

**zaphink**  
white paper

**SOA IMPLEMENTATION BEST PRACTICES**  
**COMPARING & CONTRASTING INFRASTRUCTURE PLATFORMS**



# SOA IMPLEMENTATION BEST PRACTICES

## COMPARING AND CONTRASTING INFRASTRUCTURE PLATFORMS

November 2007

Analyst: Ronald Schmelzer

### Abstract

The role of SOA is to provide an architectural approach that supports an organization's ability to support ongoing business change in the face of a heterogeneous environment. However, since SOA does not introduce a new programming language or runtime environment, organizations must implement code that underpins and exposes a Service interface somehow. Since implementation matters to computers as much as architecture matters to people, it makes sense to consider the runtime environment of the implementation to be a good place to coordinate Service interactions.

When looking at the technology buying patterns in the world of SOA, there is one common thread. The influence of the larger SOA vendors is very much a force in the market today. Within this context, and given continued consolidation, confusion, and change in the SOA marketplace, organizations should make a careful evaluation of the various vendor offerings that support SOA implementations. The consolidation of SOA implementation capabilities has yielded a collection of vendors offering the new solution stack: the "SOA Platform".

When comparing vendors, it is important to verify how their products and services address both parts of the lifecycle. If the vendor's own products or services address the full lifecycle, what is their strategy for interoperating with other vendors whose products or services fill the gap? Are vendor platforms just an amalgam of acquired, partnered, and separately developed technologies, or do they represent a cohesive collection of agile SOA infrastructure that supports the widest range of SOA capabilities? This report aims to identify SOA platform vendors and help make the evaluation using the above criteria.

**Statement of Opinion:** Although this is a commissioned report, the views expressed in this document are solely the opinions of ZapThink. Although ZapThink believes that the results, conclusions, and analysis produced in support of this research are well informed, comprehensive, and reasonable, ZapThink asserts that the views herein are opinions and not statements of fact, and does not guarantee results, accuracy of forward-looking market predictions, or applicability of conclusions to reader's particular requirements. Moreover, ZapThink does not assume responsibility for the accuracy and completeness of such statements. The information derived in this research shall be construed as statements of opinion only, and ZapThink shall not be held liable in any manner for any conclusions or actions taken pursuant to this report. The information contained herein has been obtained from sources believed to be reliable, and where possible, cited with sources. ZapThink shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. Client assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed are subject to change without notice. **Disclosure:** This White Paper was commissioned by IBM.

All Contents Copyright © 2007 ZapThink, LLC. All rights reserved. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.

## I. Service-Oriented Architecture: Satisfying the Need for Agility

In today's post-dot.com world of tight IT budgets, increased regulation, global competition, and accelerating change, companies (and governments) require quantifiable results from their investments in technology. No executive will sign off on any investment in new technology without a solid expectation for how it will deliver value to the business. As a result, the pressures on today's IT executives are enormous: cut costs and make do with existing technology, while at the same time serve customers better, be more competitive, and respond to the business's strategic priorities. Basically, executives must do more with less, not just right now, but also into the future as business needs change and new ones develop.

There are two underlying themes behind all of these pressures: *heterogeneity* and *change*. Business must rapidly adapt to survive, let alone to succeed in today's dynamic competitive environment, and the IT infrastructure must enable businesses' ability to adapt. Companies wish to enable continuous change even in an environment of heterogeneity. And this is the environment in which Service-Oriented Architecture (SOA) thrives.

### Understanding the Role of Vendors in SOA Implementations

At its core, SOA is an architectural practice, not technology nor product, nor is it a set of product features. Properly implemented, SOA should be blind to the underlying runtime infrastructure, messaging protocol, or application server environment. Yet, an abstract architecture does little for an organization unless it is actualized by technology. As such, many firms look to their technology suppliers to help them implement, understand, and achieve success with their SOA initiatives.

When looking at the technology buying patterns in the world of SOA, there is one common thread. The influence of the larger SOA vendors is very much a force in the market today. Working with an organizations' existing technology vendors is often the path of least resistance, because the relationship is already established, and organizations don't have to go through the hassle of getting to know new players, or many new players. Thus, it's easier to purchase the latest SOA stack from an existing supplier than it is to go through the detailed requirements, analysis, and design that is really required to build an effective SOA implementation.

In addition, the SOA market is consolidating in such a manner that many large vendors offer "one stop shopping" for SOA. This SOA platform approach leads to broad confusion as to how to implement SOA properly, and which vendors to consider for the task. Past integration solutions failed to live up to their grand promises, so as technology vendors fill out their SOA stacks, it has become

*SOA is an architectural practice, not technology nor product, nor is it a set of product features. Properly implemented, SOA should be blind to the underlying runtime infrastructure, messaging protocol, or application server environment.*

Thank you for reading ZapThink research! ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences 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.

Earn rewards for reading ZapThink research! Visit [www.zapthink.com/credit](http://www.zapthink.com/credit) and enter the code **SOIBP**. 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](mailto:info@zapthink.com).



timely to ask whether this time round, history will be different.

Since SOA is something that organizations do, and not something they buy, proper SOA adoption requires more than simply buying a product with features and functions. It requires the adoption of methodologies and methods that support a loosely-coupled, composite, and heterogeneous approach to enterprise architecture. Within this context, and given continued consolidation, confusion, and change in the SOA marketplace, organizations should make a careful evaluation of the various vendor offerings that support SOA implementations. Furthermore, when comparing vendor support of SOA, it is critical to identify how to leverage vendor offerings to help enterprises adopt SOA.

## II. SOA Infrastructure Patterns

Any discussion of SOA infrastructure begs an answer to the most obvious question first: why purchase any SOA infrastructure at all? The main idea behind SOA is that organizations can map their continually changing business processes and requirements to IT capabilities represented as Services, which they have implemented in an abstracted, loosely-coupled manner such that they can compose them with other Services. Web Services add to the SOA story by standardizing various aspects of Service interfaces and communication so as to abstract the implementation details of individual Services. The combination of the two implies that integration is a side-effect of composition and Services are discrete entities abstracted such that they can communicate and interact with any other Service, regardless of its implementation.

While Web Services and SOA are different and mutually non-requisite concepts, most SOA implementations utilize Web Services precisely as the mechanism to make loose coupling a reality in heterogeneous environments. So, given that SOA provides the architectural precepts for building composable Services on the one hand and Web Services provide a technological means for isolating implementations on the other, shouldn't Service endpoints simply be smart enough to communicate with any other end point, without requiring any additional infrastructure between them?

