

# zapthink foundation report

## RICH INTERNET APPLICATIONS

### *MARKET TRENDS AND APPROACHES*



# RICH INTERNET APPLICATIONS

## MARKET TRENDS AND APPROACHES

July 2006

Analyst: Ronald Schmelzer

### Abstract

Two of the often conflicting desires in IT are the need for rich user interfaces that maximize a user's productivity on the one hand and the desire to decentralize computing so that a user can gain access to the widest base of IT assets at the lowest possible cost on the other. These two forces are at odds because rich client interfaces, until recently, have only been possible in certain limited scenarios in which the business logic and computing resources were combined with the interface.

However, a new class of presentation layer is emerging in the marketplace. This *Rich Internet Application* provides an end user experience that is similar to client/server applications, with a rich graphical user interface, responsive performance and highly interactive functionality. As companies desire richer interaction between their Web Services-based applications and the users of those applications, Rich Internet Applications will continue to gain prominence in the enterprise. Users will increasingly demand the ability to present very large data sets to a dispersed audience without sacrificing the economics that Web applications or the rich user experience that traditional client/server applications provide.

### Key Points:

#### ◆ Market Overview

- Combining real-time user interaction with rich user interface capabilities, Rich Internet Applications (RIA) leverage increasingly sophisticated client-side technology to enable users to interact with and compose functionality from distributed applications no matter where they are located.
- The market for RIA solutions consists of three submarkets focused on delivering RIA components, environments, or extensions to IDE suites.
- There are four primary means for providing RIA capabilities: Flash VM-based approaches; approaches that use JavaScript and HTML, also known as Asynchronous JavaScript and XML (AJAX), approaches that use Java applets or ActiveX controls; and Custom-developed Client Code developed with Java or .NET languages.

#### ◆ Future Trends

- A set of six key business applications are motivating overall RIA spending consisting of enhancement of existing web applications, high-transaction and event-driven Internet applications, next-generation portals, enhanced business intelligence solutions, application modernization, and peer-to-peer or mashup solutions.
- ZapThink expects spending on each of these areas to increase rapidly over the next five years, exceeding \$500 million by 2011.

All Contents Copyright © 2006 ZapThink, LLC. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.



## Table of Contents

I.	The Increasing Impact of Rich Internet Applications.....	4
1.1.	Scope of Report.....	4
1.2.	From the Web to the RIA.....	5
II.	The Evolving Rich Internet Application Markets.....	6
2.1.	Rich Internet Application Submarkets.....	7
2.2.	Technology Approaches to RIA.....	9
2.3.	Factors Impacting Continued Differentiation of Submarkets.....	10
III.	RIA Market Trends.....	11
3.1.	Classes of Business Problems that RIA Addresses.....	11
3.2.	RIA Market Sizing.....	15
3.3.	Technology Adoption Patterns.....	17
3.4.	Factors Impacting Growth of RIA Markets.....	19
3.5.	Methodology.....	19
IV.	Conclusions.....	20
4.1.	Figures.....	20
4.2.	Tables.....	20

## I. The Increasing Impact of Rich Internet Applications

Users today increasingly demand more from their online user experiences. Gone are the days of static Web pages and poor interaction. Today's Internet users expect their online experiences to behave more like the desktop applications they are used to and enhance their day-to-day productivity. In place of stateless Web interactions that offer only a cheap imitation of the interactivity we've come to expect from our desktop operating systems, a new class of distributed applications is emerging – the **Rich Internet Application (RIA)**. Combining real-time user interaction with rich user interface capabilities, Rich Internet Applications leverage increasingly sophisticated client-side technology to enable users to interact with and compose functionality from distributed applications no matter where they are located on the network.

Generally, RIAs are capable of providing advanced capabilities for user interaction, including desktop-like GUI features and data integration. RIAs compose, and thus integrate, local and remote sources of data and business logic. The most powerful of these RIAs will take advantage of standards-based, Service-oriented approaches to integrate all the content, communications, and application interfaces it can physically access. However, what is most important is that RIAs loosely couple presentation from application logic, allowing developers to change server-side logic without breaking the usability of their applications. As a result, RIAs make ideal Service consumers in a Service-Oriented Architecture (SOA).

### 1.1. Scope of Report

While significant attention and hype have recently been placed on emerging RIA technologies such as **Asynchronous JavaScript and XML (AJAX)**, there are still many questions about precisely what business problems are driving RIA adoption. Furthermore, is the RIA class of applications merely a flash in the pan soon to be subsumed by a more potent solution to business problems, or is there sustainability and repeatability in RIA solutions that provide long-lasting and compelling value to businesses? As such, this report aims to tackle the following questions to help establish the current state of the RIA market, quantify business trends, and postulate the future of the RIA markets:

- What are the emerging classes of business problems targeted by RIA solutions?
- What are businesses currently spending on RIA solutions, and how will this change over time?
- What are the primary technological approaches being implemented by businesses, and how will these trends play out over the next few years?
- Who are the primary buyers of RIA solutions and how does this impact RIA deal sizes?
- What is the prospective market sizing for RIA solutions and how will this change over time?
- How are RIA solutions categorized by different segment and market approaches?

This report aims to identify emerging market trends and address the above questions, but does not aim to specifically analyze individual RIA solutions or product offerings, nor rank vendors according to how they meet specific business

requirements. Indeed, this report does not even attempt to provide comprehensive coverage of all the vendors in the market, but rather highlight a few as examples of different markets and technology positioning. Even though the RIA market is still emerging, this report uses actual customer anecdotal evidence and primary research with RIA purchasers as well as vendors who supply those businesses. Given the early state of the RIA market, significant variability exists with market projections as well as extrapolations of current results to the market as a whole.

