Web 2.0 Driving Service Oriented Architectures

Version 1.0

February 2007

White Paper

Web 2.0 Driving Service Oriented Architectures

There is no doubt that Rich Internet Applications (RIAs) have taken the Web by storm. The meteoric rise in popularity of RIA has been accompanied by the introduction of the term "Web 2.0", which conveys a whole new way of delivering Web based applications to enterprise users. Facilitated by technologies such as Ajax, RIAs deliver the power and flexibility of desktop applications but with the convenience and cost savings of running software in a standard browser.

As the rich user interface enabled by Ajax is changing the way we interact with the Web, the need to preserve existing investments in mission critical Web applications, like the Service Desk, has become essential. To ensure that such applications continue to integrate with current and emerging technologies, software vendors must evolve to embrace SOA architectures and rich interactive user interfaces, without the need for special browser plug-ins.

This paper provides an overview of SOA architectures and Ajax-based RIAs, and explores how LiveTime has successfully blended these emerging technologies into LiveTime 5.0.

Traditional Web Applications

The Internet has emerged as the default platform for application development. Adoption and usage rates of the Internet has become the driving force behind technology investment, including spending on Service Desk software. As the Web continues to extend its reach into the enterprise, traditional page-based Web Service Desk applications face a substantial challenge: the inability to visually capture and represent the complexities required for today's Service Management frameworks, including ITIL. These limitations are inherent to the page-based, stateless-communication model.

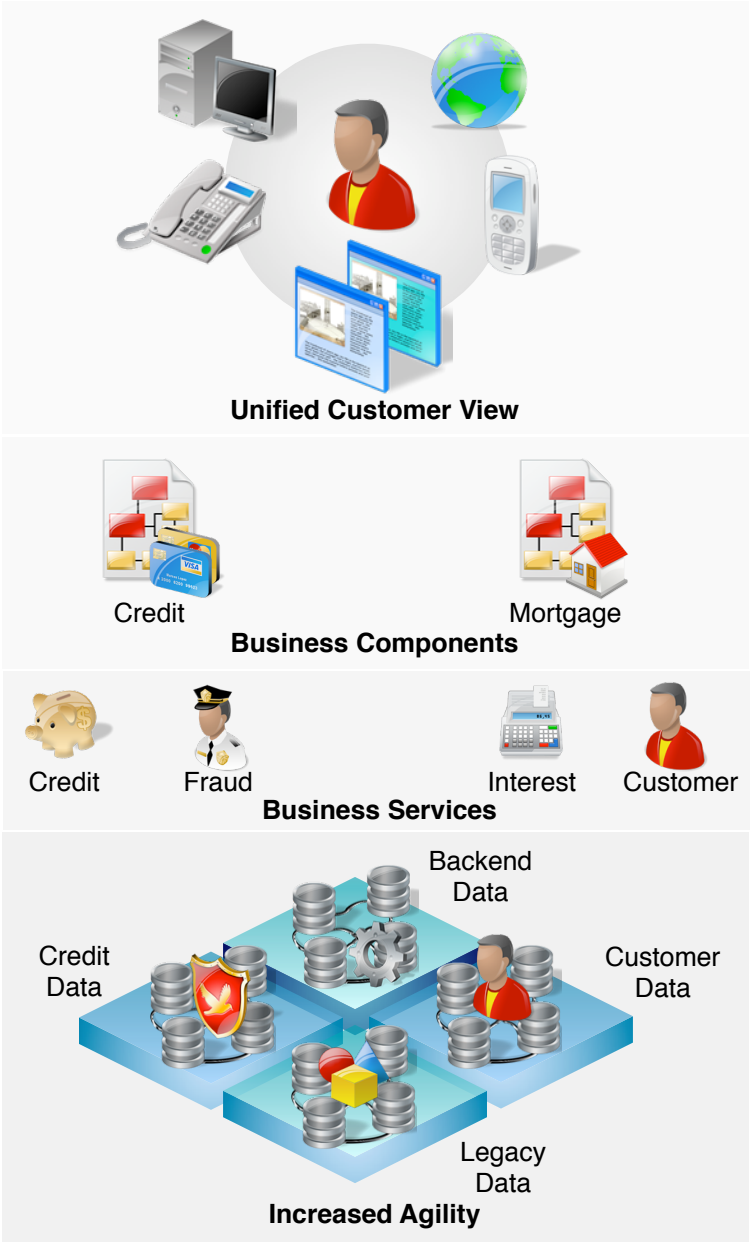
The restrictions are two fold: frustrated user experiences and excessive integration costs. In the page-based model, a page is self-contained and the minimal unit used to communicate between clients and servers. While simple and elegant in design and suitable for exchanging documents, ironically this model has become cumbersome and complex for developing modern applications like Service Desk software.

