zapthink
white paper

The SOA Implementation Framework

The Future of Service-Oriented Architecture Software

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

THE SOA IMPLEMENTATION FRAMEWORK

THE FUTURE OF SERVICE-ORIENTED ARCHITECTURE SOFTWARE

April 2004

Analyst: Jason Bloomberg

Abstract

The world of distributed computing is currently in the midst of a major transition, as tightly coupled, integration-centric approaches gradually lose favor and give way to loosely coupled, Service-oriented computing techniques. At the core of this trend are Service-oriented architectures (SOAs), which promise greater flexibility in the way that companies produce and consume IT assets, in particular when IT environments are heterogeneous and business requirements are dynamic.

However, building SOAs is challenging for most organizations, for several reasons: they involve a different way of thinking about software resources, they require a level of architectural discipline, and companies need a range of software solutions to build, run, and manage an SOA. In particular, companies need a framework of capabilities that include access to data sources and the composition of Services into business processes in a secure, managed environment.

Today, most enterprises must purchase a range of products to assemble the software they need to build an SOA. However, there is a new class of product called the *SOA Implementation Framework* that offers all the elements a company would need to build and run an SOA. Such frameworks are only now coming to market, but one such framework, Rogue Wave Platform X, from Rogue Wave Software, a division of Quovadx, is well-positioned to be a leader in this nascent market for comprehensive SOA software frameworks.

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.

Table of Contents

I. The Service Orientation Value Proposition

There is a sea change facing the world of information technology today. After years of ups and downs, from Y2K and the Internet buildout of the nineties, to the dot.com backlash and IT downturn of the first part of this decade, we’re finally raising our heads above water. Companies are finally looking to address strategic IT issues again—finally looking to build flexible, robust IT infrastructures that meet the needs of business in increasingly dynamic corporate environments.

Service orientation is the perspective of IT functionality being available as discoverable Services on the network.

The sea change that IT is currently undergoing is the shift to *Service orientation* (SO), based upon open standards-based computing. Service orientation is a way of viewing software assets on the network—fundamentally, the perspective of IT functionality being available as discoverable Services on the network. Essentially, Service orientation provides business users with understandable, high-level business Services they can call upon and incorporate into business processes as needed. The Service orientation vision is therefore one of agility and flexibility for users of technology, coupled with an abstraction layer that hides the complexity of today’s heterogeneous IT environments from those users.

Service-Oriented Architecture: the Next Step in Information Technology

Web Services provide standards-based interfaces to software functionality.

For this Service-oriented world to become a reality, however, companies must move to a new architectural approach known as *Service-Oriented Architecture* (SOA). SOA is architecture that represents software functionality as discoverable Services on the network. SOAs have been around for many years, but the difference with the SOAs we talk about today is that they are based on standards, in particular, *Web Services*. Web Services provide standards-based interfaces to software functionality. Producers of these Services may publish information about them in a Service registry, where Service consumers can then look up the Services they need and retrieve the information about those Services they need to bind to them.

Service-oriented architectures are nothing new; the Common Object Request Broker Architecture (CORBA) and the Distributed Component Object Model (DCOM) have long provided similar functionality. These existing approaches to Service orientation, however, suffered from a few difficult problems. First, they were *tightly coupled*, which meant that both ends of each distributed computing link had to agree on the details of how each participant worked. A code change to a COM object, for example, required corresponding changes to the code that accessed that object. Secondly, such Service-oriented architectures were *proprietary*. Microsoft unabashedly controlled DCOM, and while CORBA was ostensibly a standards-based effort, in practice, implementing a CORBA architecture typically necessitated the decision to work with a single vendor’s implementation of the specification, because each vendor's interpretation of the standard varied enough to prevent seamless interoperability.

Fortunately, the combination of Web Services and SOAs resolves the issues of CORBA and DCOM approaches to SOAs by enabling three key concepts ZapThink calls *idées fortes*, or powerful ideas. These *idées fortes* are loose coupling, coarse granularity, and asynchrony.

Web Services-based SOAs are loosely coupled.

