



Think BIG, Really BIG!

September 2010

Jonas Jacobi

Co-Founder: Kaazing

Co-Author: Pro JSF and Ajax, Apress, 2006

Think Different ...

KAAZING >K™



Think Different ...

KAAZING >K™

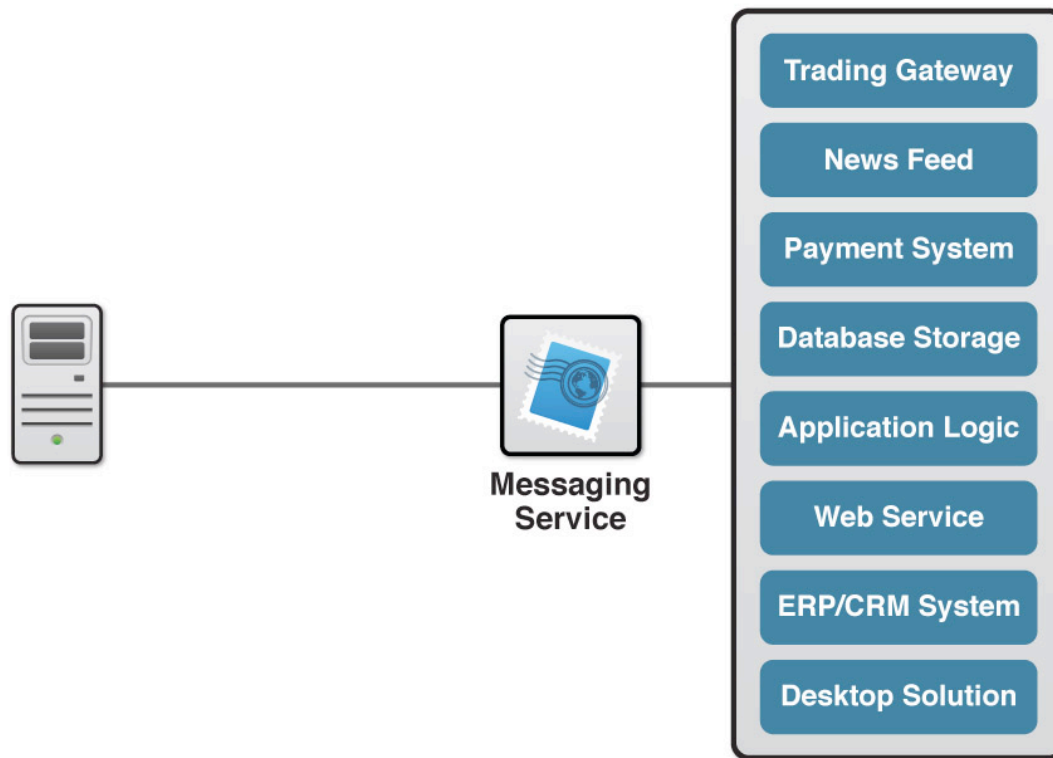


- Demand for:
 - Dynamic, Live, and Interactive Web Applications
- Today:
 - Desktops, Laptops, and Smart Phones
- Tomorrow:
 - TVs, IPTV, Blu-ray players, Cars, Fridges, Bus & Train Stops, Malls, Smart Grids, SatNavs, ...

It is just the beginning.

“If we were not restricted by the traditional limitations of HTTP, what type of Web applications would we build?”

Desktop Architecture



HTTP Is Not Full Duplex



- HTTP is designed for document transfer
 - Resource addressing
 - Request / Response interaction
 - Caching
- HTTP is bidirectional, but half-duplex
 - Traffic flows in only one direction at a time
- HTTP is stateless
 - Header information is resent for each request

