

Kaazing WebSocket Gateway - JMS Edition

KAAZING WEBSOCKET GATEWAY enables companies to quickly create a new class of fully-interactive web applications with live streaming data – enhancing the user experience, simplifying development, and reducing the cost of back-end server infrastructure.

The JMS EDITION extends the reach of your JMS message broker to browsers and mobile clients over the Web without sacrificing performance or security

Business applications are being deployed more and more not to the desktop, but to the Web and mobile devices. Today's Internet users demand the same kind of reliable real-time communication and near-zero latency of traditional desktop applications. Whether they're facilitating financial transactions, getting up-to-the-second data, playing games with others, chatting on-line, or bidding in auctions, web and mobile users demand responsive two-way communication.

Traditional web architecture is based on HTTP which wasn't designed for real-time or interactive communication. While this approach has served the Internet well for the last fifteen years, it is now outmoded for the kinds of responsive real-time applications businesses are required to deliver.

KAAZING WEBSOCKET GATEWAY is based on a new standard called HTML5 WebSocket and can connect hundreds of thousands of concurrent users directly to the real-time data flowing through an organization. The WebSocket specification was designed to address the limitations of HTTP and serve as the communication model for the next generation of interactive and streaming web applications.

The **JMS EDITION** lets your browser and mobile applications connect to your back-end JMS message broker just like a desktop application does, but over the Web through firewalls and proxies. Companies no longer need to distinguish between desktop applications, browser applications, and mobile applications in terms of publishing and subscribing data to their message brokers. You could say that Kaazing WebSocket Gateway brings client-server to the Web.

FEATURES

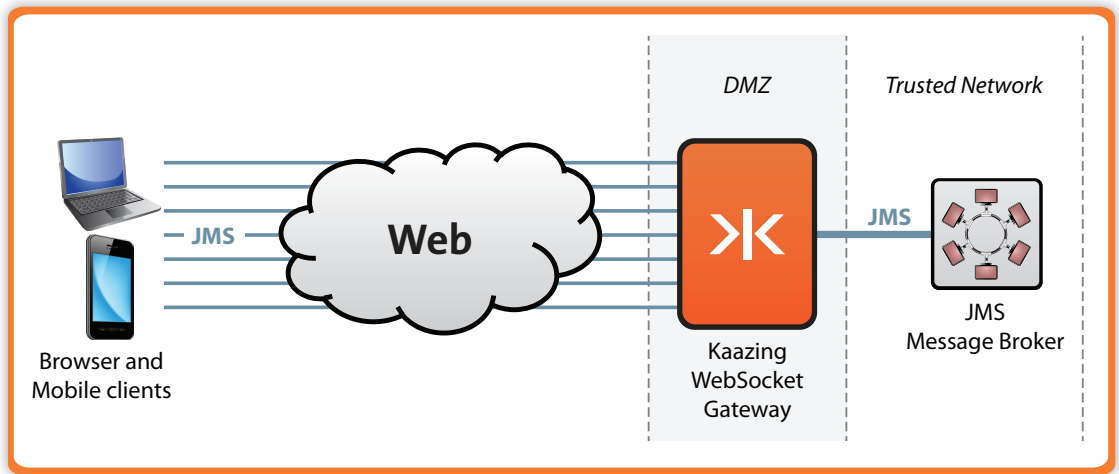
BENEFITS

- Creates a superior user experience with full interactivity and streaming live data – no plug-ins required
- Extends existing client-server based applications and JMS message brokers to the Web
- Increases productivity for developers who can use the standard JMS APIs which they are already familiar with
- Reduces operational costs significantly and simplifies Web architecture by minimizing Web-tier processing
- Unifies your architecture for both the Web and mobile devices by creating a single access point
- Connection and subscription offloading scales your JMS server far beyond its inherent capacity
- Available for a variety of client technologies, even older browsers will work seamlessly

FEATURES

- Use JMS end-to-end from browser or mobile client over WebSocket to your JMS message broker
- Works with industry standard message brokers like ActiveMQ, Informatica UMQ, IBM MQ
- Publish and subscribe to topics and queues
- Fast fail authentication ensures your data stays safe before users even communicate with your server
- Fine-grained authorization lets you specify the operations that users can perform
- Send both JMS text and binary messages
- Automatic reconnection for dropped connections
- Asynchronous and synchronous message processing
- Start and stop flow control
- Custom Message Properties
- Client-initiated Transactions

This example architecture diagram shows browser and mobile clients connecting to a Kaazing WebSocket Gateway over the Web. The Gateway offloads those connections and only uses a single connection to the JMS broker. Thus Kaazing WebSocket Gateway scales your back-end server beyond its inherent capacity, to “web scale”. Unlike traditional web



applications, here JMS is used from the client end-to-end all the way to the JMS message broker. In other words, the client is effectively talking directly to the back-end message broker like a desktop application does. Your JMS server remains protected in the trusted network while Kaazing WebSocket Gateway is the access point in the DMZ.

Best-in-industry performance and scalability

Kaazing WebSocket Gateway scales easily to enormous amounts of concurrent users. Running separately from your server, it eliminates competition for resources, dramatically enhances performance, and lowers latency. In addition, network bandwidth is a tiny fraction of competing pseudo-real-time Web products. In some cases, the bandwidth required is 1/1000th of an HTTP-based Ajax or Comet solution.

Complete browser support without plug-ins

With Kaazing WebSocket Gateway there is no need to upgrade to the latest browser versions in order to take advantage of HTML5 WebSocket. Kaazing’s WebSocket emulation technology will automatically kick-in if it detects your application running in a browser with no native support for WebSocket. The emulation is 100% transparent so developers can deploy the one application without having to worry about browser-specific logic. Browsers as old as Internet Explorer 5.5 can be used with near-native performance – all without plug-ins.

Enhanced security and stateless single sign-on

Kaazing WebSocket Gateway integrates with JAAS, supporting pluggable authentication modules, and is the world’s only implementation of Kerberos single-sign on over WebSocket. Encrypted credentials sent to Kaazing WebSocket Gateway are automatically injected into the protocol before authenticating with the back-end system, eliminating the time and risk of a long, multi-password authentication sequence without compromising credential storage.

Enterprise-grade reliability and resiliency

Message delivery from back-end systems or services are guaranteed all the way from the server to the browser and vice-versa. In case of a broken request, Kaazing WebSocket Gateway automatically reconnects, guaranteeing message delivery. Clustering adds resiliency by reducing single points of failure for high availability and lets you scale easily as demand increases.

Client technology boosts developer productivity

Developers build against the same JMS APIs they are already familiar with, since their application effectively communicates to the back-end directly. What’s more, you can use a wide variety of client-side technologies including JavaScript, Adobe Flex (Flash), Microsoft Silverlight, Java, JavaFX, and mobile devices.

LEARN MORE

Browser Requirements

- Apple Safari 3.0
- Google Chrome 1.0
- Microsoft IE 5.5
- Mozilla Firefox 1.5
- Opera 9.5

System Requirements

- 100MB of storage space
- 1.5GHz CPU or better
- Java 1.6 (Java6) certified platform
- Java Developer Kit (JDK) 1.6x (Java6) or greater

FOR MORE INFORMATION CONTACT:

HQ/United States
P: +1 (877) KAAZING
F: +1 (650) 960 8145
E: sales@kaazing.com

EMEA/United Kingdom
P: +44 (208) 895 4023
F: +1 (650) 960 8145
E: sales@kaazing.com

444 Castro Street
Suite 1100
Mountain View, CA 94041
United States

Citypoint
1 Ropemaker Street
London, EC2Y 9HT
United Kingdom