Loose Coupling –The fundamental problem with traditional distributed computing architectures is that they are *tightly coupled*, which means that the architect must design each component system with the other systems in mind. As a result, making changes to one tightly coupled system often affects the whole architecture, requiring expensive and difficult reworking. Web Services-based SOAs, on the other hand, are *loosely coupled*. Each Web Service describes how other systems, known as Web Service consumers, can connect to it and exchange information with it. Therefore, the consumers need have no knowledge beforehand about a Service, other than where to find it and that it is based on the common Web Services standards. As a result, a developer can make changes to a Web Service without breaking the Service-oriented architecture.

Coarse Granularity – The traditional approach to getting information in and out of an enterprise application (such as an ERP system) is via an application programming interface, or API. An API is little more than a set of method calls in a particular programming language that distributed systems can execute to exchange data with the application. Such APIs are typically *fine grained*, which means that each method call is a detail-oriented, technical construct for use by programmers. For two systems to communicate a complex business task via an API, they typically must exchange many of these fine-grained messages. Web Services, however, are actually inappropriate for sending fine-grained messages. Web Services are in fact much better suited to building coarse-grained interfaces. A *coarse-grained* interface rolls up the functions of many different API calls into a small number of business-oriented messages.

Web Services are at their most powerful when they are used to exchange coarse-grained information between systems.

Web Services are at their most powerful when they are used to exchange coarse-grained information between systems. Instead of large numbers of arcane API calls, an enterprise application can expose its functionality as coarse grained, business-oriented Web Services, with functions like "place order," "provision patient," or "get customer information." Coarse granularity clearly depends on loose coupling, because the Web Service consumer does not care how the Web Service puts together the information it needs.

Asynchrony – Finally, synchronous communications consist of round-trip messages in which the sender waits for a reply. Submitting a Web page form and waiting for a confirmation page is a familiar example of a synchronous operation. In contrast, with an asynchronous message, the sender can submit a request, and then go about its work. If a reply does come, then the original sender can pick it up when it wants. Email works asynchronously, for example. Web Services-based SOAs enable sending and receiving both synchronous and asynchronous messages, which is our *idée forte*.

Loosely coupled, coarse grained, asynchronous SOAs provide a layer of abstraction that hides the complexity of the underlying technical implementation details from the business user.

Loosely coupled, coarse grained, asynchronous SOAs provide a *layer of abstraction* that hides the complexity of the underlying technical implementation details from the business user who takes advantages of the Services the SOA exposes. These business-oriented Services are therefore both location independent and loosely coupled from the technology that supports them, thus enabling business agility—a primary goal for most businesses.

The Road to Business Agility

The power and flexibility that SOAs can offer the enterprise are substantial. If an organization abstracts its IT infrastructure so that it presents its functionality in the form of coarse-grained Services that offer clear business value, then the consumers of those services (whether they are at the same company or one of that company's business partners) can access those Services independent of the underlying technology that supports them. Furthermore, if Service consumers can dynamically discover and bind to available services while they are active, then the IT infrastructure behind those Services can offer extraordinary flexibility to the businesses that invoke them.

The difference between the practice of SOA and other approaches to enterprise architecture is in the business agility that SOA offers. *Business agility* is the ability of a company to respond quickly and efficiently to change, and to leverage change for competitive advantage. For the architect, building an architecture that provides business agility means creating an IT infrastructure that meets as-yet unknown business requirements—a situation that throws traditional IT planning and design out the window.

Business agility *is the ability of a company to respond quickly and efficiently to change, and to leverage change for competitive advantage.*

The current transition to Service orientation is fundamentally different from the last major distributed computing transition: the one from client/server to n-tier architectures in 1996-97. That last build-out heralded the beginning of the dot.com boom, where Internet-related investment coupled with Y2K expenditures created a kind of IT "perfect storm" so dramatic it led to a worldwide economic bubble, and subsequent downturn. Today, of course, the economic environment for technology adoption has completely changed, and this return to the "new business normal" is accelerating the move to Service orientation. Rather than promoting massive build-out or extensive rip-and-replace, Service orientation embraces heterogeneity and obtaining greater value from existing legacy technology. Today's distributed computing transition, while every bit as significant as the ones that came before, has an entirely different economic model. Instead of massive IT investment, today's IT executive is concerned with *thrift*.