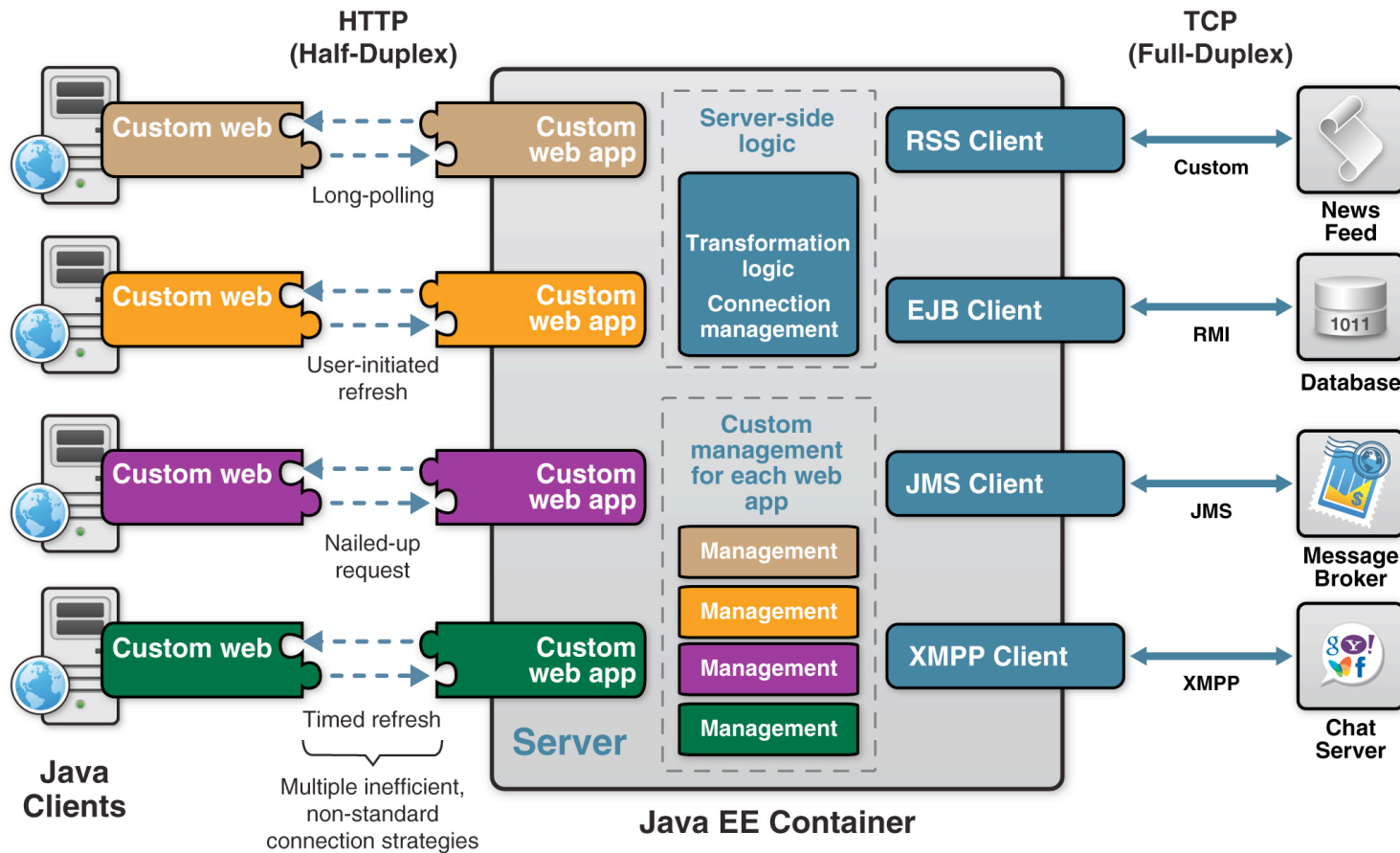
HTTP Request Headers

```
POST /gwt/EventService HTTP/1.1
Host: gpokr.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US) AppleWebKit/532.5 (KHTML, like
Gecko) Chrome/4.1.249.1064 Safari/532.5
Referer: http://gpokr.com/gwt/7F5E66657B938E2FDE9CD39095A0E9E6.cache.html
Content-Length: 134
Origin: http://gpokr.com
Content-Type: text/plain; charset=utf-8
Accept: */*
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
Cookie: __utmz=247824721.1273102477.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
JSESSIONID=E7AAE0E60B01FB88D1E3799FAD5C62B3;
__utma=247824721.1247485893.1273102477.1273104838.1273107686.3; __utmc=247824721;
__utmb=247824721.4.10.127
```

