

zapthink white paper

SOA: BUILDING THE ROADMAP



SOA: BUILDING THE ROADMAP

June 2007

Analyst: Tony Baer

Abstract

By now, most Information Technology (IT) organizations have become aware of the potential of Service-Oriented Architecture (SOA) to pierce through those silos. Just as Rome wasn't built in a day, implementing SOA should be an incremental, iterative process that should start modestly. Your first foray into a SOA implementation should be through a pilot project, where your organization has the opportunity to conduct an evaluation to determine whether to make further investments. The goal is gain experience while mitigating the risks. Consequently, the scope of the pilot should be limited. Choose a handful of Services that will make a difference, and that people will notice.

Governance is essential. Lacking governance, SOA projects become yet another example of undisciplined software development. As your organization becomes more experienced with SOA, it eventually learns to compose business Services bridging those silos, and gradually becomes more efficient to the point where SOA supports business processes to the point where you can continuously optimize your business.

From a starting point of point-to-point integration, organizations evolve to developing more flexible dynamic couplings that exploit far more effectively the Services that they have exposed. At that point, governance becomes essential if SOA is to evolve beyond isolated, discrete connections to support an environment where Service contracts drive development, Services become composable, and the agility that SOA promises becomes reality.

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.

I. Introduction

While business agility has become a top competitive goal for many organizations, information locked away inside monolithic application silos has proven a stubborn obstacle. By now, most IT organizations have become aware of the potential of Service-Oriented Architecture (SOA) to pierce through those silos. They are already cognizant that today's SOA implementations rely upon standards that promote use and reuse of information assets and business processes that already exist across the organization, and that of business partners. However, while many IT organizations have started on the road to SOA, many of them are still struggling to answer the question on how to build the roadmap that is right for their enterprise.

Selecting the proper path starts with identifying where the key business pain points reside. It can involve asking questions such as:

- Is information locked away on legacy systems that are difficult to access?
- Does your organization require a common view of data across multiple sources that may exist across different relational and legacy databases?
- Does it require better visibility of key performance indicators?
- Does your organization need to optimize business processes? Could your organization benefit from the ability to reuse valuable process assets?

II. Setting the Stage

SOA is all about exposing information and processes as self-contained Services that can communicate and interoperate with each other in a standard way, enabling the business to build flexible compositions of Services that implement business processes. Ultimately, SOA enables IT to more effectively align with the business, because it changes the way IT delivers the solutions. The self-contained nature of Services and the standard connectivity empowers IT to rapidly compose solutions by reusing existing Services, resulting in faster-time-to-benefit compared to traditional ways of developing, modifying, and integrating conventional, monolithic software applications.

Furthermore, because SOA changes the very notion of how data and functionality are deployed and architected, the business, rather than the platform, drives the functionality and processes that Services enable. In place of the tight coupling, SOA abstracts data and functionality that, in conventional IT silos, specific applications or databases own. Instead, SOA loosely couples data and functionality as Services in a manner that abstracts them from the physical

Ultimately, SOA enables IT to more effectively align with the business, because it changes the way IT delivers the solutions.

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 and enter the code **SAGTIME**. 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.



platform, application, or database. That enables IT to direct its attention to solving the business problem, eliminating much of the concern over how to integrate data or functionality.

This loose coupling also has several key architectural implications that can ultimately aid business agility and accelerate time to benefit. For instance, with loose coupling, the underlying data, software application, and platform can change without changing the Service. Secondly, the designer of a Service should not have to know anything about the consumer of that Service. Instead, policies govern consumption at run time. In turn, you can change those policies as competitive requirements dictate. Finally, you can design Services and specify policies so that you can dynamically orchestrate them into a composite that bridges individual applications and Services. Each of these characteristics of loose coupling can play key roles in improving enterprise agility because of their ability to support change.

