KAAZING

HTML 5 WebSocket

The Web Communication Revolution

Speaker Bio

John Fallows

Co-Founder: Kaazing, At the Heart of the Living Web

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

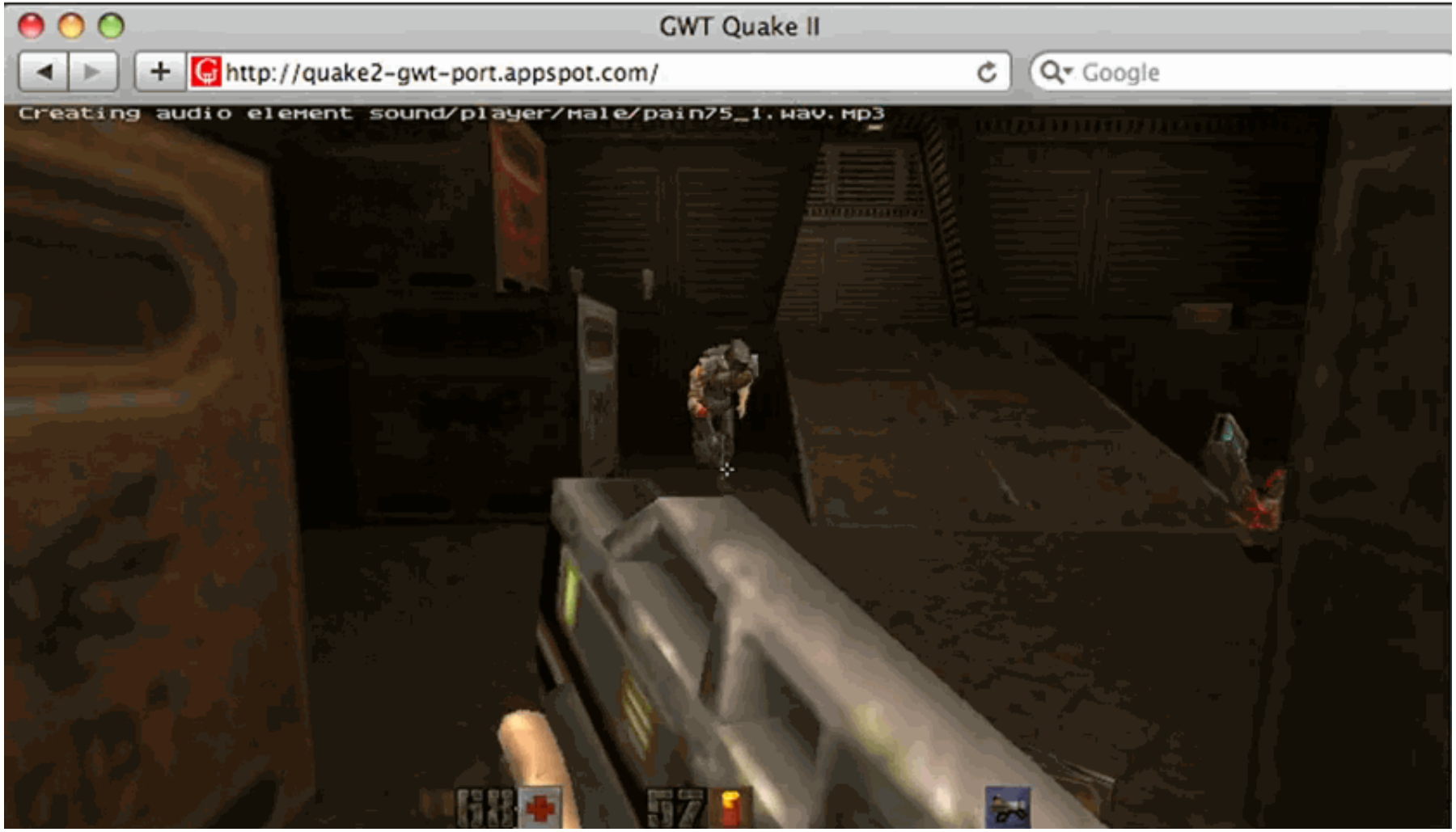
Challenge…

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

Living Web Applications

- 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
  - Instant augmented reality…