The problem is that the devil's in the differences. Proper abstraction of Service endpoint implementations requires dealing with the differences in protocol, semantics, policy, availability, and the need for security, management, quality, and governance to guarantee reliable communications. While it is certainly possible to build smart endpoints that ameliorate those differences, companies often find that they can more easily (and cheaply) make the abstraction a reality by inserting some intelligence as an intermediary between those endpoints. And this intermediation is where the confusion over SOA infrastructure begins.

There are many patterns for SOA infrastructure: application server runtime environments that host Service containers, hub-and-spoke middleware that treats Services as endpoints to be integrated with, and centralized messaging infrastructure that focuses on message passing and handling. The vendors who champion the majority of these patterns have built their current Service infrastructure from technology that preceded the SOA movement. Each of these approaches have merits, to be sure, but lost in the conversation is the fact that there are other patterns for SOA infrastructure that don't borrow from past technologies and offer a different way of thinking about Service interaction and composition.

It is important to note, therefore, that a discussion of SOA infrastructure patterns doesn't map cleanly to vendor product offerings, since many such offerings in fact implement a number of different infrastructure patterns or hybrids of SOA

*Proper abstraction of Service endpoint implementations requires dealing with the differences in protocol, semantics, policy, availability, and the need for security, management, quality, and governance to guarantee reliable communications.*

infrastructural approaches. Furthermore, it is vital to understand that vendors may label any of these approaches as an Enterprise Service Bus (ESB), highlighting the fact that there is no single definition for an ESB. However, it is imperative that architects understand the differences and merits of each infrastructure pattern before committing to a style of SOA infrastructure that might not meet the goals of the architecture.

### **The Service Container Infrastructure Pattern**

Since SOA does not introduce a new programming language or runtime environment, organizations must implement code that underpins and exposes a Service interface somehow. Since implementation matters to computers as much as architecture matters to people, it makes sense to consider the runtime environment of the implementation to be a good place to coordinate Service interactions. In this infrastructure pattern, developers implement their Services in a container that provides both a runtime environment for marshalling requests to and from the Service as well as a control environment for providing security, asynchronous message handling, Service composition, management, and other activities.

The advantage of the Service container pattern is that it provides a consistent implementation infrastructure for running Services across all the Services in an organization's immediate control. The obvious challenge with this approach is that it can only guarantee Service quality to Services that execute within its runtime containers. Services that run in the containers are managed whereas those that are external to the environment are managed only if a Service proxy is created to interact with them. This approach borrows from existing application server platforms that manage runtime code such as Servlets. Implementing a Service container-style infrastructure in an environment of significant heterogeneity, distributed Service runtimes, and wide ranges of interaction types is often a deployment challenge. However, for many enterprises this approach is the easiest to implement, especially if they already depend on the vendors for their existing, non Service-oriented infrastructure.

### **The Hub-and-Spoke Infrastructure Pattern**

Another common SOA infrastructure pattern borrows from Enterprise Application Integration (EAI) techniques: the use of integration middleware servers that interact with endpoints through adapters and other interaction mechanisms and alleviate differences through centralized mapping and coordination logic. In essence, the integration middleware server acts as the coordination point for all interactions between endpoints. One of the advantages of this approach is that endpoints can run on a wide array of runtime environments as long as an adapter or other access mechanism exists to access the functionality or data. Another advantage is that the middleware centralizes the integration logic within a single runtime environment.

Of course, the big challenge with hub-and-spoke infrastructure patterns is that the point of centralization all too often becomes a point of tight coupling. If the infrastructure forces the centralization of integration capability such that the endpoints really are Service interfaces and not heterogeneous, loosely-coupled, composable Services, then you haven't really achieved SOA at all. In fact, all you have built is standards-based integration using Web Services. So, the challenge with such infrastructure patterns is to make sure you really are getting integration as a side-effect of composition, and haven't just been sold old wine in new bottles.

### The Centralized Messaging Infrastructure Pattern (Message Bus)

The third popular approach to facilitating inter-Service communication and runtime management is to leverage message-oriented middleware and associated messaging infrastructure to coordinate messages between Services with the idea that managing the messages matters more than managing the specific runtime endpoints. So, rather than placing Services in a managed container or connecting to Service endpoints through adapters or a hub-and-spoke approach, the SOA infrastructure simply needs to instrument the endpoints to utilize a particular message bus or publish/subscribe infrastructure. As such, the bus serves as a Service highway with on and off ramps to get the messages onto and off the bus. The bus messaging protocols and servers then handle all the requirements for communication.

Vendors with products in the Message-Oriented Middleware or Message Queue markets have popularized this approach to SOA infrastructure, and for good reason: the approach lends itself well to loosely coupled, event-driven, and message-oriented styles of communication. Of course, the main challenge with this approach is that it can only manage those interactions in which messages pass through the bus. Much as the Service container approach requires proxies to work with third-party Services, Messaging infrastructure approaches require some mechanism for marshalling requests from other systems onto the bus. In addition, the message bus approach tends to imply a platform-centricity in much the same way as the Service container approaches.

*The message bus approach tends to imply a platform-centricity in much the same way as the Service container approaches.*

### The Service / Network Intermediary Pattern

Another Service infrastructure approach uses an already existing loosely-coupled infrastructure that supports heterogeneity: the TCP/IP network. In this approach, Service requests are routed through content-based routers, which then use late-binding rules to determine how to route, manage, or enforce policy on Service interactions. In the same manner that routers, firewalls, load balancers, gateways, and caches facilitate all kinds of complex network interactions ranging from email to Voice-over-IP without requiring single-vendor middleware, so too can Service interactions be facilitated by a wide variety of Service intermediary deployed in software or hardware form.

Service intermediaries consist of content-based routers that inspect layer 7 messages, registries that provide policy metadata to guide content-based routers and help determine how to bind Service consumers to providers, and active SOA management tools that provide runtime exception management, policy enforcement, and state management for long-lived, asynchronous interactions.

In order for this pattern to work, all Service interactions must pass through at least one intermediary between the Service consumer and provider, and must use late-binding as a mechanism to determine how to handle and manage Service requests. The easiest way to address the former concern is to leverage WS-Addressing as a mechanism for loosely coupling the references to Service endpoints and forcing Service requests to resolve to network intermediaries. Using hard-coded URLs with HTTP addresses in Service contracts is not appropriate for the late-binding mechanism required of a network intermediary infrastructure pattern.

