

Develop Real Time HTML5 Applications using WebSocket with Apache Camel & ActiveMQ

Camel One
16th of May - 2012

Charles Moulliard
Sr. Principal Solution Architect
Apache Committer

FuseSource
integration everywhere

Speaker : Charles Moulliard

- Engineer in Agronomy & Master in Zoology
 - 18 years of experience in IT world development
 - Specialized in new technologies Web & Integration
 - Project manager in Bank, Financial, Telco world
- Solution Architect at FuseSource
- Committer on projects : Apache ServiceMix, Apache Karaf (PMC), Apache Camel, Fuse Fabric

 Twitter : <http://twitter.com/cmoulliard>

 LinkedIn : <http://www.linkedin.com/in/charlesmoulliard>

 My blog : <http://cmoulliard.blogspot.com>

 Slideshare : <http://www.slideshare.net/cmoulliard>



Agenda

- Introduction
 - History & HTML5
 - Websocket revealed
- ActiveMQ
 - Stomp and WebSocket
- Camel
 - Camel-websocket
- Demo
- Conclusion

Agenda

- **Introduction**
 - History & HTML5
 - Websocket revealed
- **ActiveMQ**
 - Stomp and WebSocket
- **Camel**
 - Camel-websocket
- **Demo**
- **Conclusion**

Real Time HTML5 Apps - Introduction

■ History

- Started with Java Applet and Shockwave player (199x)
- Followed by Macromedia Shockwave, Adobe Flash, Microsoft Silverlight
- Goal : Develop client-server application using HTTP/RMI protocol



Microsoft®
Silverlight™



Real Time HTML5 Apps - Introduction

- Evolution : Animation → Design Rich Internet Applications (Intranet)
- Limitations about Technology based on Web 1.0
 - No push → require lot of HTTP requests
 - RMI protocol → blocked by Firewall
 - Exchanges mainly use XML - WebServices
 - Compatibility between plugin(s) / browser
 - Code should be signed (to access local resources)

Real Time HTML5 Apps - Introduction

■ Web 2.0 “Revolution”

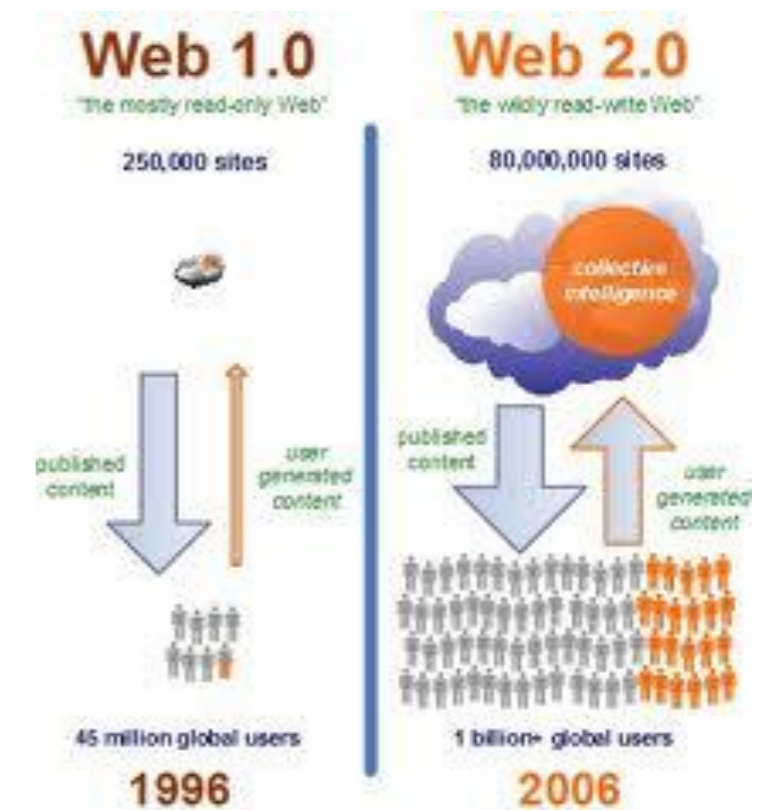
-  HTTPs requests

- Due to volume of data (video, audio, text) exchanged between “social networks”

- User becomes “actor” – “collaborating”

■ New solutions have been imagined to support that →

- Ajax technology (2005), JSON instead of XML , RestFull



Real Time HTML5 Apps - Introduction

- JavaScript client uses asynchronous communication with server
 - Ajax Reverse, but not really PUSH

