zapthink
white paper

SERVICE-ORIENTED ARCHITECTURE: WHY AND HOW?

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

SERVICE-ORIENTED ARCHITECTURE: WHY AND HOW?

May 2003

Analyst: Jason Bloomberg

Abstract

It seems like every few years there's some new computing approach that promises to solve all the ills facing IT in the enterprise. Hopeful IT executives invest time and money in the new approach, only to find that fundamental problems like complexity, inflexibility, and brittleness remain. Today, there is yet another promising new approach known as *Service-oriented architecture*. Is this new approach just another technology fad, or does it actually offer solutions to IT's perennial problems?

As a matter of fact, today's Service-oriented architectures based on Web services leverage the lessons learned from previous computing approaches and break new ground. Based on open standards, these new architectures expose IT functionality in a fundamentally more flexible, responsive manner. The strategic value that such architectures offer is thus in the form of IT environments and approaches flexible enough to enable companies to leverage changing business environments for competitive advantage at a reduced cost. In other words, Service-oriented architectures allow companies to do more with less.

I. Today's Fundamental Business Drivers

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.* Most enterprises today contain a range of different systems, applications, and architectures of different ages and technologies. This heterogeneity is more pronounced than it was as little as ten years ago. The enterprise application story of the 1990s was all about suites. Buying a few large packages with several tightly integrated modules made more sense than going with a best-of-breed, "mix and match" approach, so the argument went, because integrating products from multiple vendors was such a nightmare.

Today, companies cannot afford to take a single-vendor approach to IT.

Today, however, companies cannot afford to take a single-vendor approach to IT, because application suites are inflexible, which potentially leads to higher costs. Interoperability is the order of the day, and drives new IT investment. The appeal of a best-of-breed approach, after all, is often undeniable: assemble just those products that best meet the needs of the enterprise, and when multiple products in a particular category meet those needs, then let the vendors compete for the business. Embracing heterogeneity is therefore an inherently thrifty best practice, because a best-of-breed approach can be more affordable.

Change is the second theme underlying the questions that today's IT executives face. Change is an ever-present theme in today's IT world for several reasons:

- Competition is growing more fierce, which leads to shortening product cycles, as companies look to gain advantage over their competition.
- Customer needs and requirements change more quickly, in response this cycle of competitive improvements in products and services.
- Broad economic forces including globalization and eBusiness are accelerating the pace of change.
- Improvements in technology continue to accelerate, feeding the increased pace of changing customer requirements.

Business must rapidly adapt to survive, let alone succeed in today's dynamic competitive environment, and the IT infrastructure must follow.

Ever since applications first started communicating with one another, companies have been struggling with how to integrate systems. As business applications became more powerful and more complex, they required increasing levels of customization. Finally, the over-investment in IT in the 1990s coupled with the tough economic environment of the 2000s is putting the IT executive in an exceptionally difficult position. Such executives are asking the following questions:

- How do I make my IT environment more **flexible and responsive to changing business requirements**, without breaking the bank?
- What can I do to reduce the **complexity** of my IT environment?
- How do I get **more value out of the existing data** on my systems?

- How can I make application purchasing decisions, **without getting locked into a single vendor**, and without spending an arm and a leg on customization?
- How can I pull cost out of my IT budget, by **reducing unnecessary expenses**?
- What is the best strategy for solving all these problems, given that my IT environment is **heterogeneous**, and is going to stay that way?

Business is calling upon IT more than ever to respond quickly and efficiently to shifting requirements.

Business is calling upon IT more than ever to respond quickly and efficiently to shifting requirements—but IT is still facing a plethora of issues, including:

- **Costly, inflexible integration technologies** that present unacceptable risks of failure to the enterprise.
- **Monolithic business applications** that require expensive customization and maintenance.
- A limited ability to **participate in value networks** because the automation of business processes that involve suppliers or customers is complex, inflexible, and insecure.
- A failure of IT to address the **changing needs of customers** in a nimble, cost-effective fashion.
- The difficulty with **accessing and utilizing key business information** locked inside closed and proprietary systems.
- A lack of **visibility and control** into automated business processes for line-of-business management, combined with excessive management complexity.