## 1.2. From the Web to the RIA

Despite the advancement in architecture that the Web represents, the traditional Web browser interface and technologies, namely HTTP and HTML, are severely limited in comparison with the relatively thicker interfaces of the client/server technologies. Users quickly realize that while exhibiting excellent economics, the Web in its current form is not an adequate substitute for their existing tightly-coupled client/server technologies, because of user interaction limitations, poor performance, and poor suitability for event-based or real-time interaction. In particular, traditional web-based applications exhibit the following drawbacks:

- *User interaction limitations:* HTML was originally intended for publishing of static, text-heavy documents with limited user interaction. As such, the format lacks significant capabilities that even the most basic of client/server applications possess. HTML lacks the ability to provide windows for data, drag-and-drop capabilities, sortable and resizable data grids, spreadsheet capabilities, dynamic data visualization capabilities, and resizable viewing areas. The lack of these features hinders a user's productivity.
- *Poor performance:* The Web was designed with a stateless mode of interaction in mind. As a result, users interact with Web sites using a "click-wait-reload" pattern that requires the browser to reload a new document for every user interaction. This approach requires frequent roundtrips between browser and server in order to accomplish any multi-step process, resulting in reduced user productivity, increased server strain, and unnecessary network overhead.
- *Poor suitability for event-based or real-time interaction:* As described above, the Web is based on a stateless request/response model in which the user is responsible for determining when information is to be retrieved from servers. While there exist a number of hacks and kludges around this model that basically forces the browser to reload information on a preset timeframe, the Web communication model in essence does not support two-way, real-time communications. Event-driven, asynchronous, real-time based interaction with clients are simply not possible using today's Web browser technology.

Businesses today want to gain the operational and cost advantages of deploying applications over the Internet, but don't want the limitations that Web browsers impose on user interfaces. Unfortunately, there has been no practical way to provide rich client capabilities without reintroducing the scalability and manageability limitations of thick clients – that is, until now.

The RIA offers more than either the thick or thin client approaches of the past ever have. As companies desire richer interaction between their Web Services-based applications and the users of those applications, rich client solutions will increasingly gain prominence in the enterprise. Users will increasingly demand the ability to present very large data sets to a dispersed audience without

sacrificing the economics that either Web applications or the rich user experience that traditional client/server applications provide.

The set of capabilities for rich clients include the ability to:

- Provide advanced capabilities for user interaction, including windowing features and data navigation controls like buttons, check boxes, radio buttons, toggles, windows, palettes, etc., as well as powerful, rich media component objects like animated sprites, multi-track sound, and movies.
- Integrate local and remote sources of data and business logic, taking advantage of standards-based, Service-oriented approaches to integrate all the content, communications, and application interfaces it can physically access.
- Loosely couple presentation from application logic, giving developers the freedom not to have to make any changes to rich client functionality to make sure that the server and client can communicate. The rich client, like the standards-based Web thin client, can be loosely coupled and thus enable independent innovation of the business logic and the user interface to that business logic.
- Provide greater intelligence and efficiency in distributed computing. Rich clients aren't tethered by specific, tightly-coupled communications protocols as are most client/server thick clients and Web-based thin clients. Rather, rich clients can communicate with a wide range of distributed computing systems using both synchronous and asynchronous communication modes as needed. As a result, rich clients can surpass the inefficient request/response paradigm of thin clients and also not be confined by proprietary protocols, so as to make use of the most cost-effective and productive means to communicate. Finally, rich clients won't need to dynamically generate sets of information simply to transmit large data sets, and can afford to maintain client-side data storage and message queuing.
- Enable online and offline modes of usage, allowing users to interact with applications while they are offline on occasionally connected devices such as mobile phones, laptops, and personal digital assistants (PDAs).
- Deploy onto multiple platforms and channels. By abstracting rich clients' underlying runtime platform, users should be able to specify their presentation layer preferences without having to decide on a particular runtime environment. Rich clients also should support a wide range of user interaction channels including telephone, email, and voice interaction.

As can be seen from the above set of requirements, the new breed of Rich Internet Application solution requires more than either the thick or thin client approaches of the past ever have.

## II. The Evolving Rich Internet Application Markets

Macromedia (now part of Adobe Systems) coined the term "Rich Internet Application" when they first started rolling out products and solutions based on their popular Flash platform. However, soon after, additional vendors and solution providers with products that performed similar capabilities or aimed to address overlapping markets co-opted and extended this term to encompass additional approaches and technologies aimed at meeting the rich user

interactivity and client-side integration requirements promised by RIA approaches.

As such, with the emergence of technologies such as AJAX and virtual machine-based approaches to RIA, what might have once been a single market focused on providing RIA capabilities has in fact further developed into three separately identifiable submarkets, each with their own set of vendor participants, end-user purchasing behavior, and market growth trajectories. These markets, their growth factors, and vendor constituencies are described in detail in this section below.

## 2.1. Rich Internet Application Submarkets

In order to understand the various RIA submarkets, it is first important to understand how demand and supply for RIA solutions has changed since 2004. Prior to 2004, there were a handful of vendors selling “complete” solutions for Rich Internet Applications that included the development environment for building RIA solutions, a runtime environment for deploying and running those solutions, and templates or widgets that accelerating the development and adoption of RIA solutions by pre-bundling user interaction and integration components with the development and runtime environments. In February 2005, Jesse James Garrett of Adaptive Path coined the term “AJAX” to refer to a wide range of technologies and approaches leveraging Asynchronous JavaScript and XML as well as supporting technologies such as Dynamic HTML (DHTML) and browser-specific HTML capabilities. In this regard, AJAX is not an invented technology, but rather simply a catchphrase for a variety of technologies that had already been in existence at the time.

The resultant dramatic increase of attention on AJAX precipitated the emergence of dozens of new vendors such as **JackBe**, **ICESoft**, **General Interface** (now part of **TIBCO**), and open source technology platforms including **RICO** and **DOJO**. In addition to these AJAX-focused vendors, other solutions not based on AJAX technology from vendors such as **Macromedia** with their Flex product, **Nexaweb**, **Altio** (now part of **Integra SP**), **DreamFactory**, and **Curl** (acquired by **Sumisho Computer Systems** in 2004) caused the market to split into different submarkets each focusing on different aspects of delivering, running, and/or deploying RIA solutions. These submarkets each have specific characteristics and are defined as follows:

### 2.1.1. *RIA Components Submarket*

A large and growing submarket of RIA solutions are vendors who sell components or widgets for enhancing existing web applications with rich Internet capabilities. Many of these widgets or components perform individual tasks such as grid controls, drag-and-drop data fields, interactive menuing and navigation, and asynchronous data exchange. Vendors provide many of these components in a piece-meal fashion or as part of a toolkit that require developers to integrate them with web-based applications.