Finally, the network infrastructure pattern works by leveraging the Service registry as a mechanism for enabling late binding. In essence, the registry stores the information about not only where Service end points are located, but also all the policy and contract metadata required to successfully interact with that Service. A Service consumer makes a request to an abstract WS-Address, which is resolved by the content-based routers, passed to a SOA management intermediary, which in turn consults a registry to determine how and where to

bind to the appropriate Service. Of course, this is all cached and distributed for performance. This combination of routers, management intermediaries, and registries makes up a highly distributable, loosely coupled, and reliable infrastructure that has a very “un-ESB” feel, although it provides the very same capabilities as the SOA infrastructure patterns discussed above.

### Finding the Best Approach to SOA Infrastructure

So, what’s best approach to SOA infrastructure? Of course, experienced architects know that there’s no such thing as a single best approach that applies in all circumstances. A good architect will realize that the right answer is always “it depends.” The problem is that not enough architects have knowledge of all the possible ways to implement SOA infrastructure, and simply assume one approach will suit all needs. Good architects have a toolbelt with multiple architectural, infrastructural, organizational, and methodology approaches to suit a wide variety of needs. Trying to shoehorn one particular infrastructural approach into all scenarios is a recipe for disaster.

There isn’t any one infrastructural approach as inherently “bad” or another inherently “best,” and it is worth highlighting the fact that SOA has no infrastructural bias. Indeed, it’s theoretically possible to implement a network of smart endpoints that can communicate with each other in a peer-to-peer manner, abstracting all the differences in protocol, policy, semantics, and availability without requiring anything between them. However, it’s exactly the requirement of mediating the differences between endpoints that necessitates the need for something intelligent between endpoints that might not have the sophistication to deal with interoperability-killing differences.

Companies evaluating SOA Platforms need to ask themselves the following key questions:

- *Dependencies:* With SOA being an architectural approach that abstracts the Service from the underlying technology that physically implements it, the first question to ask is, does the vendor’s product architecture add technology or product dependencies?
- *Coherence and consistency:* Is the architecture logically coherent, or does it add unnecessary complexity such as multiple engines or servers that add interfaces, increase administrative overhead, and require additional coding? Does the technology stack rely on standards, or does it require significant proprietary content? Continued acquisition in the SOA platform brings further question in the market with the potential and pending Oracle and BEA merger, the consolidation of products in the Software AG webMethods stack, and other areas of consolidation. How do vendors define SOA and how do they position it with regard to their product or service offerings? How consistent is their messaging?
- *Support for SOA patterns:* There are multiple ways to approach SOA, including exposing and running atomic services, enabling composite, process-driven SOA, hardware-enabled Service intermediaries, event-driven styles, and legacy-enabled approaches. Not all vendors support all these styles, with IBM offering the broadest range of the SOA platforms identified above compared to TIBCO, Oracle, and Software AG webMethods.
- *Single runtime or many?* Many companies in the SOA marketplace have grown through acquisition and/or partnership. As such, many SOA solutions are hodge-podges of a mixed set of technologies that don’t inherently work well together. As such, it is an exercise left to the customer or a consulting group to tie together a vendor’s own offerings

*Trying to shoehorn one particular infrastructural approach into all scenarios is a recipe for disaster.*

together. This is clearly not beneficial from a time or cost perspective. Customers evaluating SOA platforms should evaluate whether the technology is based on a cohesive, single runtime, or is a collection of mismatched runtime technologies.

- *Interoperable or proprietary?* Sure, most SOA platform vendors support Web Services. They will claim that's all that's needed for interoperability. Right? Wrong. Many runtime environments use proprietary approaches for messaging, process runtime, and policy enforcement. So, if choosing a SOA platform requires you to use proprietary approaches to those needs, then you're locking yourself into a proprietary platform. Careful evaluation of what's truly needed for interoperability will enable you to build truly heterogeneous Service-oriented architectures.

When comparing vendors, it is important to verify how their products and services address both parts of the lifecycle. If the vendor's own products or services address the full lifecycle, what is their strategy for interoperating with other vendors whose products or services fill the gap? The following section of this report aims to identify SOA platform vendors and help make the evaluation using the above criteria.

### III. Evaluating SOA Platform Approaches

*Companies evaluating vendor offerings should consider more than simply the SOA infrastructural approach, but rather the completeness of the entire SOA environment.*

Companies evaluating vendor offerings should consider more than simply the SOA infrastructural approach, but rather the completeness of the entire SOA environment. Simply being able to run and execute Services is not sufficient to enable the range of capabilities for the full SOA lifecycle. Furthermore, the specific mechanism for exposing and interacting with Services is not sufficient to differentiate between vendor offerings. Indeed, there's no such thing as "SOA product features" because, of course, SOA is architecture, and architecture consists of best practices.

#### **Governance, Quality, and Management: the Most Critical Infrastructure in SOA:**

Companies that are well on the path to SOA adoption know full well that the technical challenges of building and exposing Services are less significant than the hurdles of building loosely coupled, business-relevant Services leveraged across their continuously changing business processes. Indeed, the challenge of making loose coupling a reality is only surpassed by the even-greater challenges associated with organizational and cultural adoption of SOA.

While SOA abstracts the complexity associated with heterogeneous, point-to-point integration and tightly-coupled application logic, it introduces a different kind of complexity: the management of distributed, loosely coupled, and dynamically composable Services. Over the past few years companies developed a number of approaches for dealing with this new form of complexity: management solutions that isolate failure and provide mechanisms for abstracting end-point differences, quality solutions that provide mechanisms for assuring changes can be propagated in environments of significant change, and governance approaches that provide oversight into the development of Service-oriented systems, mitigation of change and version management issues, and enforcement of policies core to the operation of the business as a whole.