For example, to raise an incident, users may have to open a page to search for the customer, a page to locate the relevant CI, and another page to profile the incident. Users are often forced to leave the page they are working on, and navigate through several pages. It is easy to get lost and confused, which results in unhappy customers, low productivity and lost sales.

The challenge for Web based vendors, including Service Desk vendors, is to extend the page-based model of traditional Web applications in order to overcome its limitations. The updated model, requires applications that run at the server to take care everything from parsing the request, rendering the response, routing processes that link users from one page to another through to handling user errors.

SOA

A Service-oriented architecture (SOA) is software built using loosely coupled software services to support the requirements of business processes and software users. Network resources in a SOA environment are made available as independent services that can be accessed irrespective of the underlying platform implementation.



More importantly, SOA architecture enables the creation of applications that are built by combining loosely coupled and interoperable services. These services inter-operate based on a formal definition (or contract, e.g., WSDL) that is independent of the underlying platform and programming language, and the interface definition hides the implementation of the language-specific service. SOA-based systems can therefore be independent of development technologies and platforms (such as Java, .NET etc), and can be made to interact between platforms.

SOA can support integration and consolidation of activities within complex enterprise systems, but does not specify or provide a methodology or framework for documenting capabilities or services.

Ajax – the new face of SOA applications

Over the last decade, Web Service Desk applications have evolved from static HTML pages, to Dynamic HTML pages, applets, and, finally, to Ajax technologies (Asynchronous JavaScript and XML). Unlike applets, Ajax is based on the standard browser and JavaScript, with no proprietary plug in necessary.

Ajax is a kind of new generation DHTML. Like DHTML, it relies heavily on JavaScript to listen for events triggered by user activity. Events trigger functionality that amongst other things can dynamically manipulate portions of of a page (eg. DOM) in the browser. Moreover, it takes this a step further by enabling the communication with the server asynchronously without leaving or rendering the whole page again. It breaks the page-based model by introducing light-weight communication between the client and server. With proper design, Ajax can bring rich components similar to those of desktop applications to life in Web applications.

It is the combination of Ajax UI technology, ubiquitous Web network connectivity, and SOA services – along with high levels of security, reliability, scalability and governance – which are making these new 'Web 2.0' based business solutions a reality.

This is Illustrated by LiveTime 5.0, as Ajax breaths new life into Web applications by delivering the same level of interactivity and responsiveness as desktop applications.

LiveTime: Simple and Rich "Web 2.0" Service Desk

At LiveTime Software, we have extended our page-based Web services model with Ajax architecture to deliver a richer interactive experience to our users.

In 2001, we developed an infrastructure, inspired by Apple's Web Objects Framework, for developing a Service Desk application using the J2EE technology stack. In 2004, we extended our infrastructure, inspired by CSS (Cascading Style Sheets), Hibernate and Web services, for integrating with other third party applications like systems Management tools, Asset Management systems and Authentication Servers like LDAP and Active Directory.

With LiveTime 5.0, LiveTime Software has developed Ajax components that enable LiveTime applications to provide the same rich user experiences that desktop application users enjoy. The core LiveTime solution now includes an efficient Ajax-based engine to automate interactivity, including a rich set of XHTML components to improve usability.

With LiveTime, you're investing in a feature abundant, component rich Web Service Desk application driven by events triggered by users, similar to desktop applications. LiveTime applications allow art designers the freedom to create new UI elements through the use of CSS and HTML.

LiveTime Architecture

All LiveTime business logic continues to run at the server. Events triggered by users are automatically sent to the LiveTime application running at the server. Ajax-based components running at the server automatically update the visual representation at the browser.

To ensure that LiveTime integrates with all SOA enterprise enabled infrastructure, LiveTime has been designed as a pure n-tier Web solution that supports Web services. All J2EE middleware utilities work as they used to, such as JDBC, Hibernate, Java Mail or JMS.

The efficient web-based architecture means no more terminal-like interface user frustration, Javascript headache. asynchronous hassles or replication of business logic at the client interface.

LiveTime can be made to co-exist with portals, JSP, dashboards or any other technologies. Users can make any portions of their page highly interactive by simply including LiveTime pages. Users have total control of systems integration when communicating with other applications.

Conclusion

In order to deliver a manageable Web 2.0 Service Desk solution, LiveTime Software has evolved its Web services framework with Ajax technology to deliver a much richer interactive experience that is demanded by users. LiveTime embraces SOA architectures with dynamic RIAs to deliver a scalable and flexible Service Desk solution, with the convenience and cost savings of running in a standard browser.

For more information:

For more information about LiveTime Discovery or to discuss how LiveTime can assist you in your Discovery or Service Management project, please visit our website or contact us at info@livetime.com.

Contact Us
You can find out more information
or register for a demo from our website
www.livetime.com or you can

email us at: sales@livetime.com

United States
Telephone: +1 949 777 5800

Europe
Telephone: +44 (0) 1242 580090

Asia Pacific
Telephone: +61 (0)3 9620 7588