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Expires: Thu, 06 May 2010 01:06:51 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 303
Date: Thu, 06 May 2010 01:06:50 GMT
```

- Total (unnecessary) HTTP request and response header information overhead: 871 bytes (example)
- Overhead can be as much as 2 kB

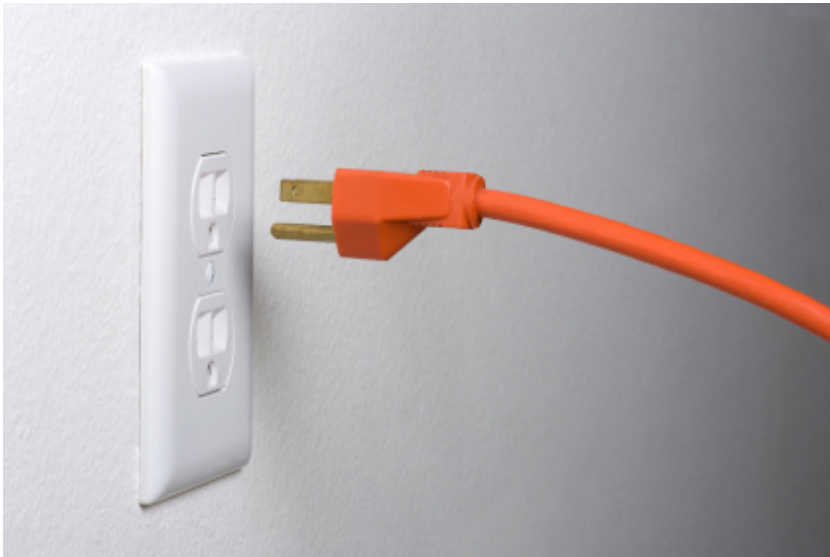
Half-Duplex Architecture



Complexity does not scale



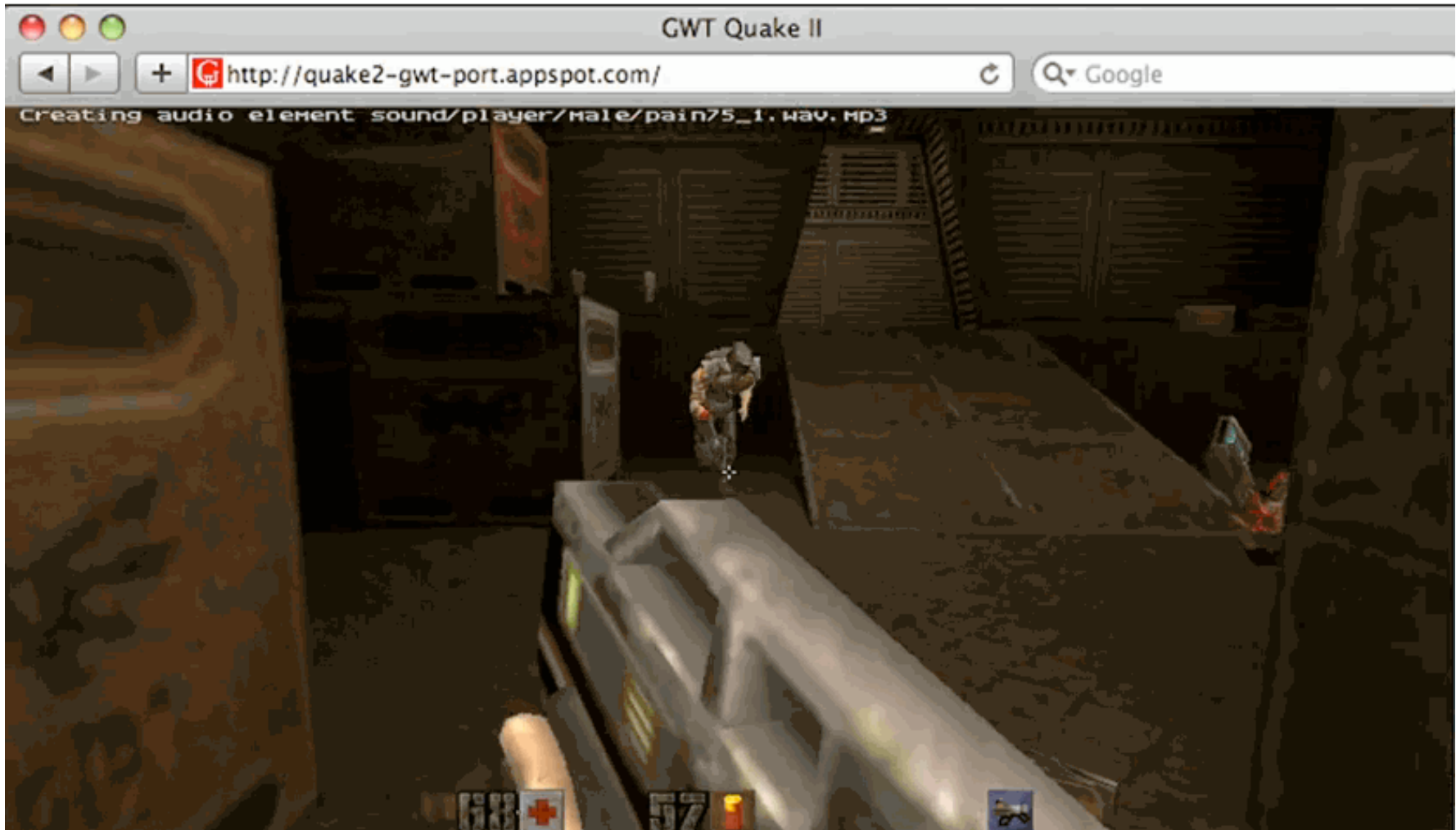
Enter HTML5 WebSocket!



- Today's Web applications demand reliable, real-time communications with near-zero latency
- Not just broadcast, but bidirectional communication
- Examples:
 - Financial applications
 - Social networking applications
 - Online games
 - Smart power grid



WebSocket: Quake II Port



<http://code.google.com/p/quake2-gwt-port>

- Initial draft TCPConnection, submitted 2007 by Ian Hickson, Google & HTML5 Editor
- New draft WebSocket, submitted summer 2008 by the Kaazing Dev Team (John, Michael, and Brian)

What do WebSocket and model trains have in common?



- Includes W3C API and IETF Protocol
 - Event-driven JavaScript API
 - Full-duplex text-based protocol
- Integrates HTTP addressing
 - <ws://websockets.org:80/tcp-for-the-web>
- Traverses firewalls, proxies, routers
- Leverages Cross-Origin Resource Sharing (CORS)
 - <http://www.w3.org/TR/access-control/>
- Allows unlimited connections per Origin

- **WebSocket**

`ws://www.websocket.org/text`

- **WebSocket Secure**

`wss://www.websocket.org/encrypted-text`

- Connection established by upgrading from the HTTP protocol to the WebSocket protocol using the same TCP connection
- Once upgraded, WebSocket data frames can be sent back and forth between the client and the server in full-duplex mode

WebSocket Handshake

