

# **POWERING FINANCIAL APPLICATIONS ON THE LIVING WEB**

A BUSINESS WHITEPAPER

# Powering Financial Applications on the Living Web

Web Communication without Limits

## Background

Today's Web users expect Rich Internet Applications (RIAs) that require the power of real-time Web communication to deliver compelling interactivity and transactional services on demand. The problem, however, is that with today's Web technology there is no such thing as a true real-time Web.

High performance, low-latency protocols such as Transmission Control Protocol (TCP), and the messaging systems that sit on top of TCP, perform extremely well within enterprise networks but the full benefits cannot be extended over the Web using Hyper Text Transfer Protocol (HTTP). For the growing number of organizations that need to provide their customers with services that rely on real-time data flows, —real-time is just a dream that is constrained by today's inefficient technologies and bound to yesterday's Web communication architecture.

## Solving the dilemma of the Real-Time Web

The next generation real-time Web is both a holy grail and a major headache for IT executives. On one hand, delivering customers an unprecedented level of real-time services through their existing Web experience means standing out amongst a crowded field, and increasing market share. On the other hand, accomplishing these tasks within the limits of today's Web architecture means quickly encountering performance snags and scalability issues that eat up budgets. Is the improvement worth the investment and risk?

To solve this dilemma for customers in the financial services industry requires a fundamental rethink of how data is delivered to Web clients. The desired solution needs to address latency and integration challenges by directly extending internal real-time data to Web-based Rich Internet Applications (RIA); this would eliminate the bandwidth overhead, conversion headaches and time delays, and make Web-based applications function with the performance and interactivity offered by desktop or client-server applications - with the ease and cost savings of Web based client deployment and management. To alleviate risk and potentially significant investment, this approach needs to leverage existing enterprise application development practices and skill sets.

Kaazing was founded to tackle these problems head-on. Kaazing turns any browser or mobile device into the equivalent of a real-time, full-function desktop client by simply and seamlessly extending any TCP-based protocol or messaging service out to the Web, and doing so without the complex and costly translation required with today's HTTP-based solutions.

The possibilities extend only to the limits of the imagination. Picture being able to monitor truly real-time data flows, from server statistics to stock portfolios and up-to-the-millisecond FX spot rates over the Web--on any device. What about the ability to stream messages with minimal latency and building out the next generation of RIA-based client applications using the existing protocols and integration approaches already familiar to your team? For the first time it can all be done.

## Extending High Performance Messaging Systems to the Web

High-performance messaging systems are an essential component of any financial services business. To pass muster in the financial services industry, while meeting the rigorous demands of real-time trading platforms, IT systems must achieve the highest measures of security, deliver superior reliability--but most of all, they must be fast. To achieve these goals, organizations within the financial services Industry have assembled their own ultra-high performance infrastructure from scratch, with a variety of highly scalable components: JMS based systems, TIBCO's Rendezvous, Informatica's Ultra Messaging products, industry standard protocols such as the Advanced Message Queuing Protocol (AMQP), and homegrown protocols devised to tackle specific problems.

To meet the evolving expectations of customers, large enterprises today seek to extend the functionality of these back-end messaging systems over the Web, which involves considerable wrangling of code, and marshaling of expensive resources. Furthermore, these messaging systems must reach customers who are now accessing data across a range of web-enabled devices such as tablets and mobile phones, different browser types, and different versions of these browsers.

What may be lightning-fast, reliable and capable of handling huge message volumes within the walls of an Enterprise becomes a snarled, unruly mess on the Web. In order to deliver services over the Web that replicate full featured enterprise functionality, financial services businesses have relied on HTTP-based workarounds that robbed applications of the performance and scalability they enjoy within the enterprise. Introducing high levels of latency on the client side, and slower deployment times on the server side were all seen as the cost of doing business.

A solution to tame the cost and complexity involved in extending these services to the Web is a welcome change, especially in an era where the popularity of Rich Internet Applications continues to soar.

## Reducing Cost and Improving Performance

Rich Internet Applications are a highly desirable alternative to the distribution and upkeep of desktop software, especially if they deliver the same functionality that desktop applications receive from back-end enterprise messaging systems. RIAs deliver attractive, highly interactive user interfaces with compelling graphics and video, and can do so within the existing framework of the browser.

The tradeoff with deployment of RIAs is that they place even more strain on overloaded Web infrastructure, because these applications still use HTTP to communicate between the Web browser and the back end service. In order to simulate the real-time functionality that happens on the server side, such as data-feeds for trading, these RIAs have to convert highly optimized TCP-based messages into HTTP, which only then can be carried over the Web.