Into this tough environment an evolutionary, standards-based approach to architecting IT application functionality is gaining traction at enterprises across many industries. Known as *Service-oriented architecture* (SOA), this new way of thinking about how to integrate IT resources and access application functionality in the enterprise promises to address all of the issues listed above.

Fundamentally, SOAs have the flexibility and responsiveness to enable business priorities to finally drive technology decisions.

SOAs have the potential to rise to the challenges of brittle application infrastructures, inflexible technology, and high-risk, high-cost IT. Fundamentally, SOAs have the flexibility and responsiveness to enable business priorities to finally drive technology decisions. On the other hand, building service-oriented infrastructures is not easy. It requires commitment and expertise. The long-term business benefits of SOAs, however, can justify such investments. Many enterprises have already implemented SOAs and achieved quantifiable benefits from their investment in this new architectural approach. This paper provides the context for SOAs, details the benefits of adopting an SOA, and provides a roadmap for enterprises who are also looking to gain the extensive business benefits that service-oriented architectures can offer.

II. The Current State of IT: The Big Picture

Today's IT executives face a number of items on their plate, including:

- **Business applications** – from large suites like enterprise resource planning, supply chain management, and customer relationship management packages to portals and desktop productivity packages, as well as industry-specific applications.

- **Heterogeneous Systems** – mainframes, midrange computers, servers of all types and sizes, desktop systems, and any number of special-purpose systems, depending on industry.
- **Distributed computing architectures** – client/server architectures for database-centric applications and n-tier architectures for Internet or intranet purposes, to name a few.
- **Middleware** – the glue that keeps systems talking to each other, including message-oriented middleware, application servers, enterprise application integration solutions, and transaction monitoring and processing systems.

This plethora of technology, while intended to address *business issues, often* presents *issues that the IT executive must resolve.*

This plethora of technology, while intended to *address* business issues, often *presents* issues that the IT executive must resolve. Most of these issues fall into three broad categories: *complexity*, *inflexibility*, and *brittleness.*

- **Unnecessary complexity** – Today's enterprise IT environment contains many kinds of systems that work in many different ways. IT shops must hire large, multi-skilled groups of workers to develop, deploy, and manage the heterogeneous collection of applications and systems. Add to the mix software upgrades, mergers and acquisitions, and the ubiquitous need for security, and many enterprises have an intractable tangle of problems on their hands.
- **Inflexibility** – Compounding the complexity in the enterprise is the inherent inflexibility of the IT companies have in place. Virtually every enterprise has existing business applications that are difficult to upgrade, difficult to interoperate with, and worst of all, impractical to replace. Furthermore, dissimilar systems tend to be difficult to integrate, each exposing different interfaces with different rules. Integration is therefore an expensive, difficult process that yields inflexible distributed systems.
- **Brittleness** – The flipside of inflexibility is brittleness: the risk of failure and other problems inherent in inflexible systems. Traditional approaches to building IT environments lead to a "spaghetti" approach to integration. As a result, when business processes or requirements change, IT departments must either undertake expensive, risky upgrade projects, or simply make do with technology that no longer meets the needs of business. It is the risk inherent in making changes to inflexible distributed systems that is of most concern for enterprises, because of the complex interdependencies among individual systems.

Today's IT executives need fresh approaches to dealing with heterogeneous environments and an increasing pace of change, in the face of tight budgets and a tough economy.

Naturally, the problems of complexity, inflexibility, and brittleness are nothing new in the enterprise. The conclusion is evident: today's IT executives need fresh approaches to dealing with heterogeneous environments and an increasing pace of change, in the face of tight budgets and a tough economy. Fortunately, there is such an approach, and it's a reality today—an approach known as *service-oriented architecture*.