Each of these distinct market segments are in actuality aspects of same problem: making the illusion of loose coupling a reality. In combination, SOA governance, quality, and management (GQM) form a perfect trifecta that can make the perceived difficulty of loose coupling in a continuously changing IT and business environment a reality. Companies that are truly interested in the

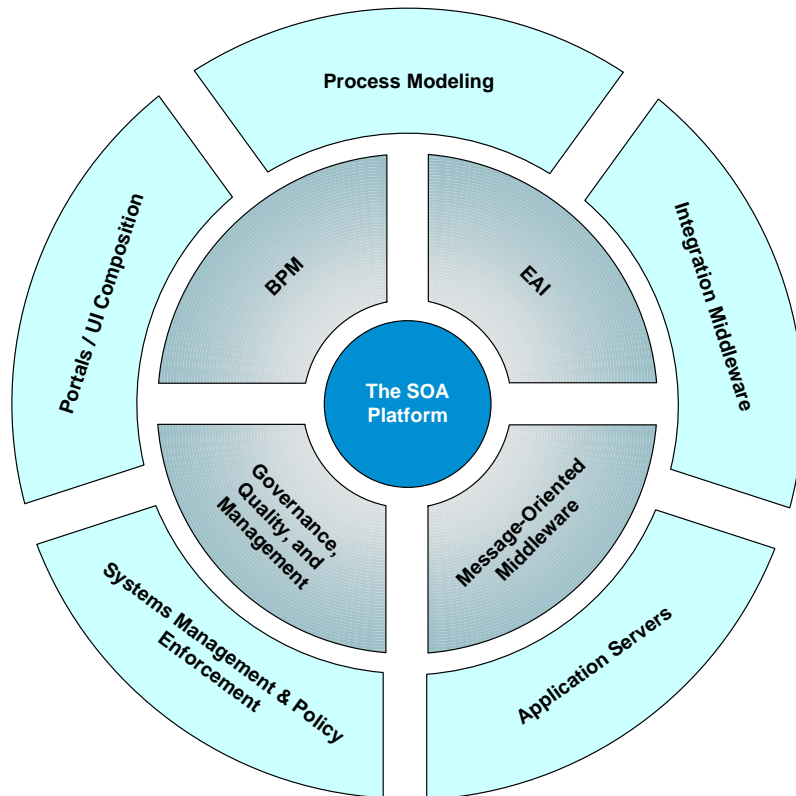


benefits of loosely-coupled, composable, business-oriented Services that achieve a flat-cost of change in an environment of heterogeneity and continuous change would be well-served to focus the technology-specific part of their efforts not solely on the ESB as the lynchpin of their SOA success, but rather on the GQM aspects. Addressing all parts of the GQM puzzle is beneficial to making the illusion of loose coupling a reality. In addition, what makes the GQM space more compelling is that it represents new technology investment that most companies haven't already made.

**The Steady Progression Towards the SOA Platform**

However, there are indeed a wide range of infrastructural capabilities that can be present in a SOA suite or platform that aims to embrace the full SOA lifecycle, as shown in the figure below.

**The Progression to the SOA Platform**



Source: ZapThink LLC

This white paper does not dive into the details about the above diagram, but the main takeaway is that the SOA Platform is an amalgam of many different technology components of SOA infrastructure. Furthermore, one of the most important things to be gleaned from the chart is that most of the existing SOA platform vendors originated from a certain center of gravity. For example, many vendors developed their SOA platforms from an EAI perspective while others originated from a message-oriented middleware centrality. Yet others originated their platforms from the legacy integration and data integration perspectives.

Knowing how the vendors developed their platforms will help to understand the context of their offerings.

Even though market forces are conspiring to push all these different aspects of the SOA infrastructure together into a single suite, why would customers want such a SOA platform? The most basic answer is that while there's no penalty for using best-of-breed technology in an SOA (and indeed, in an environment of heterogeneity it may be impossible to truly ever solidify on a single platform), companies simply feel more comfortable with their existing infrastructure vendors. In essence, before sourcing a SOA solution from a new vendor, they most often desire to check with their existing suppliers to see if their offerings meet their needs.

Furthermore, market consolidation is out of the enterprise end user's hands. As best-of-breed point solutions grow and become mature, they become highly desirable acquisition targets. The public market is not favorable at this time to an IPO, so investors seek their returns through acquisition. Despite the desire by enterprises for point solutions and best-of-breed products to remain independent, the best-of-breed vendors themselves are pushing the market towards consolidation.

### **A Comparison of SOA Platform Suites**

Many organizations are evaluating SOA platforms that consist of more than just point solutions to parts of the SOA infrastructure puzzle. Furthermore, the task of comparing these SOA platforms is becoming easier as the market continues to consolidate. A view of a selection of the key SOA platforms appears in the table on the next page.

Note that the below chart does not aim to provide a comprehensive overview of all SOA platform providers in the market, but rather illustrate a few examples of vendor offerings for the purpose of comparison. In addition, the above table doesn't aim to comprehensively evaluate each of the offerings, but rather aims to show how the offerings have matured from existing technologies, acquisitions, new development, or partnership / embedding of third-party solutions.

It is important to note that the vendors created their SOA platforms, for the most part, out of a collection of existing or acquired products that they repurposed for SOA. Very little of the vendors' SOA platforms originated from new development, begging the question as to what precisely is new about the products they are offering. Putting that question aside, it is important to note that some of the vendors' offerings are more of an amalgam of acquisitions and partnerships than others. IBM is the most consistent in its single-vendor originated SOA Platform stack while TIBCO, Oracle, and Software AG have mostly grown their infrastructure stacks through acquisition, partnership, and embedding.

These vendors approach the market from different perspectives. TIBCO entered the SOA infrastructure market from their strong position in the message-oriented middleware (MOM) market, Software AG and webMethods from their strong Legacy, XML, Enterprise Application Integration (EAI) and Business-to-Business (B2B) background, IBM from their broad support of software, services, and hardware across the IT environment, and Oracle within the context of their acquisition frenzy of the past few years. Understanding the origins of these vendor platforms will help understand the differences, cohesiveness, and weakness of their respective offerings.