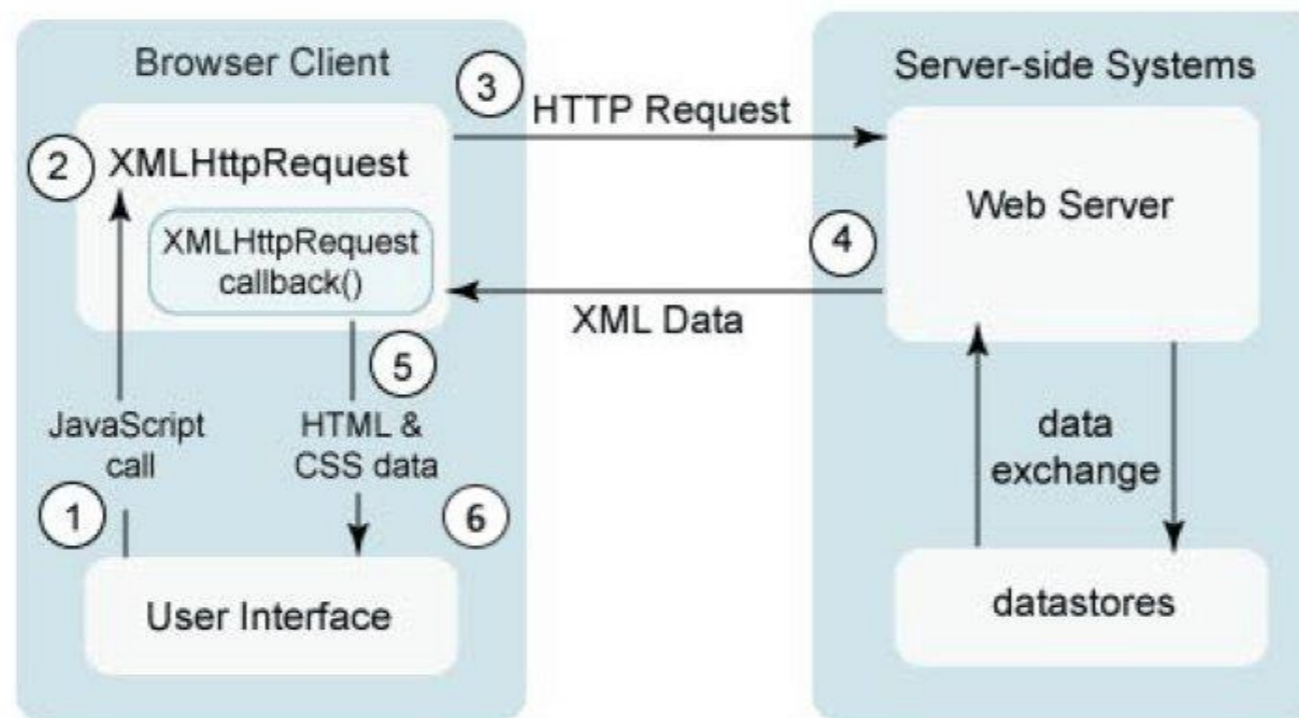


Figure 1: How AJAX Technologies Handle a User Action

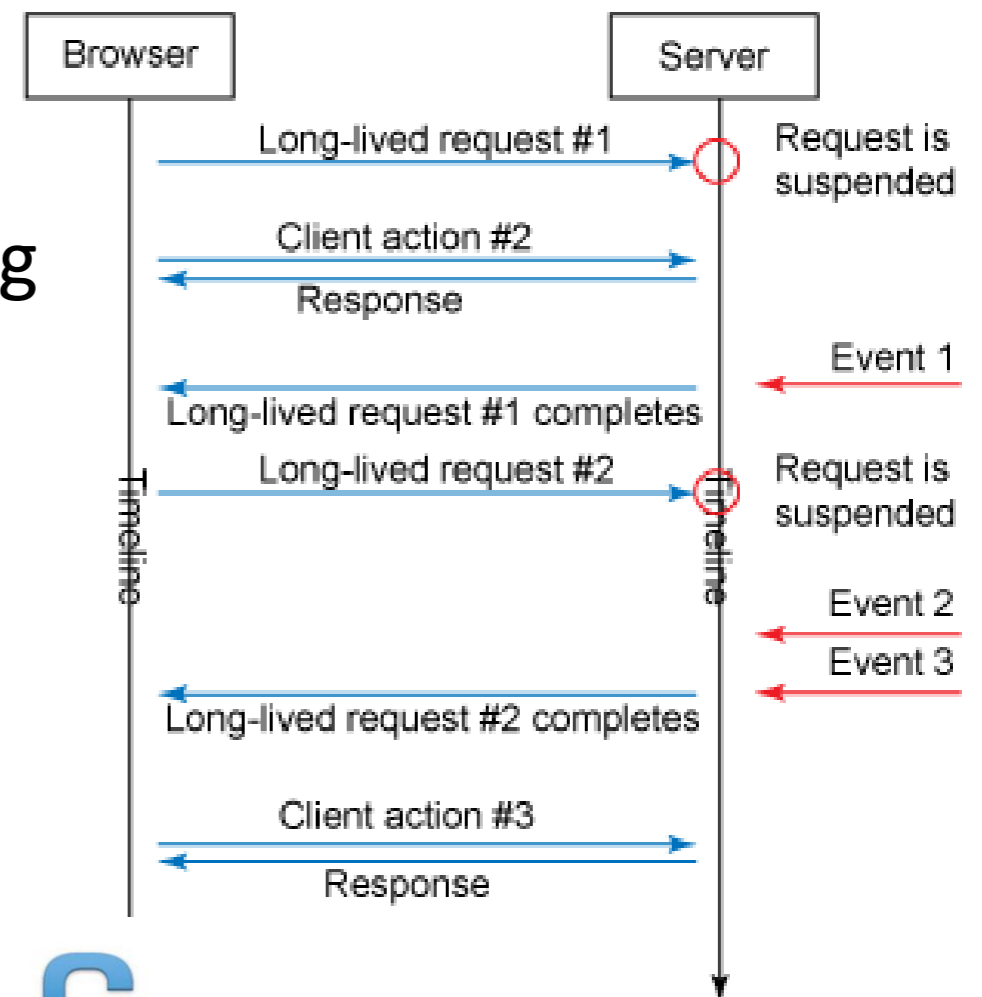


Real Time HTML5 Apps - Introduction

- First Strategy Developed
- Cometd/Bayeux protocol
- Principe



- Use a **persistent** or **long-lasting** HTTP connection with HTTP server (streaming / polling)
- **Events** based to push data to the browser
- Connection is keep alive till timeout



Real Time HTML5 Apps - Introduction

■ Drawbacks :

- Browser (HTTP 1.x spec) only allow 2 connections with HTTP server → 1 connection will be kept alive for Cometd/Bayeux server
- Any new event requests a new HTTP call
- Bayeux communication protocol has not been standardized through instances IETF, W3C
- Exchanges based on “Bayeux” messages (and not XML, JSON, ...) – more complex to handle
- One Way communication (server → client)

