KAAZING

KAAZING

Think BIG, Really BIG!

Building next-generation Web Apps with WebSocket and HTML5

Matthias Weßendorf

KAAZING

- Kaazing Corporation (http://kaazing.com)
  - Kaazing WebSocket Gateway:
  - High-performance Web communication platform
    - Advanced WebSocket Server
      - Several Client SDKs
        - .NET/Silverlight, Java, GWT, Flash, JavaScript
      - Browser support:
        - Works With Any Browser! (➔ http://kaazing.me)
      - Support for lot's of high-level protocols:
        - JMS, AMQP, XMPP, etc

- Apache Software Foundation (Member, Committer, PMC)
- Speaker, Author, Blogger:
  http://matthiaswessendorf.wordpress.com
- Twitter: @mwessendorf

Agenda

KAAZING

- HTML 5
  - Overview
- Real Time (Web)
  - Introduction
- WebSocket
  - Introduction
  - Handshake and Client API
- Remix!
  - WebSocket and other (HTML5) APIs

Agenda

KAAZING

- ***HTML 5***
  - Overview
- Real Time (Web)
  - Introduction
- WebSocket
  - Introduction
  - Handshake and Client API
- Remix!
  - WebSocket and other (HTML5) APIs

HTML5 – Facts!

KAAZING

- HTML5 is brand new
  - Indeed, it isn't finished yet!
- History
  - first draft: 1993
  - HTML 4.0.1: 1999
  - 2004: "Bringing the web to the next level"
    - Web 2.0 started around this time
    - HTML5 (by WHATWG)
  - 2006: W3C got involved – first HTML5 draft in 2008
- 2014: W3C recommendation …
- HTML5 is "done" by:
  - WHATWG
  - W3C
  - IETF


HTML5.0


JAN
California
2012
HTML5
PROTECT LAKE TAHOE

Moments in HTML…

HTML5: it's all about the user

KAAZING

- Use <header>, not *
- User is king:
  - id=myDiv VS id="myDiv" VS ID=myDiv
- Simplification
  - Simple is better. Simplify where possible
  - Doctype: <!DOCTYPE html>
- Accessibility
- No Browser Plugins…
  - Plugins can be blocked …
    - Check you phone…!

HTML5 – A few APIs*

KAAZING

- Canvas
- Geolocation
- Server-Sent events
- SVG
- *WebSocket*
- WebWorker
- XHR Level2
- Web Storage
- …

Canvas – Was kann das?

KAAZING

- Simple Demo…
- Advanced Demo:

(thanks to Gerrit Grunwald (@hansolo_))

Geolocation + Google Maps

KAAZING

- Mashup = Geolocation + Google Maps

Agenda

KAAZING

- HTML 5
  - Overview
- ***Real Time (Web)***
  - Introduction
- WebSocket
  - Introduction
  - Handshake and Client API
- Remix!
  - WebSocket and other (HTML5) APIs

Real Time (Web)