Living Web: Quake II Port



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

HTTP is not Full Duplex

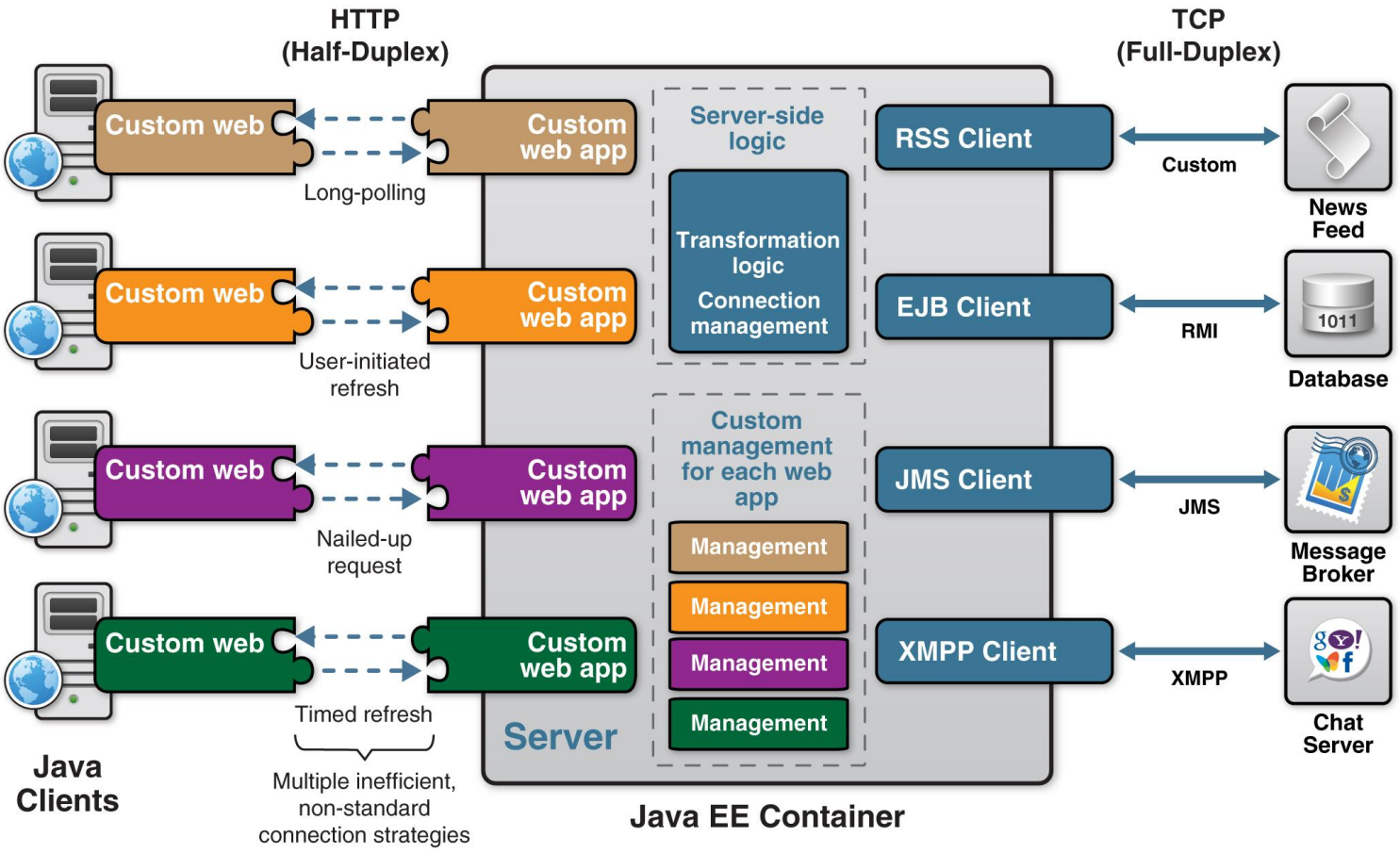
HTTP Characteristics

- 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

Legacy Web Stack

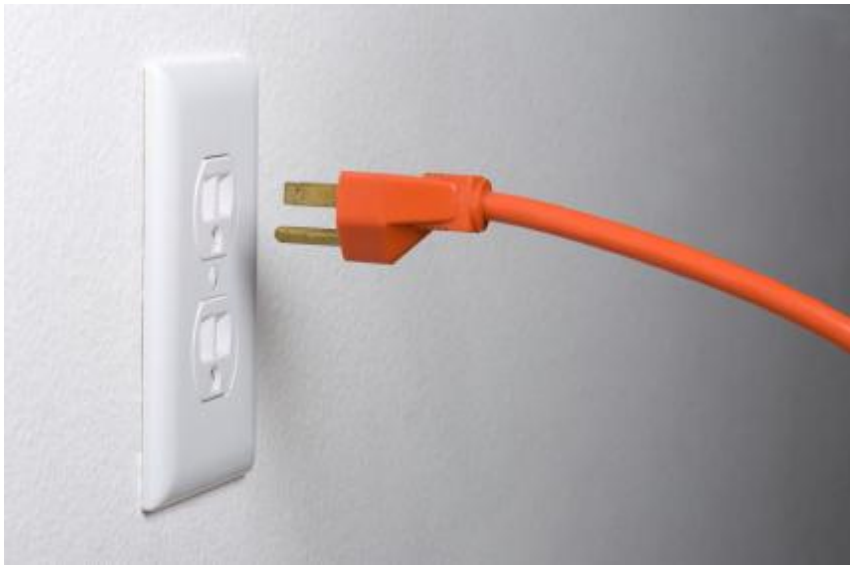
- Designed to serve static documents
  - HTTP
  - Half duplex communication
- High latency
- Bandwidth intensive
  - HTTP header traffic approx. 800 to 2000 bytes overhead per request/response
- Complex architecture
  - Not changed since the 90's
  - Plug-ins
  - Polling / long polling
  - Legacy application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

Legacy Web Architecture



Complexity does not scale

Enter HTML5 WebSocket!



HTML5 WebSocket History

- Initial draft TCPConnection, submitted 2007 by Ian Hickson, Google & HTML5 Editor
- Updated to WebSocket during summer 2008 with major contributions by Kaazing Engineering

What do WebSocket and model trains have in common?

