

Best Practices II

PRINT

EMAIL

Web Services and SOA

Cities and states tap a new IP-centric approach to data sharing.

Karen D. Schwartz

In today's do-more-with-less world, anything that increases efficiency, saves money or improves customer satisfaction is sought after.

Nowhere is that more true than in the public sector. For an increasing number of state and local governments, the road to efficiency seems to be paved with a service-oriented architecture (SOA) — a method of analyzing and categorizing government services so that common services can be shared across agencies and interdepartmental units.

Government is an especially good fit for Web services and SOA, says Yefim Natis, vice president of software infrastructure for Gartner of Stamford, Conn.

“Government agencies have to deal with massive numbers of users, so everything they do has to be standards-based because they can't impose anything on their constituents. People will be accessing these services from all types of devices and protocols. They have to go with the lowest common denominator, and Web services is that — it supports everything,” he says.

But that's just one part of the equation. In addition to serving its citizen base, agencies also must share information and communicate with one another, sometimes across city and state boundaries.

“If they choose to use Java or a specific messaging product instead of Web services, they can't assume the other side is using the same,” Natis says. “The only way to assure connection without additional cost is through Web services.”

For some government organizations, using Web services in conjunction with SOA makes good sense. That's the case for Minnesota, which currently is embarking on a lengthy project to define a state SOA that eventually will incorporate Web services.

“Web services have always been our standard, long before we introduced the concept of SOA,” says Michael P. Ryan, enterprise architect with the Minnesota Office of Technology in St. Paul, Minn. “We've been using HyperText Transfer Protocol, Simple Object Access Protocol [SOAP], virtual private networks [VPNs] and Extensible Markup Language [XML] since 2000, and Web services just seem like the de facto standards for building new applications.”

But it's a long process, and Minnesota is just beginning. “I don't think we're any further along than anyone else,” Ryan says.

Diving In

For other states, the decision on whether to incorporate Web services came early in the process. That's the case in Wisconsin, where enterprise integration architect Allen Poppe says the Division of Enterprise Technology chose to



CURRENT ISSUE
April/May 2009



Most Popular Articles

- Courting Millennials
- The Promise of Paperless
- Advanced Surveillance Networks
- Hazardous Conditions Ahead
- Out Sick
- How to Keep Top IT Talent
- Moving Maps From the Sky
- Is Municipal Broadband Right for You?
- Home Work
- IT Brings Home the Bacon

incorporate Web services into its SOA from the get-go.

“It’s one of those things where industry standards have evolved, and it’s quite mature,” he says. “We plan to do more of our interactions via the Web, so Web services is a natural extension.”

Chicago also has been on the forefront of using Web services to help propel its SOA effort. Although the city’s project did not initially begin with a focus on Web services, Chicago CIO Hardik Bhatt says that as the development team began orchestrating changes and as Web services themselves began to mature, they slowly became a fairly critical part of the SOA project.

“Early on, Web services weren’t mature enough. But as time progressed, Web services and the tools available to develop them started becoming available, and we knew it was the right way to go,” Bhatt says. “It became easier to build Web services and have heterogeneous systems talk to each other.”

In general, the benefits of using Web services in conjunction with SOA are fairly straightforward. Not only do Web services make it easier to access systems regardless of location, platform and implementation language, but various standards have been defined relating to security, policy, reliability and interoperability.

Another major driver for incorporating Web services is lower integration costs and a correspondingly high return on investment.

Because Web services generally use open standards such as SOAP or Web Services Description Language, they can run on a variety of hardware and software platforms, letting organizations leverage their existing investments in platforms and technologies.

Increased functionality is another bonus — something Chicago’s Bhatt hopes to achieve. The city’s inspection and permitting system, for example, runs a Visual Basic application and needs to interface with a 311 customer-service system built on an Oracle system. The city has been testing a Web services link that, Bhatt says, will not only allow for greater service to its citizens but will avoid a great deal of time and expense.

“Up until now, our inspectors were only able to do annual inspections generated on the license [renewal] date, and they couldn’t incorporate other reasons for inspections,” he says. “Citizens generally call 311 to complain about something such as an unstable porch, and that information would go into the CSR [computer systems research] system, which they would print out and give to the building inspectors manually. With the interface, they will be able to handle all inspections for a building, both annual and through complaints, at one time. It saves a lot of time.”

Bhatt credits the effort in his city to the push that his application architect, Clinton Chow, gave to adopting SOA and Web services. It was Chow who focused the effort and made the case for using Web services to improve data sharing.

Overcoming Hurdles

In addition to the positives, however, there are challenges when implementing Web services in conjunction with SOA.

The biggest is security, mostly in terms of authentication and authorization of users. Several standards exist to address the security issues inherent in Web services, including the Web Services Security specification from the Organization for the Advancement of Structured Information Standards (OASIS), an international consortium for e-business standards.

WS-Security defines the mechanism for including authentication, integrity and confidentiality within SOAP messages, using specifications like XML Signature and XML Encryption to provide for digital signatures, encryption and message digests. The Security Assertion Markup Language, also created by OASIS, lets partners share user authentication and authorization information across multiple trust domains, says Pradipa Karbhari, national director for Web services and SOA at SilverTrain, a business intelligence consultancy in Milwaukee.

Integration challenges also can cause roadblocks, which often arise when working with legacy systems. That’s been a problem in Wisconsin. The state’s current hurdle, Poppe says, is how to handle the many jurisdictional and agency partners that are not in a position technically to implement Web services.

“For us and most heterogeneous enterprises, the challenge is to reach critical mass with a successful Web services-based architecture while making provisions for those partners we don’t want to marginalize because they aren’t able to stay on the leading edge with us,” he says.

For Chicago, the biggest hiccup isn’t security, integration or the process itself, it’s convincing stakeholders that Web services is the right way to go.

“The biggest challenge isn’t technical, it’s business-related,” Bhatt says. “Generally [stakeholders] want things tomorrow, and Web services requires patience. We have to explain that while there is a lot of initial cost involved in building the first

interface and a lot of time involved in getting it up and running, once you expose that system, you start reaping the benefits. Now we can build interfaces very quickly, and that's where you really begin to see the ROI.”

The Bits and Bytes of Web Services

Although people often refer to Web services as a generic term, there are actually many technologies through which organizations can develop Web services.

Web services, for example, can be built using any technology, such as Java, Visual Basic or Microsoft.NET. Web services are basically software components that are loosely coupled and exposed via well-defined interfaces, allowing other applications to be built that leverage the exported functionality.

These services can be described using standard formats, such as the Web Services Description Language, and accessed or called by open standards-based mechanisms, using protocols like the Simple Object Access Protocol and Extensible Markup Language over the HyperText Transfer Protocol.

Although a single Web service by itself could provide useful functionality, substantial return on investment with minimal risk is derived by assembling multiple services into composite applications over a well-governed service-oriented architecture framework.

Hot-Ticket Items in Web Services Security Specs

- Multiple security token formats n Multiple trust domains
- Multiple signature formats
- Multiple encryption technologies
- End-to-end message content security (not just transport-level security)

SOURCE: OASIS, www.oasis-open.org