Reworking existing brittle, expensive IT infrastructures into flexible, Service-oriented environments promises substantial cost savings, not just in terms of reduced integration expense and squeezing more value out of existing IT investments, but most dramatically in terms of business agility. After all, change comes in many forms: changes in the marketplace, in technology, in the world at large. Companies that can make effective use of a changing environment are better able to compete and thrive in any business climate.

Information technology is often the area most relevant to discussions of business agility, because achieving agility begins with removing the bottlenecks that impede it, and IT has traditionally been the source of most bottlenecks. In fact, companies are so used to the fact that IT decision-making and implementations impede their organization that technology and its limitations often drive business decisions. Service orientation, however, has the potential to change this equation, and enable business decisions to finally drive their technology decisions. On the other hand, building SOAs requires investment and commitment on the part of enterprises. The long-term business benefits of Service orientation, however, can justify such investments.

Service orientation has the potential to enable business decisions to finally drive technology decisions.

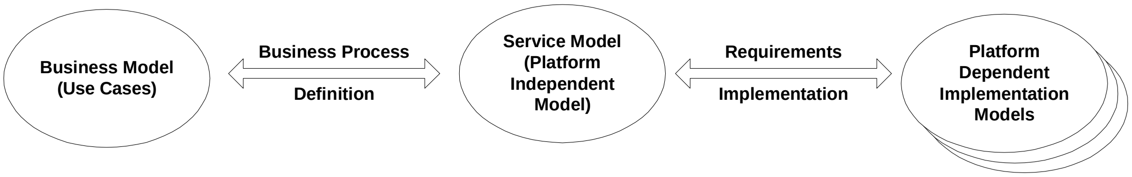
The SOA Metamodel

How, then, does a company actually build an SOA? The starting point is to realize that SOA is a form of enterprise architecture. Architecture is essentially the fundamental organization of a system embodied by its components, their relationships to each other and to the environment and the principles guiding its design and evolution. Enterprise architecture extends this definition of architecture to cover both the business and technology inherent in an organization.

The best way to visualize architecture is through the use of *models,* which are conceptual representations of actual systems. Architects can then represent the overall architecture with a *metamodel*, which is a model that itself contains models. To represent SOA, therefore, we use the *SOA Metamodel*, represented in Figure 1 below.

The core of the SOA Metamodel is the Service Model, represented as the central oval in Figure 1. The Service Model represents the Services an enterprise has in production. The left oval, then, is the Business Model, which represents the users and their requirements. On the right are several Implementation Models, which represent the technology underlying the Services.

Figure 1: The SOA Metamodel



Copyright © 2003 ZapThink LLC

In the SOA Metamodel, the business drives the Services, and the Services drive the technology.

In the SOA Metamodel, the business drives the Services, and the Services drive the technology. The SOA Metamodel connects its component models with two-way arrows that represent feedback loops. The left arrow represents the translation of business requirements into changes in the Service Model that inform the IT organization about the requirements, and the right arrow represents the process of implementing those requirements in code. It is important to note that there is no arrow directly between the Business Model and the platform dependent models—this feedback always feeds through the Service Model.

An enterprise must actively maintain the Service Model so that it can act as a central clearinghouse for business requirements from each constituency within the organization. The IT organization can then look to the Service Model as the source of the business requirements it needs to make changes to the underlying systems, as represented in the Implementation Models.

The Service Model becomes the point of contact between the business and technology views, and acts as the conduit for communication across the enterprise. Business users work with coarse-grained business Services, orchestrating them into processes as needed depending on the fluctuating requirements of the business. Technologists, on the other hand, work to build and maintain the abstraction layer between the Services and the underlying technology. The central model, representing the Services themselves, acts as the axis around which the business moves.

The most important feature of the SOA Metamodel is in fact this delineation between the business and technology realms, coupled with the explicit modeling of the two-way interaction between the two realms. This balance between the two realms enables the business to drive the technology in an environment of flux.