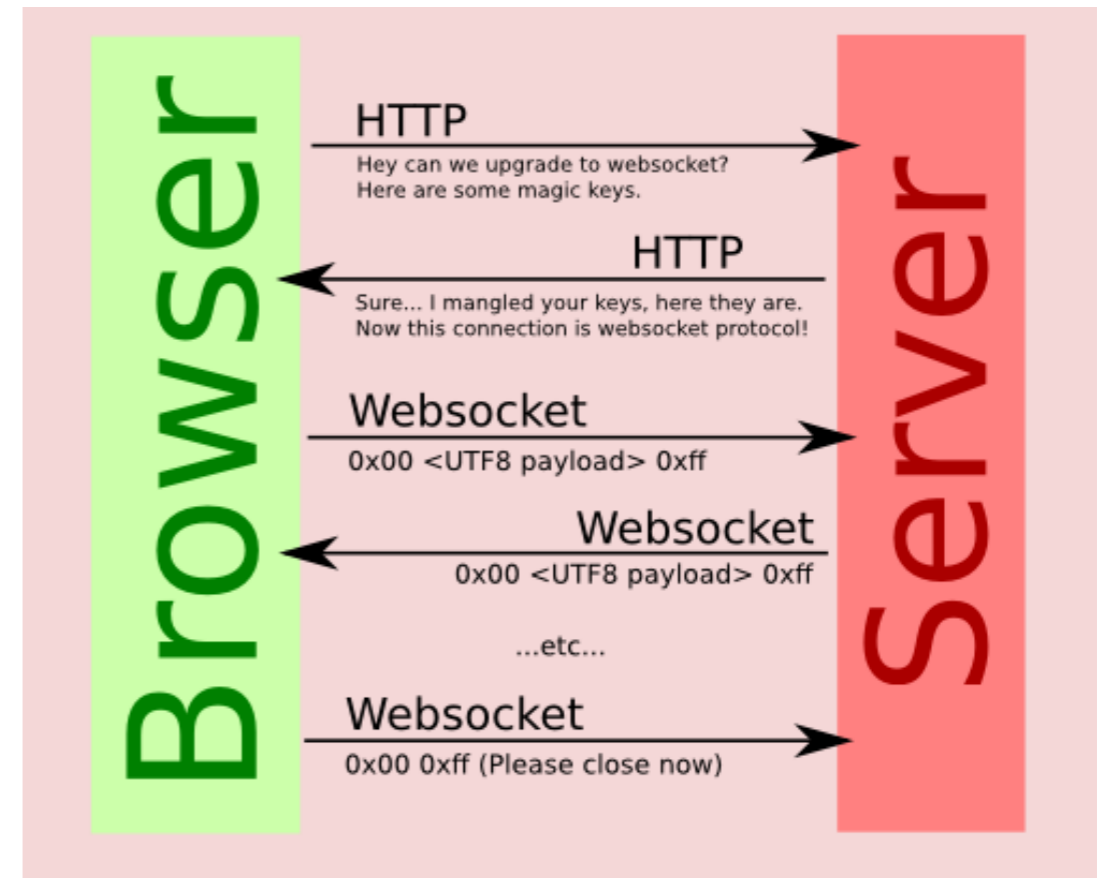
Real Time HTML5 Apps - Introduction

- 2nd approach - WebSocket
- HTML5 (2009)
 - ~ = html + css + js
 - Offline storage
 - **RealTime/Communication**
 - File/Hardware Access
 - CSS 3
 - Graphics/Multimedia
 - Supported by Tablet/Pc/Mobile



Real Time HTML5 Apps - Introduction

- Websocket revealed
 - Full-duplex single socket connection
 - HTTP request followed by WebSocket data Packets exchange
 - `ws://` and `wss://` protocol
 - Developed part of HTML5 initiative
 - Specification [rfc-6455](#) (Dec-2011) managed by IETF Task Force



Real Time HTML5 Apps - Introduction

■ What the browser sends →

▼ Request Headers view parsed

```
GET ws://localhost:9090/newsTopic HTTP/1.1
Origin: http://127.0.0.1:8080
Connection: Upgrade
Host: localhost:9090
Sec-WebSocket-Key: CTEq35HHaHuT+1Gs0MB3qQ==
Upgrade: websocket
Sec-WebSocket-Version: 13
```

Key which is base64 encoded and uses as handshake between client and server

Version of WebSocket used - allow client and server to check if they are compatible

Real Time HTML5 Apps - Introduction

■ And what it receives

▼ Response Headers view parsed

HTTP/1.1 101 Switching Protocols

Connection: Upgrade

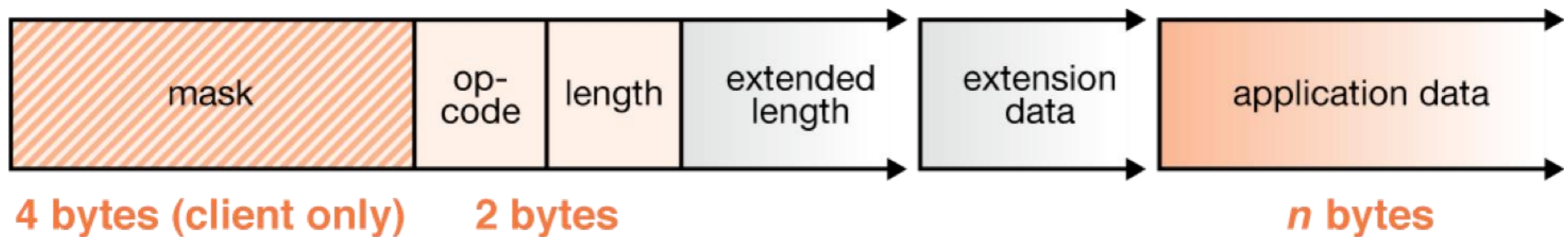
Upgrade: WebSocket

Sec-WebSocket-Accept: rygeGXZxH4n5ZWXC2emIAwi7jdA=

Response created by the server
(client key + GUI) signed SHA-1 and
encoded in base64

Real Time HTML5 Apps - Introduction

- Next Data frames are exchanged back and forth between client and server through TCP/IP connection

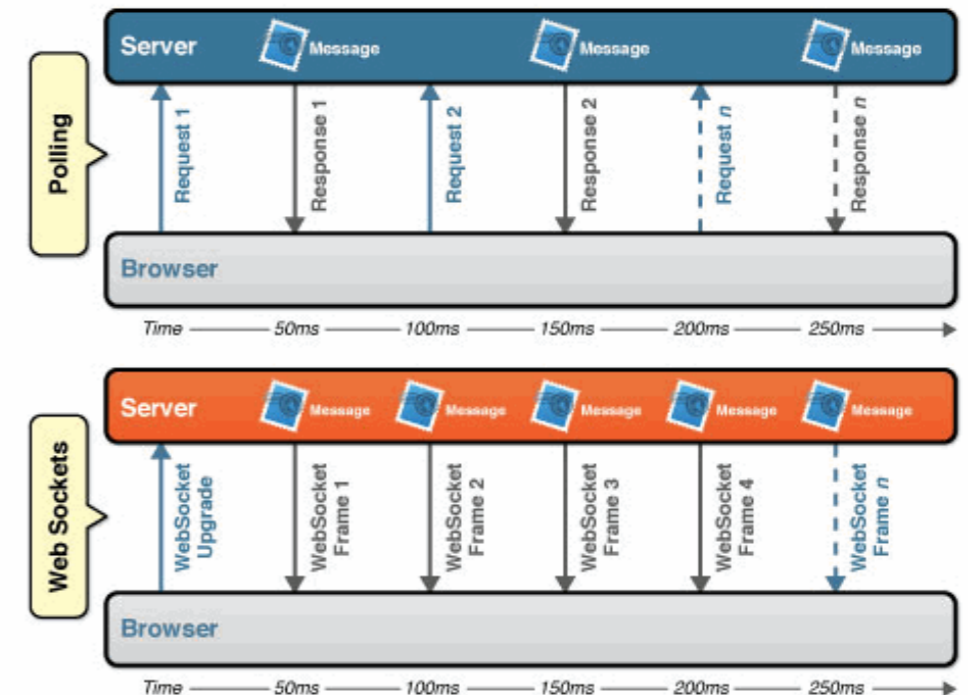


- Bytes or UTF-8 Text packets are supported
- Text is mainly used within JavaScript

