

# HTML 5 WebSocket delivers the Real-time Web

QCon, San Francisco  
November 18, 2009

# Introduction

- John Fallows
  - CTO @ Kaazing
  - Author @ Pro JSF & Ajax
  - Contributor @ W3C HTML 5
  - Contributor @ IETF Bidirectional Hypertext
- Kaazing Corporation
  - Founded May 2007
  - Based in Mountain View, California, USA
  - Provides Kaazing WebSocket Gateway

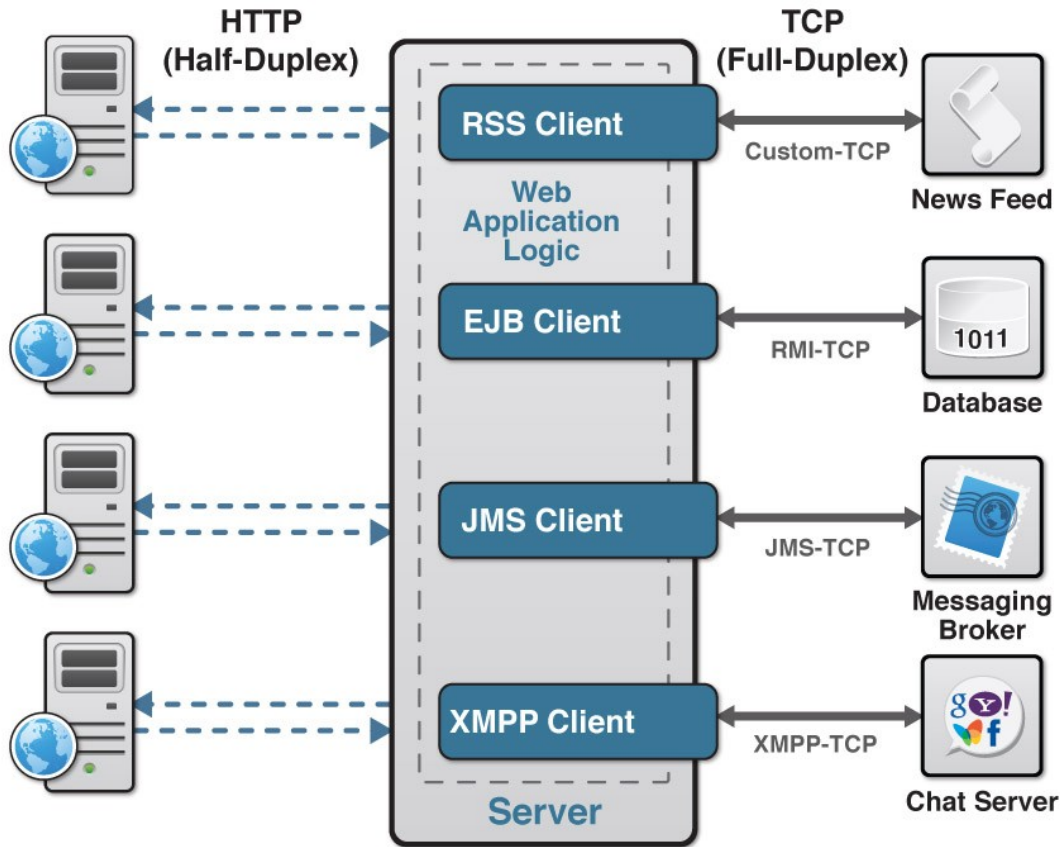
# What is Real-time?

- Hard real-time systems
  - Heart pacemaker – increased latency is useless
- Soft real-time systems
  - Multimedia streaming – increased latency is recoverable

# Real-time Web challenge

- HTTP is the foundation of the Web
  - Designed for document transfer
  - Client-initiated request-response
  - Intermediate proxies add complexity
  - Internet adds variable latency
- Web applications demand real-time
  - Social networking
  - Internet gaming
  - Financial services

# Half-duplex Architecture ☹️



# HTML 5 Overview

- Next generation application platform
  - Communication (sockets, cross-origin)
  - Graphics (2D)
  - Drag 'n' drop
  - Storage (transient, persistent)
  - Offline
  - Compatibility
- “Last Call” October 2009

