THE STENCIL GROUP

THE STENCIL SCOPE

An Analysis Memo from The Stencil Group

Defining Web Services

(June 2001) The label "web services" is incredibly generic. Like any promising and loosely defined technology trend, the concepts it describes will be subject to a great deal of speculation and bandwagoneering in the months to come. With the aim of providing a reference benchmark—and of separating posturing from reality—we provide a technology and business definition. By Brent Sleeper and Bill Robins.

The technology marketing industry has mastered the hype lifecycle, but rarely have we seen a concept emerge from obscurity to new new thing as quickly as "web services." While we fundamentally believe that the concept, promise, and implementation of web services are sound, these virtues run the risk of getting lost in a rush of marketing showmanship.

Part of the problem is that the web services label is incredibly generic; while perhaps more accessible than techno-jargon like UDDI, WSDL, and all the other acronyms that collectively define the concept, such a generic term easily can be misapplied in ways that dilute its meaning. Indeed, we frankly expect that many companies will co-opt the web services label in ways that benefit their short-term positioning, but which ultimately sow confusion in the marketplace.

To counter that confusion and to provide an introduction to basic web services concepts, this memo documents three facets of The Stencil Group's definition of this technology:

- **What Are Web Services?** A conceptual definition of the web services model.
- **The Web Services Technology Stack.** Documentation of the basic technologies that compose the web services framework.
- **Web Services FAQ** A brief list of frequently asked questions about web services.

> *Author's note:* The Stencil Group's recent article, "How Web Services Will Beat the 'New New Thing' Rap," explores the issues of web services hype and reality (http://www.stencilgroup.com/ideas_scope_200106newnew.html).

Brent Sleeper (bsleeper@stencilgroup.com) is a partner in The Stencil Group. He explores the intersection of Internet technologies and e-business strategies. Bill Robins (bill@stencilgroup.com) is also a Stencil partner. He focuses on venture financing, business models, and the emerging e-business landscape.

What Are Web Services?

The label "web services," as broadly applied, has two levels of meaning—one specific and one conceptual:

- Specifically, web services are a stack of emerging standards that describe a service-oriented, component-based application architecture.

- Conceptually, web services represent a model in which discrete tasks within e-business processes are distributed widely throughout a value net.

With these meanings in mind, The Stencil Group defines web services as:

Loosely coupled, reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols.

Let's examine this definition and deconstruct its meaning.

- First, web services are *reusable software components*. Web services continue the long ascension of object-oriented design in software development. Rather than requiring programmers to write one start-to-finish set of instructions after another, the component-based model allows developers to reuse the building blocks of code created by others to assemble and extend them in new ways.

- Second, these software components are *loosely coupled*. Traditional application design depends upon a tight interconnection of all subsidiary elements. The complexity of these connections requires that developers thoroughly understand and have control over both ends of the connection; moreover, once established, it is exceedingly difficult to extract one element and replace it with another. Loosely coupled systems, on the other hand, require a much simpler level of coordination and allow for more flexible reconfiguration.

- Third, web services *semantically encapsulate discrete functionality.* A web service is a self-contained "applet" that performs a single task. The component describes its own inputs and outputs in a way that other software can determine what it does, how to invoke its functionality, and what result to expect in return.

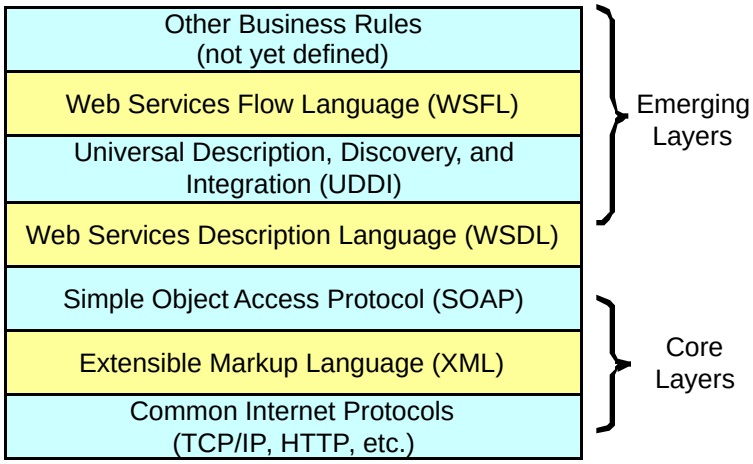
- Fourth, web services can be *accessed programmatically.* Unlike web sites and desktop applications, web services are not designed for direct human interaction, and they do not have a graphical user interface. Rather, web services operate at the code level; they are called by and exchange data with other software. Web services certainly will be incorporated into software designed for human interaction, however.

- Finally, web services are *distributed over the Internet*. Web services make use of existing, ubiquitous transport protocols like HTTP. By piggybacking on the same, well-understood transport as web content, web services leverage existing infrastructure and can comply with current corporate firewall policies.