*Vendors have created their SOA platforms, for the most part, out of a collection of existing or acquired products that they repurposed for SOA.*

SOA Platform Suite Comparison

	IBM	TIBCO	Software AG webMethods	Oracle
Infrastructure Patterns Supported	Service Container, Message Bus, Hub-and-Spoke, Service Intermediary	Message Bus	Hub and Spoke, Message Bus	Service Container, Hub and Spoke, Message Bus
Legacy Enablement / Modernization	Yes	None	Yes	Some
Registry / Repository Offering	<a href="#">WebSphere Service Registry / Repository</a>	<b>OEMs HP SYSTINET AS ACTIVEMATRIX REGISTRY<sup>1</sup></b>	<a href="#">CentraSite, CentraSite Governance Edition</a>	<b>OEMs HP SYSTINET<sup>2</sup></b>
Service Intermediary Offering	<a href="#">DataPower appliances</a>	None	None	None
Enterprise Service Bus (ESB) Offering	<b>WebSphere ESB, WebSphere Message Broker , DataPower XI50</b>	<b>ActiveMatrix, BusinessWorks</b>	<a href="#">webMethods ESB Platform</a>	<b>Oracle Enterprise Service Bus</b>
Service-Oriented Business Process Management (BPM)	<b>WebSphere Process Server, WebSphere Business Services Fabric</b>	<b>BusinessWorks, iProcess</b>	<a href="#">webMethods Business Process Management Suite (BPMS)</a>	<a href="#">Oracle BPEL Process Manager</a>
Service Management	<b>Tivoli Composite Application Manager</b>	<b>OEMs AMBERPOINT AS ACTIVEMATRIX POLICY MANAGER<sup>3</sup></b>	<b>CentraSite and RESELLS AMBERPOINT<sup>4</sup></b>	<a href="#">Oracle Web Services Manager</a>
Business Activity Management	<b>WebSphere Business Monitor</b>	<b>BusinessFactor, OpsFactor, iProcess Analytics, Spotfire</b>	<a href="#">webMethods Optimize</a>	<b>Oracle Business Activity Monitoring</b>
Systems Management	<b>Tivoli OMEGAMON XE</b>	<b>Hawk Enterprise Management Advisor, BusinessFactor</b>	None	<b>Oracle Enterprise Manager</b>
Completeness of Total Product	Software, Professional Services, Hardware, Devices	Software, Professional Services	Software, Professional Services	Software, Professional Services

Legend: [New development](#), [acquisition](#), **PARTNERSHIP/EMBEDDING**, [extension of existing offering](#) .

Source: ZapThink LLC

Most of the above offerings offer a similar range of functions related to implementing SOA, encompassing runtime Service execution, BPM, BAM, workflow, and management. All platforms offer atomic Service exposure and Service implementation runtimes, composition technologies, and process-driven styles, as well as asynchronous, event-driven, and synchronous styles of Service invocation. The major differences in platforms revolve around the vendor product architecture, degree of support for different SOA infrastructure patterns), and their reliance on third-party products.

<sup>1</sup> [http://www.webservices.org/vendors/hp/tibco\\_to\\_resell\\_systinet\\_registry](http://www.webservices.org/vendors/hp/tibco_to_resell_systinet_registry)

<sup>2</sup> [http://searchsoa.techtarget.com/originalContent/0,289142,sid26\\_gci1130547,00.html](http://searchsoa.techtarget.com/originalContent/0,289142,sid26_gci1130547,00.html)

<sup>3</sup> [http://www.amberpoint.com/releases/pr\\_152.shtml](http://www.amberpoint.com/releases/pr_152.shtml)

<sup>4</sup> [http://documentation.softwareag.com/crossvision/inm/policies/feature\\_pack.htm](http://documentation.softwareag.com/crossvision/inm/policies/feature_pack.htm)

### 3.1.1. The IBM SOA Platform

IBM is a highly diversified IT technology and services company that is one of the largest in the IT industry. Over past decade, IBM has transitioned from its systems heritage to a broad-based emphasis on platforms, software and consulting services to help organizations integrate technology and innovate their business models. IBM Software Group itself has retooled, thanks to significant investments in building its five major brands through organic development and aggressive acquisition strategy. In 2006 IBM Software delivered 40% of IBM's pretax profits.

IBM's Global Services businesses have more than doubled the revenue of their nearest rivals with broad based business and technology consulting and implementation. For the purposes of this report, we will restrict the focus to SOA-related offerings ranging from Service enablement and lifecycle best practices to the building of vertical industry SOA frameworks.

IBM has an SOA offering built around a single runtime, and a common Web-based administrative console. Each of the offerings utilizes Eclipse-based development tools. The IBM stack uses a common WebSphere runtime engine, and IBM has integrated acquisitions that have filled out the WebSphere stack, including the WebSphere Process Server, which originated with internally-developed process modeling logic enhanced with acquisition of Holosofx which brought process modeling and analysis capabilities. The company then adapted the integration flow and application adapters from CrossWorlds acquisition, rewriting it over two and a half years to support the WebSphere J2EE container.

IBM supports a variety of SOA deployment styles. Customers can deploy through a distributed bus or server-based approach, or leverage distributed network intermediaries. IBM integrates the WebSphere ESB into WebSphere Process Server, but doesn't require customers to use the bus if server-based deployment is appropriate. IBM provides additional options, including the vertical industry business Service fabrics that came through the Webify acquisition and transactional integration (part of Process Server, which came from CrossWorlds). IBM has extensive third party product and service ecosystem for offerings outside the core SOA stack.

While generally consistent in its message to the market around SOA, IBM has not been perfect in its SOA go-to-market story. IBM has consistently promoted one of the most important cornerstones of SOA: that there must be business value. It backs up the message with specific SOA entry point scenarios, case studies, online tools for analyzing potential benefits for specific entry points, and professional services for guiding clients through building the business case for SOA. However, IBM's message, especially with regard to ESB, has not always been consistent. As the market started to solidify around the ESB as an accepted approach to SOA infrastructure, IBM continued to claim that ESBs were nothing more than design patterns. More so, they had been claiming that a combination of products, including the Message Broker, fit the bill. But in response to market demand, the company developed ESB-branded offerings and adopted a more customer-responsive, balanced view when the company introduced WebSphere ESB.

The company's ESB offering now consists of a few different components. WebSphere Enterprise Service Bus is built on WebSphere Application Server to provide integrated SOA runtime capabilities. WebSphere Message Broker provides more advanced ESB capabilities and provides universal connectivity and transformation in heterogeneous IT environments. The WebSphere DataPower Integration Appliance XI50 is a purpose-built hardware ESB for simplified deployment and hardened security. WebSphere DataPower SOA Appliances are hardware-based, purpose-built integrated appliances or network

*IBM supports a variety of SOA deployment styles.*

devices that provides a different approach than putting software on a general purpose server. The company also considers the WebSphere DataPower XML Security Gateway XS40 as a Service Intermediary. By providing these different options, IBM sees the benefit that customers can mix and match them to meet their specific ESB needs. Leveraging these approaches, IBM enables event-driven approaches to SOA as part of their support for the Message Bus pattern

IBM has improved its SOA messaging since the rollout of WebSphere ESB, and today provides a more complete message and vision that positions the SOA lifecycle with adjacent processes, including SDLC and IT infrastructure management. With further refinements, IBM's vision is becoming consistent with Enterprise Architecture (EA), IT Infrastructure Library (ITIL), and IT governance best practices. IBM has also emphasized the need for vertical industry frameworks that can act as accelerators to developing Service-oriented composite business applications. IBM is already backing up its messages and vision with offerings across its software brands and global services groups.