Real Time HTML5 Apps - Introduction

■ Benefits

- Use same ports as HTTP (80 and HTTPS (473))
- Bandwidth reduction
- No long polling process
- No more unnecessary traffic
- Standard based
- Security managed through Web Server (SSL, Authentication)



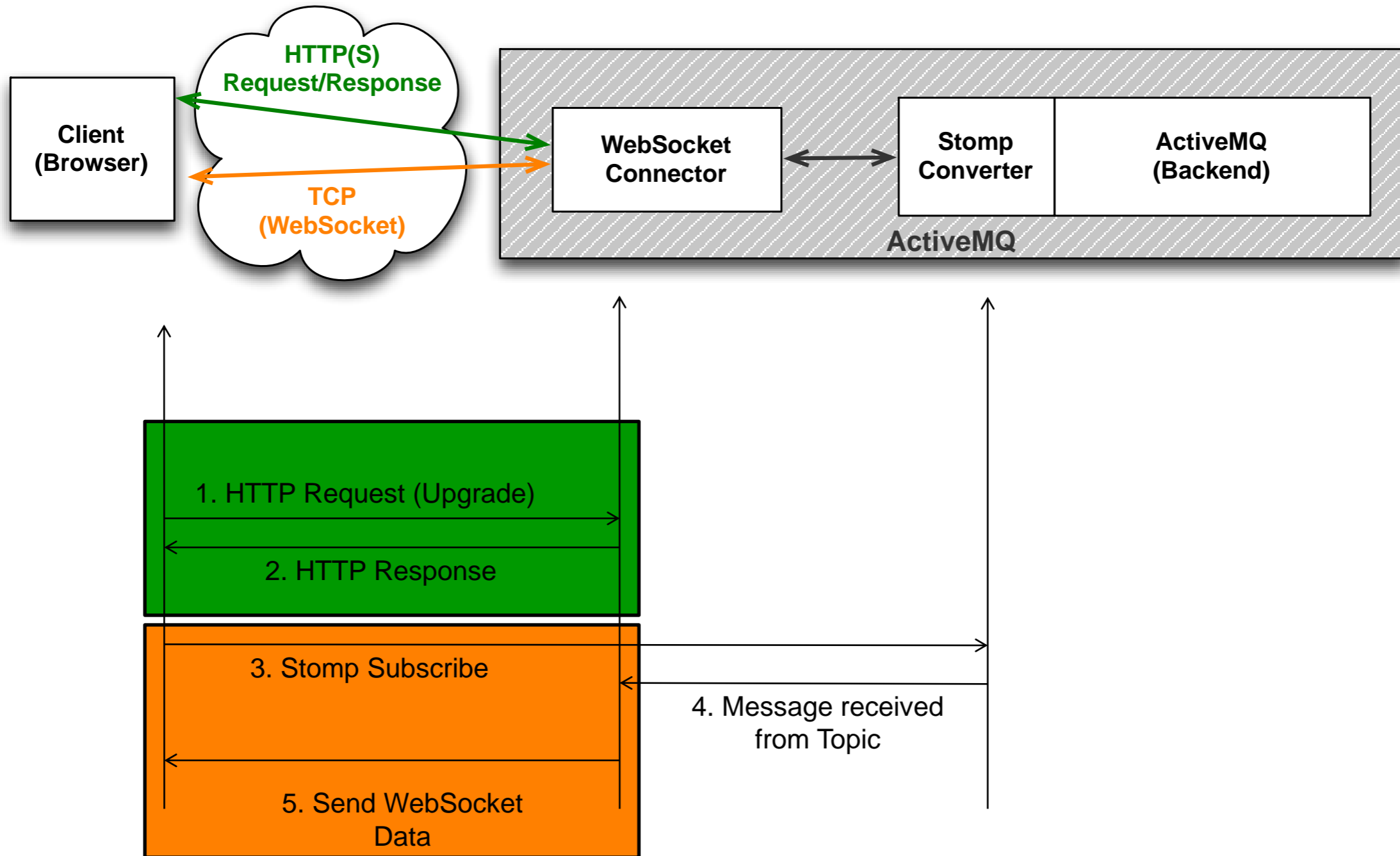
Agenda

- Introduction
 - History & HTML5
 - Websocket revealed
- **ActiveMQ**
 - Stomp and WebSocket
- Camel
 - Camel-websocket
- Demo
- Conclusion

Real Time HTML5 Apps - ActiveMQ

- Provide a websocket implementation using `ws://` transport connector
- Use Jetty WebSocket v7.5
- Is available since ActiveMQ 5.6
- Stomp is the wire format
- More info :
<http://activemq.apache.org/websockets.html>

Real Time HTML5 Apps - ActiveMQ



Real Time HTML5 Apps - ActiveMQ

■ Server side

- 1) Transport connector must be enable for websocket

```
<transportConnectors>
  <transportConnector name="openwire" uri="tcp://0.0.0.0:61616"/>
  <transportConnector name="websocket" uri="ws://0.0.0.0:61614"/>
</transportConnectors>
```

Add ws:// transport connector

- 2) Topics/Queues will be used as channels to send (or receive messages)

Nothing special ...

```
// create the connection factory
ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://localhost:61616");
Connection connection = connectionFactory.createConnection("guest", "password");
//Connection connection = connectionFactory.createConnection();
connection.start();

// Create the session and topic
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
Topic quoteTopic = session.createTopic("stockQuoteTopic");
```