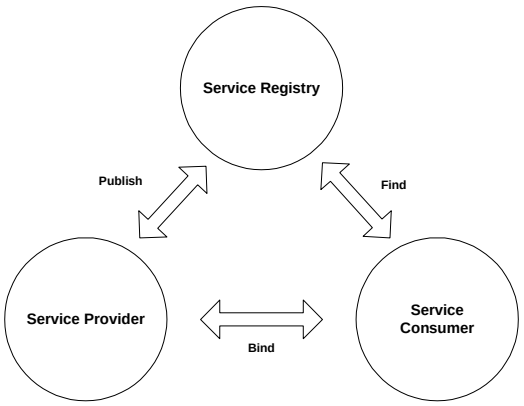
SOAs are an approach to enterprise IT that considers software resources as services available on a network.

III. What is a Service-Oriented Architecture?

Service-oriented architectures (SOAs) are an approach to enterprise business systems and applications that considers software resources as services available and discoverable on a network. Such services provide functionality to the business while hiding the underlying implementation details. Providers of these

services must be able to publish information about them in a *service registry*, where service consumers, or requestors, can then look up the services they need and retrieve the information they need about those services to find and use, or "bind" to them. This "publish-find-bind" triangle forms the core of a Service-oriented architecture, as shown in Figure 1 below:

Figure 1: The SOA Triangle



SOAs promise to address the complexity, inflexibility, and brittleness issues of existing approaches to integration, while embracing heterogeneity in the IT shop. SOAs are nothing new; the *Common Object Request Broker Architecture* (CORBA) and Microsoft's *Distributed Component Object Model* (DCOM) have long provided this functionality. These older approaches to service orientation, however, suffered from a few difficult problems. First, they were *tightly coupled*, which meant that the implementation of a service provider (the software that offered the service) had to be closely tied to the implementation of the service consumer (the software that accessed the service). Such tight coupling meant that changes to service providers and consumers were difficult, because all such changes had to be planned and carried out in a highly coordinated fashion.

Secondly, such SOAs 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. Finally, CORBA and DCOM were *fine grained*, which means that service requests and responses typically contain small amounts of specific information, requiring many round trips between the consumer and the provider of the service.

In spite of all of these issues, the concept of service orientation continued to make sense, provided that the problems of proprietary approaches, tight coupling, and fine granularity could be solved. It is within this architectural context that Web services were first imagined. Web services are an open standards-based way of creating and offering software services. SOAs implemented with Web services are a significant improvement upon DCOM and CORBA's weaknesses, for several reasons:

Web services are an open standards-based way of creating and offering software services.

- Because Web services are **based on open standards** that most software vendors agree upon, it's possible to get software that interoperates out of the box.

- Web services are designed to **promote loosely coupled interactions** between Web service providers and consumers. Such loose coupling means that provider and consumer applications can be developed independently of each other.
- Web services are **able to exchange structured documents** that contain different amounts of information.. In other words, Web services can be *coarse grained*. Web services are also able to exchange information about that information, known as metadata.

SOAs offer a different perspective on the way that an organization accesses its IT capabilities. Companies must understand this new perspective in order to get the full advantage of their Web services implementations, and build IT environments that are flexible and responsive.

SOAs, however, offer more than technical advantages over other approaches to distributed computing. Fundamentally, SOAs offer a different *perspective* on the way that an organization accesses its IT capabilities. Companies must understand this new perspective in order to get the full advantage of their Web services implementations, and build IT environments that are flexible and responsive. In particular, SOAs offer the following core principles:

- **The business drives the services, and the services drive the technology** – In essence, services act as a layer of abstraction between the business and the technology that hides the implementation details from the business user, and with them, much of the complexity. Both IT and line-of-business personnel must understand the dynamic relationships between the needs of the business and the available services on the one hand, as well as the technical underpinnings that offer the layer of abstraction required by the services on the other.

SOAs provide the ability to respond to changing requirements, which is the new "meta-requirement" for business.