The largest number of new entrants to the broader RIA market is in this submarket. Many of these RIA vendors sell their products for less than a few hundred dollars each, and there also exist many open-source efforts that offer free software. In many ways, the RIA Components market is primarily a group of developer-oriented vendors selling their wares directly to developers. As such, there is little emphasis on providing complete solutions for RIA development or for delivering complete, turn-key offerings that would be tailored to meet the specific needs of their customers. RIA Component vendors aim to solve largely tactical issues of meeting specific user interface or integration requirements

rather than being seen as a strategic solution for large-scale application development. Thus, the RIA component market is primarily a license-fee based model with a very small professional services component.

Examples of RIA Component vendors include **Adobe's Spry Framework for AJAX, EBusiness Applications, Zapatec, ZK**, and others. More vendors can be found at <http://www.ajaxmatters.com/r/resources?id=17>.

### *2.1.2. "Native" RIA Development and Runtime Environments*

The oldest and most established of RIA submarkets are solutions that provide "native" RIA development and runtime environments. This submarket is the most established with vendors such as **Curl** first providing solutions as far back as 1998. Rather than focusing on simply providing widgets and components that developers can add to their pre-existing web applications, RIA development and runtime environment vendors focus on building complete applications that leverage the capabilities of RIA. These solutions may be deployed in browser-based environments or as standalone applications deployed on desktop applications.

Most of the RIA environment vendors focus on delivering solutions that help their customers build complete solutions out of their products without having to do any coding at all, and many sell customized solutions as a result to the end-user line of business or project managers. As a result, many RIA environment implementations are focused on project-based or enterprise-wide solutions that are priced from the tens or hundreds of thousands of dollars to multi-million dollar projects. While companies like **Adobe** are not focused on delivering custom solutions to clients, vendors like **Nexaweb, ICESoft, and JackBe** are indeed delivering custom-built RIA solutions.

Some of the RIA vendors utilize browser-based technologies such as AJAX whereas others utilize either client-side plug-ins like **Adobe Flash** or virtual machine approaches such as Java or **Microsoft ActiveX**. Examples of RIA Environment vendors include **Adobe Flex, Altio (Integra SP), Backbase, Curl, ICESoft, Ideo Technologies, JackBe, Nexaweb**, and others.

### *2.1.3. RIA Extensions to Integrated Development Environments*

The final submarket of RIA solution is actually the enhancement of the existing Integrated Development Environment (IDE) market with RIA capabilities. Vendors such as **Borland, IBM, Microsoft, and Sun Microsystems** are looking to enhance the capabilities of their existing software development toolsets with RIA capabilities for outputting either AJAX or other technology-based RIA solutions in the place of traditional client/server or n-tier applications. Since these vendors are in essence members of an incumbent market, there already exist millions of developers and users for the existing products in the market, although a very small percentage of them are using such products for RIA development today. However, over time, these vendors might make RIA development through the use of their products so simple and low-cost that they might present the greatest competitive threat to RIA component or development environment vendors.

The RIA Extensions submarket is clearly focused on helping existing developers utilize the products they have already purchased and adopted to build RIA solutions. As such, it is not clear how much additional incremental dollars will be spent by users on such solutions. Vendors plying these wares are also not focused on delivering tailored solutions to their end-users, and as such, the price points will lie in the hundreds to thousands of dollars. Examples of RIA Extensions include add-ons to Microsoft Visual Studio that supports **Microsoft's Atlas Framework** capabilities. While Microsoft Atlas Framework itself will be a

free download (as of the date of this report), tools that support such capabilities will cost extra for developers. Similarly, **Sun Microsystems** is adding Ajax libraries to its development platform, as will IBM to its portal and Rational tool sets.

## 2.2. Technology Approaches to RIA

As implied in the above section discussing RIA submarkets, there are a variety of technological approaches being implemented for realizing the capabilities of rich user interaction and integration. The only commonality between different solutions is not server-side technology (which is considerably diverse and heterogeneous from solution to solution), but rather specific approaches for delivering such functionality to user clients. The following table outlines the capabilities and limitations of each of the approaches for deploying RIAs on the client.

**Table II-1: Capabilities and Limitations of Rich Client Approaches**

Rich Client Approach	Capabilities	Limitations
Flash-based Virtual Machine  Examples: <b>Adobe Flex, OpenLaszlo</b>	<ul style="list-style-type: none"> <li>➤ Can support wide range of user interaction requirements.</li> <li>➤ Consistent, reliable, predictable user interaction</li> <li>➤ Very high penetration on desktop systems.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Limited by browser sandbox and security issues.</li> <li>➤ If users do not have Flash, must have the ability to install.</li> <li>➤ Flash runtime not available for all operating systems or devices</li> </ul>
Java or ActiveX VM-based  Examples: <b>Altio, DreamFactory, Nexaweb</b>	<ul style="list-style-type: none"> <li>➤ Can support wide range of user interaction capabilities.</li> <li>➤ Java VMs available for a wide range of operating systems and devices</li> <li>➤ Supports existing development skills.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Limited by browser sandbox and security issues.</li> <li>➤ Many users do not have Java VM or have it disabled</li> <li>➤ VM implementation not consistent across browsers, desktops.</li> </ul>
Asynchronous JavaScript/DHTML (AJAX)  Examples: <b>Adobe Spry, BackBase, General Interface (TIBCO), ICESoft, JackBe, Microsoft Atlas, Nexaweb</b>	<ul style="list-style-type: none"> <li>➤ Minimal additional technology requirements at both server and client-side.</li> <li>➤ Can leverage existing knowledge in DHTML, HTML, XML, XSL, and JavaScript.</li> <li>➤ In theory, can support widest range of devices and systems.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Limited by browser sandbox, security, and cookie settings.</li> <li>➤ Hard to guarantee consistent, reliable interaction since dependent on browser versioning, capabilities.</li> <li>➤ Some implementations limited to single browser and version range.</li> </ul>
Custom developed.NET / Java  Examples: <b>Microsoft .NET or Windows Presentation Foundation</b>	<ul style="list-style-type: none"> <li>➤ Fully custom rich clients possible.</li> <li>➤ Not limited by browser sandbox and security issues.</li> <li>➤ Support for Web Services and SOA continuing to expand.</li> <li>➤ Strong developer base and tools support</li> </ul>	<ul style="list-style-type: none"> <li>➤ Must deal with issues of maintaining, developing, and deploying own code base.</li> <li>➤ Might impose requirements on clients.</li> </ul>

**2.3. Factors Impacting Continued Differentiation of Submarkets**

Buying behavior for Rich Internet Applications has yet to become firmly entrenched. As such, it is likely that new RIA submarkets might appear focusing on different aspects of delivering rich internet capabilities to end-users. One of the factors impacting market segmentation is the increasing strength of a number of open source efforts. In particular, the Dojo Toolkit, Prototype, and Rico efforts are mobilizing large communities of developers that in turn may end up focusing the technologies on different market segments.