KAAZING

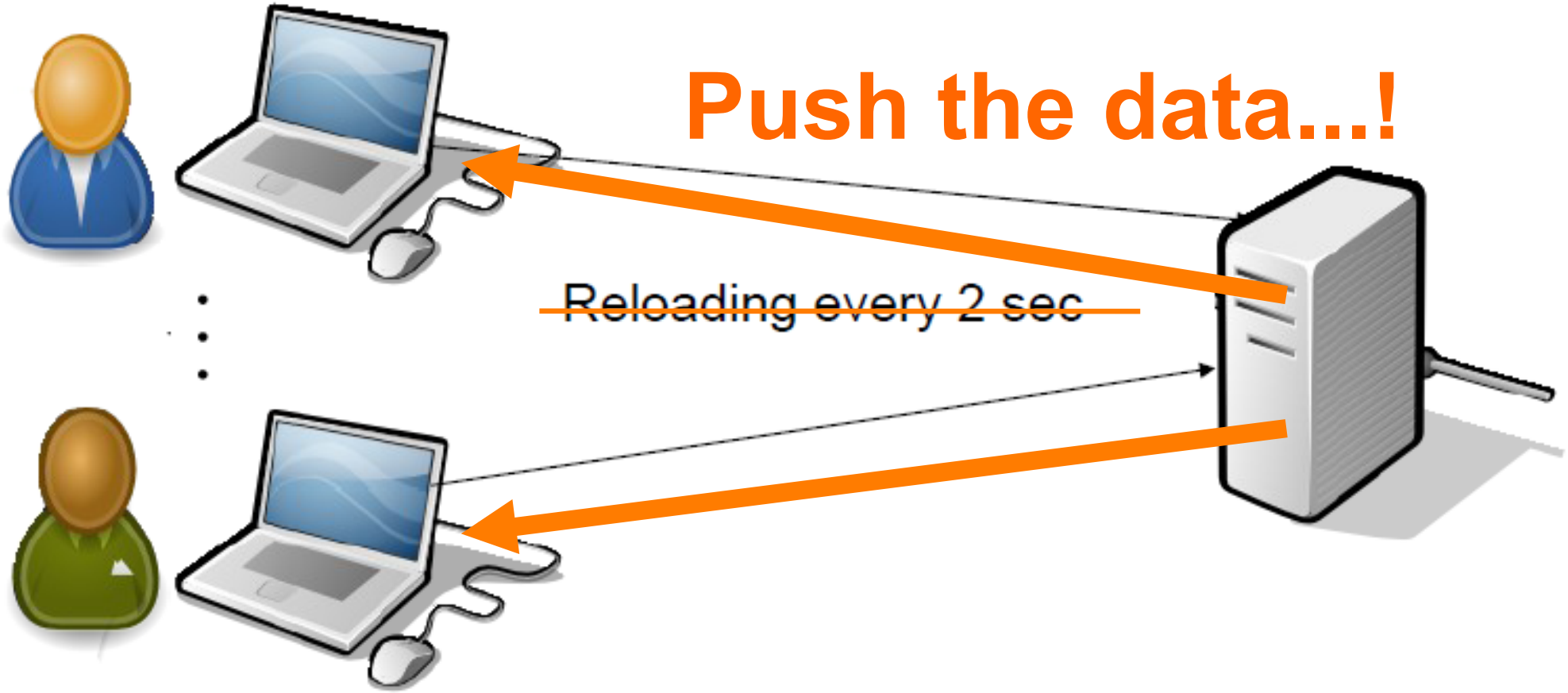
- Every application has a hook…
  - Collaborative tooling
    - Share / edit documents
    - chat
  - Gaming (web based games)
  - Streaming data (➔ football / sports)
  - Trading Systems (➔ auctions)
  - Financial apps (➔ market data)
  - Social Network app
  - (Web-based) Support Systems
  - Monitoring (➔ server console)

What is your use-case?

Drop the built-in latency

KAAZING

- Collaborative Tooling:



Think different …

KAAZING

Think different!

KAAZING

Challenge…

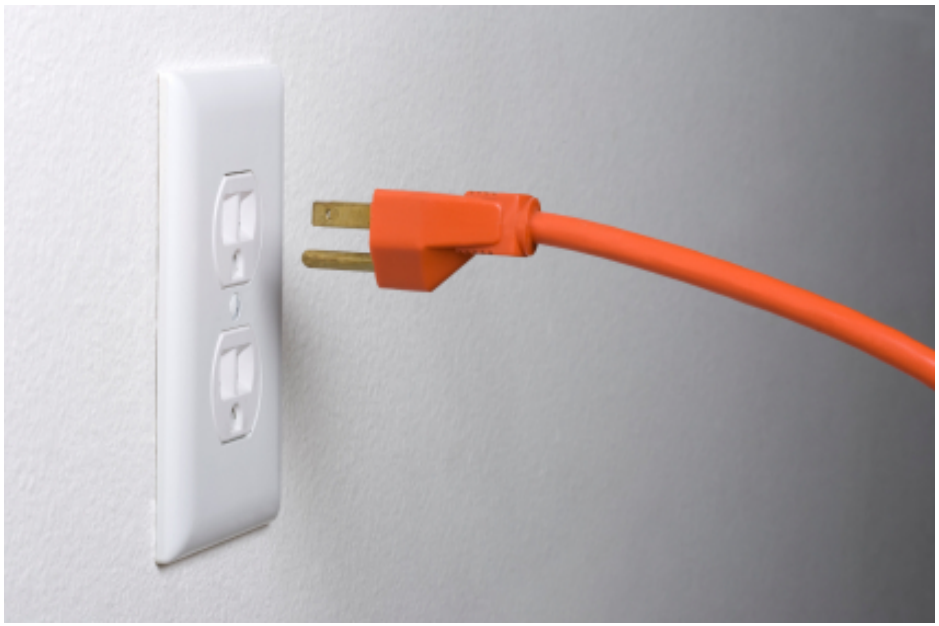
KAAZING

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

Agenda

KAAZING

- Enter HTML5 WebSocket!



WebSocket …

… is a bi-directional AND full-duplex communication STANDARD for next-generation web applications

("TCP for the web")

… and HTML5 APIs

KAAZING

Build the future!

WebSocket is full-duplex!