```
GET /chat HTTP/1.1\r\nHost: facebook.com:80\r\nConnection: Upgrade\r\nSec-WebSocket-Key2: 12988 5 Y3 1 .P00\r\nSec-WebSocket-Protocol: XMPP\r\nUpgrade: WebSocket\r\nSec-WebSocket-Key1: 4@1 46546xW%01\r\nOrigin: http://facebook.com:80\r\n\r\n[8 byte challenge]
```

```
HTTP/1.1 101 WebSocket Protocol Handshake\r\nUpgrade: WebSocket\r\nConnection: Upgrade\r\nSec-WebSocket-Origin: http://facebook.com:80\r\nSec-WebSocket-Location: ws://facebook.com:80/chat\r\nSec-WebSocket-Protocol: XMPP\r\n\r\n[16 byte response]
```

Checking Browser Support

JavaScript

```
//Checking for browser support
if (window.WebSocket) {
    document.getElementById("support").innerHTML =
        "HTML5 WebSocket is supported";
} else {
    document.getElementById("support").innerHTML =
        "HTML5 WebSocket is not supported";
}
```

JavaScript

```
// Create new WebSocket  
var mySocket = new WebSocket("ws://websockets.org/  
example");  
  
// Attach listeners  
mySocket.onopen = function(evt) {  
    console.log("Connection open...");  
};  
  
mySocket.onmessage = function(evt) {  
    console.log("Received message: " + evt.data);  
};  
  
mySocket.onclose = function(evt) {  
    console.log("Connection closed...");  
};
```

JavaScript