Traditional approaches to software architecture presuppose a traditional software development lifecycle, where users define their needs, and then IT builds and deploys the required system. However, companies have realized that this traditional, “waterfall” approach typically does not work, for a variety of reasons that boil down to risks that develop as a result of unknown or changing circumstances. As a result, companies react to the risks of the traditional approach by constraining the expectations of the business, essentially allowing technological risks and limitations to drive the business. SOA reverses this conundrum, providing sufficient flexibility to allow business to drive the technology.

II. The Shift to Service Orientation

As companies seek to achieve the business agility promised by SOAs, they face a substantial challenge, as the IT marketplace is currently in a transitional phase. Integration-based distributed computing products are gradually giving way to SOA-based approaches, but this shift is currently in progress. Furthermore, the movement toward SO computing affects many areas of IT beyond application integration. As a result, companies must be prepared to evaluate the products that are currently on the market carefully, and place them in the context of the broader shift to SO computing that the industry currently faces.

Companies must be prepared to evaluate the products that are currently on the market carefully, and place them in the context of the broader shift to SO computing that the industry currently faces.

Shifting Sands in IT

The key market reality that companies must understand is that at this point in time, the shift to Service Orientation affects all aspects of enterprise distributed computing. We are not moving toward a world where SOAs are one of several possible distributed computing architectures in the enterprise. Rather, other architectures, including n-tier, client/server, and message bus, will all come to be understood within the context of SOA. Fundamentally, SOAs provide a layer of abstraction above other architectural styles, providing an asynchronous, loosely coupled, coarse-grained Service interface on top of a heterogeneous mix of architectures.

SOAs provide a layer of abstraction above other architectural styles.

To understand how the shift to SO computing, then, will affect an individual company’s IT initiatives, it is important to understand that as any emerging market matures, two fundamental economic forces cause change within the industry: the forces of innovation and consolidation. As new technologies and approaches for solving customer problems become economically viable, new opportunities open for startups and other nimble companies to devote investment dollars into developing innovative products that capitalize on the emergence of nascent markets. Then, as those markets expand, larger players realize that in order to grow, stay relevant, and continue to serve their current customers, they must also leverage the innovative solutions the smaller players are bringing to market, either by building the innovative solutions themselves, or by acquiring the new capabilities from the new entrant vendors.

As a result, in any emerging market, there is a turning point when consolidation comes to dominate innovation. Smaller players either go out of business, merge, or become part of a larger player’s plans to enter the new market. Once this consolidation settles down, there are typically a small number of new entrants left standing—those who managed to acquire rather than be acquired, find a defensible niche, or are able to continue to innovate at the edge of the market.

However, the number of survivors is typically much smaller than the number of startups who fail or become acquired.

At this point in time, the SO marketplace is near the turning point, as established players are beginning to consolidate the market, and the window for new, innovative startups closes, until the next wave of innovation strikes.

The SOA Implementation Framework

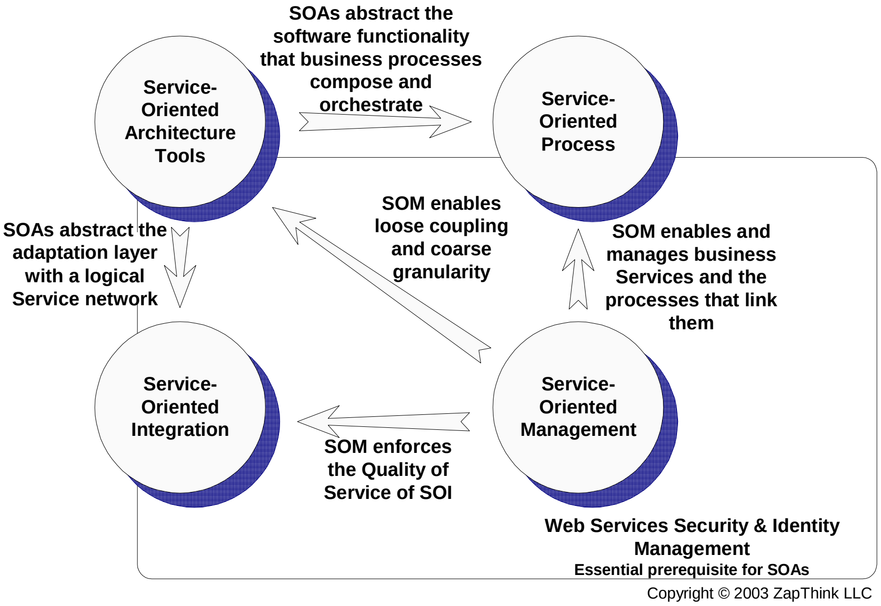
Understanding the changes to established IT market segments that the move to SO will create is therefore a critical part of the big picture of the development of the IT market as it enters the next consolidation phase. In fact, for each segment of the software market (as well as selected hardware segments), the overall movement toward SO is leading to one or more shifts in product categories, as Web Services and other SO functionality becomes an integral part of so many different products in the marketplace.