KAAZING

- Enables web pages to communicate (in full-duplex mode) with a remote host
  - Shares port with existing HTTP content (80, 443)
  - Traverses firewalls, proxies, and routers seamlessly
  - Leverages Cross-Origin Resource Sharing (CORS)
  - Work with existing Authentication and Authorization frameworks (e.g. Kerberos)

- Extend any TCP/UDP based protocol!!!
  - AMQP
  - XMPP/STOMP/JMS
  - *VNC*
  - ...

Demo: HTML5 mix!

- Canvas and WebSocket

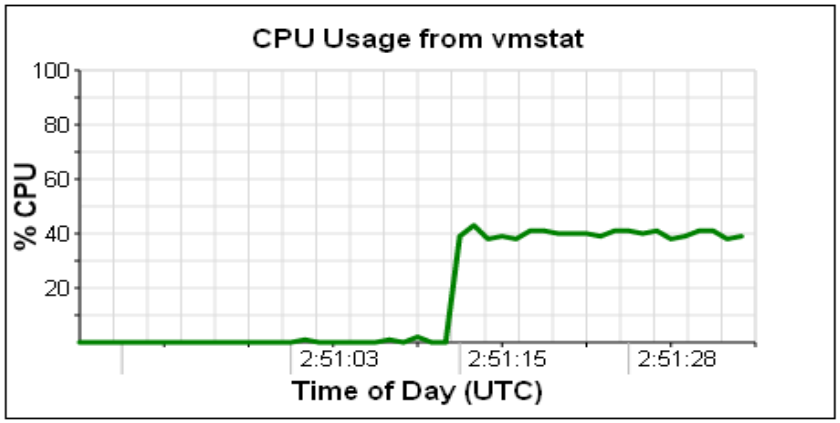
Cloud Server Monitor

Date	Time (UTC)	CPU Usage (%) (from *vmstat*)
02/10/11	2:51:29	40
02/10/11	2:51:30	41
02/10/11	2:51:31	38
02/10/11	2:51:33	39
02/10/11	2:51:34	41
02/10/11	2:51:35	41
02/10/11	2:51:36	38
02/10/11	2:51:37	39

