or administer IBM's WebSphere suite of Web Services software

ColdFusion Developer's Journal... The only publication dedicated to ColdFusion web development

WebLogic Developer's Journal... The official magazine for BEA WebLogic application server software developers, IT management & users

.NET Developer's Journal... The must read iTechnology publication for Windows developers & CXO management professionals

Wireless Business & Technology... The wireless magazine for key corporate & engineering managers, and other executives who purchase communications products/services

Recommended for a variety of offers including Java, Internet, enterprise computing, e-business applications, training, software, hardware, data back up and storage, business applications, subscriptions, financial services, high ticket gifts and much more.

NOW AVAILABLE!
The SYS-CON Media Database
304,187 postal addresses

e-POSTDIRECT
Your iMarketing Source

For email information:
contact Frank at 845-731-3832
frank.cipolla@epostdirect.com
epostdirect.com 800-409-4443 fax845-620-9035

For postal information:
contact Kevin at 845-731-2684
kevin.collopy@edithroman.com
edithroman.com 800-223-2194 fax845-620-9035

List Brokerage & Management
EDITH ROMAN
Data Processing • Email Marketing

Web Services Journal/XML-Journal Readers' Choice Awards

And the winners are...

Best Book

Understanding Web Services: XML, WSDL, SOAP, and UDDI

Addison-Wesley www.aw.com

This book introduces the main ideas and concepts behind core and extended Web services technologies and provides developers with a primer for each of the major technologies that have emerged in this space. In addition, *Understanding Web Services* summarizes the major architectural approaches to Web services, examines the role of Web services within the .NET and J2EE communities; and provides information about major product offerings from BEA, HP, IBM, IONA, Microsoft, Oracle, Sun Microsystems, and others.

Best App Server for Web Services

BEA WebLogic Server

BEA Systems www.bea.com

BEA WebLogic Server 7.0 includes several significant enhancements aimed at improving the productivity of the developer. It's about fewer manual steps, cleaner code, easy packaging, and ultimately getting the job done faster, with fewer people. It simplifies deployment of massive clustered applications by providing configuration wizards and two-phase deployment capabilities.

Best GUI for Web Services

IBM WebSPhere Platform

IBM www.ibm.com

IBM WebSphere is a high-performance and extremely scalable Internet infrastructure software, or middleware, for creating, running and integrating e-business applications across a variety of computing platforms. It is built on open technologies such as J2EE, XML, Eclipse, and the new Web services standards.

Best Portal Platform for Web Services
Best Web Services Utility
Best XML Database