# HTML 5 Communication

- WebSocket
  - HTTP-friendly text socket for your browser
- Server-Sent Events
  - Standardized HTTP streaming (downstream)
- Cross-Origin XMLHttpRequest
  - Secure cross-origin remote communication
- postMessage
  - Secure cross-document local communication

# HTML 5 WebSocket

- Full-duplex, bidirectional text socket
  - Send and receive strings
- Browser native support
  - Chrome 4.0 beta
  - Mozilla Firefox trunk
- Kaazing emulation support
  - IE 5.5+, Firefox 2.0+, Opera 9.61+, Safari 3.2+, Chrome 2.0+



# HTML 5 WebSocket API

```
[Constructor(in DOMString url, in optional DOMString protocol)]
```

```
interface WebSocket {
```

```
  readonly attribute DOMString URL;
```

```
  // ready state
```

```
  const unsigned short CONNECTING = 0;
```

```
  const unsigned short OPEN = 1;
```

```
  const unsigned short CLOSED = 2;
```

```
  readonly attribute unsigned short readyState;
```

```
  readonly attribute unsigned long bufferedAmount;
```

```
  // networking
```

```
  attribute Function onopen;
```

```
  attribute Function onmessage;
```

```
  attribute Function onclose;
```

```
  boolean send(in DOMString data);
```

```
  void close();
```

```
};
```

# HTML 5 WebSocket Handshake

```
GET /real-time HTTP/1.1\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
...\r\n
```

```
HTTP/1.1 101 WebSocket Protocol Handshake\r\n
Upgrade: WebSocket\r\n
Connection: Upgrade\r\n
...\r\n
```

# HTML 5 WebSocket Framing

- Frames can be sent full-duplex
  - Either direction at any time
- Text or binary frames
  - Lead byte indicates frame type
- Text Frames use terminator
  - `\x80Hello, WebSocket\xff`
- Binary Frames use length prefix
  - `\x00\x10Hello, WebSocket`