Load Program Control

Start

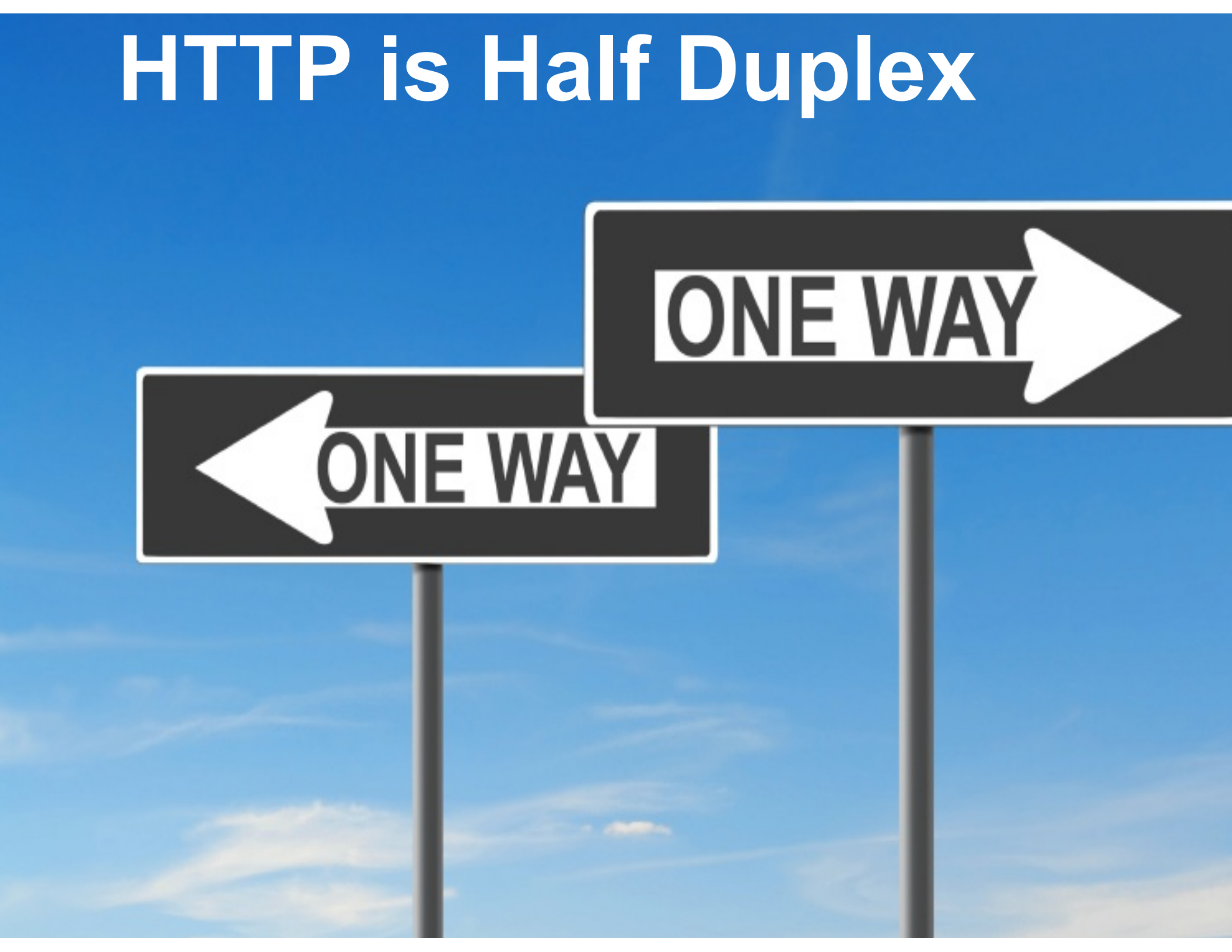
Stop



Top output

```
top - 02:51:38 up 52 days,  1:15,  0 users,  load average: 1.12, 0.27, 0.09
Tasks:  65 total,   2 running,  63 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.6%us,  0.2%sy,  0.0%ni, 96.9%id,  0.0%wa,  0.0%hi,  0.0%si,  2.3%st
Mem:   1755492k total,   810312k used,   945180k free,    92696k buffers
Swap:   917496k total,        0k used,   917496k free,   492796k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
24452 ubuntu    20   0  1540  276  224 R 37.3  0.0   0:09.37 load
    1 root      20   0  2548 1448 1104 S  0.0  0.1   0:01.78 init
    2 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      15  -5     0    0    0 S  0.0  0.0   0:01.87 ksoftirqd/0
```