WebSocket

- Includes W3C API and IETF Protocol
  - Event-driven JavaScript API
  - Full-duplex communication 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 Schemes

- WebSocket

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

- WebSocket Secure

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

WebSocket Handshake

KAAZING

Client wants *ws://example.com/chat*

Client

REQUIRED

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-Websocket-Key: 16-byte nonce, base64 encoded
Sec-Websocket-Version: 6
```

OPTIONAL

```
Sec-Websocket-Origin: http://example.com
Sec-Websocket-Protocol: protocol [,protocol]*
Sec-Websocket-Extensions: extension [,extension]*
Cookie: cookie content & other cookie-related headers
```

Server accepts

Server

REQUIRED

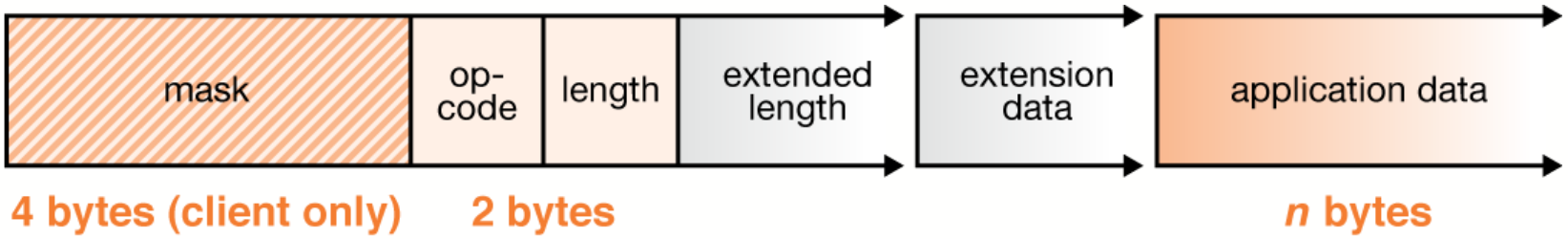
```
HTTP/1.1 101 "Switching Protocols" or other description
Upgrade: websocket
Connection: Upgrade
Sec-Websocket-Accept: 20-byte MD5 hash in base64
```

OPTIONAL

