intalio

HTTP, WebSocket, SPDY, HTTP/2.0

Evolution of Web Protocols

Simone Bordet
sbordet@intalio.com
Thomas Becker
tbecker@intalio.com

Intalio

- **Intalio|Jetty**
  - Services, Training and Support for Jetty and CometD
- **Intalio|BPMS**
  - Business Process Management System
- **Intalio|Create**
  - The modern way to build business applications
- **http://intalio.com**

Simone Bordet
sbordet@intalio.com

Who are we

- **Simone Bordet**
  - sbordet@intalio.com
  - @simonebordet
- **Open Source Contributor**
  - Jetty, CometD, MX4J, Foxtrot, LiveTribe, JBoss, Larex
- **Lead Architect at Intalio/Webtide**
  - SPDY and HTTP client lead
- **CometD project leader**
  - Web messaging framework
- **JVM tuning expert**
- **Runner, triathlete**

- **Thomas Becker**
  - tbecker@intalio.com
  - @tbecker00
- **Open Source Contributor**
  - Jetty, Grails
- **Senior Engineer at Intalio/Webtide**
  - Jetty's SPDY maintainer
- **JVM tuning expert**
- **Mountain biker, climber**

Agenda

- Some History of Web Protocols
- WebSocket – Bidirectional Web
- SPDY – A better Web
- HTTP 2.0 – the Future
- Q&A