HTTP is Half Duplex

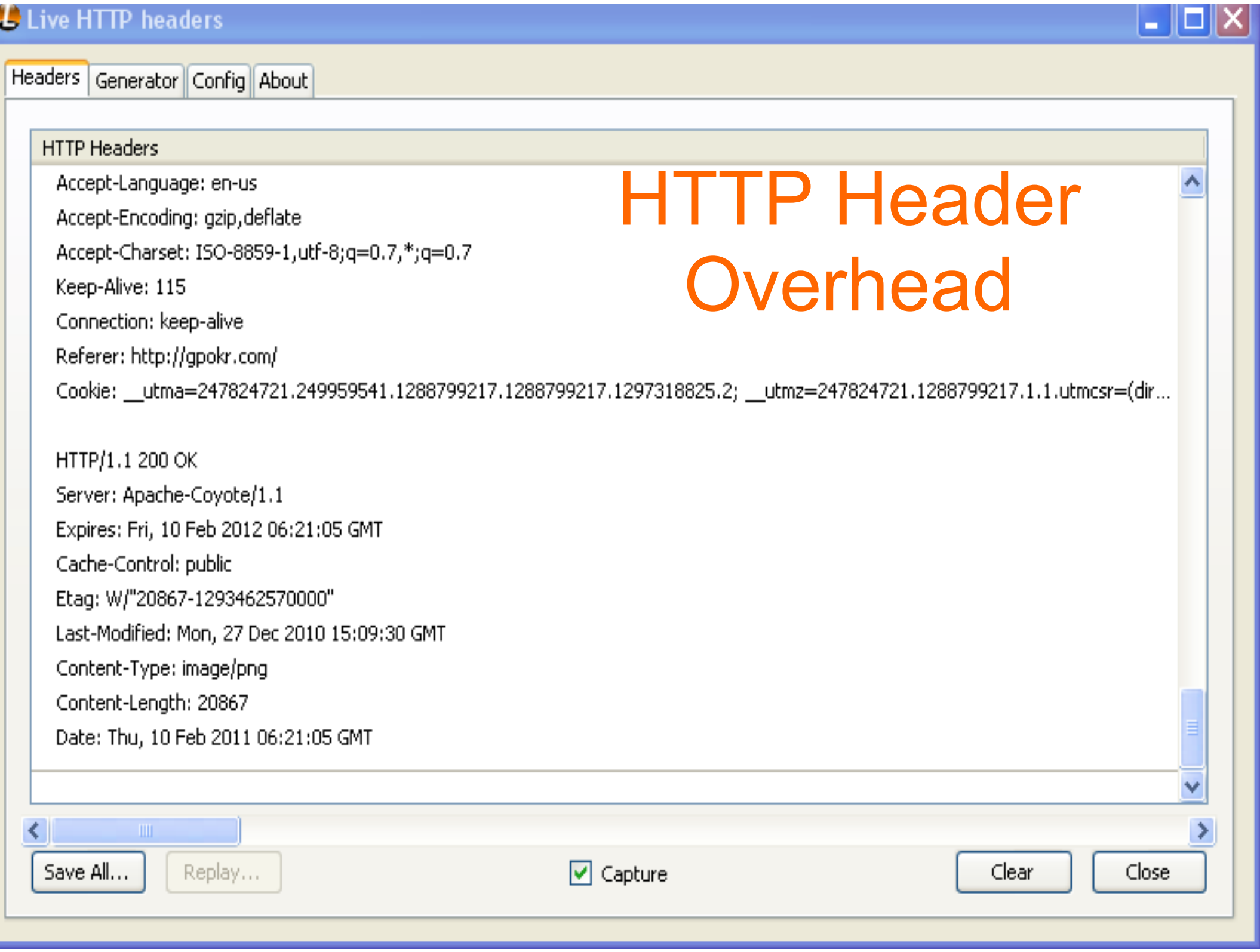


Half Duplex Communication is Primitive

Question for the Audience!

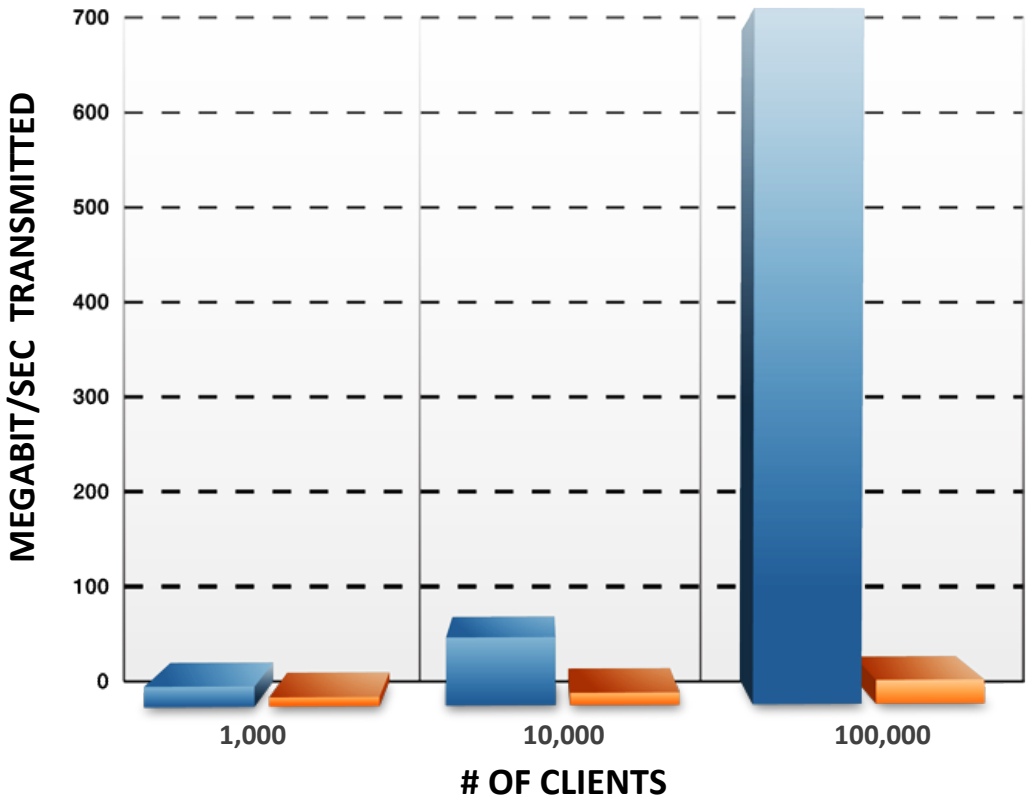
KAAZING

- What are some common names for Ajax Polling implementations?


Live HTTP headers
Headers
Generator
Config
About
HTTP Headers
Accept-Language: en-us
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://gpokr.com/
Cookie: __utma=247824721.249959541.1288799217.1288799217.1297318825.2; __utmz=247824721.1288799217.1.1.utmcsr=(dir...
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Expires: Fri, 10 Feb 2012 06:21:05 GMT
Cache-Control: public
Etag: W/"20867-1293462570000"
Last-Modified: Mon, 27 Dec 2010 15:09:30 GMT
Content-Type: image/png
Content-Length: 20867
Date: Thu, 10 Feb 2011 06:21:05 GMT
HTTP Header Overhead
Save All...
Replay...
Capture
Clear
Close