Implementing SOA should be an incremental, iterative process that should start modestly. As with any significant architectural project, IT should begin with an initial architectural plan and governance framework, in addition to internal tests and prototypes in order to learn the building blocks and assess readiness for SOA. Specifically, that means determining:

- Business readiness: Does the organization have business requirements that SOA could help address? If so, how receptive is the enterprise or specific business units or departments to change?
- IT Infrastructure readiness: Which assets could the organization expose as Services to support the business requirements?
- IT organizational readiness: Does the IT organization have the proper skillsets to specify and implement SOA, or has it identified qualified consulting partners who are ready to assist?

Governance Matters

SOA is a journey that begins with the task of piercing the application silos that isolate data and processes, thereby limiting organizational agility. As your organization becomes more experienced with SOA, it eventually learns to compose business Services bridging those silos, and gradually becomes more efficient to the point where the organization can harness Services to optimize their business.

Lacking governance, SOA implementations become yet another example of undisciplined software development. Consequences could include barriers to reuse, where lack of architectural standards governing Service creation reduce the likelihood of reuse. Furthermore, when organizations don't apply security policies governing authentication, authorization, and access uniformly, breaches and compliance violations grow more likely. Consequently, governance is necessary for ensuring that when the organization exposes a Service in a new scenario that might serve a different class of user, that the new usage won't violate corporate policies for confidentiality or privacy.

Not surprisingly, as Services begin to proliferate, some of the following questions that inevitably arise include:

- How to identify whether Services exist within an organization and whether there are consumers of a particular Service?

Lacking governance, SOA implementations become yet another example of undisciplined software development.

- How to determine which consumers are entitled to which versions of a Service, and what is the process for authorizing a new version of a Service?
- How to document new Services, and where to publish them?
- How to choose the right SOA tools for the job?
- How can technology architecture and business policies be enforced regarding all stages of the Service *lifecycle*, from creating, testing, classifying, granting and managing access, modifying, and retiring a Service?

The Starting Point

Your first foray into your SOA implementation should be through a pilot project, where your organization has the opportunity to conduct an evaluation to determine whether to make further investments. There are many reasons why a company would want to undertake an SOA pilot project. Among these reasons are the following:

- To build acceptance for SOA within the organization
- As a means to work through some of the harder aspects of architecture and Service modeling
- To bring the technical team up to speed on the intricacies of SOA
- To develop and fine-tune methodologies and approaches for implementing SOA
- To evaluate various commercial products that purport to help with SOA rollouts.

Pilot projects are a great way to reduce the risks of moving to a new architecture. However, today's perennially constrained IT budgets don't allow for high-risk projects, so it's vitally important to build acceptance for SOA with the powers-that-be, particularly because architectures don't have features—only implementations do. As a result, the best SOA projects solve some real business problem, and do so cost-effectively. After all, the best way to build acceptance for your new approach is to solve a problem on a shoestring budget. Furthermore, since implementing SOA in an iterative fashion is actually a best practice, the SOA pilot serves as a low-risk first step in the ongoing, iterative process of implementing SOA.

Consequently, the goal of the pilot is to gain experience while mitigating the risks. However, while delivering a meaningful business result is important, the primary goal of an SOA pilot isn't delivering the functionality. It's all about proving the feasibility of the architecture. Consequently, you should limit the scope of the pilot, choosing a handful of Services that will make a difference, and that people will notice. They could involve integrating several related systems, integrating a key business process, or modernizing specific legacy applications.

The result is that you should focus most of the effort on SOA architecture, as opposed to designing as many Web Services as possible. That involves:

- *Identify the key people that could eventually form the center of excellence for SOA.* They could be the principals behind the first project, who could subsequently become part of the steering committee for other SOA projects. Identifying the necessary skills and the right

The primary goal of an SOA pilot isn't delivering the functionality. It's all about proving the feasibility of the architecture.

resources for SOA projects is a key issue that organizations should address early on.