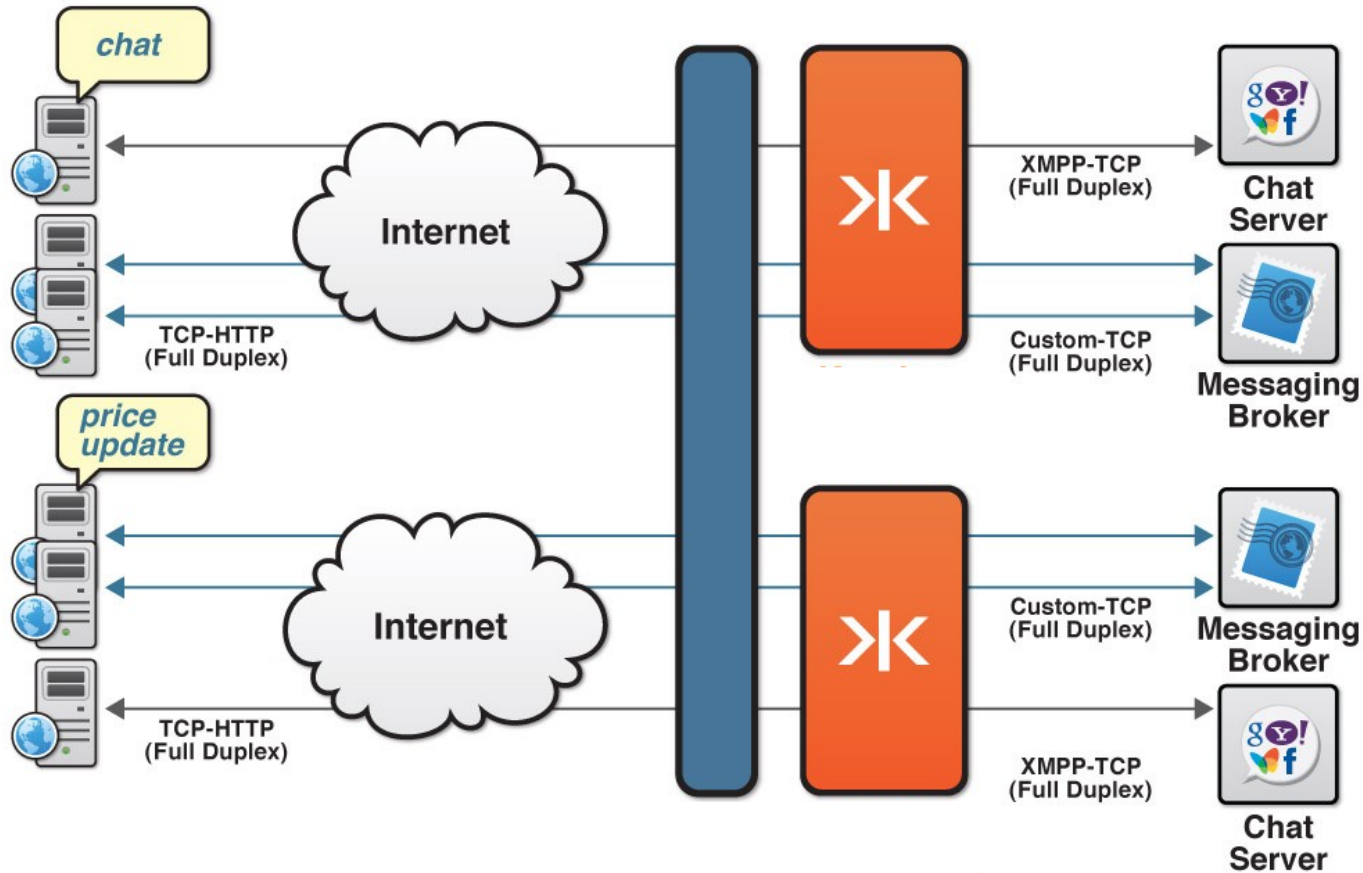
# HTML 5 WebSocket Efficiency

- Send “Hello, world” message client-to-server
  - “Hello, world” + 2 bytes framing (WebSocket)
  - “Hello, world” + ~1K headers (Ajax)
    - Plus the response overhead – network, memory & CPU
    - Plus request queuing delays at client
- Send “Hello, world” message *server-to-client*
  - “Hello, world” + 2 bytes framing (WebSocket)
  - “Hello, world” + ~1K headers (Ajax)
    - Plus polling or long-polling empty request-reponse overhead
    - Plus buffer memory between polls at server

# HTML 5 WebSocket Security

- Wire encryption via TLS / SSL
  - ws://kaazing.net/clear
  - wss://kaazing.net/encrypted
- Cross-Origin WebSocket connections
  - Sandbox execution model
  - Origin: http://www.kaazing.com:80
  - WebSocket-Origin: http://www.kaazing.com:80
- HTTP authentication and authorization
  - WebSocket handshake is traditional HTTP

# Full-duplex Architecture ☺



# DEMO

*“Web Sockets in action”*

# Kaazing WebSocket Gateway

- High-Performance WebSocket Server
  - Optimized for minimal overhead per connection
- High availability clustering
  - Optimized for minimal intra-cluster communication
- Protocol-specific acceleration
  - Driven by customer demand



# Kaazing WebSocket Client

- Emulation via HTTP streaming
  - Long-polling is a configuration error (!)
  - Cross-origin also supported by emulation
- HTTP proxy detection
  - Dynamically adapt to proxy behavior
- Recover from server failure
  - Automatically reconnect
  - Fallback to disaster recovery site
- Support plug-in technologies too
  - Flash, Silverlight, Java(FX)

# Kaazing ByteSocket

- HTML 5 WebSocket is text-only
  - Send and receive strings
- Kaazing ByteSocket supports binary
  - Send and receive ByteBuffers
  - ByteBuffers contain different types
    - Fixed-width integers
    - UTF8 strings