The Web Services Technology Stack

By intent, web services are not implemented in a monolithic way, but rather represent a collection of several related technologies. At a bare minimum, any web service entails a connection between two applications—in programmers' parlance, a remote procedure call (RPC)—in which queries and responses are exchanged in XML over HTTP. The more generally accepted definition, however, implies implementation of a stack of specific, complementary standards (see Figure 1, below).

Figure 1: The Web Services Technology Stack
(The Stencil Group, 6/2001)



Today, the core layers that define basic web services communication have been widely accepted and likely will be implemented quite uniformly. Higher-level layers that define strategic aspects of business processes remain an open question, however, and it is possible that divergent approaches will emerge.

The development of generally open and accepted standards is a key strength of the coalitions that have been building web services infrastructure. At the same time, these efforts have resulted in a dizzying array of jargon and acronyms. We have provided high-level descriptions of the most important ones below.

Core Layers of the Web Services Stack

- **Common Internet Transport Protocols.** Although not specifically tied to any transport protocol, web services build on ubiquitous Internet connectivity and infrastructure to ensure nearly universal reach and support. In particular, web services will take advantage of HTTP, the same connection protocol used by web servers and browsers.

- **Extensible Markup Language (XML).** XML is a widely accepted format for exchanging data and its corresponding semantics. It is a fundamental building block for nearly every other layer in the web services stack.

- **Simple Object Access Protocol (SOAP).** SOAP is a protocol for messaging and RPC-style communication between applications. It is based on XML and uses common Internet transport protocols like HTTP to carry its data. SOAP has been submitted to the World Wide Web Consortium (W3C) standards body and will emerge later this year as "XML Protocol (XP)."

Higher-Level Layers of the Web Services Stack

- **Web Services Description Language (WSDL).** WSDL is an XML-based description of how to connect to a particular web service. A WSDL description abstracts a particular service's various connection and messaging protocols into a high-level bundle and forms a key element of the UDDI directory's "green pages." IBM recently submitted WSDL to the W3C, and it will likely be adopted in some form.

- **Universal Description, Discovery, and Integration (UDDI).** UDDI represents a set of protocols and a public directory for the registration and real-time lookup of web services and other business processes. UDDI's sponsors, chiefly IBM and Microsoft, officially released the first public version of UDDI in May 2001. A few more revisions to the specification are planned before UDDI is turned over to a standards organization some time during the next 12 months. (For more information on UDDI, please see The Stencil Group's recent report, "Why UDDI Will Succeed, Quietly.")

- **Web Services Flow Language (WSFL).** WSFL is the least developed of the current web services layers. Sponsored by IBM, the WSFL team hopes to define a framework that implementers of web services can use to describe the business logic required to assemble various services into an end-to-end business process.

- **Other Business Rules.** Additional elements that support complex business rules must still be implemented before web services can automate truly critical business processes. Indeed, we expect that mechanisms for security and authentication, contract management, quality of service, and more will soon follow—some as standards, others as value-added solutions from independent software vendors.

Other Related Technologies

- **XML-RPC.** XML-RPC represents the simplest form of web service style connections. As the name indicates (XML-RPC means "XML Remote Procedure Call"), the protocol was a loosely-defined spin-off from the efforts that ultimately led to the SOAP specification. Although not part of the "classic" web services stack we define earlier in this document, XML-RPC can be used in to achieve similar benefits for less structured connections.

- **ebXML.** Efforts to define a standard XML format for exchanging e-business related information predate the recent rise of web services. Although sometimes presented as an alternative to a web service model, ebXML is focused more specifically on EDI-style information exchange. OASIS, the group developing ebXML, recently adopted SOAP as a key element of its specification, and we expect many businesses will incorporate

ebXML into their overall web services strategies. ebXML was adopted by UN/CEFACT standards body in May 2001.

Web Services FAQ

In the course of The Stencil Group's web services research, we have heard a number of common questions raised in conversations, at informational seminars, and in the press. We have addressed some of these frequently asked questions below.

- **What are web services?**

  Web services are a group of closely related, emerging technologies that describe a service-oriented, component-based application architecture that is based on an open, Internet-centric infrastructure. Web services represent a model in which discrete tasks within e-business processes are distributed widely throughout a value net. Web services components can be recombined by other companies to meet the needs of their own software applications or business processes.

- **What is an example of a web service?**

  One simple example of a web service may be an auction engine like eBay's. The eBay web site offers a highly successful auction service. Today, however, if Widgets-R-Us, a business that sells surplus widgets, wants to add auction functionality to its own business model, it needs to develop its own auction software from scratch or redirect customers to a site like eBay. With web services, eBay could syndicate its auction functionality and make it available to other web sites or applications (presumably for a fee). Companies like Widgets-R-Us would simply subscribe to eBay's web service, add a few lines of code to their own applications to incorporate the web service, and they instantly have private-labeled auction functionality available on their own sites. Other customer-facing examples include stock quotes, content syndication, mapping services, and so on. Some more enterprise-centric services may include payroll management, shipping and logistics, business intelligence, credit scoring, etc.