Real Time HTML5 Apps - ActiveMQ

■ Client side

- Need js client

A) stomp.js : Author → Jeff Mesnil

- Code : <https://github.com/jmesnil/stomp-websocket>

- Doc : <http://www.jmesnil.net/stomp-websocket/doc/>

B) stompie : Author → Karl Krukow

- Code : <https://github.com/krukow/stompie>

Real Time HTML5 Apps - ActiveMQ

- 2) Create a WebSocket connection with ActiveMQ using Stomp.js client

```
$(document).ready(function() {  
    var client, destinationQuotes;  
  
    $('#connect_form').submit(function() {  
        var url = $("#connect_url").val();  
        var login = $("#connect_login").val();  
        var passcode = $("#connect_passcode").val();  
        destinationQuotes = $("#destinationQuotes").val();  
  
        client = Stomp.client(url);  
  
        // this allows to display debug logs directly on the web page  
        client.debug = function(str) {  
            $("#debug").append(str + "\n");  
        };  
        // the client is notified when it is connected to the server.  
        var onconnect = function(frame) {  
  
        };  
        client.connect(login, passcode, onconnect);  
    });  
});
```

JQuery helps us to display result in HTML page

A. Create a client to connect to the server

Destination : /topic/stockQuoteTopic
url : ws://localhost:61614/stomp :

C. Establish connection using login, password

C. Wait response ...

Real Time HTML5 Apps - ActiveMQ

- 3) Subscribe to the topic (= channel)

Subscribe to the topic and listen on messages

```
client.subscribe(destinationQuotes, function(message) {  
  var quote = JSON.parse(message.body);  
  $('#.' + "stock-" + quote.symbol).replaceWith("<tr class=\"stock-" + quote.symbol + "\">" +  
    "<td>" + quote.symbol + "</td>" +  
    "<td>" + quote.open.toFixed(2) + "</td>" +  
    "<td>" + quote.last.toFixed(2) + "</td>" +  
    "<td>" + quote.change.toFixed(2) + "</td>" +  
    "<td>" + quote.high.toFixed(2) + "</td>" +  
    "<td>" + quote.low.toFixed(2) + "</td>" +  
    "</tr>");  
});
```

Parse JSON messages

- 4) When job is done → close connection

```
$('#disconnect_form').submit(function() {  
  client.disconnect(function() {  
    $('#disconnect').fadeOut({ duration: 'fast' });  
    $('#connect').fadeIn();  
    $('#send_form_input').attr('disabled');  
  });  
  return false;  
});
```

Close here

Real Time HTML5 Apps - ActiveMQ

■ Benefits

- Acts as real time data feed provider (topic)
- Infrastructure can be secured
 - Web Server level (SSL, Authentication)
 - ActiveMQ “security plugin”
- By combining Scalability/High Availability features of ActiveMQ, feed are persisted, distributed

Agenda

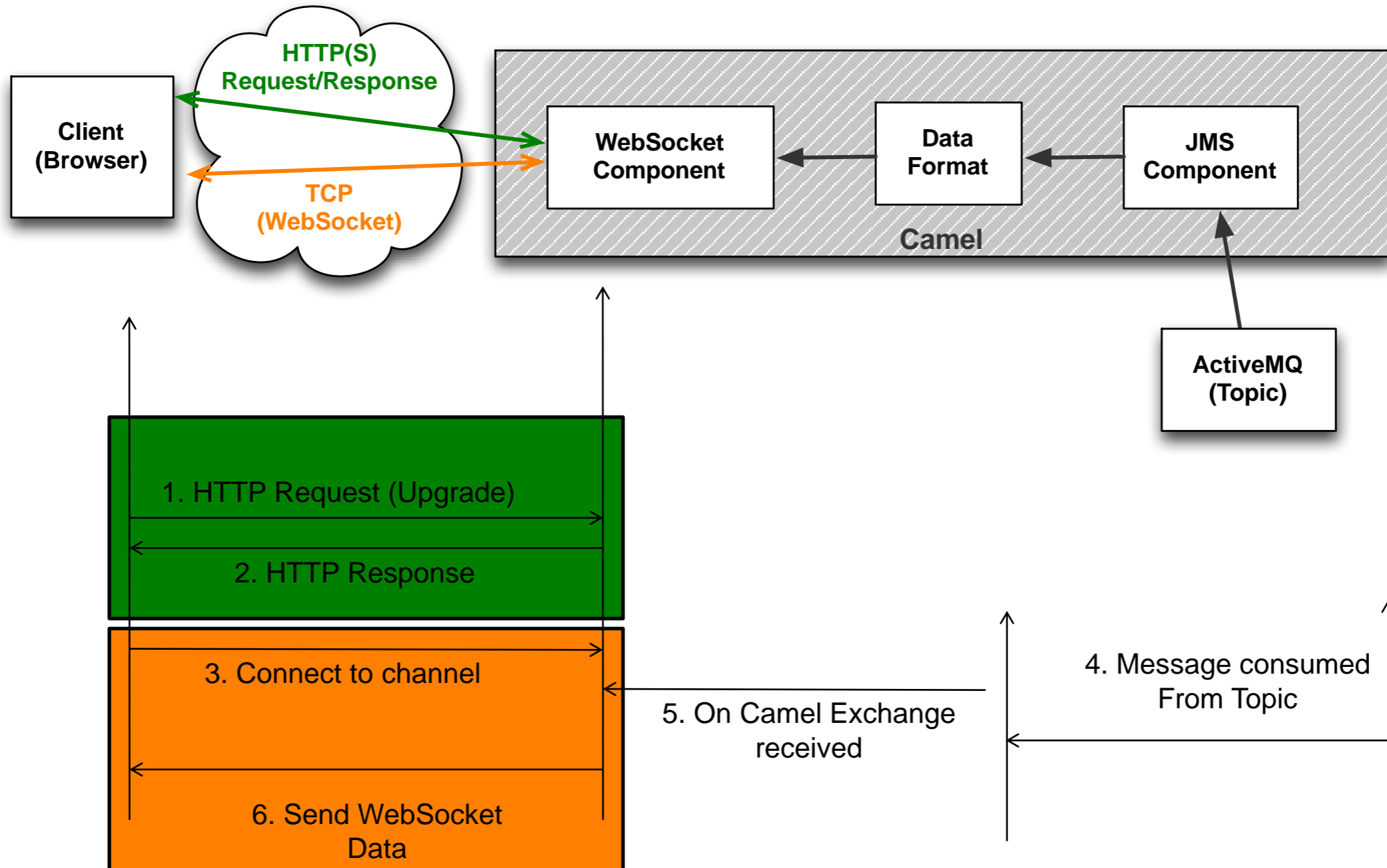
- Introduction
 - History & HTML5
 - Websocket revealed
- ActiveMQ
 - Stomp and WebSocket
- **Camel**
 - Camel-websocket
- Demo
- Conclusion

Real Time HTML5 Apps - Camel

- Camel-websocket component - part of Camel 2.10 (<http://camel.apache.org/websocket.html>)
- Use Jetty WebServer v7.5
- Can be used to produce or consume
- Allow to send packets to multiple clients (“sendToAll”) or to individual
- Should be combined with EIP patterns to aggregate/split/enrich data and/or other components (RSS, Atom, Twitter, JSON, ...)