- *Specifying an architectural plan.* Enterprise SOA is enterprise architecture that provides a comprehensive, high-level plan for how the business interacts with IT. Specifically, SOA should help a company build IT that can respond to unpredictable change. An SOA pilot, therefore, must take some subset of this enterprise-wide plan and put it through its paces.
- *Process Scope.* Implementing SOA includes identifying the right Services to build. The best way to accomplish that identification is to decompose specific business processes in order to identify areas of redundancy that might lend themselves to reusable Services.
- *Mitigating risk.* Don't try boiling the ocean at pilot stage. Choose a process (or handful of processes) where both the risk and scope are clearly limited. Don't select one that could seriously disrupt the business should the pilot fail.
- *Resource Scope.* Your approach to SOA should be both top-down (through process decomposition) as well as bottom-up (exposing existing functionality as fine-grained Services and composing them into coarse-grained business Services). For your pilot, don't simply build Services based on the systems you already have—think about the Services you need in order to make your pilot a success
- *An iterative methodology.* Admittedly, figuring out which resources are in scope before you decompose your processes may sound like a Catch-22. The solution is taking an iterative approach that starts with a broad architectural plan, and then drills down to process composition and Service definition.
- *Clear acceptance criteria.* Hopefully, the pilot will be a starting point for future SOA initiatives. To get there, you must define criteria ahead of time that delineates a clear finish line, after which results can be analyzed.

Finally, the architecture team should generally avoid promoting the initial project as a new architecture, or as SOA itself. Business users are interested in goals such as improving productivity or agility; they are not interested in being guinea pigs. Therefore, describe the project as an innovative approach for achieving the business goal. Don't draw attention to SOA until the success of the project has been proven.

III. Preparing for the Mainstream

Once the pilot has been proven successful, it is time to start planning an architecture for extending SOA to other portions of the enterprise to ensure that each project does not reinvent the wheel. The first step is dissecting the results of the pilot to identify lessons that the team has learned covering areas such as whether you have properly defined Services, was performance an issue, whether there is potential for reuse, and what benefits or issues will apply as SOA takes hold across the enterprise.

With lessons in hand, it pays to adopt a pragmatic approach that picks your early battles wisely: don't try force-fitting SOA onto every project in IT's backlog.

Don't try force-fitting SOA onto every project in IT's backlog.

Enabling certain business users to manage and evolve business processes without direct IT involvement is one of the most ambitious of SOA goals.

ZapThink has identified several scenarios that illustrate how IT organizations can promote SOA adoption:

- *Pragmatic Governance.* When planning a SOA governance initiative, IT should work with the business to prioritize SOA governance policies based upon a risk/benefit analysis. For instance, early iterations of SOA governance initiatives might focus on security policies for Services, followed by reuse and quality of Service policies, and eventually, Service consumption policies – depending on enterprise needs.
- *Pragmatic Reuse.* While many people tout reuse as the primary benefit of SOA, achieving reuse can be difficult. Effective reuse also requires governance, and an understanding that achievement of reuse builds with experience. Pragmatic reuse requires IT organizations to manage expectations realistically.
- *Pragmatic Legacy Enablement.* It is important that the SOA architect identify those business processes that require the special flexibility, agility, and user empowerment benefits that SOA provides. Chances are, only a small portion of all the business processes the organization executes every day can justify the extra expense and overhead that SOA requires.
- *Pragmatic User Empowerment.* Enabling certain business users to manage and evolve business processes without direct IT involvement is one of the most ambitious of SOA goals. Such a vision requires bulletproof governance. In the near term, a more pragmatic approach to user empowerment leverages end-user tools that are available today, including browsers, spreadsheets, and mobile devices, combined with the power of SOA-enabled Services to offer a richer, more functional interface for users. In turn, the more challenging composition tasks are left to IT.

As the organization mounts the learning curve, it becomes more adept at identifying opportunities where SOA can deliver clear, competitive benefits, and it becomes more ambitious in the way that it governs and utilizes SOA. In the long run, a pragmatic start makes long term goals more achievable.