Simone Bordet
sbordet@intalio.com

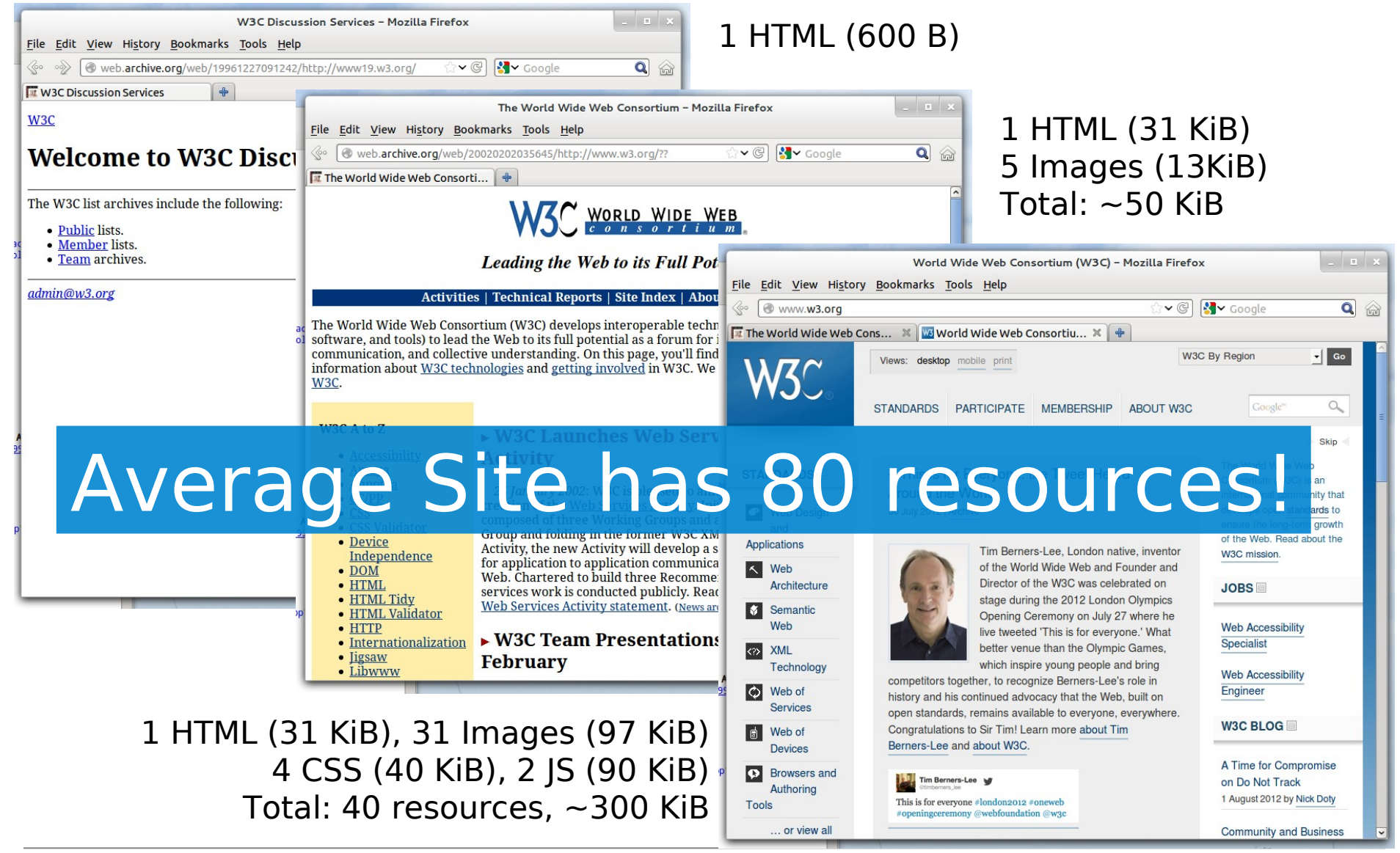
w3c.org 1996, 2002, 2012

1 HTML (600 B)