ZapThink sees a consolidation of most SO functionality into a single market category that contains broad functionality, including features that are currently associated with the security, management, process, integration, and tools segments.

Furthermore, as the market consolidates, so too do the feature sets of the various products in each segment of the market. In fact, ZapThink sees a consolidation of most SO functionality into a single market category that can be delivered as individual products, product suites, or service offerings that contain broad functionality, including features that are currently associated with the security, management, process, integration, and tools segments. For want of a better term, ZapThink refers to this full-featured SO market as the *SOA Implementation Framework*, or SOAIF.

ZapThink uses the term “implementation framework” because it provides all the elements needed to craft a solution (hence, “framework”), and it offers both design time and runtime capabilities (hence, “implementation”). Fundamentally, a SOAIF offers all the software functionality a company needs to build and run an SOA, as shown in Figure 2 below:

Figure 2: The SOA Implementation Framework


SOAs abstract the software functionality that business processes compose and orchestrate
Service-Oriented Architecture Tools
Service-Oriented Process
SOAs abstract the adaptation layer with a logical Service network
SOM enables loose coupling and coarse granularity
SOM enables and manages business Services and the processes that link them
Service-Oriented Integration
Service-Oriented Management
SOM enforces the Quality of Service of SOI
Web Services Security & Identity Management
Essential prerequisite for SOAs
Copyright © 2003 ZapThink LLC

As shown above, the SOAIF comprises functionality from the five key SO market segments: security, management, process, tools, and integration. However, Figure 2 does not represent five discrete markets; in fact, there are many elements of functionality found in more than one of the component market segments included in the SOAIF.

As the SO market consolidates, ZapThink believes that many incumbents will offer SOAIFs. Others will offer products that contain most of the functionality included in the SOAIF, but will leave out certain functions, preferring instead to leave those to partners. Still other vendors will continue to focus on offering products that contain functionality that remains within a specific market niche like security or management, preferring to offer customers a best-of-breed solution rather than a soup-to-nuts offering.

The SOAIF, however, differs in a fundamental way from the enterprise software packages and comprehensive middleware platforms found in today's mature markets. The difference, of course, is that SO leverages interoperability based upon open standards, most notably the Web Services family of standards. Gone for good are the days of proprietary, tightly coupled software suites that lock customers into a single vendor. Another key difference is the adherence to the SO *idées fortes*: loose coupling, coarse granularity, and asynchrony. It's the combination of standards on the one hand with the architectural principles of SO on the other that make the SOAIF a unifying implementation framework. Yet, while the SOAIF is feature complete, it is also modular, allowing customers to switch out parts of the SOAIF for best-of-breed point products as desired.

Gone for good are the days of proprietary, tightly coupled software suites that lock customers into a single vendor.