Real Time HTML5 Apps - Camel



Real Time HTML5 Apps – Camel - Produce

- 1) Create a Camel Route to produce websocket packets → Http Client

```
public class WebSocketStockPricesRoute extends RouteBuilder {  
  
    @Override  
    public void configure() throws Exception {  
  
        from("activemq:topic:stockQuoteTopic")  
            .log(LoggingLevel.DEBUG, ">> Stock price received : ${body}")  
            .to("websocket:stockQuoteTopic?sendToAll=true");  
  
    }  
}
```

Data consumed from a topic are send
to all the WebSocket clients
connected

Real Time HTML5 Apps – Camel - Consume

- 1) or Create a Camel Route to consume - produce websocket packets

```
public class WebSocketChatRoute extends RouteBuilder {  
  
    @Override  
    public void configure() throws Exception {  
  
        from("websocket:chat-room")  
            .log(LoggingLevel.INFO, ">> Message received : ${body}")  
            .to("websocket:chat-room?sendToAll=true");  
  
    }  
}
```

WebSockets packets are received by the websocket:chat-room endpoint

Packets send by client in the channel "chat-room" are resend to all the clients connected to the "chat-room"

Real Time HTML5 Apps - Camel

- 2) Client side - Will use WebSocket js script in combination with jquery, jquery-ui, ... to communicate

```
var host1 = $("#url1").val();  
var host2 = $("#url2").val();
```

```
socket1 = new WebSocket(host1);  
socket2 = new WebSocket(host2);
```

```
// Add a connect listener  
socket1.onopen = function () {  
    $('#msg').append('<p class="event">Socket News Status: ' + socket1.readyState + ' (open)</p>');  
}
```

← First. Create a WebSocket connection

← 4 methods are available :

- onopen
- onmessage
- onerror
- onclose

Real Time HTML5 Apps - Camel

■ 2) Parse messages

Every websocket "message" is consumed here

```
socket1.onmessage = function (msg) {  
  
    // $('#msg').append('<p class="message">Received: ' + msg.data + "</p>");  
  
    var news = JSON.parse(msg.data);  
  
    // extract the news fields  
    var symbol = news.symbol;  
    var title = news.title;  
    var picture = news.picture;  
    var info = news.info;  
  
    var $res = $("<li>");  
    $res.append("<img src='" + picture + "' width='50' height='50' />");  
    $res.append("&nbsp;" + symbol + " : " + title + "<p/>");  
    $res.append("Description : " + info);  
    $res.append("</li>");  
    $res.appendTo('#news');
```

Message is parsed
using JSON (text →
Object)

Next we display the result in the HTML page

Real Time HTML5 Apps - Camel

■ 2) Close connection

Close event can be used to stop communication with server

```
$('#disconnect_form').submit(function () {  
  
    socket1.close();  
    socket2.close();  
  
    $('#msg').append('<p class="event">Socket News Status: ' + socket1.readyState + ' (Closed)</p>');  
    $('#msg').append('<p class="event">Socket Tweet Status: ' + socket2.readyState + ' (Closed)</p>');  
  
    $('#disconnect').fadeOut({ duration:'fast' });  
    $('#connect').fadeIn();  
    $('#send_form_input').removeAttr('disabled');  
  
    return false;  
});
```


Real Time HTML5 Apps - Camel

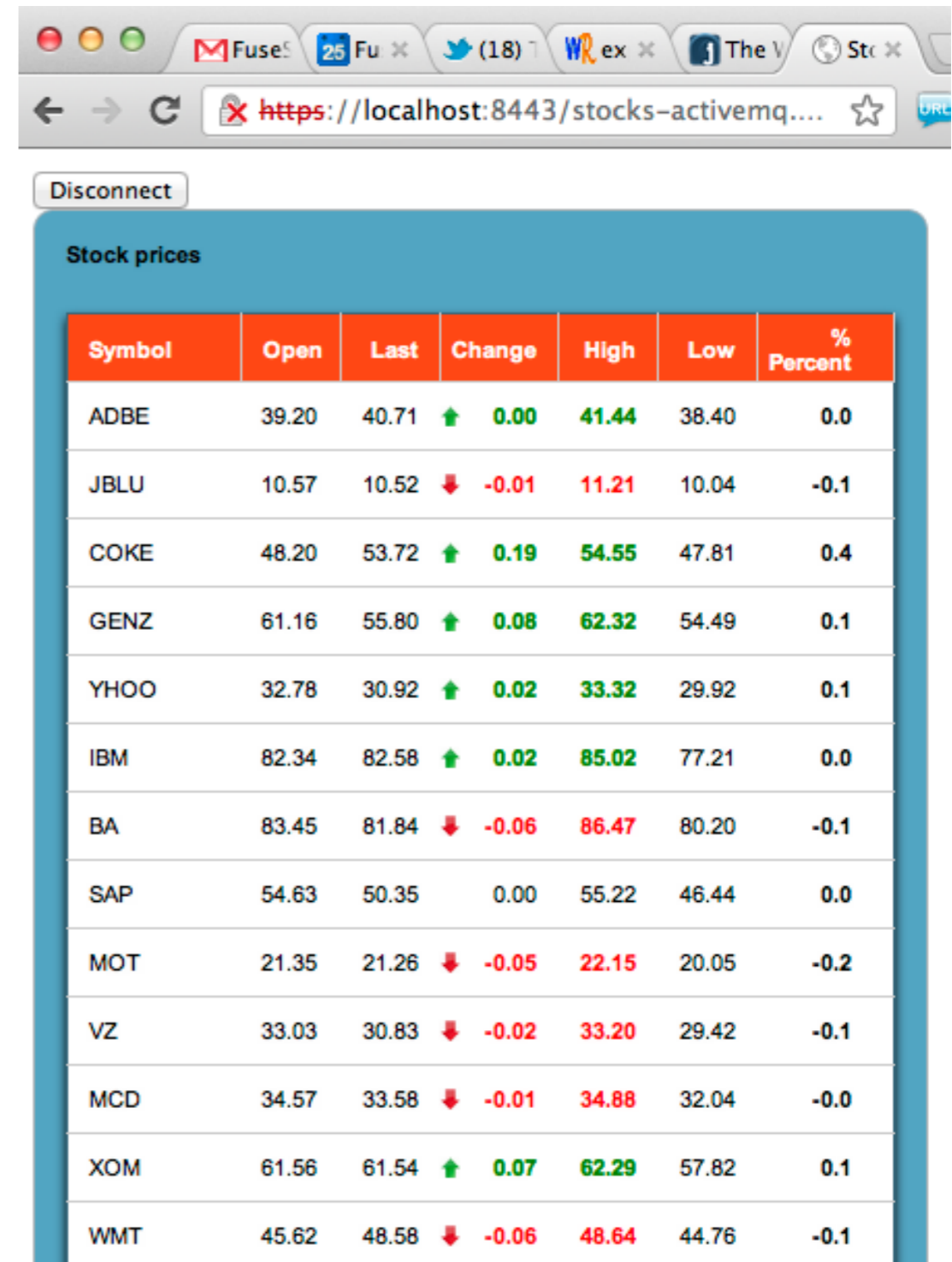
- Benefits Provided by Camel framework →
 - Camel acts as a mediation router between “feed” providers and “feed” consumers
 - Will facilitate “aggregation”, “enrichment”, “filtering” of the feeds receive before to distribute them
 - Will provide the components required to interconnect “platforms”
 - Camel + ActiveMQ + Fabric = Scalability, High Availability, Cloud of data Feed