1 HTML (31 KiB)
5 Images (13KiB)
Total: ~50 KiB

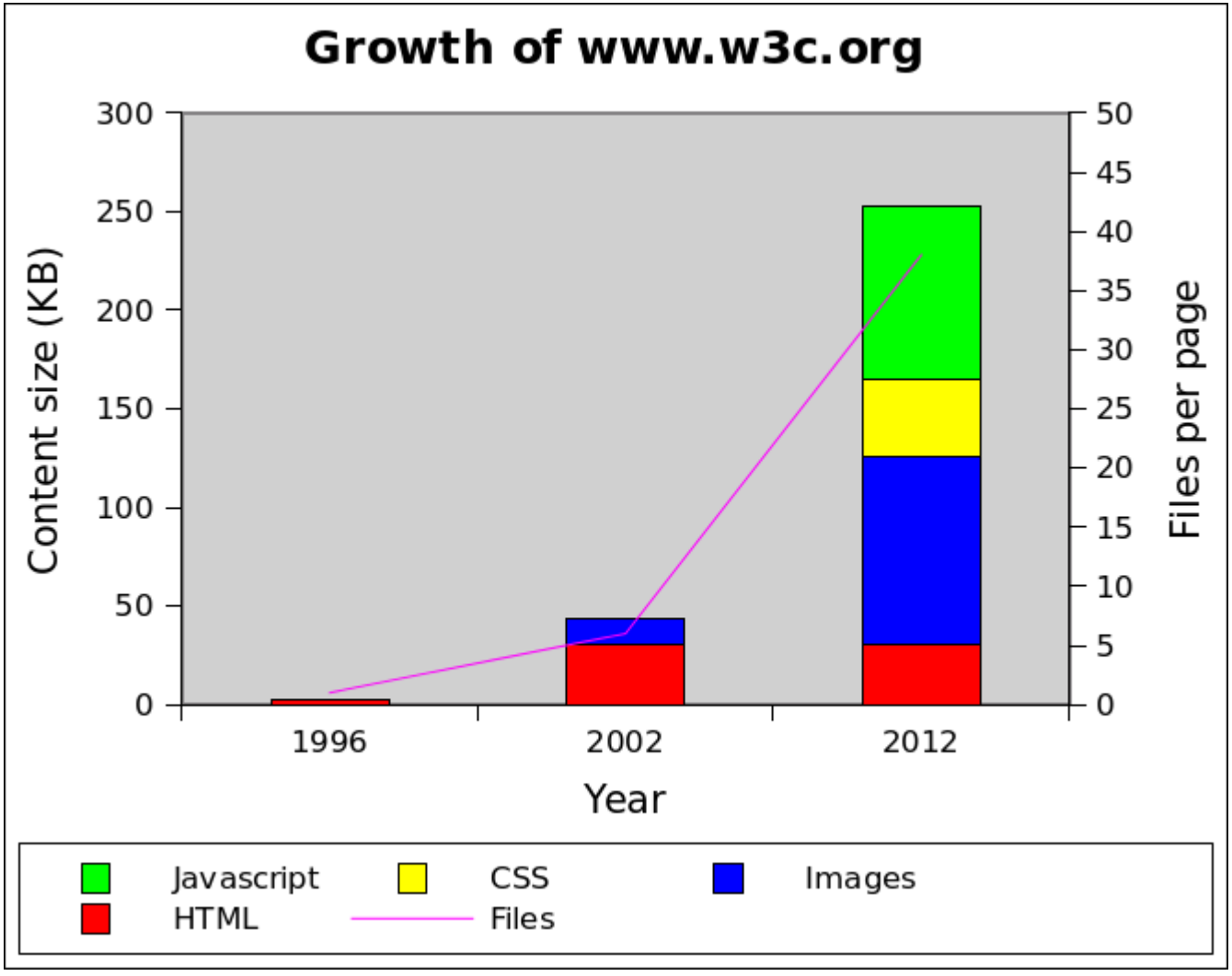
Average Site has 80 resources!