HTTP vs. WebSockets

KAAZING



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

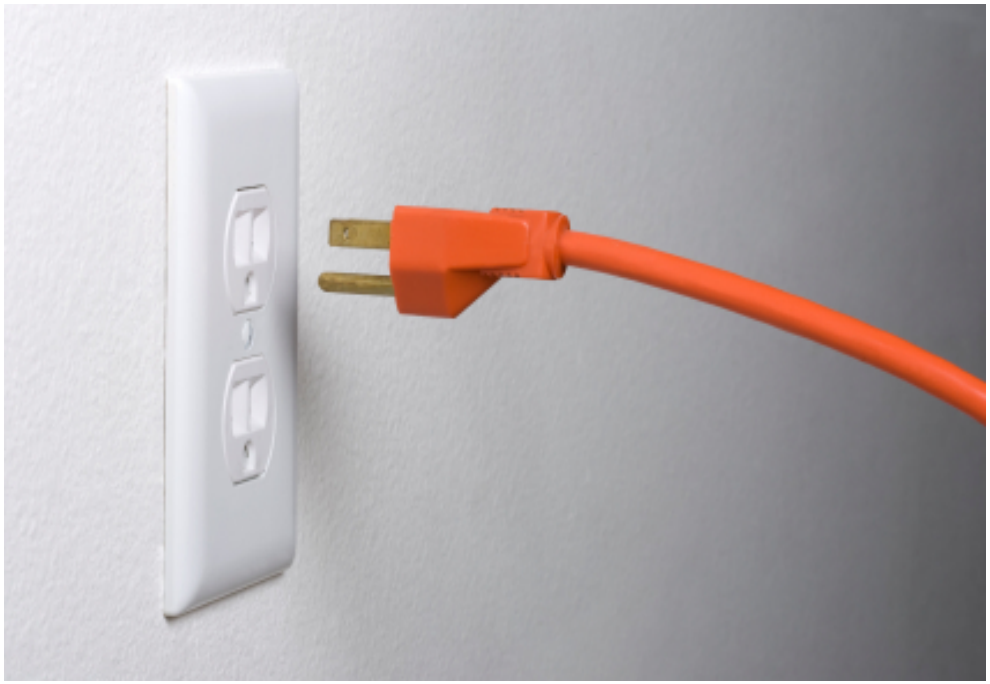
Enter WebSocket!

KAAZING

From Complexity…

(spell it with the letters C,O,M,E,T!)

… to simplicity!



WebSocket Standard

KAAZING

- Client API (W3C)
  - (simple) JavaScript API

- Network Protocol (IETF)
  - Under development
    - draft-hixie-thewebsocketprotocol-76
    - ...
    - draft-ietf-hybi-thewebsocketprotocol-14*

WS Protocol / Handshake

KAAZING

GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 17

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: superchat

WebSocket Traffic

KAAZING

- 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
  - 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: \0x00Hello, WebSocket\0xff
- There is no defined maximum size, but JavaScript does not allow >4GB of data)


Less
Overhead...
A lot less!!!
\0x00Hello, WebSocket\0xff

WebSocket Client API

KAAZING

JavaScript

```
// Create new WebSocket
var mySocket =
        new WebSocket("ws://echo.websockets.org");

// Attach listeners
mySocket.onmessage = function(evt) {

    doSomeFancyDhtml(evt.data);
};

mySocket.onopen = function(evt) {...};

mySocket.onclose = function(evt) {...};

mySocket.onerror = function(evt) {...};
```

WebSocket Client API

KAAZING

JavaScript

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


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

DEMO

KAAZING

- UDP broadcaster meets WebSocket!

WebSocket Support

KAAZING

- Browsers:
  - Firefox 7 (enabled since FF5)
  - Chrome/Chromium
  - Opera
  - Safari 5 & iOS4
  - IE 10 (preview 3, from September 13th)
- *To test support visit:* http://websocket.org
- Servers:
  - Kaazing WebSocket Gateway
  - Node.js
  - mod_pywebsocket
  - Netty
  - Resin

WebSocket Misconception

KAAZING

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

WebSocket Misconception

KAAZING

- When you build Desktop applications (client server solutions) do you develop your application using raw TCP sockets?

- Or, are you using higher level protocols and APIs, such as XMPP, JMS, JDBC?