- **Flexibility and responsiveness are the fundamental business requirements** – Instead of dealing with rigidly-defined, concrete requirements from business, loosely coupled services in an SOA provide the ability for applications to respond to changing requirements, because of the layer of abstraction the SOAs provide between the services and the underlying technology.
- **An IT environment organized as an SOA is always susceptible to change** – To visualize how an SOA is supposed to work, it's better to think of a living, growing organism rather than the traditional "building a house" metaphor that gave software architecture its name. The everyday normal state of affairs is an IT environment that is undergoing constant change, and as a result, the work of the IT department is never done. House-building assumes a state of completion and the ability to craft a design that remains the same over time, which is rarely the case in any business environment. As a growing entity, SOAs provide business value incrementally as they are phased in.. To continue this metaphor, businesses participate in an "ecosystem" containing suppliers, partners, and customers. In fact, a company need not transition all of its application functionality to a SOA to get the benefits of such an architecture. Building an SOA is a progressive approach that can mix service-oriented and non-service-oriented elements.

Building an SOA in the enterprise, however, is a difficult task, because there are many aspects of the underlying IT infrastructure that must be in place for the SOA to operate as advertised. For example, building and maintaining an SOA may require several of the following:

- Deep architectural expertise.
- Web services development and testing tools.
- Integrated development environments.

- Integration brokers or message buses.
- Systems management and Web services management capabilities.
- An enterprise security infrastructure based on corporate security policies.
- Business process automation and management tools.

Companies should take a step-by-step approach to achieving service orientation, first by laying out a broad architectural plan based upon clear business requirements, followed by creating and exposing individual services that provide standards-based access to individual systems and other existing assets, and then by composing those services into services that provide value to the business. Only then can companies access and combine the available business services into flexible business processes that provide the flexibility and responsiveness that businesses require. Such business processes are themselves services that companies can orchestrate into more complex processes.

IV. What Are the Benefits of an SOA?

SOAs are about thrift.

The SOA story is about *thrift*—both cost savings as well as getting more value out of existing systems. It's no wonder that Web services and SOAs have moved from hype to reality during a severe IT downturn—after all, in 2002, most companies using Web services were applying these new technologies to reduce integration costs. Hundreds of enterprises have already learned that taking a Web services-based approach to integration reduces the cost of a project dramatically. This is one story that executives love to hear.

SOA Benefit: Leverage Existing IT Assets

Thrift, however, means more than simple cost savings. True thriftiness means making do with what you have—squeezing value out of every asset. One of the clear benefits of an SOA is that such an architecture helps companies get more value out of existing resources, by componentizing legacy applications, wrapping those components in Web services interfaces, and then making those services available and discoverable on the network.

Take, for example, one of the largest investment firms in the world. This company had a large existing IBM mainframe infrastructure. They considered themselves an "early adopter" of new technology, while at the same time, their systems were mission critical, and had to keep running 24/7/365. They wanted to leverage legacy investments across the enterprise, in particular, over 23,000 live CICS (Customer Information Control System) applications that were running in a mainframe environment.

They realized that 90% of the cost of their IT was infrastructure and integration--essentially, the overhead—and the remaining 10% of the cost of IT was business logic, where the business value lay. Previous attempts to provide better access to legacy data were too slow and brittle. Therefore, they established an XML-focused enterprise initiative that included the following elements:

- A SOA for exposing all 23,000 CICS transactions as Web services.
- A program analyzer for determining the specific functionality of their applications.
- A directory for registering and finding the services.

- Monitoring, logging, and testing tools.

The benefits of their SOA initiative are as follows:

- **Improved response times** for CICS transactions by 10 to 20 times, by building a sophisticated caching mechanism that brought information from the mainframes into a centralized location.
- **Increased throughput** by a factor of 10 times, by implementing a SOAP interface in COBOL directly on the mainframe. The mainframe therefore hosted the Web services directly.
- **Reduced the percentage of cost for infrastructure** and integration from 90% to 65%, increasing the percentage devoted to business logic to 35%, by migrating middleware functionality from Intel-based servers to Linux virtual environments on the mainframe. This migration also reduced software licensing costs.
- **Reduced the time** to write programs to get data off of mainframe from 3-4 months to a matter of minutes, by exposing CICS transactions as Web Services that were discoverable via a UDDI registry.