1 HTML (31 KiB), 31 Images (97 KiB)
4 CSS (40 KiB), 2 JS (90 KiB)
Total: 40 resources, ~300 KiB



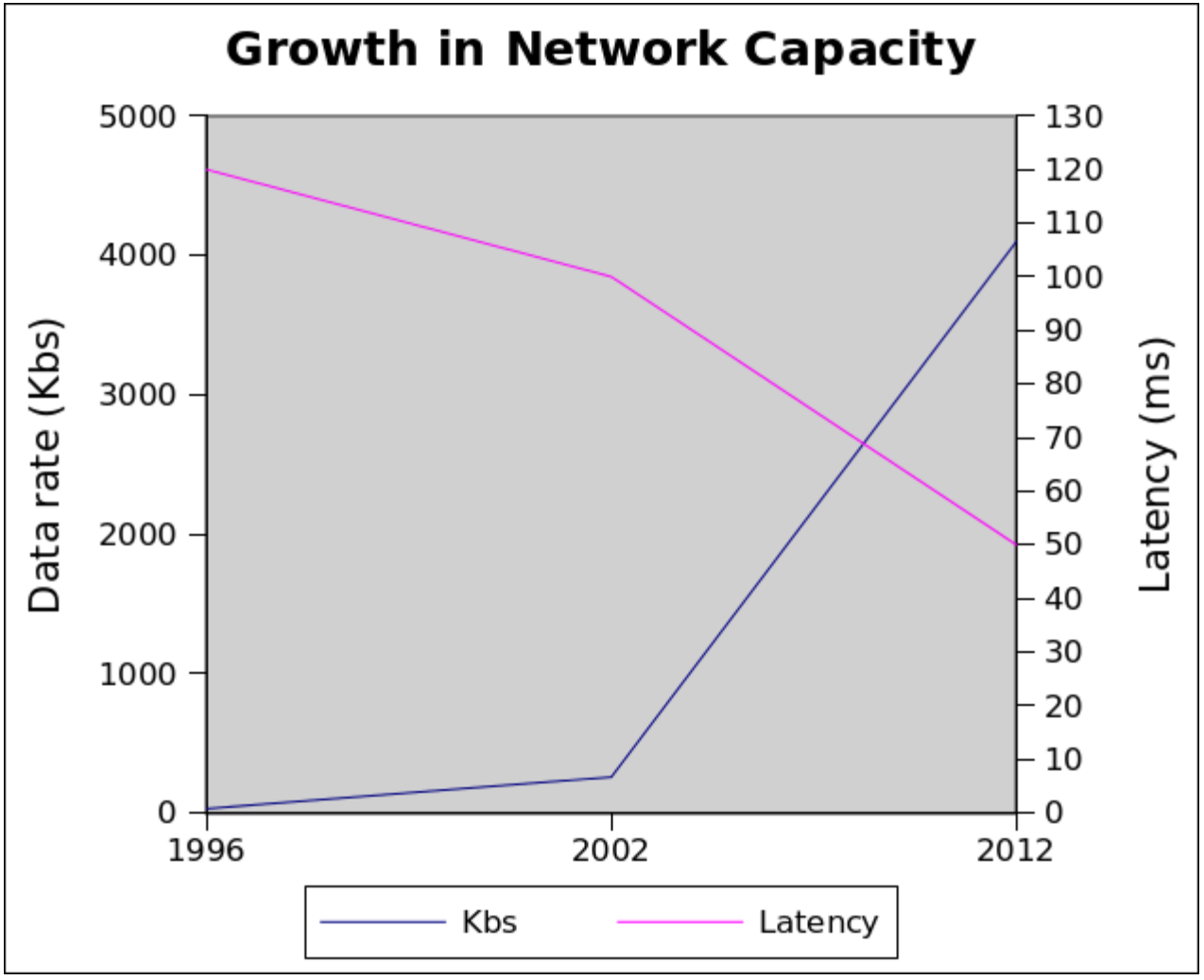
Simone Bordet
sbordet@intalio.com

Growth of w3c.org



Simone Bordet
sbordet@intalio.com

Growth network capacity



Simone Bordet
sbordet@intalio.com

Content Vs Network Growth

- **Calculated Page Load Time**
  - With HTTP/1.0

YEAR	1996	2012
Protocol	HTTP/1.0	HTTP/1.0
Feature		
max Connections	2	2
Connections	1	38
Data per connection(KB)	2	7
Request per connection	1	1
establish(ms)	*120*	*950*
requests(ms)	*120*	*950*
data(ms)	*556*	*495*
Slow start (ms)	*278*	*247*
Load Time (ms)	**1073**	**2642**

Content Growth
>
Network Growth

Simone Bordet
sbordet@intalio.com

Improved HTTP

- **Calculated Load T**
  - With features

38/2 = 19 Round Trips

Semantic issues Turned off by default

Connections have a cost

YEAR	1996	012	*2012*	2012
Protocol	HTTP/1.0	HT 1.1	*HTTP/1.1*	HTTP/1.1
Feature			***Pipeline***	
max Connections	2	2	*2*	**6**
Connections	1	2	*2*	6
Data per connection	2	127	*127*	42
Request per connection	1	19	*19*	6
establish(ms)	*120*	*50*	*50*	*50*
requests(ms)	*120*	*950*	*50*	*317*
data(ms)	*556*	*495*	*495*	*495*
Slow start (ms)	*278*	*16*	*16*	*47*
Load Time (ms)	**1073**	**1510**	***610***	**908**

Simone Bordet
sbordet@intalio.com

Multiple Connection Issues

- **Server resources**
  - Buffers and kernel data
  - Threads or selectors
- **Load balancers must be sticky**
  - Avoid distributed session
- **Difficult concurrency issues**
  - Multiple simultaneous accesses
  - Maybe out of order via different paths
- **Starts an arms race**
  - If 6 are good, surely 12 are better?
  - Already happening with domain sharding!

Simone Bordet
sbordet@intalio.com

New Protocols Needed

- **Content Growth continues > Network Growth**
  - HTTP has taken the easy wins
  - HTTP does not support the rich semantics needed
  - We've started to “bend the rules”

- **The natives are getting restless**
  - New & Experimental protocols have been deployed!
  - Websocket
    - *New Semantics*
  - SPDY
    - *Better Efficiency*

Simone Bordet
sbordet@intalio.com

WebSocket

Simone Bordet
sbordet@intalio.com

WebSocket

- **Effort Initiated by Browser Vendors to**
  - Bidirectional low latency communications
  - Replace Comet HTTP “hacks”
  - Implement rich web applications within the browser
    - *JavaFX? Silverlight? Flash?*

- **Standardized**
  - IETF - RFC6455 wire protocol
  - W3C - HTML5 Javascript API
  - JCP - JSR356 Java API (in progress)

Simone Bordet
sbordet@intalio.com

Web Socket JS API

```
interface WebSocket : EventTarget
{
    readonly attribute DOMString url;

    attribute EventHandler onopen;
    attribute EventHandler onerror;
    attribute EventHandler onclose
    attribute EventHandler onmessage;

    void send(DOMString data);
    void send(Blob data);

    void close(...);
};
```