```
// Send data  
mySocket.send("HTML5 WebSocket Rocks!");
```

```
// Close WebSocket  
mySocket.close();
```


- Chrome 4.0+
- Safari 5.0 & iOS 4
- Firefox 4:Beta

“We really really want to support WebSockets in the next version of Firefox.”

–Christopher Blizzard, Mozilla

- It is NOT a better Ajax or XHR, nor was it designed to be!
- WebSocket was NOT designed to be a one-size-fits-all replacement of HTTP or XHR!

- When you build Desktop applications (client server solutions) do you develop your application using raw TCP?
- Or, are you using higher level protocols and APIs, such as XMPP, JMS, JDBC?

You should think of WebSocket the same way:

- Binary payloads
 - Optimized integer and string payloads
- Multiplexing
 - Connection reuse
- Extend client-server protocols to the Web
 - XMPP, Jabber
 - Pub/Sub (Stomp/AMQP)
 - Gaming protocols
 - Any TCP-based protocol



JavaScript

```
// Stomp Client
var myStomp = new StompClient();
myStomp.onopen =
function(headers) {
    myStomp.subscribe("/topic/destination");
}

myStomp.onmessage =
function(headers, body) { alert(body); }
myStomp.connect("ws://www.websocket.org/
stomp");
myStomp.send("Hello Stomp!",
    "/topic/destination");
```

Stomp offers the following client commands:

ABORT

ACK

BEGIN

COMMIT

CONNECT

DISCONNECT

SEND

SUBSCRIBE

UNSUBSCRIBE

JavaScript

```
// XMPP Client
var client = new XmppClient();
client.onopen = function() { ... }
client.onclose = function() { ... }

// Got a message from the server... display it
client.onmessage = function(msg) { ... }
// Presence update... display it
client.onpresence = function(p) { ... }