IBM has consistently emphasized that SOA requires a full lifecycle management approach, and that governance plays a pivotal role. IBM has in turn outlined an iterative SOA governance life cycle with four distinct stages (plan, define, enable, and measure). The company takes a broader view that SOA governance is not an island, but a connected process that should fit with the SDLC and with infrastructure management. It backs its message/vision with a broad selection of products and services (from across the WebSphere, Rational, and Tivoli brands; and its global services units, respectively) that support its vision of the SOA lifecycle. And, with the Webify acquisition, IBM has focused efforts to develop vertical industry frameworks and custom applications based on SOA.

IBM has particularly strong capabilities in the breadth of the total product offering. In fact, they have the largest solution delivery force of any SOA vendor on the market, with experience with over 5,700 clients. The company leverages this experience and the size of their consulting force as a big differentiator,

### 3.1.2. The TIBCO SOA Platform

Narrower in overall focus than IBM, TIBCO specializes in enterprise integration middleware. TIBCO built its reputation as a pioneer in event management capabilities, literally making its name as "The Information Bus Company." It has been a leader in publish/subscribe multicast integration built initially for financial services companies whose businesses were built on interpreting rapid streams of events, integrating information from multiple sources. Not surprisingly, TIBCO has extended that event-oriented approach to SOA. It gradually extended beyond its proprietary TIBCO Rendezvous architecture, first by adding JMS support, and more recently with its XML-based BusinessWorks ESB.

TIBCO has acquired and developed products designed to support its vision of SOA. Its principal SOA-related acquisition was BPM provider Staffware, whose offerings are now branded as TIBCO iProcess Suite. As a software, not a professional services company, TIBCO's professional services center on implementation and integration of its own products. It relies on a large network of consulting partners which include most of the industry's leading names. TIBCO maintains partnerships with major technology partners including HP, Sun, Cisco and Intel, but not IBM. It has roughly 20 software and 45 OEM partners who include AmberPoint, IDS Scheer, Oracle, SOA Software, Solstice Software, and HP Software.

The TIBCO offering leverages a collection of runtime engines and administrative consoles, utilizing a virtualization layer to achieve cohesion. Its development tools use a mix of Eclipse and Windows-based offerings. TIBCO requires an ESB for all SOA deployment cases, and has a strong bias towards perceiving SOA

*TIBCO requires an ESB for all SOA deployment cases, and has a strong bias towards perceiving SOA infrastructure as event-based processing.*

infrastructure as event-based processing. While this is effective for transactional Services that are intended for consumption in a wide variety of real-time and batch-mode interactions, this event-driven perspective does not adequately support data-driven and information integration oriented use cases such as content or information Services.

TIBCO's SOA and BPM stacks are not as comprehensively integrated as IBM's offerings as of the date of this paper, but still work to combine a number of solutions into a process-enabled SOA stack. TIBCO BusinessWorks ESB and iProcess Suite (BPM) each run on separate runtime servers and use separate web-based administrative consoles, and they require a special adapter to integrate with each other. The company uses an added layer (Active Matrix ServiceGrid) to virtualize Services created from different platforms before deploying to TIBCO's ESB, and uses two separate administrative consoles for management of the disparate products.

TIBCO has based its Service messaging on its heritage as a provider of event-driven technology. That message is valid within that particular SOA use case. But TIBCO's vision may add complexity for customers that may not need event-driven approaches for all of their integration or business agility needs. Furthermore, the ESB-centricity of TIBCO's approach might not be appropriate for firms looking for a Service intermediary style of approach in which XML gateways and/or management agents are better suited for the task of Service mediation.

TIBCO takes a narrower view than IBM on the Service lifecycle. The TIBCO perspective on the Service lifecycle has four states: build and deployment of Services, liking and composition of Services, operation of Services, and the retirement of processes. As such, this lifecycle has a predominately IT-oriented focus that sees SOA mostly as an effort to enable IT and business collaboration for Service development, associated policies, delineating roles regarding Service approvals, and runtime monitoring, access, and modifications.

TIBCO One is TIBCO's technology convergence strategy for its various integration engines. Through the end of 2008, TIBCO will phase in a common framework<sup>5</sup> based on common design tool for deploying processes to each of these products, a common front end shell to provide a common user experience, and a set of workflows for deploying processes spanning multiple engines (e.g., a complex event processing workflows that trigger EAI processes). This SOA platform convergence strategy involves using Business Studio 2.0 to deploy to ActiveMatrix Service Grid, which in turn virtualizes TIBCO's (and other vendors') various runtimes to enable processes that span multiple runtime Service engines.

TIBCO One uses Web Services to link or rationalize different pieces of the stack, but it does not do so in a Service-oriented fashion in that it relies on proprietary frameworks to piece the puzzle together. A Service-oriented perspective would allow organizations to leverage third-party runtimes to stitch together disparate Services without requiring consolidation on a single platform. This is something ZapThink calls "middleware for middleware" – an approach to integrating disparate runtime infrastructures by introducing yet another integration infrastructure. Companies should carefully consider and evaluate whether this makes sense for them. While TIBCO One rationalizes the development tooling, it does not rationalize the proliferation of servers and administrative consoles within its suite.

Not surprisingly, the first phase of the TIBCO One strategy, which will involve developing a common user interface shell that accesses these products, is set to roll-out through 2008. Initially, it the new UI will extend to Active Matrix,

<sup>5</sup> [http://www.infoworld.com/article/07/05/01/Tibco-One-interface\\_1.html](http://www.infoworld.com/article/07/05/01/Tibco-One-interface_1.html)

TIBCO's go-to-market message around SOA has been inconsistent.

BusinessWorks, and the EAI adapters. The next phase will target the messaging products, master data management, complex event and rules processing; while the last phase of UI convergence will extend to the mainframe offerings.

In addition, TIBCO continues to rely on an OEM strategy for critical pieces of its stack, including utilizing HP's Systinet as registry/repository as a core part of their SOA governance strategy, AmberPoint for administering run time policy enforcement (but not service management, which is covered by TIBCO Hawk) with a native interface to Active Matrix, and Arjuna for distributed transaction monitoring. It is important to note that AmberPoint and HP Systinet maintain similar relationships with TIBCO rivals such as BEA and Oracle.