Simone Bordet
sbordet@intalio.com

WebSocket Upgrade

- Runs on port 80 (or 443 for wss)
- Uses HTTP/1.1 Upgrade mechanism

REQUEST

```
GET / HTTP/1.1
Host: localhost:8080
Origin: http://localhost:8080
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Key: SbdIETLKHQ1TNBLeZFZS0g==
Sec-WebSocket-Version: 13
```

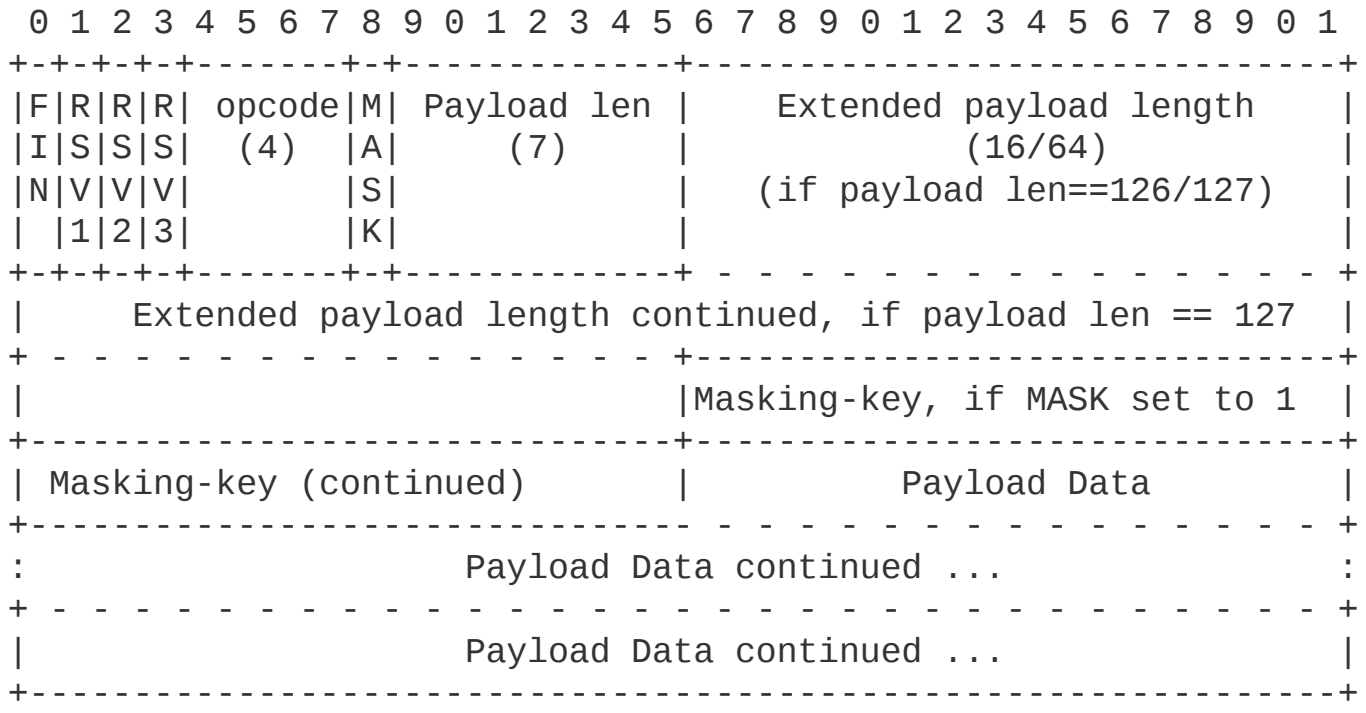
RESPONSE

```
HTTP/1.1 101 Switching Protocols
Connection: Upgrade
Upgrade: websocket
Sec-WebSocket-Accept: y4yXRUolfnFfo3Jc5HFqRHNgx2A=
```