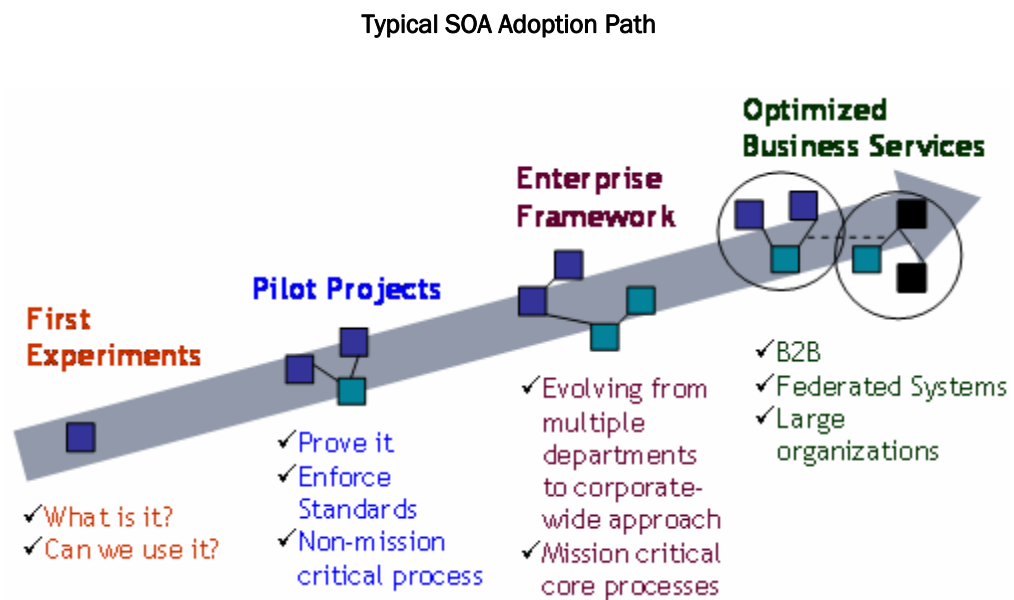
Developing the SOA Roadmap

There are many ways that your enterprise could benefit from SOA, and consequently, step one is to outline your organization's goals. They could include one or more objectives, ranging from increasing reuse to improving information integration, improving process flow across internal units, integrating with business partners, simply lowering the costs of software development, and others.

Beginning with point-to-point integration, organizations evolve to developing more flexible dynamic couplings that exploit far more effectively the Services that they have exposed. At that point, governance becomes essential if SOA is to evolve beyond isolated, discrete connections to support an environment where Service contracts drive development, Services become composable.

Ultimately, SOA success leads to an optimized state where critical processes inside the business and with business partners are Service-enabled. The organization enters a virtuous cycle where the business can combine, modify, or recombine processes readily without the usual time lag associated with developing or procuring new software, or installing new platforms. In order to get

there, you will need an SOA roadmap. A typical SOA adoption path appears in the figure below.



Source: Software AG

An SOA roadmap is critical. SOA is not only about generating Services because they offer a quick and easy way for improving application interoperability. Lacking a plan, your organization will wind up with a more modern version of the same tangled ball of point-to-point connections. Lacking architecture, Web Services will prove just as unmanageable, costly to maintain, and impossible to reuse as the spaghetti code they were designed to replace. Lacking the right people, SOA will proliferate through “worst practices” and degenerate to chaos. It’s important to note that although many people associate SOA with Web Services, the two are not synonymous. While SOA is software architecture, Web Services are only a set of standards for implementing SOA.