Tamino XML Server 4.1

Software AG www.softwareag.com

Best App Server for Web Services

–Winner–

BEA WebLogic Server

BEA Systems www.bea.com

–First Runner Up–

IBM WebSphere Application Server v5

IBM www.ibm.com

–Second Runner Up–

Oracle 9i Applic ation Server

Oracle Corporation www.oracle.com

–Third Runner Up–

Sun ONE application Server 7.0

Sun Microsystems www.sun.com

Best Book

–Winner–

Understanding Web Services: XML, WSDL, SOAP, and UDDI

Addison-Wesley www.aw.com

–First Runner Up–

XMLSPY Handbook

Wiley www.wileypub.com

–Second Runner Up–

Building Web Services with Java

Sams www.pearsoned.com

–Third Runner Up–

XML and Java: Developing Web Applications, Second Edition www.aw.com

Best Framework for Web Services

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

BEA WebLogic Workshop

BEA Systems www.bea.com

–Second Runner Up–

IBM WebSphere Platform

IBM www.ibm.com

–Third Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

Best GUI for Web Services Product

–Winner–

IBM WebSPhere Platform

IBM www.ibm.com

–First Runner Up–

XMLSPY

Altova www.altova.com

–Second Runner Up–

Oracle 9i JDeveloper

Oracle Corporation www.oracle.com

–Third Runner Up–

Borland Delphi 7 Studio

Borland Software Corp. www.borland.com

Best Integrated Services Environment

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

BEA WebLogic Platform

BEA Systems www.bea.com

–Second Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Third Runner Up–

IBM WebSphere Application Server v5

IBM www.ibm.com

Best Portal Platform for Web Services

–Winner–

Tamino XML Server 4.1

Software AG www.softwareag.com

–First Runner Up–

BEA WebLogic Portal

BEA Systems www.bea.com

–Second Runner Up–

IBM WebSphere Portal v4.2

IBM www.ibm.com

–Third Runner Up–

Oracle 9i AS Portal

Oracle Corporation www.oracle.com

Best Service-Oriented Architecture

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Second Runner Up–

ebXML

OASIS

–Third Runner Up–

Novell exteNd

Novell, Inc. www.novell.com

Best Web Service Security Solution

Tamino XML Server is a high-performance XML Server for reliably storing, managing, publishing, and exchanging XML documents in their native format based on open-standard Internet technologies. It is available on all major operating systems from Windows to Unix. including Linux. Tamino XML Server provides advanced high-availability features and offers many easy-to-use interfaces, tools and functionalities that help increase developer and administrator productivity for companies of all sizes.

Best Framework for Web Services
Best Integrated Services Environment
Best Service-Oriented Architecture
Best Web Services Platform
Best Web Services Automation Tool
Best Web Services Integration Tool
Best Web Services Legacy Adapter
Best Web Services Management Tool/Platform

EntireX 7

Software AG www.softwareag.com

EntireX is Software AG's leading-edge integration server providing an efficient way to create reusable services from existing systems, turn back-end services into Web services, and manage the growing complexity of XML-based information exchange. While many vendors view a services-oriented architecture as a new technology, EntireX was designed from the ground up to support the creation of a services view of corporate IT systems. This patented approach has been validated by years of customer success.

Best Web Service Security Solution

IBM Tivoli Access Manager

IBM www.ibm.com

IBM Tivoli Access Manager for e-business expands on IBM's open standards–based security platform by offering interoperability with third-party e-business tools. The software helps customers integrate security across e-business infrastructures, including Web services applications, allowing companies to manage security across collaborative networks that span millions of users employing Web services technologies.

Best Web Services BPM Engine

BEA WebLogic Integration

BEA Systems www.bea.com

BEA WebLogic Integration is a single solution delivering application server, application integration, business process management, and B2B integration functionality for the enterprise. Designed to speed time to value, reduce the costs of IT initiatives, and future-proof businesses. BEA WebLogic Integration relies on a standards-based, "build to integrate" approach that enables companies to rapidly develop, deploy, and integrate new Web and wireless applications, streamline complex business processes, and connect with business partners.

Best Web Services IDE

BEA WebLogic Workshop

BEA Systems www.bea.com

BEA WebLogic Workshop is an integrated development framework that empowers all application developers, not just J2EE experts, to rapidly create, test, and deploy enterprise-class Web service applications on the BEA WebLogic Platform. It provides a unified development platform that enables developers to easily build and connect components, data, and application business logic, while insulating them from the complexities of J2EE.

Best Web Services or XML Site

The Tamino Developer Community

Software AG www.softwareag.com

The Tamino Developer Community Web site provides useful information, soft-

–Winner–

IBM Tivoli Access Manager

IBM www.ibm.com

–First Runner Up–

Netegrity SiteMinder and TransactionMinder

Netegrity www.netegrity.com

–Second Runner Up–

RSA ClearTrust 5.0

RSA Security www.rsasecurity.com

–Third Runner Up–

VordelSecure

Vordel www.vordel.com

Best Web Services Platform

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Second Runner Up–

XMLSPY

Altova www.altova.com

Novell exteNd

Novell, Inc. www.webmethods.com

Best Web Services Automation Tool

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

IBM WebSphere Studio v5

IBM www.ibm.com

–Second Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Third Runner Up–

XMLSPY

Altova www.altova.com

Best Web Services BPM Engine

–Winner–

BEA WebLogic Integration

BEA Systems www.bea.com

–First Runner Up–

IBM WebSphere Busines Integration v4.2

IBM www.ibm.com

–Second Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Third Runner Up–

Savvion BusinessManager

Savvion, Inc. www.savvion.com

Best Web Services IDE

–Winner–

BEA WebLogic Workshop

BEA Systems www.bea.com

–First Runner Up–

IBM WebSphere Studio
(Application Developer v5.0)

