

ZAPTAKE: WEB SERVICES

Brought
to you by:



There are few topics in information technology to have come along since the dot.com bust that have garnered as much hype as *Web Services*. Overstated, poorly named, and often misunderstood, Web Services have come to represent either a new approach to computing or little more than a way to simplify integration. Unfortunately, many industry pundits have done little but muddy the waters surrounding Web Services, so here is ZapThink's take.

Fundamentally, Web Services are standards-based interfaces to software functionality. If all applications spoke the same language, there would be no need for Web Services. Unfortunately, the reality is that in today's heterogeneous IT environment, there is no consensus on how applications should communicate. That's where Web Services come in. What makes Web Services special is the fact that the entire IT industry has settled on open standards for application-to-application communication. Because of these standards, it's possible—in theory—to interact with functionality exposed as a Web Service knowing little about that Service except for the interface it exposes. Therein lies the power of Web Services.

The Brass Tacks

A strict definition of Web Services is *encapsulated, loosely coupled, contracted software objects offered via standard protocols*. Essentially, Web Services are interfaces to application functionality residing on systems that accept requests from other systems locally or across the Internet. Breaking down the definition should make it clearer:

- *Encapsulated* means that the implementation of each Web Service is invisible from outside the Web Service. Its functionality is known only by the interface it exposes.
- Web Services can be *loosely coupled*. Loosely coupled means that Web Services and the programs that invoke them (known as Web Service consumers) can be changed independently of each other, instead of requiring a redesign of the involved components.
- *Contracted* means that the Web service's behavior, as well as how to connect, or bind to it, and its input and output parameters, are available to those consumers who are able to access it.
- Web Services are built upon *standard protocols*, including XML (eXtensible Markup Language), as well as SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery, and Integration), which are all standard protocols based upon XML.

This strict definition, however, only tells a part of the story. On the one hand, Web Services reduce the cost and complexity of integration. On the other hand, their loosely coupled, contracted nature provides a *layer of abstraction* that masks the underlying complexity of heterogeneous technology while providing business-oriented Web Services to the business.

All Contents Copyright © 2003 ZapThink, LLC. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.



The Services Layer of Abstraction: Revealing the True Power of Web Services

Layers of abstraction find their roots in the very earliest software. The first programmable digital computers dealt in the world of zeroes and ones—that is, *only* zeroes and ones. Into this black-and-white world came programs called compilers that let programmers work with English-like languages like COBOL that created a layer of abstraction that both masked the underlying complexity of the technology while at the same time provided more power to the users of that technology. The history of IT contains layers of abstraction placed one on top of another, including the graphical user interface, object-oriented programming, and the Internet. At every step, software allowed people to work with relatively simple tools that accessed complex systems behind the scenes. That's the power of abstraction: enabling the tools people use to get simpler as they become more powerful.

Web Services, then, are currently enabling an evolutionary step in this inexorable progression to the next level of abstraction for distributed computing. Software functionality can now be thought of as location-independent Services on the network—an abstraction called *Service orientation*. Service-oriented architectures leverage the loosely coupled, contracted nature of Web Services to enable today's heterogeneous collections of applications and systems to provide flexible, resilient IT functionality that meets ever-changing business requirements.

The ZapThink Take

Ironically, Web Services are not strictly required for the loosely coupled Service-oriented architectures that offer the framework for this vision of Service orientation that enables business agility. Web Services, however, have paved the way for standards-based, loosely coupled distributed computing. Broadly speaking, then, the term *Web Services* has come to represent the industry-wide movement to Service orientation.

The key part of the term *Web Services*, therefore, is not *Web*, it's *Services*. The word *Web* likely comes from the fact that in the early days, Web Services went over HTTP, the protocol that underlies the World Wide Web. Today, however, Web Services can go over any communications protocol. Nevertheless, the phrase is here to stay, where Web Services are distinguished from software services in general because of their standards-based, loosely coupled nature.

ABOUT ZAPTHINK LLC

Thank you for reading ZapThink research! ZapThink is an IT market intelligence firm that provides trusted advice and critical insight into XML, Web Services, and Service Orientation. We provide our target audience of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

Take Credit for reading ZapThink research! Visit www.zapthink.com/credit and enter the code **WSTAKE**. We'll reward you with ZapCredits that you can use to obtain free research, ZapGear, and more! For more information about ZapThink products and services, please call us at +1-781-207-0203, or drop us an email at info@zapthink.com.