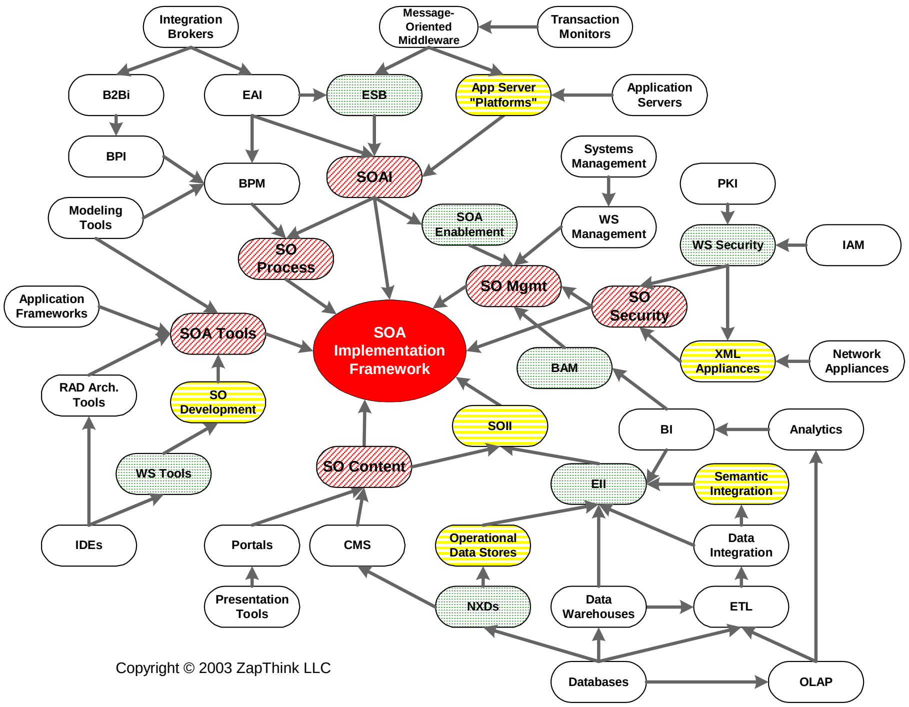
The March Toward the SOAIF

While the security, management, process, tools, and integration markets form the core transitional markets whose functionality the SOAIF will subsume, these five markets are actually part of a software ecosystem that is gradually coalescing toward the SOAIF. Traditional markets will remain, of course, but as those markets become Service-oriented, the vendors who play in those markets will find themselves competing with SOAIF vendors. Figure 3 below illustrates the various IT markets that ZapThink feels will be most affected by the consolidation of the IT market as companies adopt SO.

At the center of Figure 3 is the SOAIF. Around the perimeter of the figure, in white, are selected incumbent markets. The ones that appear in the figure are not intended to be an exhaustive list, but rather a selection of key established markets that will be particularly affected by the movement toward the SOAIF. The white boxes represent established markets that are currently undergoing transition into other markets as they become Service-oriented. These markets will likely not exist at all in their current form once SO has been adopted by the majority of enterprises.

In green dots are transitional Web Services-related markets, including Enterprise Service Buses (ESBs), Web Services tools, Native XML Databases (NXDs), Web Services Security solutions, and Business Activity Monitoring (BAM) solutions. ZapThink considers markets to be transitional because they have emerged recently, but will be subsumed by other, SO market segments. The markets indicated in yellow horizontal stripes in Figure 3 represent markets that are likely to remain distinct from the SOAIF market, even as the overall market consolidates into the SOAIF market. In some of these cases, like the Operational Data Store (ODS) market, strategies that involve embedding the solution within other vendors' offerings will become increasingly important, while in other cases, like the XML Appliance market, a different category such as firewalls or content-based routing will likely subsume the smaller market. Finally, the red diagonally hatched boxes represent core SO market segments that will continue to exist for a long time as companies come to adopt the SOAIF.

Figure 3: The SOAIF Market Transition Map



Copyright © 2003 ZapThink LLC

Because the SOAIF Market Transition Map represents market trends more so than technology trends, vendors have some measure of flexibility regarding their positioning relative to the map. Clearly, markets connected by arrows will contain products that share functionality, so a single product might belong in two or more markets as identified on the map. What ZapThink is seeing in the marketplace are incumbent vendors who are shifting their marketing to move toward the center of the map, as they add Web Services and SO functionality to their products, while at the same time remaining in the established markets that have been their traditional bases. This trend is important to highlight: companies do not need to move off of existing markets to participate in new ones in the short term, unless they believe that it will be strategically advantageous for them to do so.