Ostensibly, what the user gets is an experience that unfolds in real-time. Under the hood, the use of HTTP is neither real-time nor is it as streamlined as using TCP. The cost is considerable, an expense which must be borne by the Web infrastructure, the application developer and the support organization. As adoption rates increase, hardware and software costs will spiral out of control, performance will degrade and operational overhead soars.

In the absence of a truly viable alternative, business as usual continues.

## Introducing the Kaazing WebSocket Gateway

The Kaazing WebSocket Gateway (KWG) is a high-performance Web communication platform that enables bi-directional and full-duplex connectivity that is ideal for building real-time RIAs quickly and cost effectively. At the core of KWG is Kaazing's patent-pending WebSocket Acceleration™ technology, which reliably and securely connects any Web-enabled device and JavaScript-enabled browser to real-time data flowing through an organization.

---

*“Kaazing provided a stable and scalable out-of-the-box solution that allowed us to rapidly deliver our Prime Services Rich Internet Application in a most cost-effective manner. The rapid integration of the Kaazing WebSocket Gateway into our infrastructure dramatically reduced our architectural complexity and delivered superior latency, highest performance and scalability. Kaazing provided key infrastructure to support our high-speed, mission-critical web application. They were a clear choice as a critical component of our customer portal.”*

- David Kim, Director of Development, Cantor Fitzgerald Prime Services

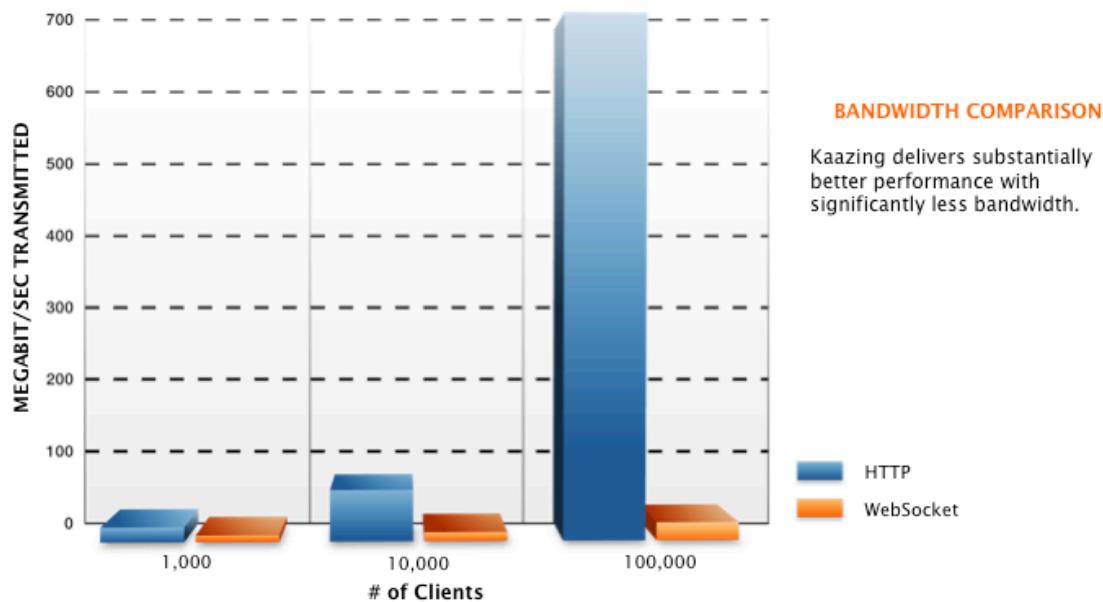
---

WebSocket Acceleration eliminates the overhead and latency inherent in HTTP by extending the use of any TCP-based messaging format to any browser, delivering ultra-

high performance communication over the Web. It does this while co-existing alongside current infrastructure inside your organization, working seamlessly with the HTTP communications fabric that is between you and your customers, and without the need for any additional client side software.

WebSocket Acceleration requires no browser plug-ins, and integrates quickly and easily with enterprise messaging platforms such as JMS. It does not rely on costly —long-polling techniques that eat up server resources and induce latency. Furthermore, it works with a wide variety of client-side technologies, including Microsoft .NET, JavaScript, Adobe Flash, Microsoft Silverlight, Java, Java-applet and JavaFX.