Simone Bordet
sbordet@intalio.com

WebSocket Framing

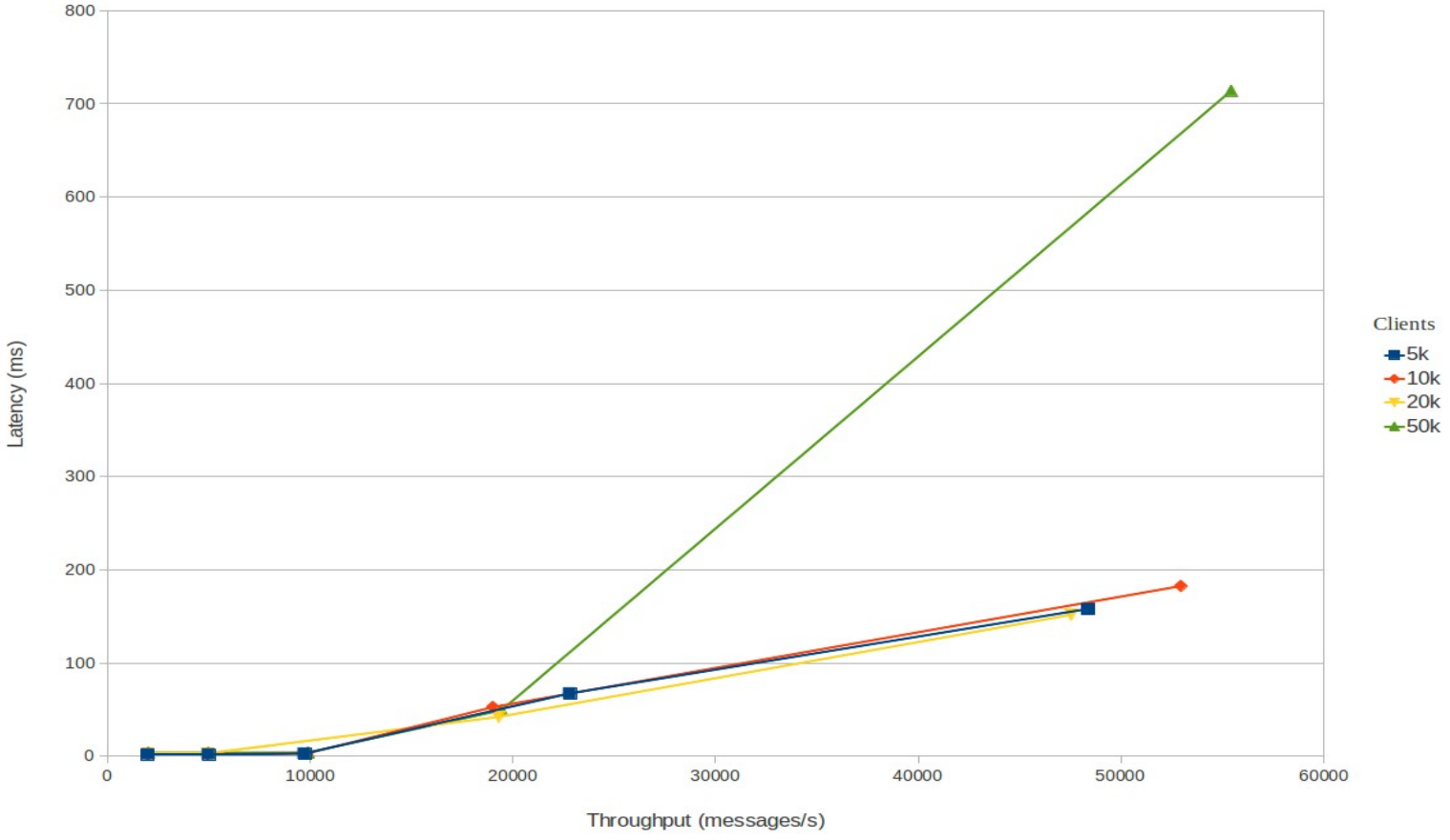
- **WebSocket frames of two types**
  - Control frames (Close, Ping, Pong)
  - Data frames (UTF8 Text & Binary)
- **Protocol is very compact**
  - Smallest data frame has only 2 bytes overhead