Key building blocks of SOA methodology include:

- *Taking an iterative approach.* As mentioned earlier, SOA involves a learning process where IT and the business constantly learn to identify the most promising targets for SOA, and evolve their knowledge on how to plan, implement, and govern it.
- *Defining the rules of engagement between IT and the business.* This best practice covers the request, specification, and management of Services; and ensuring that Services are managed in accordance with IT and corporate governance policies.
- *Creating a governance framework.* Again, as noted previously, governance becomes critical in order to attain effective use and reuse of Services. And, because SOA promotes utilization of enterprise data and processes, it can act as an essential extension of applicable IT and corporate governance initiatives that are already in force.
- *Identifying appropriate Services.* This task involves deciding the *granularity* of Services: how detailed a Service should, or should not be. For instance, if a Service is too granular, it may not be useful to the business unless it is combined with other Services. By contrast, if a

Lacking architecture, Web Services will prove just as unmanageable, costly to maintain, and impossible to reuse as the spaghetti code they were designed to replace.

Although one of the goals is SOA is fostering reuse, it is important to set priorities.

Service is not granular enough, it may become difficult, if not impossible to reuse it.

- *Developing Service reuse policies and best practices.* Although one of the goals is SOA is fostering reuse, it is important to set priorities. While reuse can reduce costs and time-to-benefit, it can also result in Services that are designed according to the lowest common denominator, thereby limiting the value that they can deliver. Consequently, the organization should set criteria or formal policies regarding when reuse is required at design or deployment stage. Yet, it should keep that policy flexible enough so it can be refined as the organization grows more Service-proficient.
- *Determining when Services are appropriate.* The criteria may vary. For instance, because reuse is one of the major benefits of SOA, designing a highly specialized Service that might be used in only one scenario might not be cost effective. Conversely, if that unique Service generates a huge cost saving or provides a significant competitive differentiator, developing that one-off Service would be well justified.
- *Creating awareness that Services have a lifecycle.* You should design Services to evolve over time. Define the lifecycle, along with the processes you use for managing Services, from requirements definition through development, deployment, versioning, modification, reuse, and retirement. In many ways, the Service lifecycle has parallels with the software application lifecycle.
- *Setting criteria for Service metrics (QoS).* In consultation with the business, you should define what levels of Service are acceptable.
- *Creating architectural blueprints.* These patterns help development teams design consistently which could address areas such as security, Service orchestration, metadata management, process integration, and compliance.

Like any IT project, risk assessment is critical. It is especially important for SOA because of the degree to which IT assets may be exposed in use case scenarios not foreseen by the designers or architects of source systems. At the top of the list is security, since as Services expose data in new contexts, risks may arise with tampering or unauthorized use. Start planning on how to address user authentication, message integrity, content accuracy & reliability, acceptable/unacceptable uses, and plan for auditing Service creation, communication, and consumption.

Building the SOA Team

Before starting the journey, make sure that you have the right people. Besides developers, you will need a strong team of SOA architects who can define and enforce architectural standards for Services. Those standards could range from granularity of Service definitions to technical specifications regarding use of Web Services standards and compliance with enterprise architecture. Additionally, your team should include business analysts who can interpret the needs of the business and translate them to requirements that IT can use for designing SOA infrastructure and the Services themselves. This team can form the core of an SOA Center of Excellence that fosters development and spreading of best practices. Ultimately, the right combination of SOA architects, business analysts, and developers can ensure that the SOA migration is aligned with business needs, and that IT transforms from cost center to strategic differentiator.

With the team in place, it is time to start designing the roadmap. It starts with a general vision of the desired Service architecture and the goals. For instance, are you implementing SOA to reduce IT costs, improve organizational efficiency, accelerate responsiveness, or adaptability? Obviously, setting the goals is not a solo act; IT must engage the business.

A methodology, which describes the rules of engagement and the process of design, is essential for supporting the SOA roadmap, and to set the ground rules, cultivate best practices, and elaborate the process for setting architectural standards. The methodology should evolve with business priorities and changes in underlying technology, such as the evolution of Web Services standards. It will also evolve over time as your organization grows older and wiser in deploying Services.

IV. Choosing the SOA Path

Because SOA represents a new approach to managing, deploying, and consuming the information and processes that are already present inside your organization, and facilitating the creation of new processes that liberally reuse existing assets, there are many ways to benefit from SOA. Consequently, while simply exposing an asset as a Web Service may be acceptable for initial prototyping and testing, the right SOA path is one that addresses a key business goal and/or resolves a critical information bottleneck. Here are a few examples.

