HTML5 WebSockets - the Web Communication revolution, making the impossible, possible

Brad Drysdale

Picasso

State of Scala
Venkat Subramaniam

Matejko + Canaletto
Don't code - create software!
Paweł Lipiński

Malczewski + Chelmonski
Java Boilerplate Busters
Hamlet D’Arcy

Main sponsor

LUXOFT
Engineering Business Performance
HTML5 WebSockets
The Web Communication Revolution

Brad Drysdale
Director of Technology - Kaazing
Birth of a new idea...

What do WebSocket and model trains have in common?
Client-Server Architecture

Java Client

Firewall (Port 1234)

Internet

Firewall (Port 5678)

Java Messaging Service (JMS or Stomp)

Customer

Service
HTTP Is Not Full Duplex
Half-Duplex Architecture

HTTP (half-duplex)
- Custom web client
- Custom web app
- Long-polling

Custom web client
- Custom web app
- User-initiated refresh

Custom web client
- Custom web app
- Nailed-up request

Custom web client
- Custom web app
- Timed refresh

TCP (Full-Duplex)
- RSS Client
  - Client-specific authentication & recovery
- EJB Client
  - Client-specific authentication & recovery
- JMS Client
  - Client-specific authentication & recovery
- XMPP Client
  - Client-specific authentication & recovery

Java EE Container

Server-side logic
- Transformation logic
- Connection management

Management

News Feed
- Custom-TCP

Database
- RMI-TCP

Messaging Broker
- JMS-TCP

Chat Server
- XMPP-TCP
The Legacy Web Stack

- Designed to serve static documents
  - HTTP
  - Half duplex communication
- High latency
- Bandwidth intensive
  - HTTP header traffic approx. 800 to 2000 bytes overhead per request/response
- Complex architecture
  - Not changed since the 90’s
  - Plug-ins
  - Polling / long polling
  - Legacy application servers
- Expensive to “Webscale” applications
Hack the Web for Real-Time

- Ajax applications use various “hacks” to simulate real-time communication
  - Polling - HTTP requests at regular intervals and immediately receives a response
  - Long Polling - HTTP request is kept open by the server for a set period
  - Streaming - More efficient, but not complex to implement and unreliable
- Excessive HTTP header traffic, significant overhead to each request response
Hack the Web for Real-Time

Polling

Long-Polling

Streaming

Request Response Overhead

Google Instant search single key press = 649 Bytes
Yahoo single character search = 1432 Bytes
HTTP Request Headers

GET /PollingStock//PollingStock HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv: 1.9.1.5) Gecko/20091102 Firefox/3.5.5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8
Accept-Language: en-us
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://localhost:8080/PollingStock/
Cookie: showInheritedConstant=false; showInheritedProtectedConstant=false; showInheritedProperty=false; showInheritedProtectedProperty=false; showInheritedMethod=false; showInheritedProtectedMethod=false; showInheritedEvent=false; showInheritedStyle=false; showInheritedEffect=false;
HTTP Response Headers

- Total (unnecessary) HTTP request and response header information overhead: 871 bytes (example)
- Overhead can be as much as 2000 bytes

HTTP/1.x 200 OK
X-Powered-By: Servlet/2.5
Server: Sun Java System Application Server 9.1_02
Content-Type: text/html;charset=UTF-8
Content-Length: 321
Date: Sat, 07 Nov 2009 00:32:46 GMT
HTTP Header Traffic Analysis

- Example network throughput for HTTP request and response headers associated with polling
  - **Use case A**: 1,000 clients polling every second:
    - Network throughput is \((871 \times 1,000) = 871,000\) bytes = \(6,968,000\) bits per second (~6.6 Mbps)
  - **Use case B**: 10,000 clients polling every second:
    - Network throughput is \((871 \times 10,000) = 8,710,000\) bytes = \(69,680,000\) bits per second (~66 Mbps)
  - **Use case C**: 100,000 clients polling every second:
    - Network throughput is \((871 \times 100,000) = 87,100,000\) bytes = \(696,800,000\) bits per second (~665 Mbps)
About Ajax and Comet

• Great toilet cleaners…

• Ajax (Asynchronous JavaScript and XML) is used to build highly interactive Web apps
  • Content can change without loading the entire page
  • User-perceived low latency

• "Real-time" often achieved through polling and long-polling

• Comet lack of a standard implementation

• Comet adds lots of complexity
Comet Problems

- Your RIA client application
- Custom code to simulate a real-time 2-way connection
- Silverlight or Flash plug-in
- Browser

Lots to build

Costly server resources devoted to translating LAN protocol to HTTP

Messy, slow, error prone HTTP (Long polling, etc.)