```
Sec-Websocket-Protocol: protocol
Sec-Websocket-Extensions: extension [,extension]*
```

WebSocket Frames

- Frames have a few header bytes
- Data may be text or binary
- Frames from client to server are masked (XORed w/ random value) to avoid confusing proxies



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";
}
```

WebSocket API

KAAZING

JavaScript

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

// Attach listeners
ws.onopen = function(evt) {
    console.log("Connection open…");
};

ws.onmessage = function(evt) {
    console.log("Received message: " + evt.data);
};

ws.onclose = function(evt) {
    console.log("Connection closed…");
};
```

WebSocket API

KAAZING

JavaScript

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


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

WebSocket Support

KAAZING

- Chrome 4.0+
- Safari 5.0 & iOS 4+
- Firefox 4+
- Opera 10.7+
- Internet Explorer 10+

"In a nutshell, we love HTML5, we love it so much we want it to actually work."

— Dean Hachamovitch, Microsoft





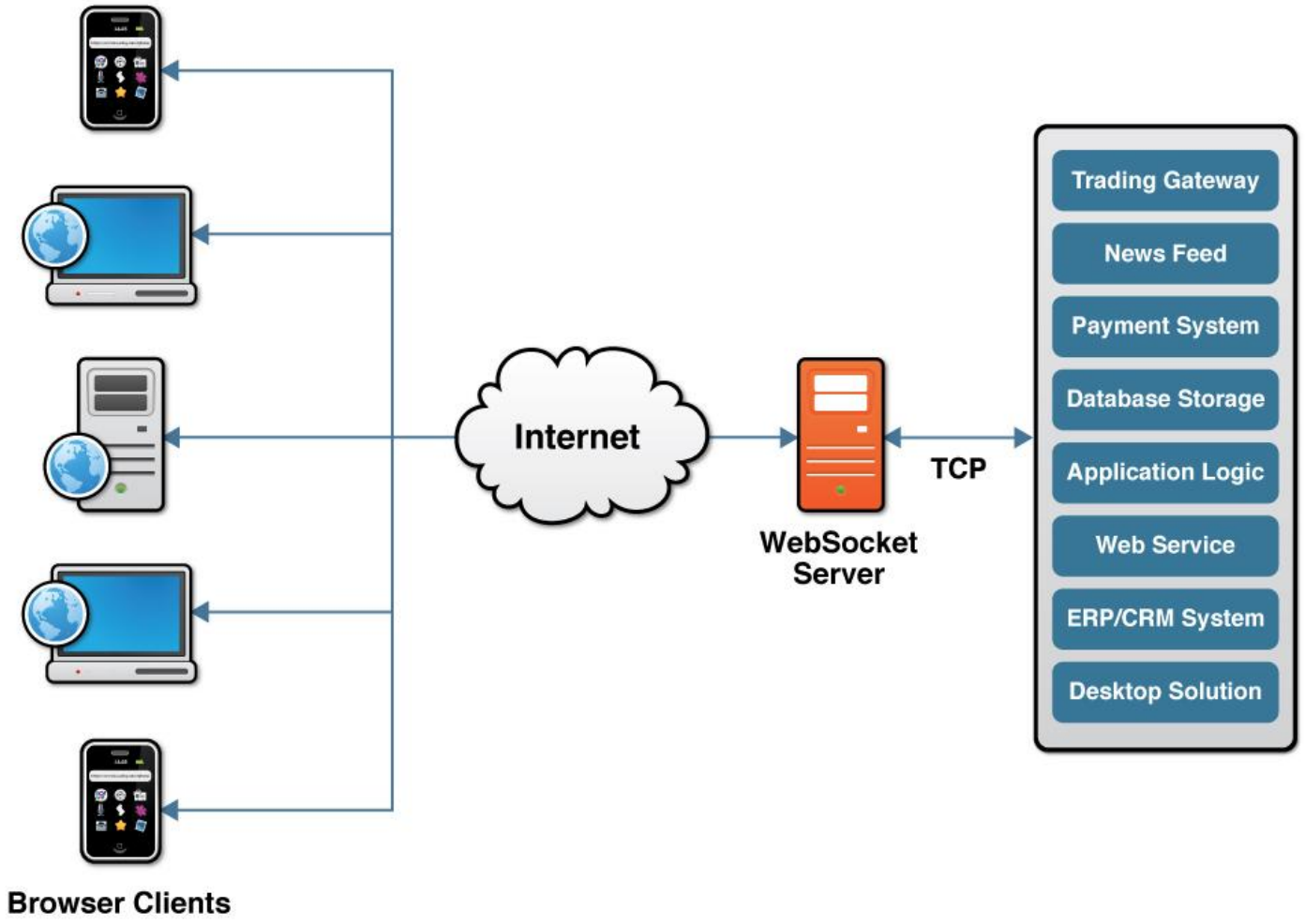
WebSocket Emulation

- Kaazing WebSocket Gateway
  - http://www.kaazing.com/download
  - Makes WebSocket work in all browsers today
    - IE6-9, Android, …
  - Makes WebSocket work in non-JavaScript clients
    - Flash, Silverlight, Java, .Net, …

Living Web Stack

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

Living Web Architecture



WebSocket Misconceptions

KAAZING

- WebSocket API is ***not*** a drop-in replacement for XMLHttpRequest, nor was it designed to be!
- WebSocket API is ***not*** intended for direct use
- WebSocket wire protocol is ***not*** designed to be a one-size-fits-all replacement for HTTP!
- WebSocket wire protocol is ***secured*** from even the *perceived ability* to negatively impact transparent HTTP caching proxies

Beyond WebSocket

So we have WebSocket, now what?

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

JMS for HTML5

DEMONSTRATION

Using the XMPP API

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")
```