The right SOA path is one that addresses a key business goal and/or resolves a critical information bottleneck.

Modernizing and integrating legacy assets

With an estimated 70% of the world's data residing in legacy systems, they are valuable, often underutilized resources that can be extended and integrated without interrupting your core business. With SOA, you can create the infrastructure that automatically wraps mainframe processes into Services without coding. It could automate bi-directional triggering of events and actions spanning mainframe and external systems. For instance, if your company seeks to grow sales channels, you could expose a Service that validates new business partner credentials and automates order fulfillment.

Combining data across multiple sources into a single view

This task involves creating views of information without concern over where data resides. In turn, you can orchestrate integration of data from multiple sources to provide 360-degree views that are driven by business semantics. And, by exposing data as Services, consolidated data views can be reused for addressing different business problems. Through master data management, you can ensure that all of your distributed sources of data remain in sync, supporting a single version of the truth.

For instance, with SOA, you can build a consolidated view that tracks the ordering history of all customers in a specific geography, expose it as a Service, and then repurposed when conducting regional sales rollups.

Rationalizing and integrating business processes

This task elevates Services to a higher level, by managing and aggregating them as composable business processes. SOA enables new processes to be composed through dynamically orchestrating existing processes and integrating human workflows. For instance, a financial institution could use automate payment processing for commercial customers by exposing processes that

automatically convert incoming customer file formats, then routing them to appropriate target systems, and tracking process execution.

Putting the Building Blocks in Place

Using SOA as the architecture for abstracting information and processes away from application and database silos, there are many tools and solutions available that can help your organization derive more value from those Services. They can include:

- *Application Composition.* Configuring applications on the fly from Services to support a rapidly changing business need, application composition tools provide the means to rapidly assemble composite applications minus the overhead involved with conventional software development.
- *Business Process Management (BPM).* Traditionally trapped inside application silos, BPM solutions enable developers and business analysts assemble and manage business processes over their full lifecycle.
- *Information Integration.* Combine data from multiple databases into a single virtualized view, providing a single target against which to submit queries.
- *Legacy Integration.* Make your mainframe a first-class citizen by creating new Services from legacy platforms, eliminating reliance on legacy programming languages and mastering hard-to-use proprietary APIs.
- *Service Orchestration.* Multiply the possibilities of SOA by eliminating point to point connections by composing and orchestrating Services

V. The ZapThink Take

SOA is not an all-or-nothing project. Your organization can adopt an incremental approach to implementing SOA as it targets specific benefits, applies the lessons learned to create and communicate best practices as it manages the Service life cycle and adds new Services, expanding their footprint throughout the organization.

Adopting a maturity model provides a yardstick for setting goals and evaluating SOA progress. For instance, SOA can start with Service-enabling the mainframe, where the core of your business processes may reside. As your organization gains experience by publishing formerly inaccessible mainframe functionality as Services, it can start taking a higher level view that abstracts processes from Services, providing yet a new level of business agility. As the portfolio of Services and processes grows, your organization can steadily phase in governance to manage all stages of the Service lifecycle.

Software AG has developed a pragmatic approach that began with a core strength in integration technology, and complemented with deep experience in helping customers modernize legacy assets. It recognizes that, while SOA governance is an evolutionary process. Having just completed the acquisition of webMethods, Software AG has a broad array of enterprise integration, business process management, Service composition, legacy modernization, and SOA governance solutions that provide customers multiple paths for embarking on their Service evolution. Most importantly, Software AG has made governance a forethought in its product line, and it is providing a roadmap that can help customers implement SOA and governance incrementally.

Software AG has made governance a forethought in its product line, and it is providing a roadmap that can help customers implement SOA and governance incrementally.

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 Road
Baltimore, MD 21210
Phone: +1 (781) 207 0203
Fax: +1 (815) 301 3171
info@zapthink.com