This trend may end up creating new submarkets focused on nuances of the RIA market such as mobile applications, offline usage, and embedding capabilities within traditional non-RIA applications. Currently, these efforts have focused their efforts on enhancing either the RIA Component or RIA Environment submarkets, but companies with a vested interest in RIA would be well served to observe the open source efforts.

### III. RIA Market Trends

Despite the relative immaturity of the RIA market, end-users are spending millions of dollars in aggregate across all the various solutions offered by RIA vendors. As such, a number of distinct trends and early data points can be gathered from this buying behavior that might suggest how the overall market will develop over the long haul. This section of the report will focus on what those emerging buying patterns are and how they might reflect on future growth of the RIA market as a whole and individual submarkets.

#### 3.1. Classes of Business Problems that RIA Addresses

One of the earliest challenges facing the RIA market was that to many end-users and outside observers, offerings by RIA vendors seemed to be solutions in search of a problem. That is to say, it was not clear which specific problems RIA solutions were meant to address. In many cases, RIA efforts were simply different technologies for performing many of the same capabilities that other approaches also solved, such as Web-based or client/server based approaches.

However, as the RIA offerings began to distinguish their capabilities by providing not just better Web interactivity, but also the capability to deploy client-side logic that leverages the scale and efficiency of the Internet, new classes of applications could be created from RIA solutions that simply would not be possible with purely Web or client/server based approaches alone. The following lists a few of those RIA-specific solutions now being addressed in the market.

- *High Touch Web Interaction* – One of the earliest applications of RIA technology was to enable a greater degree of interaction with consumers for data, visualization, and/or transaction-intensive web applications. Existing HTML and HTTP-based web applications are not able to easily power interaction-intensive applications such as product configurators, trading and asynchronous, event-driven transactions, and applications requiring intensive data manipulation and visualization. In the early years, Java applets and Flash-powered websites dominated this RIA application type, but increasingly AJAX-based RIA solutions are finding their way to delivering high-touch web interaction in both business-to-consumer (B2C) as well as business-to-business (B2B) realms. Examples of RIA applications in this problem domain include the Mini Cooper and Hummer product configurators, Home Depot/Behr Paint selector, and Ameritrade's trading cockpit. ZapThink has seen projects in this category of RIA spending range from \$30,000 for a single web project to over \$500,000 for a B2C application overhaul, and some report multi-million dollar projects if all services revenue is counted.
- *Next-Generation Portals* – Web-based portals have a number of significant limitations that have prevent them from providing much of the value that businesses demand. In particular, portals provide Web-only access to aggregated data that the IT department has configured on behalf of users beforehand. RIA solutions promise to break the mold

of today's inflexible and online-only portal offerings by allowing end-users to control their own composition of business logic, integrate portal information with local sources of data, and even take portal information off-line. Applying RIA solutions to the concept of the portal will either make today's portals obsolete or bring them to the next generation. In either case, it is only through the application of RIA approaches that portal solutions will achieve this next step in value. A number of examples exist in this RIA problem domain across a wide range of vertical industries implemented for a wide range of customers for their internal or B2B portals. While there are not as many "Next-Gen Portal RIA" projects as B2C Web applications, ZapThink has seen already a few examples, including some notable United States federal and state government projects in the six-figure pricing range.

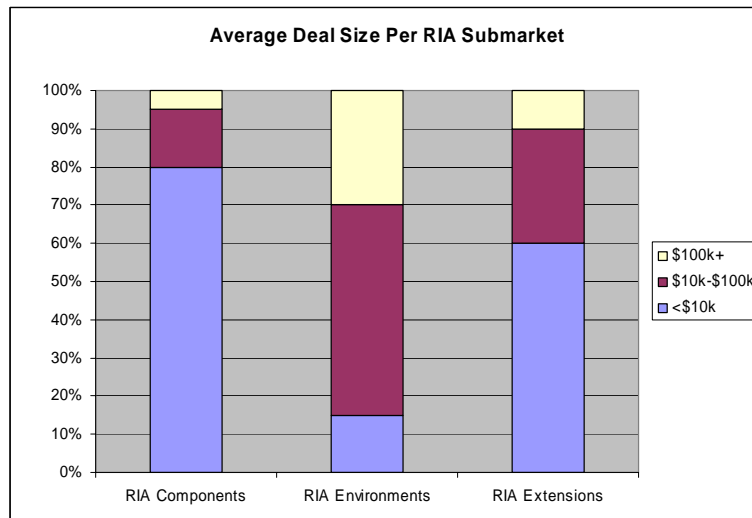
- *Advanced Business Intelligence / Analytics* – In much the same way that today's portals are ill-suited to the rich user interaction requirements that end-users desire, many business intelligence and analytics applications provide good data aggregation but poor user interaction with consolidated data. A number of RIA solutions vendors have built new classes of dashboard and situational awareness type applications that provide users a way of getting continuously updated, event-driven, asynchronous, and highly interactive access to complicated and changing data. So far, many of these dashboard or situational-awareness applications have been for intelligence-driven needs, especially in the federal government and military. As such, project sizes range from the hundreds of thousands to millions of dollars for these applications and can take months to years to fully develop.
- *Application Modernization* – Many vendors are focusing their RIA solutions not just on applications already Internet or Web-enabled, but also on those applications that are still primarily client/server, mainframe, or 4GL-based. For many companies, their existing investment in legacy systems represents a conundrum: they are too valuable to replace and too expensive or brittle to maintain. In this vein, RIA solutions have come to the rescue by allowing companies to maintain the business logic and user interface styles that they have come to depend on while simultaneously leveraging the economic benefit of the Internet and new capabilities that RIA solutions offer. These modernized applications with RIA capabilities can then be deployed internally or over the Internet as required. A number of RIA vendors, such as **Nexaweb** and **JackBe**, have focused specifically on modernizing mainframe, PowerBuilder, Visual Basic, and other legacy application types. The average cost for doing this sort of RIA-based modernization is in the \$50,000 to \$250,000 range, but can go much higher for highly complex and widely distributed applications.
- *Peer-to-Peer Web Applications and Mashups* – The final category of RIA solutions increasingly adopted are a new class of applications that seek to compose Service functionality from multiple systems to create new, dynamic applications that can communicate with disparate Services on the network or with other clients similarly enabled. In the Service-oriented context, the sort of enterprise mashups that businesses require combine rich interface capabilities with Service-Oriented Business Applications (SOBAs) that enable not only the use, but also the creation and configuration of the SOBAs themselves. Furthermore, the Services that make up the SOBA composition might abstract capabilities on either the client or the server. One of the reasons why enterprise mashups are so compelling is because enabling business users to