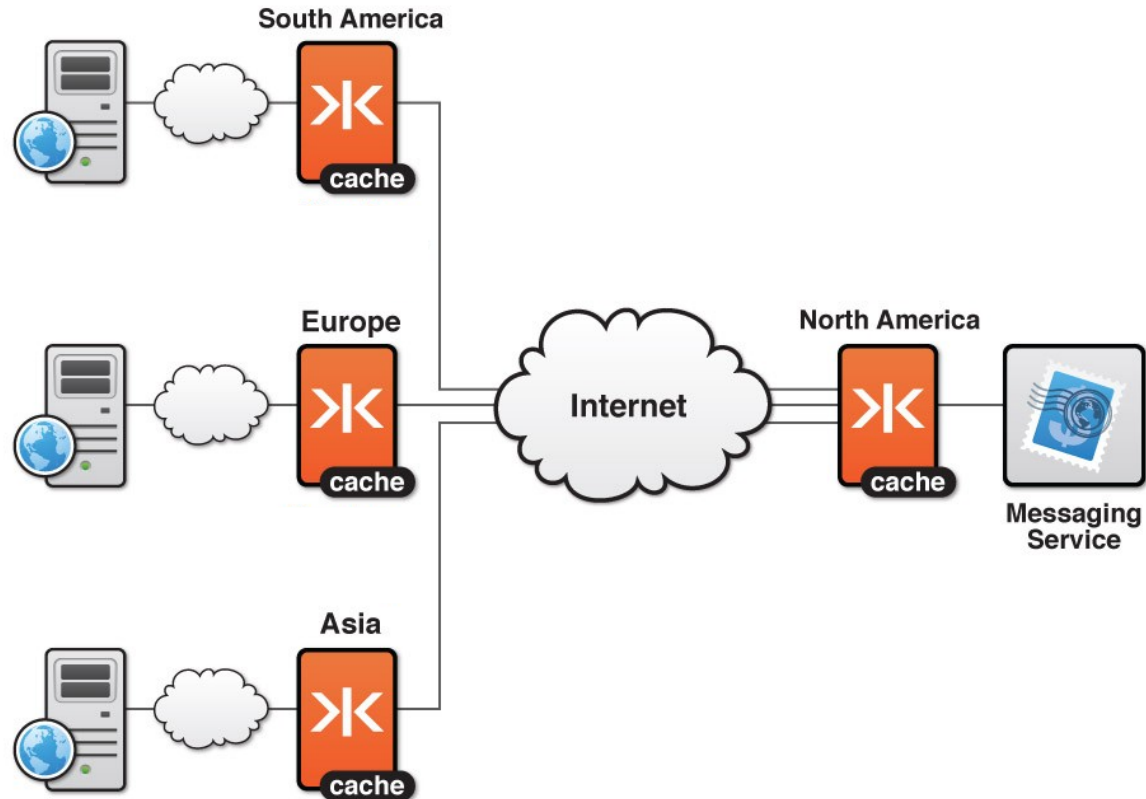
# Kaazing Protocols

- JMS (Stomp)
- AMQP
- XMPP
- IRC
- Telnet
- ...

# Kaazing WebSocket Acceleration

- Connection-offloading
  - Reduce target server kernel overhead
  - Minimize redundant network traffic
- Global message delivery
  - Deploy Kaazing WebSocket Gateway at edge
  - Dramatically reduce message latency
  - No need for VPN or leased line solutions
  - Use secure WebSocket to reach central Gateway

# Kaazing WebSocket Acceleration



# Kaazing High Availability

- Layer 4 (TCP) load balancing
  - In-band for entire WebSocket connection
  - Compatible with existing half-duplex deployments
  - Potential bottleneck for full-duplex real-time
- Layer 7 (HTTP) load balancing
  - Used during connection attempt only
  - Redirected location used directly
  - Eliminates balancer bottleneck
  - Reduces balancer scalability requirements

# Kaazing Disaster Recovery

- Disaster eliminates entire data center
  - Examples – earthquake, flood
  - Recovery site available
- Existing WebSocket clients
  - Automatically reconnect to recovery site
  - Recovers interaction state seamlessly
- New clients
  - Download cached application
  - Connect to disaster recovery site

# DEMO

*“Kaazing WebSocket Gateway”*



# Summary

- HTML 5 WebSocket delivers the Real-time Web
- Kaazing delivers HTML 5 WebSocket
  - Kaazing WebSocket Gateway
    - 60-day free trial
    - <http://www.kaazing.com>
    - 24x7 support available
  - Kaazing WebSocket client emulation
    - Older generation browsers supported
    - Browser plug-ins NOT required
    - Browser plug-ins also supported directly

# Q & A

*“Is that a WebSocket in your browser?”*

THE EXPERT'S VOICE®

# Pro HTML5 Programming

*Powerful APIs for Richer Internet  
Application Development*

Peter Lubbers, Brian Albers,  
Frank Salim, Ric Smith

Apress®