IBM www.ibm.com

–Second Runner Up–

Tamino Mobile Studio

Software AG www.softwareag.com

–Third Runner Up–

Oracle9i JDeveloper

Oracle Corporation www.oracle.com

Best Web Services Integration Tool

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

BEA WebLogic Integration

BEA Systems www.bea.com

–Second Runner Up–

IBM WebSPhere MQ Integrator Broker

IBM www.ibm.com

–Third Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

Best Web Services Legacy Adapter

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

IBM WebSphere Host Integration Solution

IBM www.ibm.com

–Second Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Third Runner Up–

Novell exteNd Composer

Novell, Inc. www.webmethods.com

Best Web Services Management Tool/Platform

–Winner–

EntireX 7

Software AG www.softwareag.com

–First Runner Up–

webMethods Integration Platform

ware, guidance, and help to developers who intend to build XML-based software applications, solutions, or products, probably by using Software AG's Tamino XML Server. The Developer Community offers public discussion forums on a variety of topics around the Tamino XML Server.

Best Web Services Testing Tool
IBM alphaWorks Web Services Testing Area

IBM www.ibm.com

alphaWorks provides a unique opportunity for developers around the world to experience the latest innovations from IBM. These emerging "alpha code" technologies are available for download at the earliest stages of development – before they are licensed or integrated into products – allowing users to evaluate and influence IBM research and development.

Best Web Services or XML Training
Software AG's DemoZones

Software AG www.softwareag.com

Software AG established two DemoZone sites focusing on benefits that can be achieved using XML, Web services, and Software AG's products for integration and XML storage. The sites offer visitors a look into specific real-life scenarios to experience the solution space of Software AG's XML and integration technology.

Best XML Parser
IBM XML Parser for Java v4.1.2

IBM www.ibm.com

XML Parser for Java is a validating XML parser and processor written in 100% pure Java; it is a library for parsing and generating XML documents. This parser easily enables an application to read and write XML data.

Best XSLT Processor
XMLSPY

Altova www.altova.com

XMLSPY has a blazing fast XSLT processor that supports XSLT debugging and visual XSLT design.

Most Innovative Application of XML
Tamino Mobile Appllications

Software AG www.softwareag.com

Software AG's Tamino Mobile Application for Field Service Automation (FSA) provides service-oriented companies that maintain and repair their customers' technical equipment or devices with a complete, ready-to-run solution for ubiquitous access to service-relevant data. The Tamino Mobile Application for FSA provides specific services for service-order management so that starting from a service-order entry the whole service process can be managed.