XMPP Client Commands

XMPP offers the following client commands:

- Connect and disconnect
- Register
- Check roster
- Send messages
- Set status
- Communicate presence
- Many other extensions

Google Talk

Google Talk

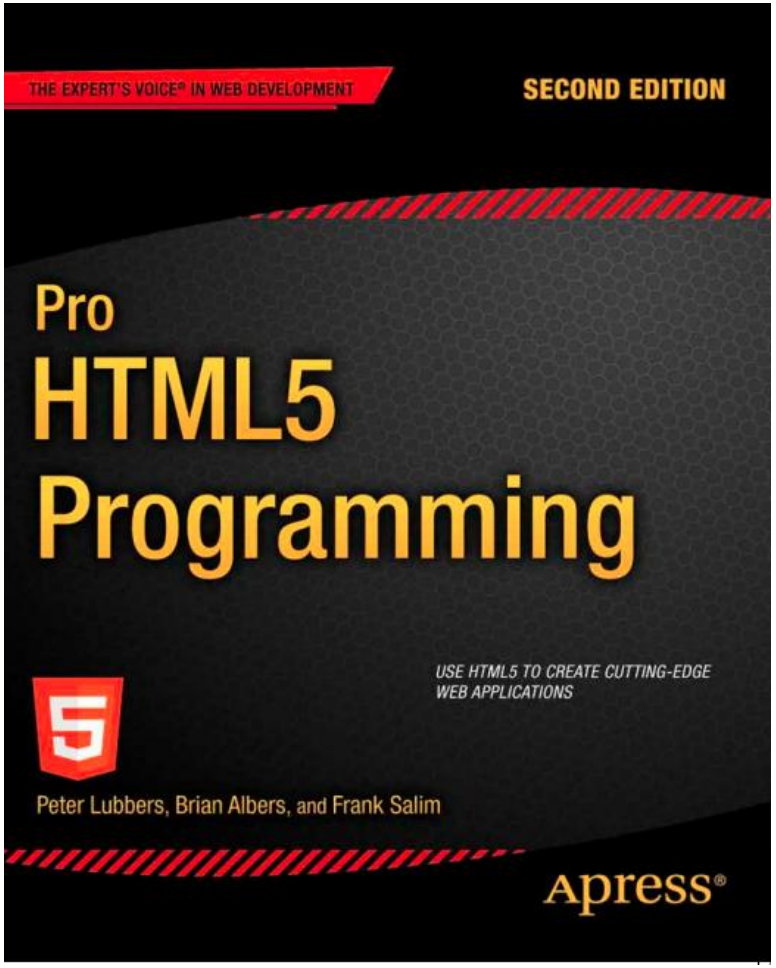
- Encrypted (XMPP over TLS)
- Supports integration with any service provider that uses the XMPP protocol
- Google Talk service is hosted at talk.google.com on port 5222
- Authentication through SASL PLAIN

Challenge, Again …

“Now that you are not restricted by the traditional limitations of HTTP, what type of Living Web applications will you build?”