assemble Services, and thus create new applications, in flexible, innovative ways, within the context of a governance framework, has broad, untapped business value. The notion of peer-to-peer Web applications or enterprise mashups is still significantly immature and ZapThink hasn't seen that many actual end-user implementations leveraging RIA solutions. However, the promise of distributing business logic to the client and allowing those clients to dynamically compose functionality from multiple, disparate Services represents a business opportunity so great that this application category might represent the bulk of RIA spending the not-to-distant future.

**3.1.1. *Customer Spending Behavior by Application Type***

Based on data collected from a variety of sources including conversations with actual RIA end-users and RIA solution providers, ZapThink observes the following trends with regards to end-user spending behaviors based on different RIA submarkets and RIA application types. The chart below illustrates that the RIA Component market consists mostly of RIA solutions priced below \$10,000 with considerably fewer in the upper pricing ranges. However, this situation is reversed in the RIA Environments submarket in which the majority of deals are in the five- to six-figure price range. The RIA Extensions submarket lies somewhere between these two extremes with roughly half of all solutions priced at \$10,000 or below and the remainder above.

**Figure III-1: Average Price Point Prevalence for RIA Submarkets**



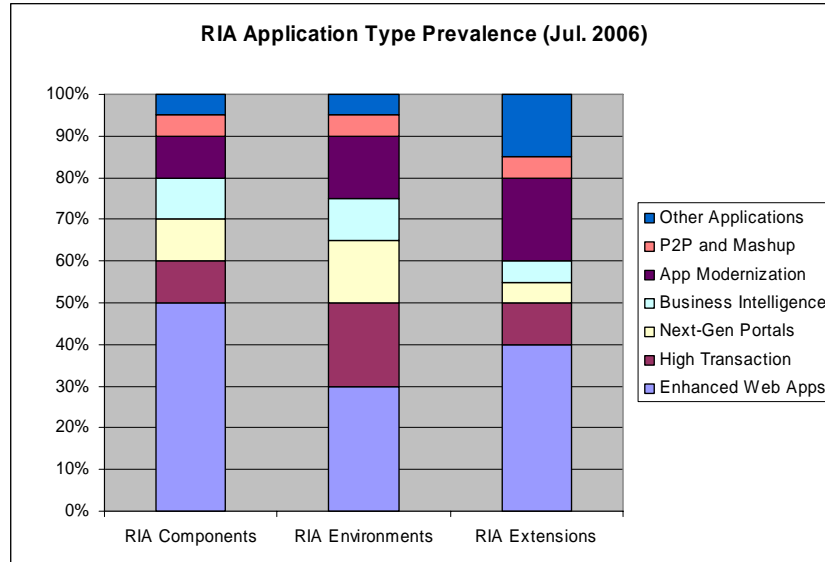
Source: ZapThink

Figure III-1 above implies that most RIA spending is of a tactical, project-by-project nature rather than enterprise-wide. This behavior is not expected to significantly change over the next few years, unless new RIA application types of an enterprise-wide nature become established as common.

Figure III-2 below illustrates that each RIA submarket exhibits different prevalence of RIA application types. In particular, RIA Components lend themselves most to enhancing existing web applications while RIA environments have a higher degree of prevalence for high transaction volume applications, application modernization, and portal applications. Specifically, half of all implementations that utilize RIA Components will be for Web enhancement,

while only 3 out of every 10 RIA Environment implementations will be for that reason. Regardless of the differences, enhancement of web applications currently seems to be the most prevalent of all RIA application types.

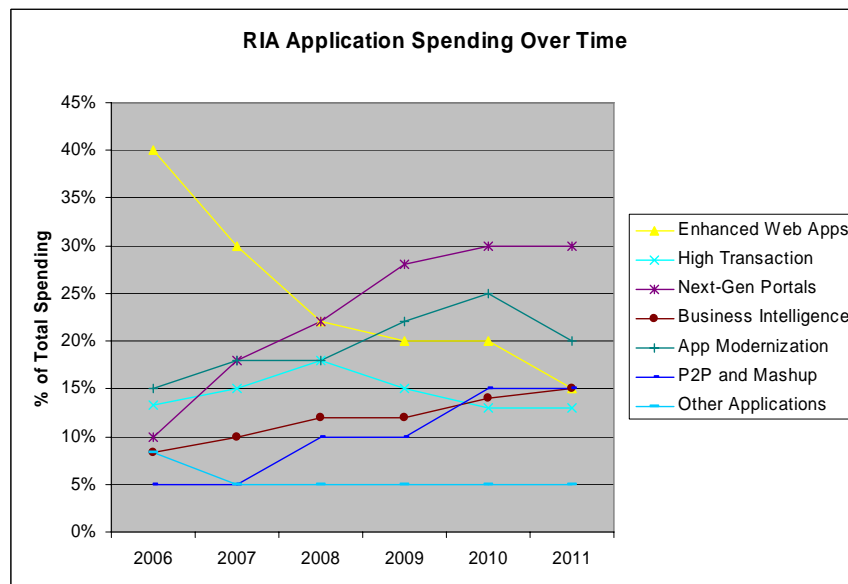
**Figure III-2: RIA Application Type Prevalence**



Source: ZapThink

However, ZapThink expects these RIA application percentages to shift dramatically over time as Figure III-3 below suggests:

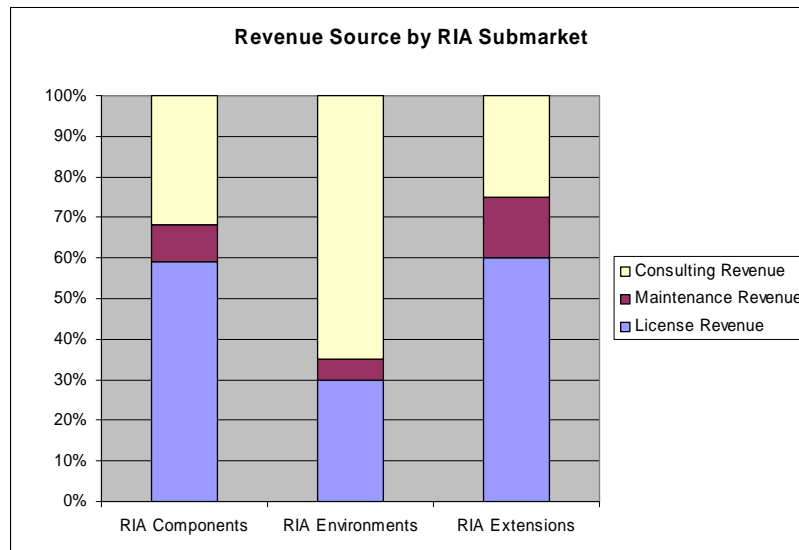
**Figure III-3: RIA Application Spending Over Time**



Source: ZapThink

Correspondingly, each RIA submarket exhibits different characteristics from a license, maintenance, and support revenue perspective. Most RIA Component and RIA Extension revenue is derived from license sales (with the notable exception of open source efforts, in which case all license revenues are allocated to consulting revenue), whereas the bulk of RIA Environment dollars are spent on application-specific consulting. The distribution of each of those revenue sources by RIA submarket is illustrated in Figure III-4 below:

**Figure III-4: Revenue Source by RIA Submarket**



Source: ZapThink

The revenue source split in the chart above suggests that even in the case of RIA Component expenditure, significant dollars are being spent by end-users on application customization. While companies are increasingly looking to third-party firms for tooling to assist their RIA development, a very large portion of firms today are building their own RIA technologies. This conclusion is especially the case when companies are leveraging open source RIA approaches. Part of the reason for the heavy build vs. buy approach is that much of the RIA application emphasis is on Web application enhancement by developers. Only when the bulk of RIA spending is by project managers and upper levels of IT or business decision makers on RIA application types that are not developer centric will the balance shift decidedly to purchased RIA solutions rather than self-built RIA applications.

### 3.2. RIA Market Sizing

In order to get an aggregate sizing for RIA Market expenditure over time, it's important to first understand how the various different submarkets will grow and how different vendors will bring their products to market. In particular, how will vendors enter the market with new solutions where they had not produced any before (what ZapThink calls "new entrants") versus how vendors who have existing products in similar or related markets will apply their products to solve the problems addressed by the RIA market (so-called "incumbents").

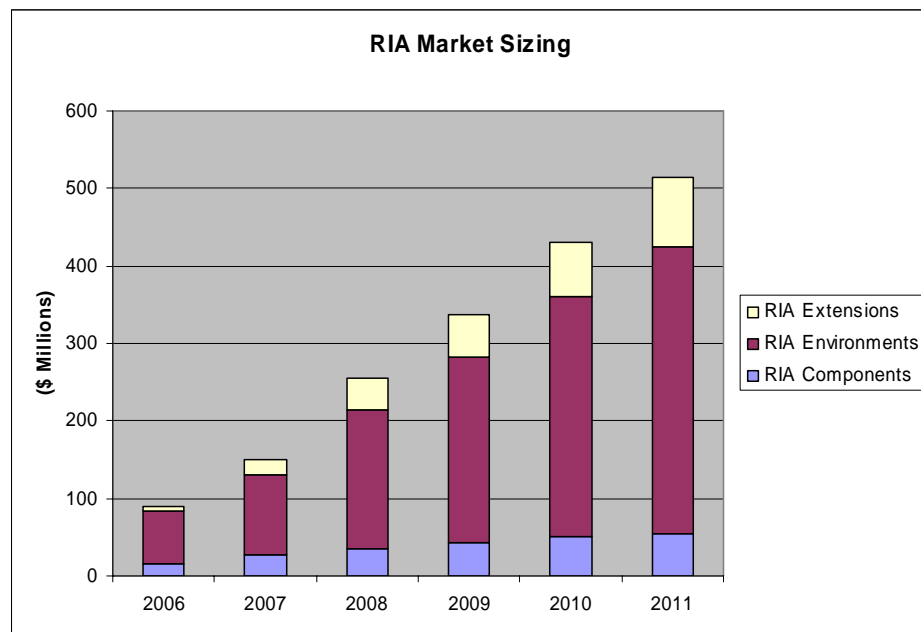
The current set of new entrants include vendors such as **Backbase, ICEsoft, JackBe, Nexaweb, TIBCO (General Interface product line)** and others who have created new businesses or business lines to address the RIA market. These

vendors will provide solutions that address one or more RIA submarkets. Market incumbents include those vendors who provide web application development, client/server applications, or 4GL suites and seek to increasingly either transition those products to deliver RIA capabilities or build new applications that will leverage their existing customer base to grow their RIA capabilities. Examples of market incumbents include **Adobe, Borland, Microsoft, and Oracle.**

Over time, as the RIA markets grow in aggregate value, many new entrants will become consolidated by market incumbents and likewise, a greater percentage of incumbent revenue will be derived from RIA markets rather than their traditional non-RIA areas such as web development, client/server applications, or 4GL suites.

ZapThink predicts that the growth of each RIA submarket in total aggregate spending will follow the general trends as detailed in the chart below:

**Figure III-5: RIA Market Sizing**



Source: ZapThink

One can observe that RIA Environment revenue will make up the majority of RIA spending for the foreseeable future. Part of the reason for this is that most of the RIA Environment vendors sell not just licensable components, but rather complete solutions that are often specifically tailored to their customer needs. As such, per-project revenue will remain high. The above graph does not reflect the total volume of customer projects, however. In that case, RIA Component vendors will find that they are addressing a considerably larger number of total customers, even though the revenue per customer will be much smaller for RIA Component sales versus RIA Environment solutions.