// Now, connect to the chat server
client.connect(url, "localhost")
```

Stomp offers the following client commands:

Connect and disconnect

Register

Check roster

Send messages

Set status

Communicate presence

Many other extensions

Google Talk

Encrypted (XMPP over TLS)


Supports integration with any service provider that uses the XMPP protocol

Google Talk service is hosted at talk.google.com on port 5222

Authentication through SASL PLAIN

Real-time is instantly refreshing



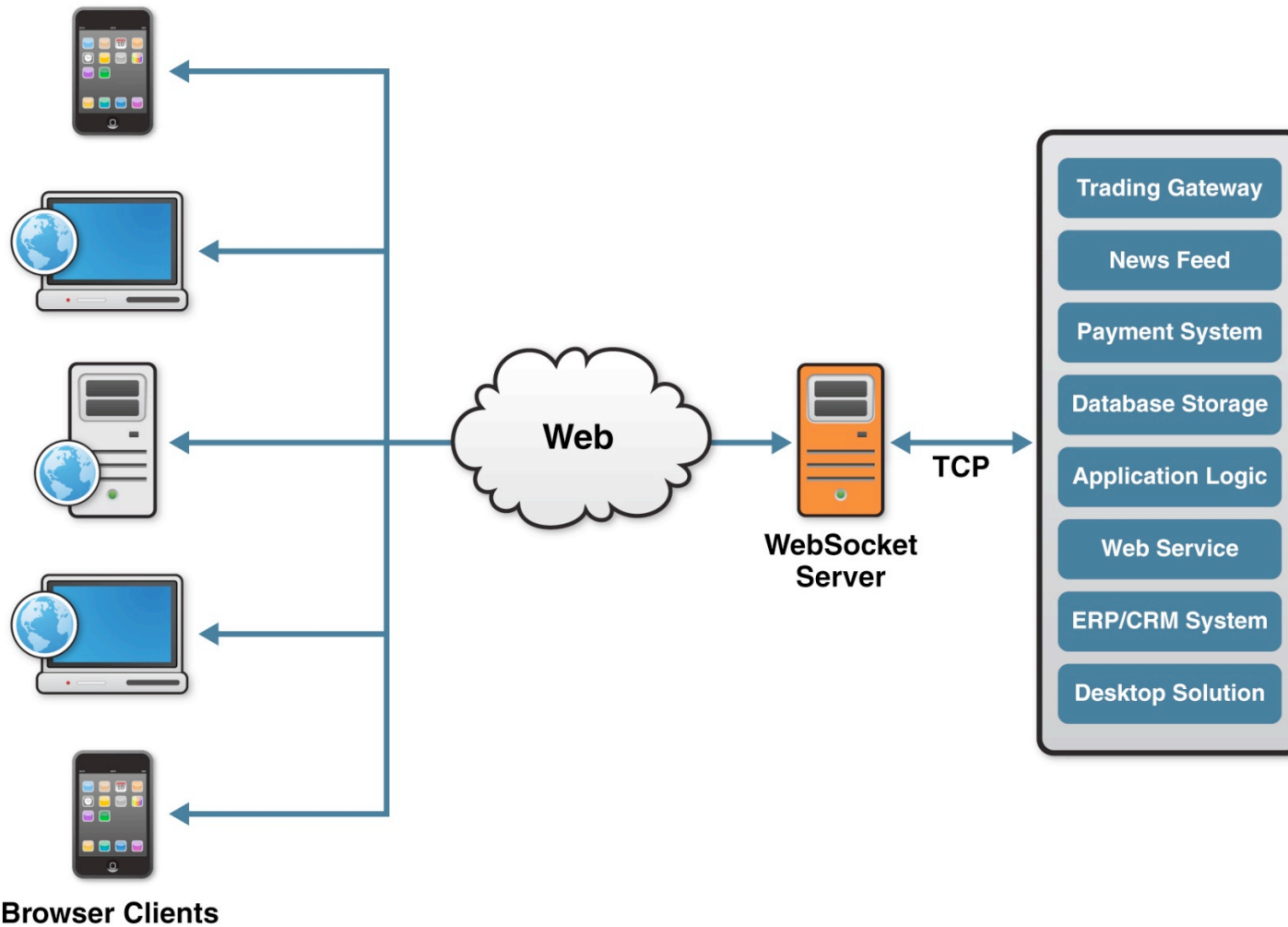
Hello Palo Alto | Thursday July 29 | 10:11:43 AM | 

Google Talk Account

Password

About