Agenda

- Introduction
 - History & HTML5
 - Websocket revealed
- ActiveMQ
 - Stomp and WebSocket
- Camel
 - Camel-websocket
- **Demo**
- Conclusion

Real Time HTML5 Apps - Demo

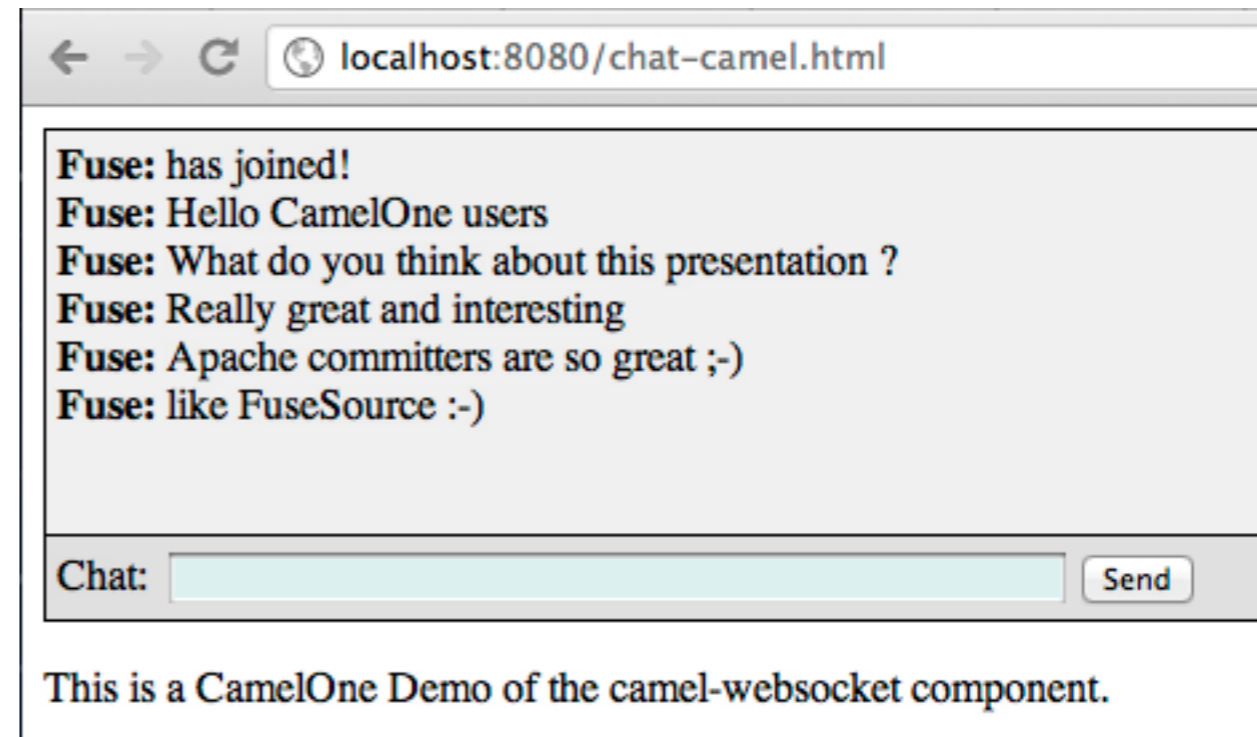
ActiveMQ – Stock Trader Demo 1



Symbol	Open	Last	Change	High	Low	% Percent
ADBE	39.20	40.71	↑ 0.00	41.44	38.40	0.0
JBLU	10.57	10.52	↓ -0.01	11.21	10.04	-0.1
COKE	48.20	53.72	↑ 0.19	54.55	47.81	0.4
GENZ	61.16	55.80	↑ 0.08	62.32	54.49	0.1
YHOO	32.78	30.92	↑ 0.02	33.32	29.92	0.1
IBM	82.34	82.58	↑ 0.02	85.02	77.21	0.0
BA	83.45	81.84	↓ -0.06	86.47	80.20	-0.1
SAP	54.63	50.35	0.00	55.22	46.44	0.0
MOT	21.35	21.26	↓ -0.05	22.15	20.05	-0.2
VZ	33.03	30.83	↓ -0.02	33.20	29.42	-0.1
MCD	34.57	33.58	↓ -0.01	34.88	32.04	-0.0
XOM	61.56	61.54	↑ 0.07	62.29	57.82	0.1
WMT	45.62	48.58	↓ -0.06	48.64	44.76	-0.1

Real Time HTML5 Apps - Demo

Camel – Chat Room Demo 2



localhost:8080/chat-camel.html

Fuse: has joined!
Fuse: Hello CamelOne users
Fuse: What do you think about this presentation ?
Fuse: Really great and interesting
Fuse: Apache committers are so great ;-)
Fuse: like FuseSource :-)

Chat:


This is a CamelOne Demo of the camel-websocket component.