- **What are the technologies that make up web services?**

  Web services are not a specific technology, but rather a group of established and emerging communication protocols that include HTTP, XML, Simple Object Application Protocol (SOAP), Universal Description Discovery and Integration (UDDI), and Web Services Description Language (WSDL). A web service can be developed on any computer platform and in any development environment, as long as it can communicate with other web services using these common protocols.

- **How are web services different from earlier models of distributed computing?**

  Web services draw extensively from previous models of component-based computing, including CORBA and others. The key differences are that web services are (1) loosely specified and coupled and (2) built on top of existing, ubiquitous infrastructure like

HTTP and XML. Web services seem poised for success, because the technology's backers have, so far, focused on incremental technology changes and the mantra of "keep it simple."

- **How is this different than EAI?**

  Some applications of web services are related to the broad category of enterprise application integration (EAI) solutions. The differences are three-fold. First, EAI solutions link existing, monolithic applications into a common infrastructure, while web services are designed to allow for smaller, modular functionality that can be assembled and reassembled into dynamic processes. Second, most EAI technologies are designed to form discrete, pre-specified connections, while web services enable open-ended, one-to-many connections. Finally, EAI solutions' "all or nothing" models require a significant commitment of strategy and resources, while web services can be deployed with incremental cost and effort.

- **Are web services a variation of the application service provider (ASP) model?**

  Although ASPs and web services both implement the concept of "software as a service," the similarities end there. ASPs deliver entire applications from a central hosting location, while web services are distributed components. ASPs form a closed "black box," while web services are inherently extensible. ASPs are as much business model as technology solution; web services may enable new forms of business models, but are fundamentally a technology solution.

- **Are web services a business model?**

  Web services are not a business model. Although they will enable a shift in the way software companies deliver software and price software, web services are a model for technology development that is not limited to any particular business model

- **Who developed the standards for web services?**

  Web services represent a number of complementary (and sometimes coordinated) efforts by individual companies and coalitions of software vendors. Some of the web services protocols have been formally standardized by independent organizations like W3C. HTTP and XML are the existing foundation upon which DevelopMentor, UserLand Software, and Microsoft developed SOAP. After the initial SOAP specification, IBM and Ariba joined Microsoft to develop UDDI. IBM has since put its muscle behind a number of additional web services specifications, including WSDL, WSFL, and others.

- **What companies are the major forces in web services?**

  This is an open question. IBM and Microsoft have taken the early leadership mantle in establishing the web services stack. Hewlett-Packard, an early proponent of a proprietary services model, now is reorienting towards web services. Oracle, Sun, and a host of smaller companies (e.g. BEA, Bowstreet, and others) have also put hats into the ring.

- **What are the relationships between web services and HP's NetAction, Microsoft's .NET, Sun ONE, etc.?**

  Each of these labels represents the individual vendor's strategy and marketing efforts to establish a role for their products in the web services world. Some of these strategies clearly are articulated and developed, while others are little more than unspecified vision and "slideware" today.

Related Articles

- Karl Gottschalk et al, "Web Services Architecture Overview," IBM developerWorks, http://www-106.ibm.com/developerworks/webservices/library/w-ovr/.
- Mary Kirtland, "Web Services Essentials," Microsoft Developer Network, http://msdn.microsoft.com/library/techart/webservicesessentials.htm.
- Bill Robins, "How Web Services Will Beat 'The New New Thing' Rap," The Stencil Group, http://www.stencilgroup.com/ideas_scope_200106newnew.html.
- Brent Sleeper, "Why UDDI Will Succeed, Quietly: Two Factors Push Web Services Forward," The Stencil Group, http://www.stencilgroup.com/ideas_scope_200104uddi.html.
- Venu Vasudevan, "A Web Services Primer," XML.com, http://www.xml.com/lpt/a/2001/04/04/webservices/index.html.

For More Information

- ebXML.org, www.ebxml.org
- SoapWare.org, www.soapware.org
- UDDI.org, www.uddi.org
- Webservices.org, www.webservices.org
- XML.com, www.xml.com

About The Stencil Group

The Stencil Group provides business strategies and implementation services to companies focused on the business Internet. We help our clients—early-stage companies and established companies entering new markets—go to market quickly, professionally, and with the proper strategy in place. Stencil's services help compress the time required for our clients to scale their operation. The result is accelerated customer acquisition, retention, and revenue.

Web Services Research

To learn more about The Stencil Group's web services research plan, or if you would like to **participate in future memos**, please visit our web site at www.stencilgroup.com. We are available to discuss this memo in more detail; please feel free to contact us directly at (415) 615-0636 or stencils@stencilgroup.com.