Book Promo

Pro HTML5 Programming
Version 2 available soon!
www.prohtml5.com
HTML5 Live Coupon
50% Off E-Book:
HTML5CONF

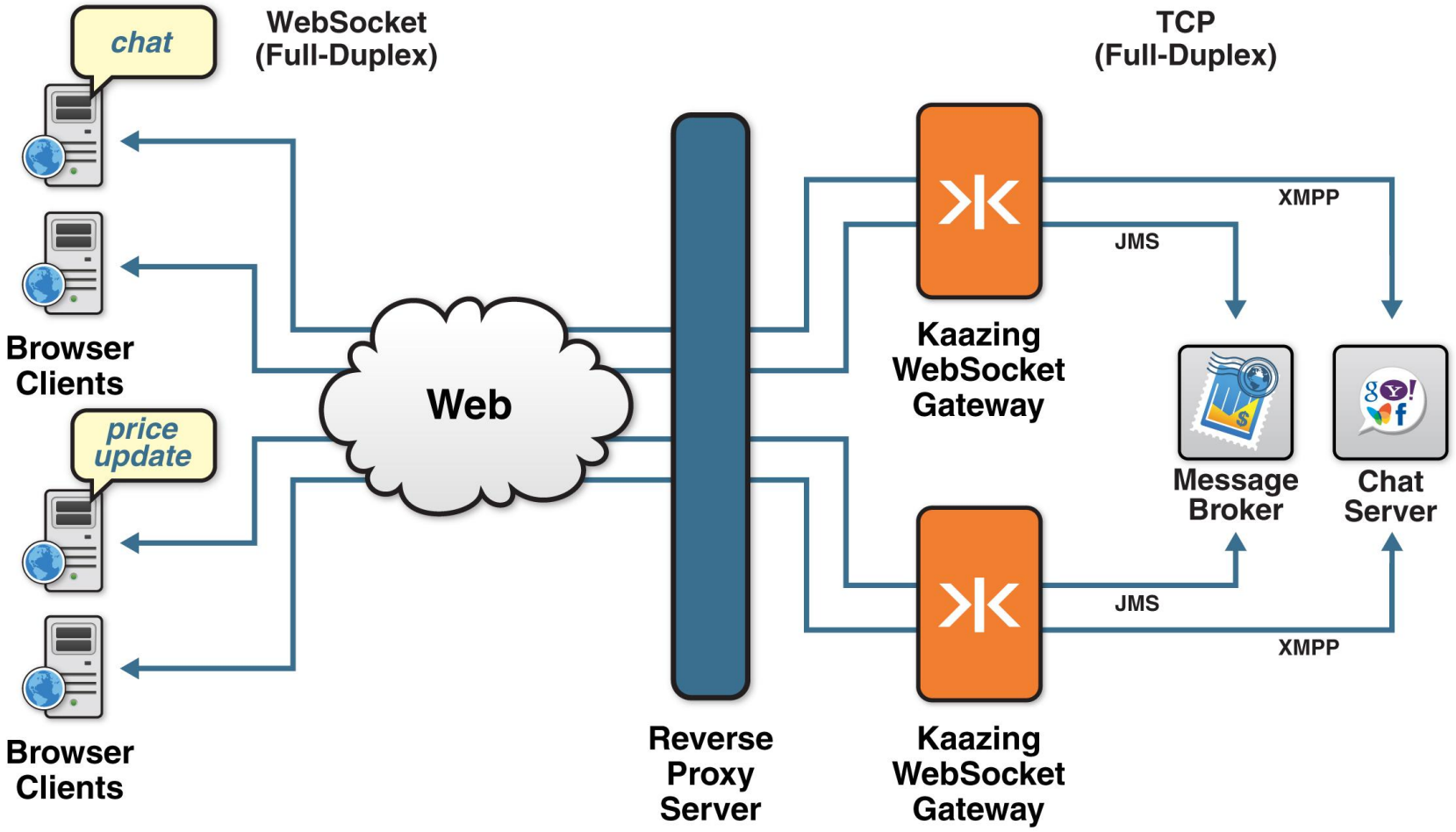


KAAZING

Server Support

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

WebSocket Architecture



Connection Offloading