webMethods www.webmethods.com

–Second Runner Up–

IBM Tivoli Monitoring for Transaction Performance

IBM www.ibm.com

–Third Runner Up–

XMLSPY

Altova www.altova.com

Best Web Services or XML Site

–Winner–

The Tamino Developer Community

Software AG www.softwareag.com

–First Runner Up–

IBM developerWorks

IBM www.ibm.com

–Second Runner Up–

XML.org

OASIS www.oasis-open.org

–Third Runner Up–

XMethods

XMethods www.xmethods.net

Best Web Services Testing Tool

–Winner–

IBM alphaWorks Web Services Testing Area

IBM www.ibm.com

–First Runner Up–

XMLSPY

Altova www.altova.com

–Second Runner Up–

iON Remote: QoS for Web Services

Santra Technology www.santra.com

–Third Runner Up–

Oracle 9i JDeveloper

Oracle Corporation www.oracle.com

Best Web Services or XML Training

–Winner–

Software AG's DemoZones

Software AG www.softwareag.com

–First Runner Up–

BEA dev2dev

BEA Systems www.bea.com

–Second Runner Up–

IBM Web Services toolkit v3.3

IBM www.ibm.com

–Third Runner Up–

Mindreef SOAPscope

Mindreef, Inc. www.mindreef.com

Best Web Services Utility

–Winner–

Tamino XML Server 4.1

Software AG www.softwareag.com

–First Runner Up–

IBM Web Services Toolkit v3.3

IBM www.ibm.com

–Second Runner Up–

XMLSPY

Altova www.altova.com

–Third Runner Up–

iON Remote: QoS for Web Services

Santra Technology www.santra.com

Best XML Database

–Winner–

Tamino XML Server 4.1

Software AG www.softwareag.com

–First Runner Up–

Oracle XML DB

Oracle Corporation www.oracle.com

–Second Runner Up–

Ipedo 3

Ipedo, Inc. www.ipedo.com

–Third Runner Up–

XML Global GoXML DB

XML Global Technologies, Inc.
www.xmlglobal.com

Best XML Parser

–Winner–

IBM XML Parser for Java v4.1.2

IBM www.ibm.com

–First Runner Up–

XMLSPY

Altova www.altova.com

–Second Runner Up–

webMethods Integration Platform

webMethods www.webmethods.com

–Third Runner Up–

Oracle XDK

Oracle Corporation www.oracle.com

Best XSLT Processor

–Winner–

XMLSPY

Altova www.altova.com

–First Runner Up–

IBM LotusXSL

IBM www.ibm.com

–Second Runner Up–

Oracle XDK

Oracle Corporation www.oracle.com

–Third Runner Up–

Stylus Studio

Sonic Software www.sonicsoftware.com

Most Innovative Application of XML

–Winner–

Tamino Mobile Applications

Software AG www.softwareag.com

–First Runner Up–

IBM WebSphere Portal v 4.2

IBM www.ibm.com

–Second Runner Up–

XMLSPY

Altova www.altova.com

–Third Runner Up–

ColdFusion MX

Macromedia www.macromedia.com

LOOK FOR YOUR FREE...

SYS-CON MEDIA

›Linux ›Java ›Web Services ›.NET ›XML ›Wireless ›Storage ›Security

The Premier Resource for Today's Corporate & IT Decision Makers

VOL: 1 ISSUE: 1 SPRING 2004

IT solutions ›GUIDE

WWW.SYS-CON.COM/IT

› Delivering Software as Service

› Leveraging Linux/Open Source

› Moving to a Service-Oriented Architecture

› Desktop Software: Migrating from Server to Client

› Using Developer Tools to Drive Cost Out of Software

› Application Integration

› Storage & Security

TECHNOLOGIES YOU NEED NOW!

How to Manage Your Ideas Using Today's *i*-Technologies

Reaching 135,000 Corporate and IT Decision Makers

Coming this **SPRING!**

Grand Central Communications Launches New Edition of Network

(San Francisco) – Grand Central Communications, Inc., provider of the Business Services Network, has announced the immediate availability and deployment of Grand Central 4.0, the newest release of its managed network service that delivers "Integration-as-a-Service". This release offers the complete capabilities of a standards-based integration technology platform in a simple-to-use, self-service, Web-based interface that changes the complex work of integration into a simple configuration task, and enables business processes to be rapidly built and deployed. New enhancements include Grand Central's Business Services Directory, self-service provisioning, advanced Web-based business process orchestration tools, and expanded support for connectivity protocols. www.grandcentral.com

iSpheres Releases Version 4 of Real-time Event Server

(Oakland, CA) – iSpheres Corporation, maker of real-time event management platforms, has announced the availability of Version 4 of iSpheres Halo Event Server. iSpheres Halo allows organizations to rapidly deploy applications that monitor critical events from distributed sources and business processes and trigger alerts according to user-defined criteria.

iSpheres also announced the iSpheres BAM (Business Activity Monitoring) Pilot Program for IT developers who want to download iSpheres Halo and configure BAM applications in one week. www.ispheres.com

BEA, Microsoft, TIBCO Software Publish New Web Services Specification

Microsoft, BEA System, and Tibco have released a proposed specification to help communicate events between Web services. Events are occurrences in the real world that can trigger actions in software: a phone rings, an order is placed, a package is shipped, a printer runs out of paper, your favorite team scores, a stock hits a new high, etc. These real-world events need to be mirrored within the technologies that have become fundamental to our personal lives and businesses.