Beyond WebSockets

KAAZING

You should think of WebSocket the same way:

- Extend client-server protocols to the Web
  - XMPP, Jabber
  - Pub/Sub (Stomp/AMQP)
  - Gaming protocols
  - Any TCP-based protocol

Using Stomp/JMS API

KAAZING

JavaScript

```
...
var stompConnectionFactory = new StompConnectionFactory(
            "ws://localhost:8000/jms");

var connectionFuture = stompConnectionFactory.createConnection
(function () {
    if (!connectionFuture.exception) {
        connection = connectionFuture.getValue();
        connection.setExceptionListener(handleException);

        session = connection.createSession(false,
Session.AUTO_ACKNOWLEDGE);
        topic = session.createTopic("/topic/destination");

        // creating some consumers!
        var consumer1 = session.createConsumer(topic);
        consumer1.setMessageListener(handleMessageCallback);

        connection.start(someCallback);
    }
....
});
```

Using Stomp/JMS API

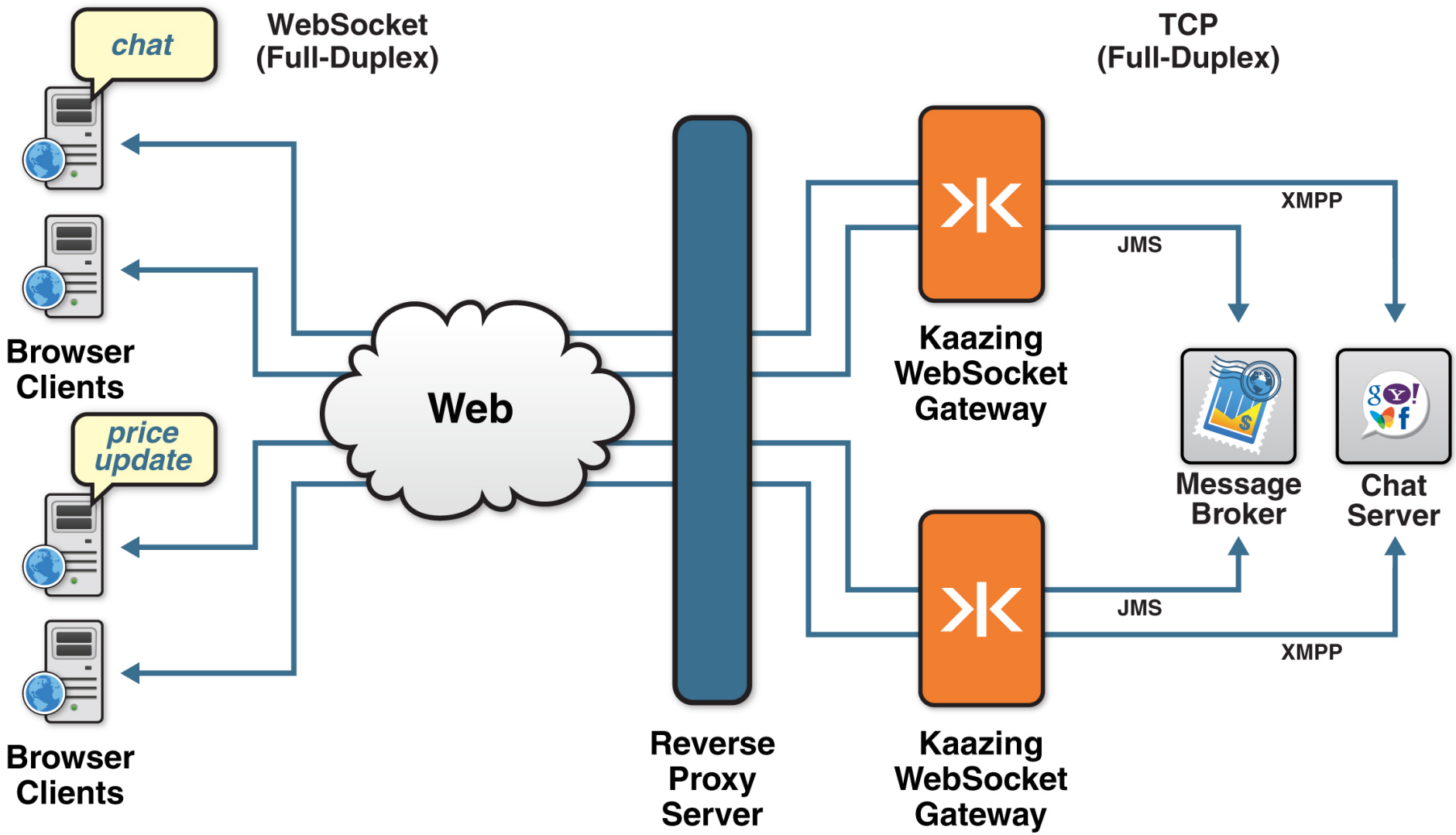
KAAZING

JavaScript

```
function handleMessageCallack(message) {

// did Apache ActiveMQ send us a JMS TextMessage?

  if (message instanceof TextMessage) {
     var body = message.getText();

     // do more stuff...

  }
}
```

WebSocket Architecture

KAAZING



Agenda

KAAZING

- HTML 5
  - Overview
- Real Time (Web)
  - Introduction
- WebSocket
  - Introduction
  - Handshake and Client API
- ***Remix!***
  - WebSocket and other (HTML5) APIs