The technology implements the latest HTML5 communication standard – WebSocket, which many analysts and industry experts believe will be the foundation of future Web development. However with Kaazing you get the benefits of the latest HTML5 communications that is totally compatible with your existing Web infrastructure investments - we even ensure that older, pre-HTML5 browsers such as Internet Explore 6 can benefit from the same HTML5 communication features and can even work with your existing real-time messaging services without the need for additional client software.



	HTTP Traffic*	WebSockets Traffic*
Google	788 bytes, plus 1 byte	1 byte, plus 1 byte
Yahoo	1,737 bytes, plus 1 byte	1 byte, plus 1 byte

\* Header information for each character entered into search bar

In the above figure (figure 1) an FX application that receives 1 update per second shows the amount of bandwidth used by HTTP and WebSocket. Using HTTP to deliver the FX spot rates requires wrapping of the HTTP request and response headers that, in this example, adds 6,300 bits (788 bytes x 8) of overhead in network traffic per update. With

1,000 clients, 6,3Mbps (1 update/sec x 6,300 bits x 1,000 clients) of unused header data is sent. Compare this to WebSocket traffic where the header data equals 16 bits (2 bytes) of additional data traffic per update. Using the same data points as in the HTTP example, now with 100,000 clients, adds 1.6Mbps (1 update/sec x 16 bits x 100,000 clients) of data compared to 630Mbps in the HTTP example.

As shown in the above example, KWG saves a tremendous amount of network bandwidth compared to traditional HTTP polling traffic. This also impacts the overall performance of the FX application and significantly reduces the latency of delivering the actual FX spot rates and the CPU/Memory resources on the server host that delivers the data. Plus, KWG provides a ubiquitous upstream delivery mechanism for traffic flowing back to the server based on executed trades, upstream chat, or other such responses from the dashboard user.

## Product Features and Benefits

Developers using the KWG can build services faster, CIOs can streamline operations and meet the needs of the business at significantly lower cost, and end user clients enjoy a dramatically improved application experience.

Consider just some of the benefits:

Choice	Integration	Performance
Ships with development libraries for Microsoft .NET, HTML/JavaScript, Java, Adobe Flex/Flash and Microsoft Silverlight providing you with your choice of Rich Internet Application (RIA) Development platforms. KWG supports all major browsers such as IE5.5+, Firefox 1.5+, Opera 9.5+, Safari 3.0+ and Chrome 1.0+	Provides easy integration with many popular messaging services such as Informatica Ultra Messaging, TIBCO EMS, IBM MQ, ActiveMQ and RabbitMQ. KWG can also be extended to support any number of TCP-based standard and custom transport protocols that may be required	Offers a high-performance Web architecture that streams more data with less latency, ensuring you meet your Service Level Agreements (SLA), comply with performance requirements and deliver a much more satisfying experience to end users.
Scalability	Reliability	Standards Compliance
Delivers up to 100 times the capacity of other approaches while reducing capital outlay an operational expenses and also preparing your organization for rapid adoption and traffic spikes. In cloud computing or virtualized environments this also means fewer virtual machines and lower monthly bills.	Meets the exacting reliability requirements of Financial Services ensuring that Web clients are as dependable as your existing messaging systems, and enabling guaranteed message delivery. KWG can traverse any network, tunneling automatically through proxy servers and firewalls so that you can be sure your services are delivered.	Kaazing is spearheading work on industry standards, ensuring availability of skilled development resources and protecting your applications against vendor lock-in while ensuring their longevity and maintainability. Kaazing has taken a leadership position in the HTML5 specification, ensuring our clients have access to the latest technology today.

## Summary

Kaazing lets you build, deploy, and manage real-time Web applications that can handle very large numbers of users and very high message volumes, using standard browsers and protocols, without changing the way you build and integrate software. We are paving the road for the next iteration of the Web – one in which millions of people and devices communicate with one another instantly and efficiently in real-time.

The Kaazing WebSocket Gateway was created with the needs of the financial services industry in mind. We connect in-house messaging protocols directly to browser-based Rich Internet Applications, offering previously unattainable levels of performance and reliability that make browser-based applications perform as well as desktop software.

At Kaazing we believe that our approach to building scalable, real-time solutions for Rich Internet Applications in the financial services community will become the standard to which others will aspire. We are actively pushing the boundaries of what will be possible by building out the next generation of web communication solutions.

To learn more, please visit us at [www.kaazing.com](http://www.kaazing.com) or contact [sales@kaazing.com](mailto:sales@kaazing.com).