WebSocket Architecture

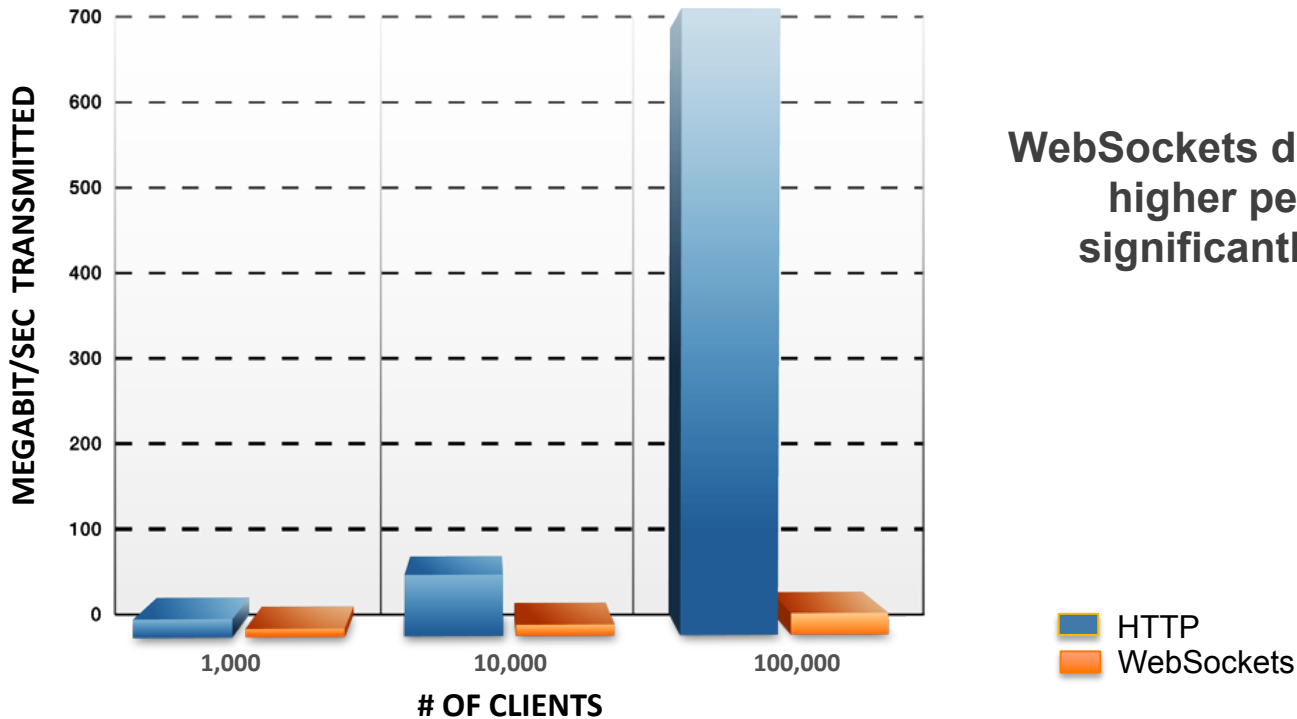


Overheard...

"Reducing kilobytes of data to 2 bytes...and reducing latency from 150ms to 50ms is far more than marginal. In fact, these two factors alone are enough to make WebSocket seriously interesting to Google."

—Ian Hickson (Google, HTML5 spec lead)

HTTP vs. WebSockets

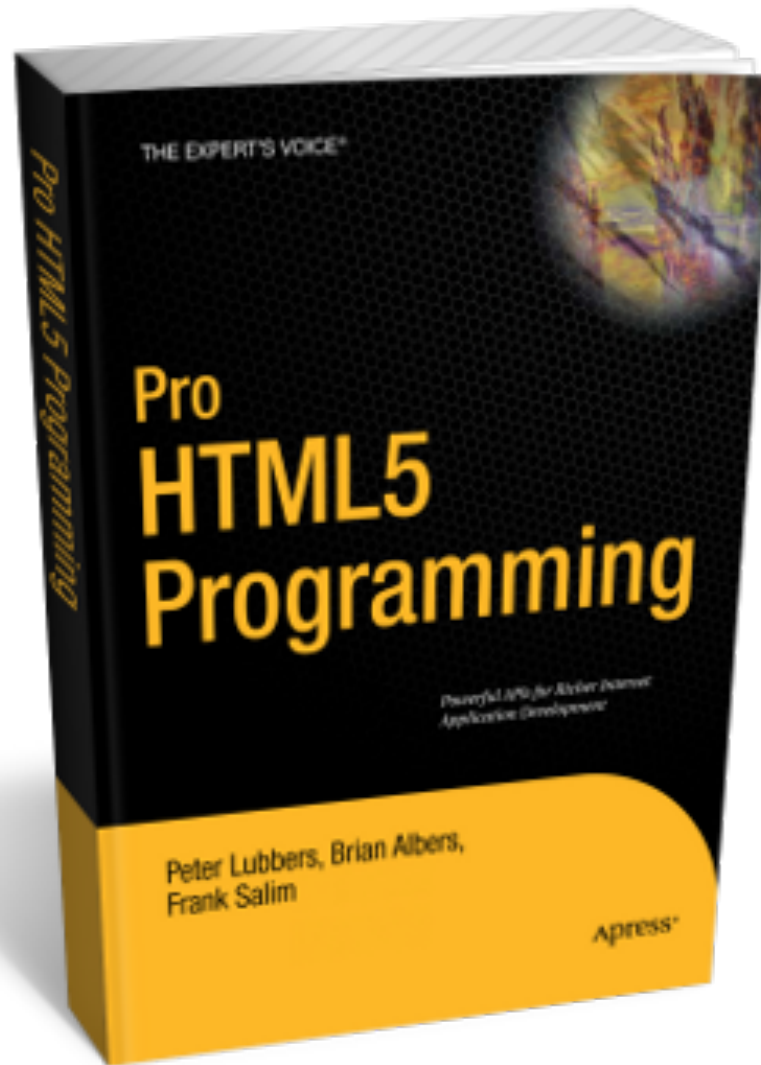


WebSockets delivers substantially higher performance with significantly less bandwidth

	HTTP traffic*	WebSockets Traffic*
Google	788 bytes, plus 1 byte	1 byte, plus 1 byte
Yahoo	1737 bytes, plus 1 byte	1 byte, plus 1 byte