TIBCO's go-to-market message around SOA has also been inconsistent. Over the last few years, its messaging has shifted from promoting SOA as an IT simplification strategy that can accelerate change to denigrating rival SOA solutions with characterizations such as "Sound Ominously Arrogant," "Sell Old Applications," or provide a "Source Of Apprehension." Today TIBCO's SOA message is one based on events and business process, and it insists that SOA requires an ESB. In its own words:

*"ESBs form the foundation of the SOA and can be complemented by additional productivity capabilities such as service orchestration and registries."<sup>6</sup>*

And:

*"TIBCO strongly recommends a mediated approach, in which all service requests are processed through an ESB, because this can be more flexible, scalable and reliable than a point-to-point configuration."<sup>7</sup>*

And finally:

*"Are you implementing the right SOA for the long term? If you're implementing an ESB that supports event processing and messaging for any event driven communication, the answer is yes. If you're not implementing an ESB - for example, if you're implementing an SOA using application servers, integration brokers, and code - the answer is no. You need an ESB ...SOA and event processing are both required for an optimized business."<sup>8</sup>*

One of the most over-hyped technology terms *du jour* is the Enterprise Service Bus (ESB). No one quite agrees on what it is, but it seems that everyone thinks they need one to make an SOA work. Indeed, the ESB represents a set of capabilities that combine distributed Service intermediaries with reliable message transport. However, many companies are realizing that they already have many such capabilities, and so can implement an ESB using their existing infrastructure. There's a good chance you already have sufficient middleware—so looking for an ESB (or any other three-letter acronym, for that matter) is often the wrong place to begin. Instead, start with the architecture, determine the best approach to implementing loosely coupled, distributed, composable Services, and only then hammer out the best infrastructure to make that happen.

While the TIBCO platform continues to increase in robustness, capability, and size, the legacy of the TIBCO technology and the marketing thrust around ESB-centricity should be factored in to any company's evaluation of this platform. TIBCO's approach might be appropriate and applicable for many customers' SOA infrastructural requirements, but ZapThink frequently cautions firms to start with their architectural needs first before considering vendor offerings.

Start with the architecture, determine the best approach to implementing loosely coupled, distributed, composable Services, and only then hammer out the best infrastructure to make that happen.

<sup>6</sup> Service Mediation: The Role of an Enterprise Service Bus in an SOA, TIBCO white paper

<sup>7</sup> UDDI and Beyond: The Role of Service Registries and Repositories in Your SOA, TIBCO white paper

<sup>8</sup> Event-Driven SOA: A Better Way to SOA, TIBCO white paper

### 3.1.3. *The Software AG SOA Platform*

Software AG has more than 37 years of global IT experience and over 3,700 employees serving customers in 70 countries. The company, headquartered in Germany and the US, offers software and professional services that “increase the value of Enterprise IT Systems.” With the recent acquisition of webMethods, the company is now acknowledged as a global leader in mission-critical software infrastructure solutions based on open standards. They have long championed and pioneered SOA solutions, especially in heterogeneous IT environments.

With a rich legacy background, their products Adabas, webMethods, Natural, and Tamino help more than 4,000 customers uncover the full potential of their IT and add value to existing systems. Software AG focuses their solution suite on creating flexible business applications and processes, extending the value and life of core and legacy systems, managing data effectively across the enterprise, and enabling control and governance of Service-oriented applications.

The Software AG webMethods suite grew as a combination of independent development and acquisition. The Software AG Crossvision line built on decades of experience in integration, database, and legacy enablement technology, and the pace of innovation around SOA continued to gain steam since 1999. Likewise, webMethods pioneered the EAI market and added SOA support with their Fabric product in 2004. The company acquired Infravio in 2006 to enhance their SOA governance and metadata management capability, and soon after, Software AG acquired webMethods to form the current SOA platform offering. Since the acquisition, Software AG embarked on a program to consolidate the various product offerings and work towards a single, cohesive stack. This process is still underway as of the time of writing this report, but already has shown positive value in the latest release of their Centrasite and webMethods offerings.

Software AG has a rich perspective on the SOA lifecycle. They see SOA as enabling enterprises to achieve control and transparency across all IT assets within the organization and monitor SOA implementations with reporting capabilities. Fundamentally, Software AG approaches SOA Governance as being critical for achieving greater business agility, because achieving the benefit of agility impacts how organizations manage and govern their SOA environments. In fact, CentraSite’s core benefits support the other SOA value propositions that this paper discusses.

The company views all IT assets, Services, policies and processes holistically, requiring effective SOA solutions to provide visibility across heterogeneous metadata and processes aids with regulatory compliance, improve business efficiency, and reduce time to value for innovative products and services. In their perspective, SOA infrastructure is focused on optimizing and managing the full SOA lifecycle for Services, policies and processes. Furthermore, the company sees the role of SOA solutions as enabling rapid and low-risk change management that changing business needs require. The company sees that most firms wish to implement SOA based on proven methodologies and standards, so they have a legacy-enablement perspective on SOA, rather than a “green field” Service development approach. The organization incorporates and encourages established SOA best practices, which help companies to leverage Services properly, reduce redundancy, and improve the time to value of their new products and services.

On the business side of the things, Software AG continues to add offerings that improve visibility, regulatory compliance, business efficiency, and risk mitigation, as well as improve collaboration between business and IT, as well as between Service providers and consumers. In particular, CentraSite’s collaboration capabilities help organizations in many ways, including improving customer value

*Software AG acquired webMethods to form the current SOA platform offering. Since the acquisition, Software AG embarked on a program to consolidate the various product offerings and work towards a single, cohesive stack.*



and business efficiency, and can also help with regulatory compliance for regulations that affect multiple silos. The product enables versatile cataloging that help organizations obtain more value from legacy, reduce redundancy, and deal with unexpected change.

The company has a strong partnership and third-party enablement approach to their SOA platform, and claims the ability to integrate with any third-party solution by plugging into the open and standards-based architecture of CentraSite, as well as integrating with over twenty SOA infrastructure vendors and CentraSite community partners, such as AmberPoint, Fujitsu, Ilog, IDS Scheer, Mindreef and Novell. Its open standards support and integration with third party products reduce dependency on expensive middleware and enable organizations to obtain more value from their existing legacy systems.