Real Time HTML5 Apps - Demo


Camel – News Feed Demo 3

News list


News Feed Tweets



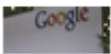
APPLE : Apple earnings: Will big numbers rev up its stock?
Description : Apple reports its second quarter results tomorrow, and bigger sales of iPhone stock trajectory.



MACINTOSH : Apple iPad 3 expected on 7 March following press event
Description : Apple has announced an event on 7 March at which the company is expected read: We have something you really have to see. And touch. While not officially confirming:



GOOGLE : Google to launch online storage service for consumers
Description : San Francisco: Google Inc is preparing to roll out a service to let consumers s the matter said, pushing into a market now dominated by the likes of Dropbox and Box.The



localhost:8080/news-camel.html

Disconnect
Socket News Status: 1 (open)
Socket Tweet Status: 1 (open)

News list

News Feed Tweets

Wed Apr 25 16:45:18 CEST 2012 (ViDa_y_Mas_Na) Mamá mamá, en la escuela me dicer Roberto...

Wed Apr 25 16:45:17 CEST 2012 (vic_die) RT @hausoframez: Gaga Deserves A Cheese

Wed Apr 25 16:45:26 CEST 2012 (Jorge_xperience) Lady Gaga estaria orgullosa de la ma

Wed Apr 25 16:45:32 CEST 2012 (Walkdats) En serio quiero que Lady Gaga Venga con s

Wed Apr 25 16:45:31 CEST 2012 (Eliezer_Bonilla) RT @TodoEsJoda: No sé hablar much MEN MEN ME ME ME ME LO LO LO LO.

Wed Apr 25 16:45:31 CEST 2012 (poohyukatin) @GaGa_saaaa わたしなんてもうネイル

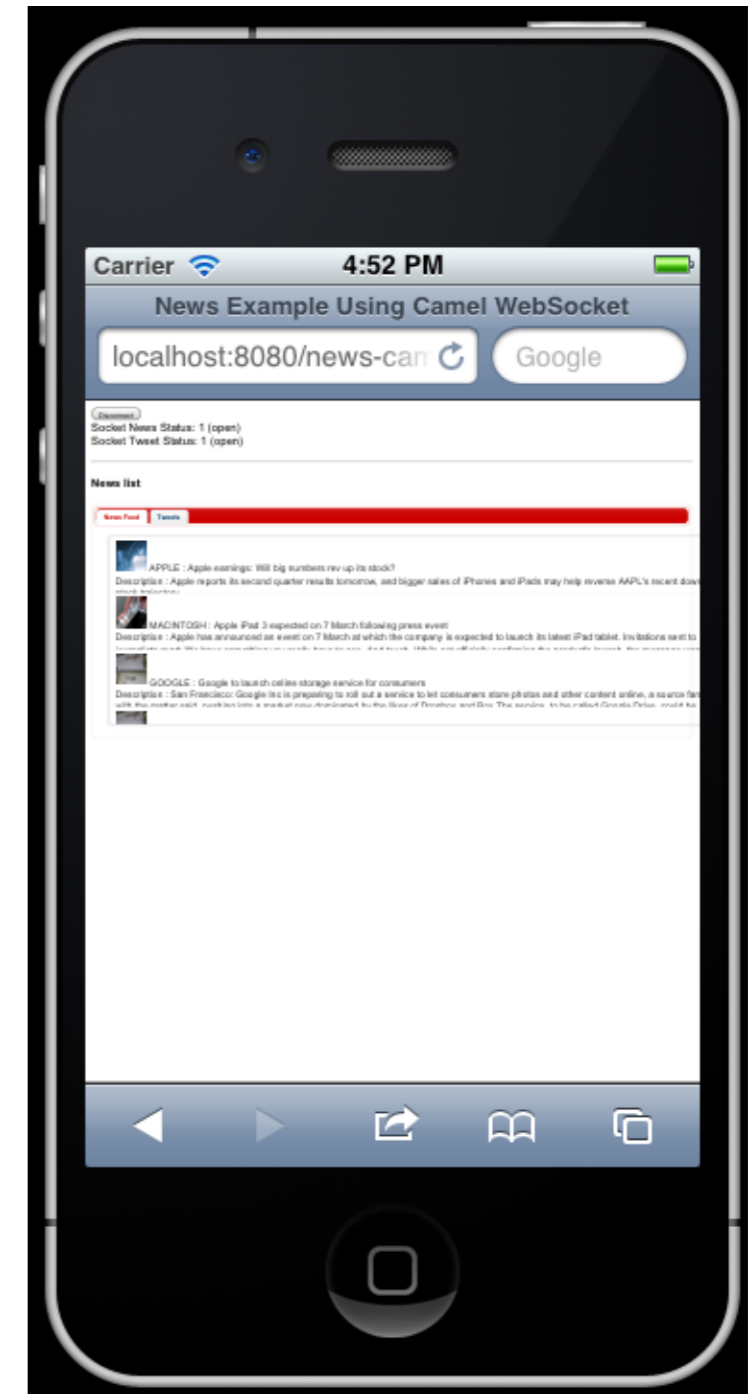
Need help :
Send messages to
[@fusenews](https://twitter.com/fusenews)

Agenda

- Introduction
 - History & HTML5
 - Websocket revealed
- ActiveMQ
 - Stomp and WebSocket
- Camel
 - Camel-websocket
- Demo
- **Conclusion**

Real Time HTML5 Apps - Conclusion

- ActiveMQ and Camel are ready for “Real Time HTML5 Application Development”
- Provide best of the Fuse - Apache technology for integration (EIP and components)
- Will help you to drive your business (transport, tracking,...)
- Available for any device
Tablet/iPhone/Android/Pc/Mac



Any Questions?



- Twitter : @cmoulliard
- More info : <http://fusesource.com>

- Demo code : <https://github.com/FuseByExample/websocket-activemq-camel>