Can’t manage the actual client—end user and data source aren’t really connected

Web

LAN

HTTP Server

App Server

JMS Client

Message Broker
Desktop vs. Browser

• Desktop Networking
  • Full-duplex bidirectional TCP sockets
  • Access any server on the network

• Browser Networking
  • Half-duplex HTTP request-response
  • HTTP polling, long polling fraught with problems
  • Lots of latency, lots of bandwidth, lots of server-side resources
  • Bespoke solutions became very complex over time
Complexity does not scale
The Web gets a new Superhero
Enter HTML5 WebSocket!
HTML5 WebSocket

- HTML5 is the next set of W3C HTML standards backed by Google, Apple, Mozilla, Opera, Microsoft, Cisco, etc.
- Consists of W3C API and IETF Protocol.
- WebSockets provides a full-duplex, single socket over the Web.
- Traverses firewalls, proxies, and routers seamlessly.
- Share port with existing HTTP content.
HTML5 WebSocket Schemes

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

- WebSocket Secure
  
  `wss://www.websocket.org/encrypted-text`
// 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://www.WebSocket.org");

// Associate listeners
mySocket.onopen = function(evt) {
    alert("Connection open...");
};

mySocket.onmessage = function(evt) {
    alert("Received message: " + evt.data);
};

mySocket.onclose = function(evt) {
    alert("Connection closed...");
};
// Sending data
mySocket.send("WebSocket Rocks!");

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

GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBsb2NvdXQgY29udHJvbC
Sec-WebSocket-Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 5

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYUMLzq3GjDDmD0=

WebSocket Traffic

- Connection established by upgrading from HTTP protocol to WebSocket protocol using the same TCP connection
- Once upgraded, WebSocket data frames can be sent back and forth between client and server in full-duplex mode
- Frames can be sent full-duplex, in both directions at the same time
  Each frame of data:
    - Starts with a 0x00 byte and ends with a 0xFF byte
    - Contains UTF-8 data in between:
      \x00Hello, WebSocket\0xff

There is no defined maximum size, but JavaScript does not allow >4GB of data
WebSocket Revolution

• Dramatic reduction in unnecessary network traffic and latency
  • WebSocket requires only single byte framing
    • 500:1 or even 1000:1 bandwidth reduction
  • WebSocket does not necessitate new TCP connections for each or group of messages
    • Faster response, even more so with TLS/SSL

• Vastly simplified, more straight through architecture
Polling Introduces Extra Latency!
HTTP versus WebSockets

Example: Entering a character in a search field with auto suggestion

<table>
<thead>
<tr>
<th></th>
<th>HTTP traffic*</th>
<th>WebSocket Traffic*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>788 bytes, plus 1 byte</td>
<td>1 byte, plus 1 byte</td>
</tr>
<tr>
<td>Yahoo</td>
<td>1737 bytes, plus 1 byte</td>
<td>1 byte, plus 1 byte</td>
</tr>
</tbody>
</table>

* Header information for each character entered into search bar

**WebSockets reduces bandwidth overhead up to 1000x**
Polling vs. Web Sockets

The graph compares the performance of Polling and Web Sockets in three different use cases:

- **Use Case A**: 6,968,000 bits per second with 16,000 connections using Polling versus 1,600,000 bits per second for 160,000 connections using Web Sockets.
- **Use Case B**: 69,680,000 bits per second with 16,000 connections using Polling versus 696,800,000 bits per second for 160,000 connections using Web Sockets.
- **Use Case C**: 696,800,000 bits per second with 16,000 connections using Polling versus 696,800,000 bits per second for 160,000 connections using Web Sockets.
“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)
The New Web Stack

- Designed for full-duplex high performance transactional Web
  - HTTP & HTML5 WebSocket
  - Full duplex communication
- Lower latency
- Reduced bandwidth
- Simplified architecture
- Massive scalability
Browser and Server Support

- **Browsers:**
  - Chrome
  - Safari
  - Firefox 4
  - Coming in Opera and “on the list” for IE

- **Servers:**
  - Kaazing WebSocket Gateway
  - Apache mod_pywebsocket
  - phpwebsockets
  - web-socket-ruby
  - More…
WebSocket Emulation

• Kaazing WebSocket Gateway
  • http://www.kaazing.com/download
  • Makes WebSocket work in all browsers today (including I.E. 6)

• Flash WebSocket implementation
  • http://github.com/gimite/web-socket-js
  • Requires opening port on the server's firewall
Discovering WebSockets
Got WebSocket. Now What?