Trying to split the above spending patterns on incumbent versus new entrant vendors proved to be too difficult to predict for this report. One can reasonably assume that the majority of spending in 2006 is on RIA new entrant vendors, since these form the bulk of the companies in the market to date. However, over time, RIA incumbent vendors will increasingly consume the majority of the RIA market as they migrate their existing customer base and acquire RIA specialist

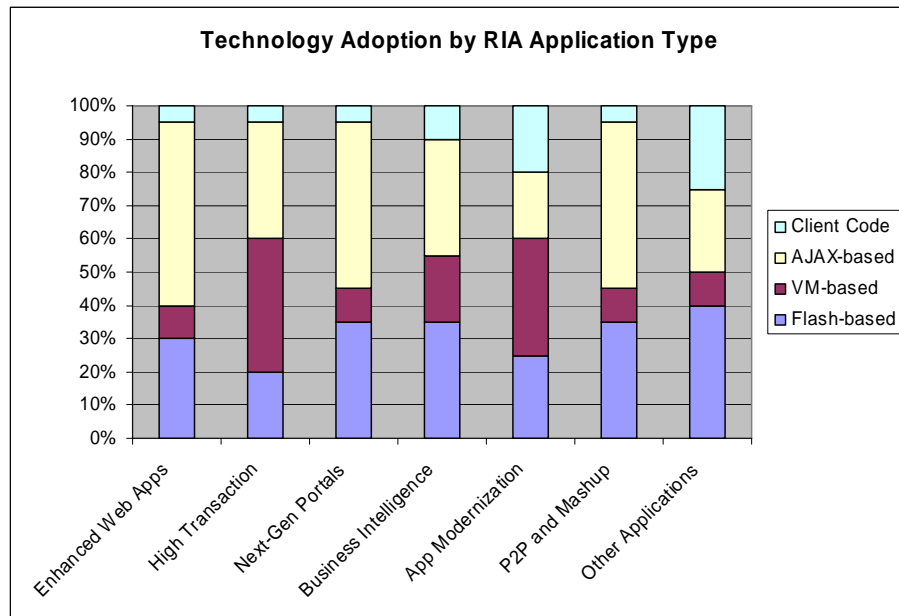
vendors. However, the percentage of new entrant vs. incumbent spending will differ by RIA submarket over time.

Since the RIA markets are considerably immature and many factors will impact the growth and overall success of various RIA submarkets, the above numbers should be viewed as a conservative estimate and general trends towards overall market growth.

### 3.3. Technology Adoption Patterns

In addition to differences in spending by RIA application type and RIA submarket, considerable differences in adoption exist for the various different technological approaches to RIA implementation. As detailed in Section 2.2 of this report, ZapThink sees four primary categories of RIA technology adoption: Flash-based, Java or ActiveX Virtual Machine-based approaches, Browser-native technologies (collectively known as AJAX), and Custom Code approaches (notably Windows Presentation Foundation and XAML-style). The relative adoption of each of these technology approaches by RIA application type *as of the date of this report* can be seen in Figure III-6 below:

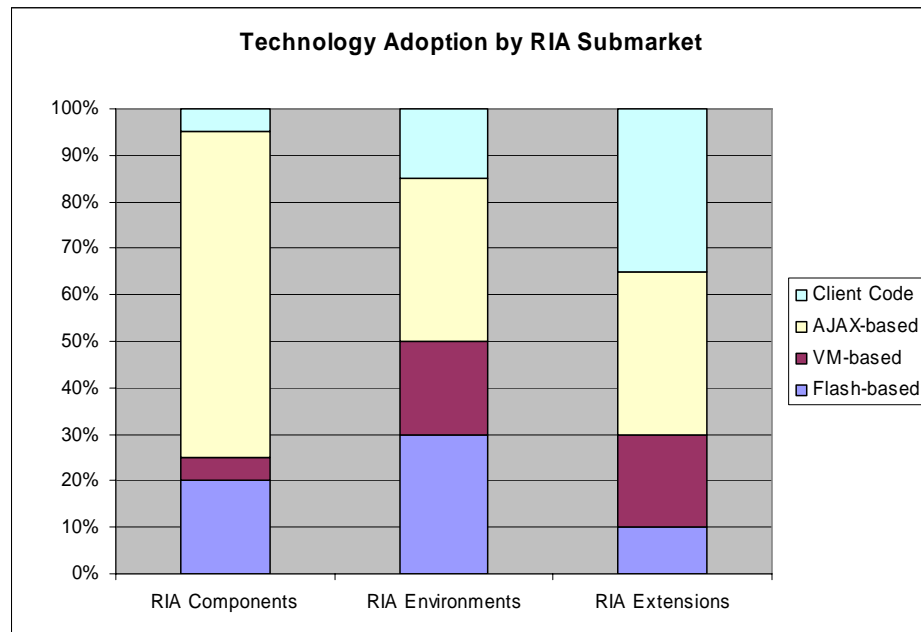
**Figure III-6: Technology Adoption by RIA Application Type (Jul 2006)**



Source: ZapThink

Likewise, it's possible to analyze the technology adoption patterns by RIA submarket, as shown in Figure III-7 below:

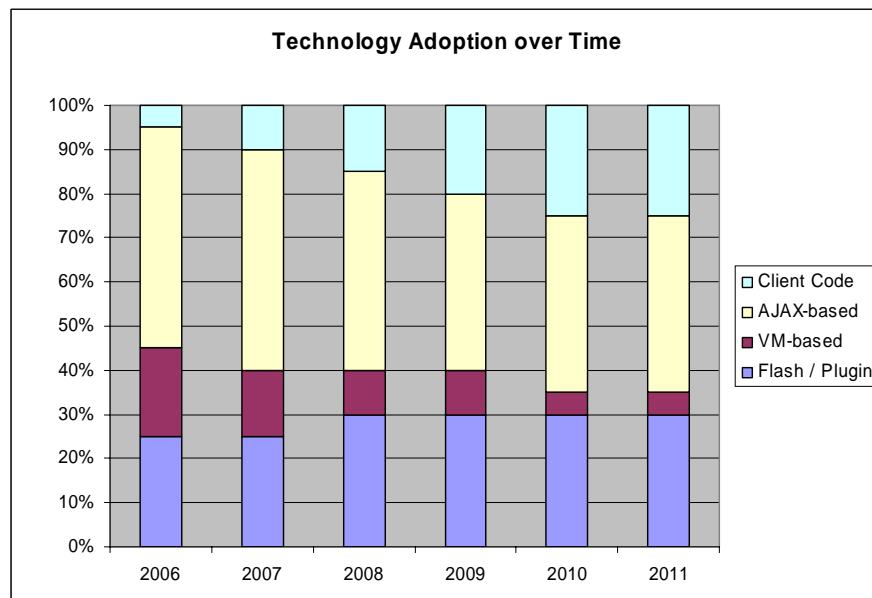
Figure III-7: Technology Adoption by RIA Submarket (as of July 2006)