While the company has not yet finalized its integration of the webMethods stack into the Software AG SOA Platform offering, the company has already shown to its customers a desire to accelerate the pace of this cohesion and its recent release of Centrasite Governance Edition shows how the Infravio technology, an acquisition of webMethods, can be quickly integrated and composed with existing Software AG capabilities.

#### IV. The ZapThink Take

A principle of SOA is that the Service should be abstracted from the physical implementation. The challenge with the SOA platform landscape is that many software vendors entered the SOA market with a product line chock full of proprietary, tightly-coupled integration middleware. The challenge is that integration license revenue is suffering as customers look for more flexible, less expensive ways of leveraging their diverse information technology (IT) resources. As a result, these firms looked towards SOA. Yet, their software, however, does not lend itself to SOA best practices: loose coupling, composable Services, and flexibility in general are all capabilities that not present in tightly-coupled integration middleware. What to do?

Reinventing the company and rearchitecting software from the ground up is far too expensive and time-consuming. For many vendors, the option chosen was to slap Web Services interfaces on existing products, call it an Enterprise Service Bus (ESB), and sell it as SOA middleware. Hopefully your customers won't notice the old wine in new bottles. After all, that's what marketing is for!

But not every SOA platform vendor has taken this route. IBM's solutions show the greatest breadth of capability and support for SOA Infrastructure Patterns. The company has a reputation for complexity, and therefore must show through both its market messaging and its product cohesion that the collection of products performs as a unified set rather than a collection of discrete technologies. Despite this, IBM has created significant intellectual property around SOA methodologies and design, and built a strong professional services group that backs up their broad SOA infrastructure capabilities. Furthermore, IBM exhibits more flexibility on the centrality of the ESB to the SOA solution and a wide perspective on SOA and its applicability to the business.

The TIBCO platform has a narrower breadth of SOA Infrastructure Pattern support, but deep capability in the area of event-driven Service interaction and ESB-centrality, which is appealing for companies that desire those aspects of SOA infrastructure. TIBCO should widen their perspective on SOA's business centrality and messaging around business, rather than IT-oriented, SOA solutions.

*IBM's solutions show the greatest breadth of capability and support for SOA Infrastructure Patterns.*

*Software AG webMethods is rapidly becoming a force to contend with in the SOA marketplace.*

Software AG webMethods is rapidly becoming a force to contend with in the SOA marketplace. Their recent acquisitions and long-standing capability around legacy-enabled, standards-based Service-orientation makes them worth considering by a wide range of firms with a heterogeneous environment. Software AG doesn't as yet support as wide a range of SOA infrastructure patterns as IBM, nor does it have as broad a business-oriented solution or depth of professional services as the IBM organization, but all indications show that the company aims to grow its capabilities here.

The Oracle SOA Platform is a disparate collection of a number of acquisitions slowly forming itself into a cohesive SOA product stack. What makes evaluating the Oracle platform vexing is that the information that is available seems to portray the SOA platform as mostly a collection of existing Oracle technologies. The company needs to do a better job of communicating it. At the time of writing this report, Oracle has made an unsolicited offer to acquire BEA. Since this acquisition would have significant repercussions on the Oracle SOA platform stack, it is quite likely firms will have to wait to see how this will shake out. The combination of these factors explains why we were unable, for this report, to provide the same depth of analysis as provided to the other platforms detailed here. This will be possible once the Oracle SOA solution suite settles.

You can identify the vendors that have taken this approach when the demand that ESBs are necessary for SOA, and when their marketing deemphasizes the more significant SOA infrastructure challenges of governance, quality, and management. SOA is not a tactical project where you implement a technology, expose the right interfaces, and then move on. SOA involves two interrelated cycles: the underlying software lifecycle and the Service lifecycle, which takes place at the composition/configuration layer. Both should be highly iterative.

### **Making the SOA Infrastructure Choice**

The SOA vendor landscape is steadily creeping towards single-vendor suites and platforms. Oracle's potentially pending acquisition of BEA will only further cement in some customers minds that SOA infrastructure, many currently branded as ESBs, are best offered by large vendors as part of platform suites that address a wide range of SOA runtime issues.

*Many people who purchase ESBs don't understand the first thing about SOA, nor their own requirements and business drivers for that matter.*

Unfortunately, many people who purchase the technology don't understand the first thing about SOA, nor their own requirements and business drivers for that matter. Needed is effective knowledge transfer to understand the basics of the process of implementing SOA, including the process of figuring out an organization's requirements, which encompasses a semantic-level, Service-level, and process-level understanding of the problem domain or enterprise. Then, and only then, should you begin selecting SOA vendors. This could result in the selection of a single-vendor SOA platform, or a collection of technology from many vendors.

All SOA initiatives are different: different business requirements, different ways of driving business processes, managing services, and thus very different technology solutions. These differences require understanding your own business drivers, or the business reasons for implementing SOA in the first place. Define just what success will be, the proper amount of investment, strategic considerations around the growth of the business, and committed resources for the execution toward SOA.

## Copyright, Trademark Notice, and Statement of Opinion

All Contents Copyright © 2007 ZapThink, LLC. All rights reserved. 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.

## About ZapThink, LLC

ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences 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.

ZapThink helps its customers in three ways: by helping companies understand IT products and services in the context of Service-Oriented Architecture (SOA) and the vision of Service Orientation, by providing guidance into emerging best practices for Web Services and SOA adoption, and by bringing together all our audiences into a network that provides business value and expertise to each member of the network.

ZapThink provides market intelligence to IT vendors and professional services firms that offer XML and Web Services-based products and services in order to help them understand their competitive landscape, plan their product roadmaps, and communicate their value proposition to their customers within the context of Service Orientation.

ZapThink provides guidance and expertise to professional services firms to help them grow and innovate their services as well as promote their capabilities to end-users and vendors looking to grow their businesses.

ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into the best practices for planning and implementing SOA, including how to assemble the available products and services into a coherent plan.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOA by vendors, end-users, and the press. Respected for their candid, insightful opinions, they are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry. ZapThink was founded in November 2000 and is headquartered in Baltimore, Maryland.

### **ZAPTHINK CONTACT:**

ZapThink, LLC  
108 Woodlawn Rd  
Baltimore, MD 21210  
Phone: +1 (781) 207 0203  
Fax: +1 (815) 301 3171  
[info@zapthink.com](mailto:info@zapthink.com)