- Major upgrade for web traffic, use it!
  - Simple text
  - JSON
  - Existing protocols (why reinvent the wheel?)
    - Text Protocols: XMPP, STOMP
    - Binary Protocols: AMQP, IRC, Pub/Sub

- Build high performance, scalable messaging for web apps
- Extend the reach of *any* TCP-based protocol you want, all the web to the browser
- The browser is a true client of that protocol – powerful paradigm shift
Example: Financial Apps

<table>
<thead>
<tr>
<th>COMPANY</th>
<th>SYMBOL</th>
<th>PRICE</th>
<th>CHANGE</th>
<th>SPARKLINE</th>
<th>OPEN</th>
<th>LOW</th>
<th>HIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>THE WALT DISNEY COMPANY</td>
<td>DIS</td>
<td>27.45</td>
<td>.36</td>
<td></td>
<td>27.09</td>
<td>24.34</td>
<td>29.16</td>
</tr>
<tr>
<td>GARMIN LTD.</td>
<td>GRMN</td>
<td>34.30</td>
<td>-.49</td>
<td></td>
<td>34.79</td>
<td>31.32</td>
<td>38.26</td>
</tr>
<tr>
<td>SANDISK CORPORATION</td>
<td>SNMX</td>
<td>19.62</td>
<td>-1.42</td>
<td></td>
<td>20.24</td>
<td>10.22</td>
<td>22.26</td>
</tr>
<tr>
<td>GOODRICH CORPORATION</td>
<td>GR</td>
<td>51.50</td>
<td>-.44</td>
<td></td>
<td>52.14</td>
<td>47.11</td>
<td>57.57</td>
</tr>
<tr>
<td>NVIDIA CORPORATION</td>
<td>NVDA</td>
<td>13.46</td>
<td>-.39</td>
<td></td>
<td>13.65</td>
<td>12.47</td>
<td>15.23</td>
</tr>
<tr>
<td>CHEVRON CORPORATION</td>
<td>CVX</td>
<td>67.69</td>
<td>-.41</td>
<td></td>
<td>60.30</td>
<td>61.40</td>
<td>75.12</td>
</tr>
<tr>
<td>THE ALLSTATE CORPORATION</td>
<td>ALL</td>
<td>32.62</td>
<td>1.61</td>
<td></td>
<td>31.62</td>
<td>27.92</td>
<td>34.11</td>
</tr>
<tr>
<td>EXXON MOBIL CORPORATION</td>
<td>XOM</td>
<td>67.73</td>
<td>1.21</td>
<td></td>
<td>66.12</td>
<td>59.87</td>
<td>73.17</td>
</tr>
<tr>
<td>H E T L I F E  I N C .</td>
<td>MET</td>
<td>35.64</td>
<td>-.69</td>
<td></td>
<td>35.73</td>
<td>32.16</td>
<td>39.38</td>
</tr>
<tr>
<td>J. C. PENNEY COMPANY INC.</td>
<td>JCP</td>
<td>32.66</td>
<td>-.29</td>
<td></td>
<td>32.15</td>
<td>29.66</td>
<td>36.24</td>
</tr>
<tr>
<td>OFFICE MAX INCORPORATED</td>
<td>CMX</td>
<td>12.22</td>
<td>-.19</td>
<td></td>
<td>12.41</td>
<td>11.17</td>
<td>13.65</td>
</tr>
<tr>
<td>AETNA INC.</td>
<td>AET</td>
<td>27.30</td>
<td>.43</td>
<td></td>
<td>26.87</td>
<td>24.11</td>
<td>29.36</td>
</tr>
<tr>
<td>CONOCOPHILLIPS</td>
<td>COP</td>
<td>43.59</td>
<td>-.63</td>
<td></td>
<td>46.62</td>
<td>41.96</td>
<td>51.27</td>
</tr>
<tr>
<td>UNITEDHEALTH GROUP INC.</td>
<td>UNH</td>
<td>24.43</td>
<td>.67</td>
<td></td>
<td>24.36</td>
<td>21.93</td>
<td>26.79</td>
</tr>
</tbody>
</table>
### Example: Financial Apps