Source: ZapThink

Over time, however, ZapThink sees aggregate adoption of the various RIA technology approaches shifting as implementation behavior settles on specific approaches and certain technologies, notably ActiveX and Java on the client, focus increasingly on niche applications whereas AJAX and Flash-based styles will increasingly dominate larger RIA markets. These trends can be observed in Figure III-1 below:

Figure III-8: Technology Adoption Over Time



Source: ZapThink

As noted above, please note that these trends are based on extrapolating current buyer purchasing behavior and might be subject to significant change if new application types emerge that shift spending behavior to different technology types.

### 3.4. Factors Impacting Growth of RIA Markets

ZapThink sees a number of factors that might contribute to a slowing of the adoption of RIA expenditure. In particular, the following issues may surface to cause companies to think twice about implementing rich or smart clients from new entrants and incumbents alike:

- *Lack of Business Buy-in For RIA Expenditure* – The single greatest threat to RIA expenditure is that much of RIA spending to date is by developers or project managers for individual, point projects, especially for the RIA Component submarket. In order for the RIA markets to grow as predicted in Section 3.3 above, companies must be able to justify the RIA expenditure in the context of significant business problems they aim to solve, such as those addressed by the RIA application types discussed earlier. Yet, buying behavior is still particularly immature and many businesses have not yet justified spending on those RIA application types.
- *Lack of interoperability among RIA implementations* – There is considerable flux and lack of standards when it comes to formats for specifying user interface and client-side business logic, especially in the AJAX technology arena where little commonality exists between RIA component vendors. Even if transformations exist to convert between similar formats, there is a high likelihood that any interface designed on a particular RIA technology will not be compatible with applications that use other approaches. As such, many RIA implementations may remain proprietary, slowing the growth of the market, if developers realize that they might be pigeonholed into development on a specific platform with no opportunity for reuse.
- *Potential emergence of serious security concerns* – If new entrants and incumbents expose serious security flaws in their approaches and do nothing to solve those problems, then end users might balk from implementing rich or smart clients at all in their enterprises, thus stunting market opportunity and growth – perhaps fatally.

### 3.5. Methodology

ZapThink bases its new entrant market numbers on an analysis of existing revenue figures provided by the current new entrants in each market segment, adjusting for possible errors and uncertainty, including under- and over-reporting of revenues and best-guess estimates for companies not briefed for this report. Those numbers are then placed into the context of the overall changes affecting the market for Web Services Presentation Layers as covered in ZapThink's existing research.

ZapThink then calculates a range of market sizing numbers for incumbent markets based on existing third-party research for each of these incumbent markets. We then multiply the low, high, and average incumbent market numbers by the percentage of the incumbent market predicted to be SO by ZapThink's analysis of the SO market trends. Therefore, it is also possible to input other third-party numbers in the place of the numbers in this report to

obtain adjusted SO incumbent percentages based on third-party analyst research.

Based on these calculations, ZapThink is then able to estimate the total market size for the markets described above, which is the sum of the new entrant contribution and the incumbent SO portion to each core SO market. From those values we can then calculate the portion of each SO market that is due to the new entrants in that market.

#### IV. Conclusions

Rich Internet Applications will quickly come to supplant the current base of static Web applications, and web-based portals in particular. The resulting new breed of rich client offers more than either the thick or thin client approaches of the past ever have. As companies desire richer interaction between their Web Services-based applications and the users of those applications, rich client solutions will increasingly gain prominence in the enterprise. Users will increasingly demand the ability to present very large data sets to a dispersed audience without sacrificing the economics that either Web applications or the rich user experience that traditional client/server applications provide.

In the current RIA market, a set of six key business applications are motivating overall RIA spending consisting of enhancement of existing web applications, high-transaction and event-driven Internet applications, next-generation portals, enhanced business intelligence solutions, application modernization, and peer-to-peer or mashup solutions. Furthermore, the market for RIA solutions actually consists of three submarkets focused on delivering RIA components, environments, or extensions to IDE suites. Finally, ZapThink expects spending on each of these areas to increase rapidly over the next five years, exceeding \$500 million in a conservative estimate by 2011.

##### 4.1. Figures

- Figure III-1: Average Price Point Prevalence for RIA Submarkets
- Figure III-2: RIA Application Type Prevalence
- Figure III-3: RIA Application Spending Over Time
- Figure III-4: Revenue Source by RIA Submarket
- Figure III-5: RIA Market Sizing
- Figure III-6: Technology Adoption by RIA Application Type
- Figure III-7: Technology Adoption by RIA Submarket
- Figure III-8: Technology Adoption Over Time

##### 4.2. Tables

- Table II-1: Capabilities and Limitations of Rich Client Approaches

## Copyright, Trademark Notice, and Statement of Opinion

All Contents Copyright © 2006 ZapThink, LLC. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. ZapThink disclaims all warranties as to the accuracy, completeness or adequacy of such information. ZapThink shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice. All trademarks, service marks, and trade names are trademarked by their respective owners and ZapThink makes no claims to these names.

## About ZapThink, LLC

ZapThink is an IT advisory and analysis firm that provides trusted advice and critical insight into the architectural and organizational changes brought about by the movement to XML, Web Services, and Service Orientation. We provide our three target audiences of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

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

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

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

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

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

### **ZAPTHINK CONTACT:**

ZapThink, LLC  
108 Woodlawn Rd, Suite 200  
Baltimore, MD 21210  
Phone: +1 (781) 207 0203  
Fax: +1 (786) 524 3186  
[info@zapthink.com](mailto:info@zapthink.com)