Advice for Companies Exploring SOAs

Many of the products on the market today are generally mature enough to provide the capabilities companies need to begin their SOA initiatives.

Even though the markets for SOA-related products are currently in a state of transition, that doesn’t mean that companies should hesitate when starting or proceeding with their existing SOA plans. On the contrary, many of the products on the market today are generally mature enough to provide the capabilities companies need to begin their SOA initiatives. It is important, however, for companies to understand how the various elements of the SOA solution set fit together, in order to understand what they need and how to work with vendors.

While it is possible to assemble several products from a number of vendors, it usually makes sense to select a single vendor who can provide a leadership position for your specific initiative.

While it is possible to assemble several products from a number of vendors, it usually makes sense to select a single vendor who can provide a leadership position for your specific initiative. While it’s not necessary (and often not desirable) to purchase all necessary software from this single vendor, it often makes economic and logistical sense to follow a single vendor’s lead regarding the best approach to the SOA buildout.

It is imperative, therefore, to select the right primary vendor. In addition to the obvious criteria of features and quality, companies should also look for a vendor who offers a product suite that supports interoperability, and is inherently modular in nature. Rogue Wave Platform X, from Rogue Wave Software, a QUOVADX™ division is just such a product

III. Rogue Wave: Taking the Lead in SOA Frameworks

Quovadx' acquisition of Rogue Wave combines a healthcare vertical industry SOA leader with a horizontal SOA tools vendor. The combined entity is now in the market with Platform X, a pioneer in the nascent SOA Implementation Framework (SOAIF) market, combining application integration, SO process capabilities, and SOA tools into a single, modular, standards-based product.

Rogue Wave® Platform X: SOAIF Leader

Rogue Wave**®** Platform X helps companies with heterogeneous IT environments streamline, simplify, and integrate disparate processes, systems, and business applications. Built on an XML foundation as well as other open standards, Platform X leverages Web Services to help companies create new, composite applications from legacy systems.

Platform X is a collection of Services called micro-services *that work jointly to compose existing Services together into composite applications.*

Rather than using a traditional platform approach, Platform X is a collection of Services called *micro-services* that work jointly to compose existing Services together into composite applications. Micro-services also orchestrate applications by deploying new Services based on defined processes and business rules. These micro-services include the following types of Services:

- *Enterprise Access Services* – Encompass information from numerous data sources into a single, consistent interface without translating the core data from their original format. This consolidation makes working with data simple and efficient. Platform X provides numerous Enterprise Access Services that connect to Web Services as well as legacy data sources. This consistent interface to data streamlines the creation of evolving SOA applications that leverage existing business assets, while exposing those assets as Web Services.
- *Dynamic Process Services* – once Enterprise Access Services have exposed mission-critical applications and data as Web Services, Dynamic Process Services use those Web Services to build new processes. Platform X supports long-running processes, complex decision and routing criteria, and any type of data transformation.
- *WebAccel® Presentation Services* – create a front-end user interface on the new composite application, or in other words, provide the view in a Model-View-Controller (MVC) architecture.

What distinguishes Platform X from many other vendors' attempts to build an SOAIF is that the Rogue Wave® solution is built as a Service-oriented framework rather than a platform.

What distinguishes Platform X from many other vendors' attempts to build an SOAIF is that the Rogue Wave® solution is built as a Service-oriented framework rather than a platform. The typical platform approach to building an SOA execution environment is to create tightly coupled components that expose Service interfaces. In Rogue Wave's case, however, Platform X is built with micro-services, thus incorporating loosely coupled components at all levels of the product.

Platform X is also unusual in that it provides the "nuts and bolts" integration capabilities of a message bus or EAI solution, but also provides the process definition and management capabilities of a business process management tool.

Rogue Wave then rounds out the Platform X product with management, and user interface capabilities.