There are a few important lessons to be learned from this case study. First, they didn't feel they needed to take tentative steps toward building an SOA; instead, they tackled a large, mission critical problem. Such an approach is usually not recommended; it often makes more sense to tackle smaller, pilot projects first. It's also important to emphasize that the benefits they achieved weren't measured in small increases, but rather by dramatic, business-changing jumps. Such substantial improvements are only possible when taking a new architectural approach to accessing IT resources.

SOA Benefit: Embracing Heterogeneity

Instead of "rip and replace," moving to an SOA means building bridges among different systems and applications, rather than throwing them out.

A second thrift benefit that SOAs provide is that they facilitate the operation of heterogeneous IT environments. Instead of "rip and replace," moving to an SOA means building bridges among different systems and applications, rather than throwing them out. In fact, the business value of SOAs is so dramatic that enterprises across the world are considering how to transition their existing IT infrastructures to SOAs.

Heterogeneous environments are nowhere more prevalent than in a multiple-company, business-to-business (B2B) situation. Take, for example, **E2open**, an electronics industry multi-company process management firm that provides a software infrastructure service for business partner integration based on an SOA. This integration framework sought to address many problems facing the electronics industry, including:

- The **lack of visibility** into supply chain issues and business processes.
- **Reduced control** over business processes.
- **Management complexity.**
- **Increased interaction costs.**
- **Business process overhead** due to information fragmentation.

The company identified several root causes for these problems, including the following:

- **Increasing number of companies** – a greater number of suppliers and customers, not to mention increased competition, complicated doing business.
- **Larger number of applications** – a greater number of applications including enterprise resource planning (ERP), supply chain management (SCM), and product data management (PDM) required increased integration across multiple enterprises.
- **Many data formats** – Data are stored and moved in many different formats, some electronic, while others are still paper-based, leading to communication problems.
- **Control and security issues** – communication among businesses typically took place via email and fax, two insecure protocols that are hard to manage and monitor.

In order to address these issues, E2open created a standards-based integration platform or framework based on an SOA. Their integration framework included the following characteristics:

- Used the standard transport protocols underlying the Web and eMail, but included a secure "tunneling" approach that provided secure communication over the Internet, providing a secure approach to implementing an SOA in a B2B environment.
- Used a Web services registry to enable loose coupling among participating companies.
- Created external process interfaces at both suppliers and manufacturers, and developed a framework for synchronizing these processes that supports multiple message and document formats.
- Used industry standard vocabularies like RosettaNet to describe data formats as well as standard processes, but allow for multiple vocabularies as required by their customers.

By providing a registry for services as well as standard business processes, E2open enabled participating electronics industry companies to migrate from static, point-to-point connections to a more cost-effective, many-to-many dynamic network. The quantifiable benefits from this solution include:

- The ability to **deliver business partner integration 50% faster** than alternative methods.
- A **reduction in overall costs for E2open customers by as much as 70%** by enabling multiple company participation in spite of heterogeneous application environments.

In addition, participants are able to take advantage of the "network effect," obtaining substantially increasing value as the number of participants grows.

SOA Benefit: Preparing for the Future

Once the enterprise has composed its individual Web services and other software functionality into coarse-grained business services and exposed them as an SOA, how does the business make use of those services? Line of business users are interested in creating, executing, modifying, and managing business processes, which are essentially sequences of actions or events that can include interactions that are modeled within the SOA. Because the business services on

the network are built following the principles of SOAs, they are inherently flexible. As a result, the business processes that incorporate those services tend to be flexible as well.