Geolocation and WebSocket

KAAZING

- A real-time tracker application
  - Logistics
  - Sports (running)
- Using Geolocation API and WebSocket

Geolocation and WebSocket

KAAZING

JavaScript

```
// register 'update' handler
geolocation.watchPosition(updateLocation, handleError);

function updateLocation(pos) {
  // get the current position
  var lat = pos.coords.latitude;
  var lon = pos.coords.longitude;

  var data = JSON.stringify([userId, lat, long]);

  // tell the server where I am ...
  myWebSocket.send(data);
}

// onmessage handler can be used to observe other peers
```

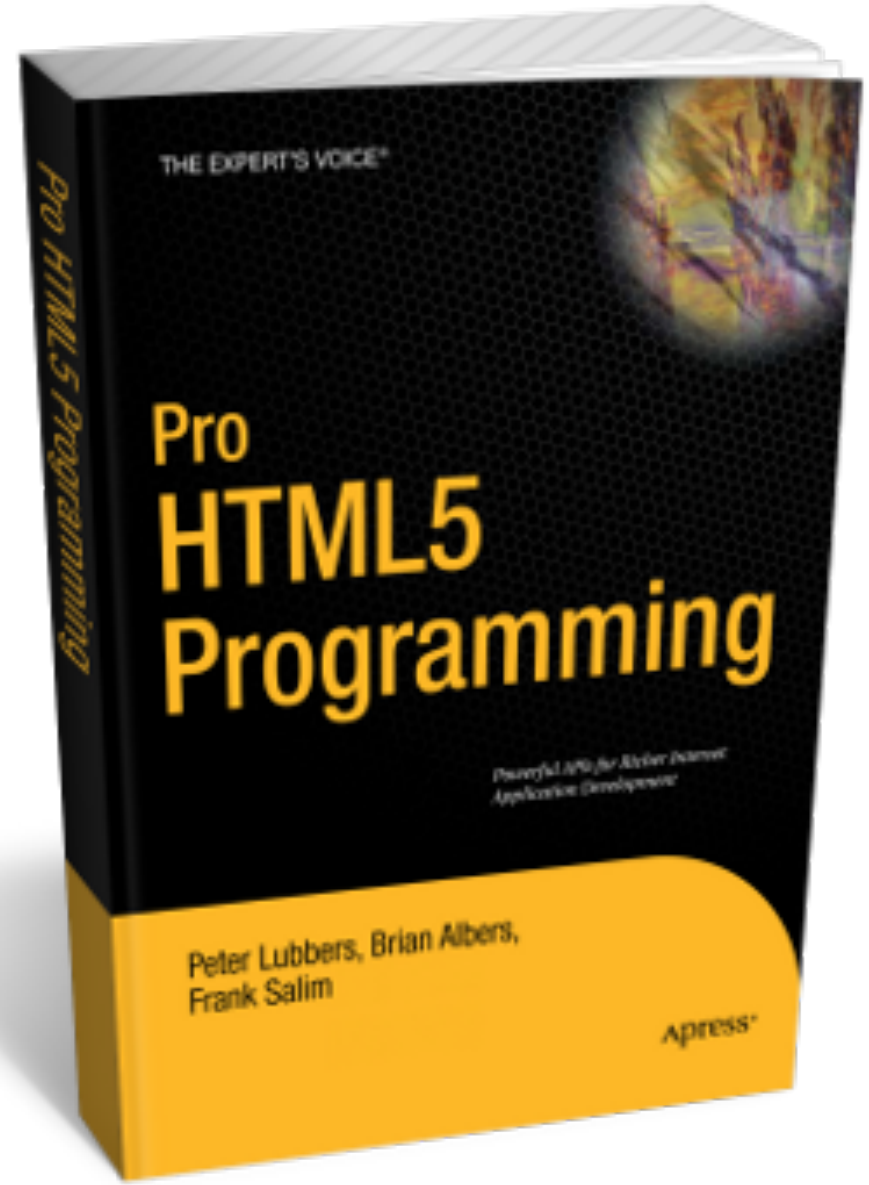
DHTML and WebSocket

KAAZING

- Stomp/JMS over WebSocket
  - Apache ActiveMQ
    (or any JMS broker, e.g. Tibco EMS, OpenMQ, ...)

- Stock Ticker:
  - HTML Table
  - DHTML

- You can use jQuery, Dojo, etc….


THE EXPERT'S VOICE
Pro HTML5 Programming
Pro
HTML5
Programming
Peter Lubbers, Brian Albers,
Frank Salim
Apress

THANK YOU!

http://kaazing.me

http://kaazing.com/download.html

KAAZING