* Header information for each character entered into search bar

Book Promo





KAAZING

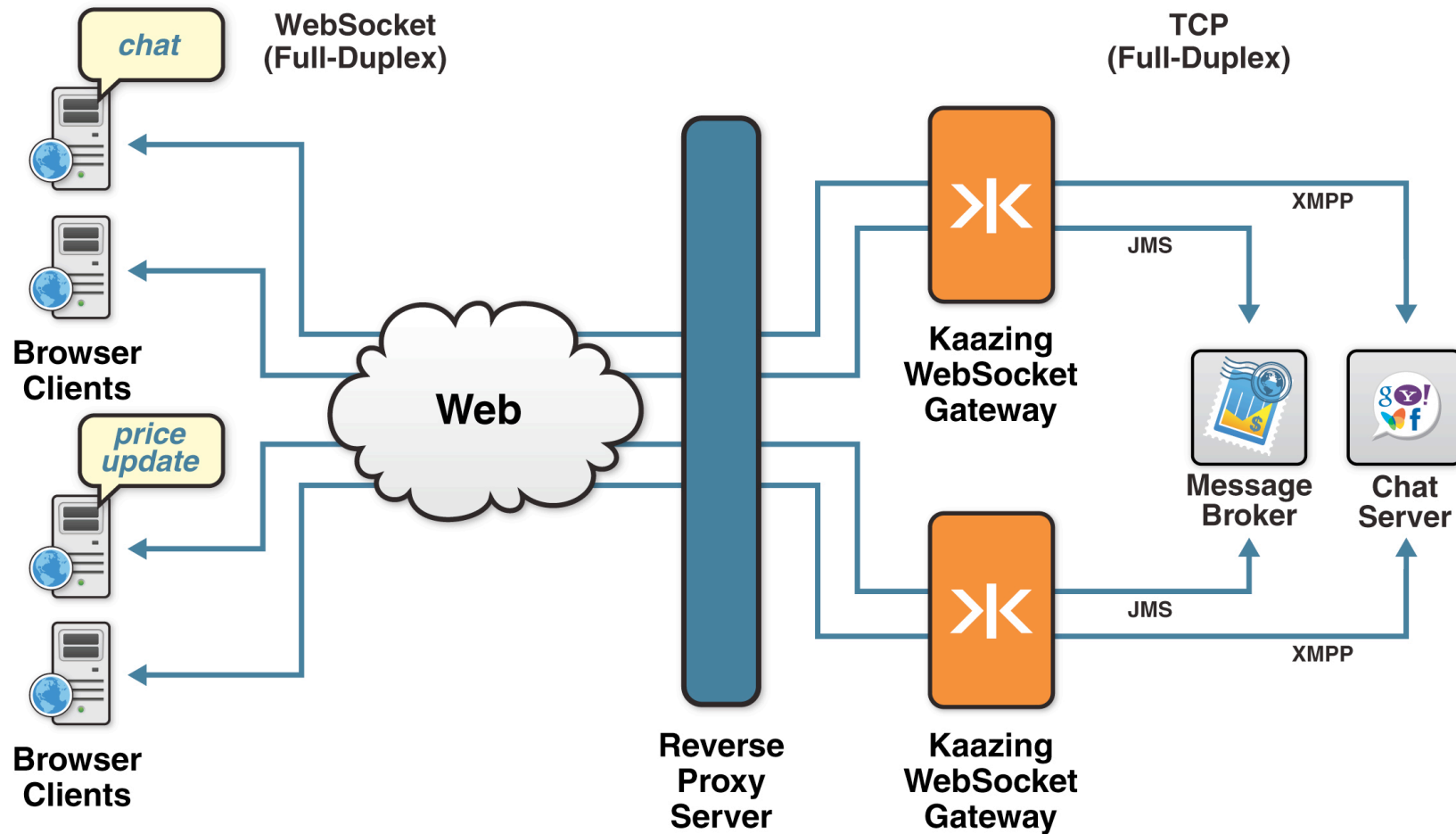
Copyright © 2010 Kaazing Corporation, All rights reserved.

All materials, including labs and other handouts are property of Kaazing Corporation. Except when expressly permitted by Kaazing Corporation, you may not copy, reproduce, publish, or display any part of this training material, in any form, or by any means.

Server Support

- Kaazing WebSocket Gateway
- Apache mod_pywebsocket
- Jetty
- phpwebsockets
- web-socket-ruby
- Yaws (Erlang)
- And more

WebSocket Architecture



Connection Offloading