Therefore, Platform X provides a SOA implementation framework for developing SOA applications from existing systems. It's a lightweight solution that enables businesses to leverage large investments in existing technology infrastructure, while at the same time providing the agility to keep pace with the changing needs of the organization. As an illustration, Platform X boils down composite application development to three steps:

- Expose existing systems and data as Web Services.
- Use the Dynamic Process Services to quickly develop new applications based upon new business requirements.
- Create a user interface for the new composite application with WebAccel Presentation Services, targeting any device for delivery.

Because Platform X is fundamentally a collection of Services, companies can deploy and manage it through a phased, component-based implementation, installing only that functionality that meets their current business needs. Furthermore, Rogue Wave doesn't tie Platform X to a traditional yearly release cycle, enabling them to provide updates to various component Services in response to customer requirements.

Rogue Wave Platform X is well-positioned to become a leader in the SOA implementation framework market because they are one of the few vendors who work on both sides of the SOA Metamodel by tying SO process creation, management and user interface capabilities to the underlying SO integration framework. Many vendors who are struggling to offer SO capabilities today are either focusing on integration or business process, but Rogue Wave understands that both capabilities are important elements of a fully functional SOA.

Many vendors who are struggling to offer SO capabilities today are either focusing on integration or business process, but Rogue Wave understands that both capabilities are important elements of a fully functional SOA.

IV. The ZapThink Take

The promise of Service-oriented architecture is undeniable: finally, standards-based distributed computing can be flexible and dynamic enough to enhance business agility, instead of IT restricting the capabilities and flexibility of the business. However, SOAs are a challenge to implement properly—they require many different IT elements to work together, and an overall discipline that affects both business and technology. Furthermore, companies don't want to be locked into a single vendor, but they also don't want to buy enterprise IT solutions piecemeal and then have to assemble them. Fortunately, we finally have broad industry agreement on open standards that should provide a level playing field for vendors to provide comprehensive, yet truly modular solutions.

The answer to the SOA implementation challenge is to buy a broad framework with modular elements that can be added as needed. This need for a broad framework is especially important as companies move toward building SOAs, because building an SOA requires so many different elements, including security, management, process, architecture and user interface tools, and integration capabilities. ZapThink believes that the broad distributed computing market is beginning a major shift, as vendors Service-enable their software and work to coordinate point solutions into frameworks that offer customers who are building SOAs the flexibility and agility they require. Rogue Wave Platform X provides such a framework today.

V. About Rogue Wave Software

Rogue Wave Software is a division of Quovadx, Inc. (Nasdaq: QVDX), a software and services firm based in Englewood, Colo., that helps more than 20,000 enterprise customers worldwide develop, extend and integrate applications based on open standards. The Rogue Wave Software division, based in Boulder, Colo., provides reusable software components and services that facilitate application development. Today, more than 300,000 developers at some of the world's leading companies use Rogue Wave® products to develop enterprise-level applications. Rogue Wave Software's newest offering, Lightweight Enterprise Integration Framework™ (LEIF), uses XML and Web services to unlock C/C++ business logic so it can be leveraged in new business applications. Rogue Wave Software operates throughout the United States, Europe and Asia and can be found on the Internet at http://www.roguewave.com.

Copyright, Trademark Notice, and Statement of Opinion

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 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.

ZapThink's role is to help companies understand these IT products and services in the context of SOAs and the vision of Service Orientation. ZapThink provides market intelligence to IT vendors who offer XML and Web Services-based products to help them understand their competitive landscape and how to communicate their value proposition to their customers within the context of Service Orientation, and lay out their product roadmaps for the coming wave of Service Orientation. ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into how to assemble the available products and services into a coherent roadmap to Service Orientation. Finally, ZapThink provides demand intelligence to IT vendors and service providers who must understand the needs of IT users as they follow the roadmap to Service Orientation.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOAs by vendors, end-users, and the press. 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 October 2000 and is headquartered in Waltham, Massachusetts. Its customers include Global 1000 firms, public sector organizations around the world, and many emerging businesses. ZapThink Analysts have years of experience in IT as well as research and analysis. Its analysts have previously been with such firms as IDC and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, CPExchange, ebXML, EIDX, and CompTIA.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how XML and Web Services impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
11 Willow Street, Suite 200
Waltham, MA 02453
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com