With WS-Eventing, every Web service can send and receive information about events that have occurred, regardless of whether the event originates in the firmware of a simple device or in large–scale enterprise systems. The co-authors are proposing a set of fundamental protocols, message formats, and interfaces for a Web service to subscribe to events coming from another Web service. The resulting specification is flexible enough to be applied to scenarios that span the enterprise, the home, and devices and can form the basis of more complex vertical solutions in the future. http://dev2dev.bea.com/technologies/webservices/standards.jsp, http://msdn.microsoft.com/ws/2004/01/ws-eventing

DreamFactory Launches Tool for Developing Web Service Client Interfaces

(Los Gatos, CA) – DreamFactory Software has announced DreamFactory 6.0, a suite of browser-based software tools that enable organizations to develop rich client user interfaces for enterprise Web applications powered by native XML documents and Web services. The DreamFactory tools streamline user interface design, enhance end-user experience, reduce network traffic, and lower development costs by enabling the aggregation of multiple data sources on a single browser page without additional server software.

DreamFactory's approach enables the delivery of software as a service. It allows the aggregation of external information sources into information centers. www.dreamfactory.com

Mindreef Announces Availability of SOAPscope 3.0 Web Services Diagnostics System

(Hollis, NH) – Mindreef, Inc. has announced the availability of Mindreef SOAPscope 3.0, a Web services diagnostics system to include life-cycle interoperability testing and troubleshooting throughout the development, deployment, and support of a Web service. This new release also offers Microsoft Visual Studio .NET integration and the ability to graphically query message logs to quickly pinpoint performance bottlenecks and other problems.

New message analysis and WS-I Testing Tools integration, combined with WSDL analysis that was released in version 2.0, complete SOAPscope's interoperability solution. WSDL and message analysis give developers, testers, and support personnel the ability to quickly detect and solve Web services interoperability issues throughout an application's life cycle by highlighting the exact part of a message or WSDL that is invalid and providing a detailed, understandable description of each problem. In addition to its built-in interoperability checking, SOAPscope also integrates and simplifies the use of WS-I Java- and C#-based tools. www.mindreef.com

Confluent Web Services Management Platform 3.5 Reaches Everywhere

(Sunnyvale, CA) – Confluent Software Inc., a provider of Web services management solutions for the enterprise, has announced that Confluent Web Services Management Platform 3.5 is now deployed at leading Global 500 manufacturing and financial services companies. The company also announced it will deliver operational best practice enforcement components for TIBCO and C++ environments.