<table>
<thead>
<tr>
<th>Markets</th>
<th>Daily Change</th>
<th>Sell</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK 100 - Daily Rolling Future</td>
<td>+26.2</td>
<td>5995.0</td>
<td>5996.0</td>
</tr>
<tr>
<td>UK 100 - Daily Rolling Cash</td>
<td>+30.4</td>
<td>6035.5</td>
<td>6037.5</td>
</tr>
<tr>
<td>WALL STREET - Daily Rolling Future</td>
<td>+55</td>
<td>12376</td>
<td>12381</td>
</tr>
<tr>
<td>S&amp;P 500 - Daily Rolling Future (per 0.1)</td>
<td>+6.3</td>
<td>1333.0</td>
<td>1333.3</td>
</tr>
<tr>
<td>DAX - Daily Rolling Future</td>
<td>+44.5</td>
<td>7241.5</td>
<td>7242.5</td>
</tr>
<tr>
<td>EURUSD - Rolling Spot</td>
<td>+0.0070</td>
<td>1.4302</td>
<td>1.4303</td>
</tr>
<tr>
<td>GBPUSD - Rolling Spot</td>
<td>-0.0015</td>
<td>1.6289</td>
<td>1.6291</td>
</tr>
<tr>
<td>AUDUSD - Rolling Spot</td>
<td>+0.0044</td>
<td>1.0379</td>
<td>1.0380</td>
</tr>
<tr>
<td>GOLD Rolling Spot (per 0.1)</td>
<td>+3.7</td>
<td>1458.0</td>
<td>1458.4</td>
</tr>
<tr>
<td>BRENT CRUDE - Daily Rolling Future</td>
<td>+97</td>
<td>12253</td>
<td>12256</td>
</tr>
</tbody>
</table>

**Market Commentary**

- FTSE100 set to open +16 6023, Dow Future +11 12338, £/$/ 1.6342 +49, E/$ 1.4268 +47, Gold 1453 +2.5, Brent May 12190 -35

  3 hours ago · reply · retweet · favorite

China shrugs off interest-rate hike but analysts remain deeply divided on further scope for tightening as inflationary pressures persist.

3 hours ago · reply · retweet · favorite

- Dow cash closed 12393 -6, FTSE100 set to open 6017 +10. £/$/ 1.6285 +161, E/$ 1.4223
WebSocket-Based Quake II

http://code.google.com/p/quake2-gwt-port
Example: Earth Control Game

http://apps.facebook.com/earthcontrol
Example: Sketchpad

http://mrdoob.com/projects/multiuserpad/
WebSocket in Action

Real-time is instantly refreshing

WebSocket in Action

New York Times

WASHINGTON — The Senate on Wednesday rejected an
work on a sweeping financial regulatory bill as two key D
sufficiently tighten rules on Wall Street.

Non-Crew in Cockpit of Doomed Polish Flight

WICKS — Aviation authorities offered new details of the plane crash that killed Poland’s
president and dozens of other top officials last month, including the revelation that two or more
passengers were in the cockpit shortly before the pilots tried to land the aircraft in dense fog.

British Coalition Offers Reform Plan

LONDON — Defying those who said it might be paralyzed by internal divisions, Britain’s new
coalition government of Conservatives and Liberal Democrats on Wednesday unveiled the most
ambitious plan in decades for expanding the highly centralized and often intrusive way the country
is governed.

State Dinner Chef Tweets About Day of Creation

WASHINGTON (AP) — Got my fingers crossed there are no hitches

In Ambush, a Glimpse of a Long Afghan Summer

BAGRAM, Afghanistan — Minutes after surviving the first ambush, Col. John M. Boone, a Marine
sniper, called it a radio. We have got a civilian here who got shot in his leg, he said.

EIKOS PARTNERS

LinkedIn

Monitor

Server Log

Monitor for Data Center

Higher bar = longer ratio

Try our full featured web-based chat client.

Log on now »
Possibilities…

- Low latency Financial and Trading apps
- Online in-game betting and live auctions
- Social networking
- Performance and monitoring dashboards
- RFID and GPS Tracking
- Sports and news broadcasting applications
- Supply chain and inventory management
- Smart meters
- Next generation web application of your choice!
Your cool [HTML5 WebSocket] App Here...

http://iseeaday.blogspot.com/
Mobile Aggregation

Reduce dependencies on portal servers and portal farms.
Unconstrained Web
- Financial Services
- Transportation and Logistics
- Telecommunications
- Utilities
- Social Networking

Cloud Computing
- Server to Server communication
- Distributed Internet applications over any TCP protocol
- Services on demand

3G & 4G Mobile Networking
- Significant bandwidth reduction
- New Service Delivery
- New Customer Experience
Arquillian: Real Java enterprise testing
Dan Allen

Picasso
Fractal TDD: Using tests to drive system design
Steve Freeman

Matejko + Canaletto
Pro Groovy
Dierk König

Malczewski + Chelmonski
“Same Data, Any API”, making sure your application scales
Guy Korland
KAAZING™