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. In reality, this traditional, "waterfall" approach typically does not solve many of the challenges IT faces when responding to business requirements, 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. SOAs reverse this predicament, providing sufficient flexibility to allow business to drive the technology—and once business truly drives the technology, IT is finally able to enable flexible business processes, enabling an enterprise to evolve to an "on demand" or "real-time" business.

SOAs provide sufficient flexibility to allow business to drive the technology.

V. How to Migrate to an SOA: Incremental ROI

Moving an enterprise's application functionality to an SOA is very different from the painful, risky "big bang" approaches of earlier generations of new IT architectures. Remember that SOAs promote thrift, eschewing "rip and replace" for squeezing more value out of existing technology. As a result, there is no one right strategy for adopting an SOA—no specific timeline or checklist that applies to the average enterprise. Instead, the best approach to adopting an SOA is a pragmatic one:

1. Create a **broad architectural roadmap** that provides a high-level direction for the enterprise architecture of the company. This roadmap should serve as a guide, without specifying implementation details.
2. It then makes sense for many companies to **start with a pilot project**. Such a project will both develop relevant technical skills as well as build acceptance for SOAs across the enterprise.
3. Continue to identify **high "bang for the buck" projects** in the enterprise that are suitable for a service-oriented solution, and solve it the service-oriented way. "Bang for the buck" means selecting projects with an eye toward achieving substantial ROI—in other words, look for projects with small investments that can yield relatively large returns.
4. Repeat step three as necessary, tackling individual projects that each **yield a positive ROI**. Taking this step-by-step approach builds acceptance for SOAs in the enterprise, enables the IT shop to acquire the necessary skills, and lowers the overall risk of the SOA initiative. Continue to improve and refine the architectural roadmap as the SOA falls into place.

Fundamentally, this white paper doesn't simply omit a preferred timeline for adopting an SOA, it explicitly recommends against using one at all. Instead, companies must first take a flexible approach to thinking about the problem of SOA—that is, to accept the fact that the business environment is in a state of constant flux, and plan accordingly. In fact, there is also no deadline for completing the transition; it may be appropriate for many companies to remain only partly service-oriented indefinitely.

As a company moves to an SOA, issues of complexity, inflexibility, and brittleness will gradually be resolved—and each step provides measurable ROI.

There is an important economic advantage to taking this pragmatic, flexible approach to SOA migration—the migration takes place in manageable steps, and each step will have measurable ROI. SOAs, after all, are appropriate for solving many of today's knottiest IT problems. As a company moves to an SOA, issues of complexity, inflexibility, and brittleness will gradually be resolved and business becomes increasingly flexible.

Copyright, Trademark Notice, and Statement of Opinion

About ZapThink, LLC

Founded in October 2000, ZapThink, LLC (http://www.zapthink.com) is an industry research and analysis firm that provides quality, high-value, focused research, analysis, and insight on emerging technologies that will have a high impact on the way business will be run in the future. ZapThink focuses on XML and Web services technologies that provide open, standards-based, loosely-coupled systems and represent an evolutionary advancement in computing that requires a new way of thinking about computing resources, capabilities, development methodology, and architecture.

ZapThink produces and sells XML-focused research and analysis reports. The company has a comprehensive, thorough, and up-to-date understanding of XML and Web services technologies, the markets in which they exist, the vendors that operate in various market segments, the growth of the various segments, and applicable ROI and revenue models. ZapThink's research has been often cited and quoted by numerous industry analysts, media, venture capital firms, vendors, and end users. The company is considered to be a reliable, credible, and comprehensive source of research and analysis. The company is more uniquely qualified and better positioned to provide this advice than any other company in the world due to its combination of technical focus and ability to apply detailed and rigorous methodology to research and data sources, while maintaining independence and objectivity.

ZapThink also provides strategic eBusiness consulting services to a select clientele, helping a broad range of companies formulate their business strategies by leveraging XML and Web services. ZapThink advises hundreds of clients on the usage, adoption, and challenges of applying XML to their business needs. The company focuses on providing its clients advice on strategic planning and business growth as it applies to the use of XML.

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