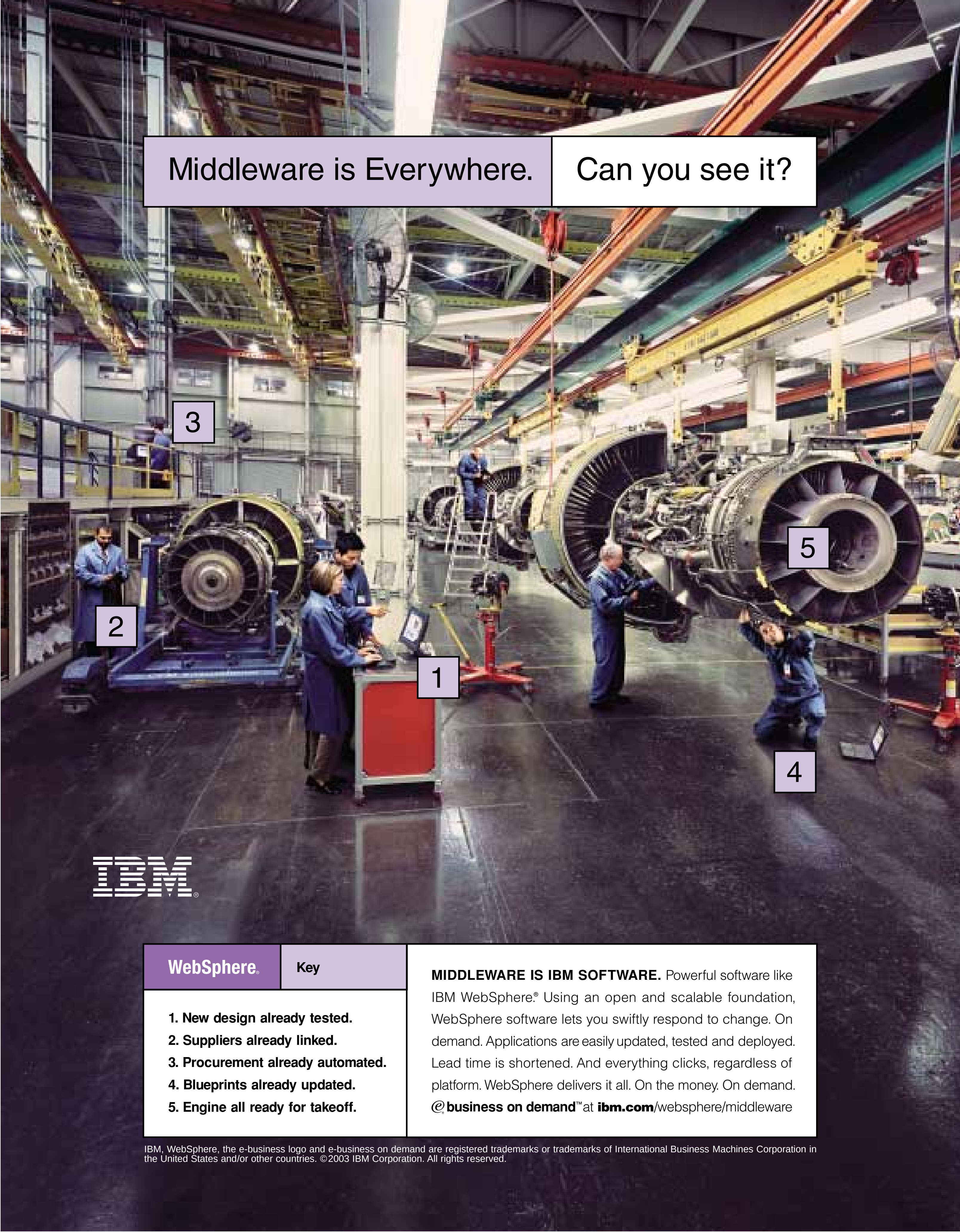
The introduction of Confluent Agent for TIBCO and Confluent Agent for C++ extends Confluent's support for multiple platforms and transports already available in this release of the Confluent Web Services Management Platform. Confluent Agent for TIBCO helps users enforce operational rules and monitor interactions between business applications connected by TIBCO BusinessWorks, allowing customers to use Confluent to ensure every service request, to or from business applications participating in a managed business process, complies with security and quality-of-service best practices. www.confluentsoftware. com

Parasoft Releases SOAPtest 2.5 for Comprehensive Web Services Testing

(Monrovia, CA) – Parasoft, a provider of Automated Error Prevention (AEP) software solutions, has announced the release of SOAPtest 2.5, a comprehensive Web services testing product, verifying every aspect of a Web service from WSDL validation to client/server unit and functional testing to performance testing.

This version offers support for WS-Security, MIME attachments, enhanced load-testing features, and other features designed to help development teams prevent errors and accelerate time to market for their Web service initiatives.

SOAPtest 2.5 is available for Windows 2000, Windows XP, Linux and Solaris. Pricing starts at $3995. www.parasoft.com


Middleware is Everywhere.
Can you see it?
3
2
1
5
4
IBM