Simone Bordet
sbordet@intalio.com

HTTP Performance

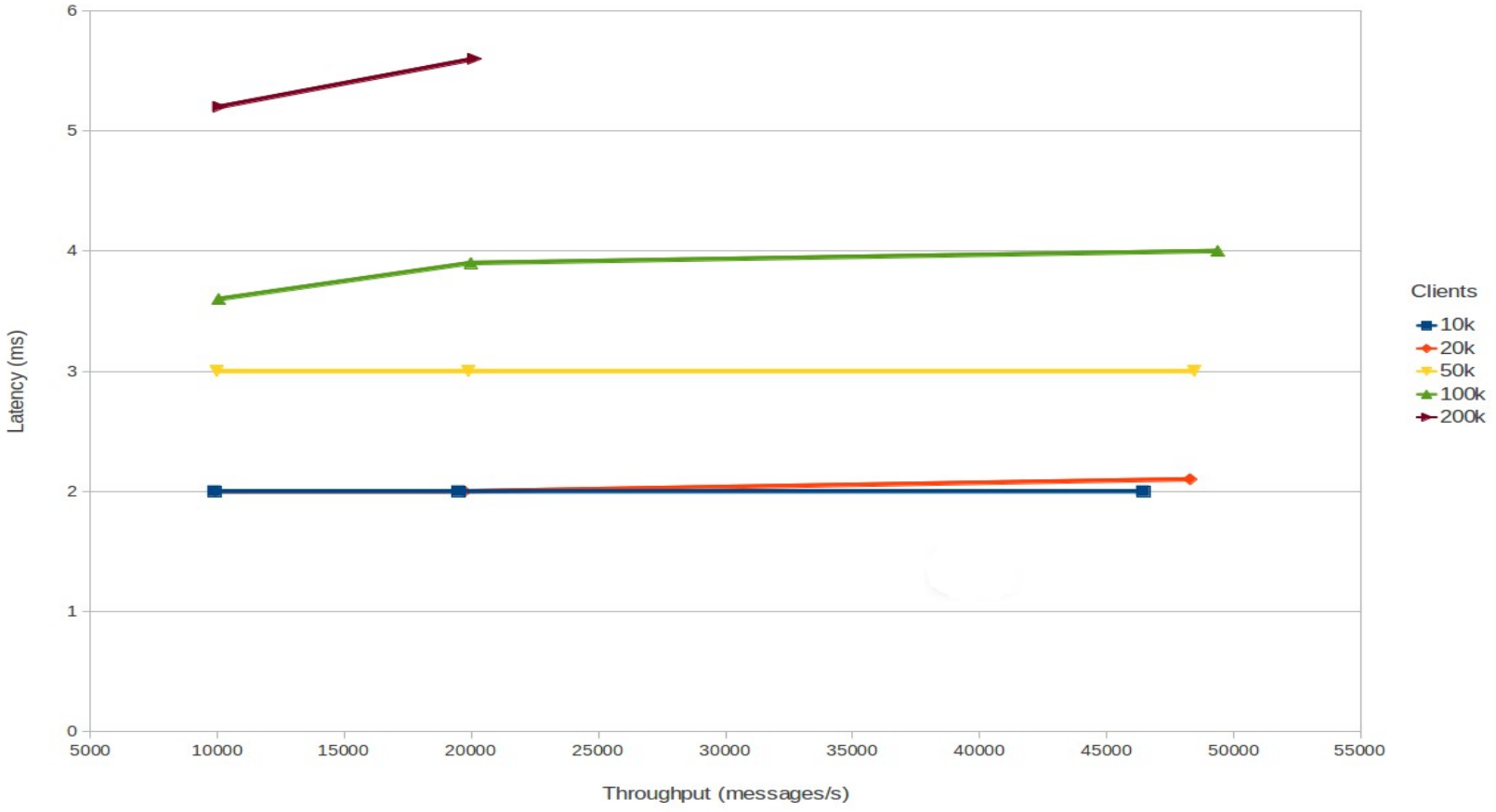
- Jetty-7.6, CometD-2.4, Chat simulation
- HTTP



Simone Bordet
sbordet@intalio.com

WebSocket Performance

- Jetty-7.6, CometD-2.4, Chat Simulation
- WebSocket



Simone Bordet
sbordet@intalio.com

WebSocket Deployment

- **Browsers**
  - Firefox 11 (android 15)
  - Internet Explorer 10 PP5
  - Chrome 16 (android 18)
  - Safari 6 (iOS 6)
  - Opera 12.1 (mini N/A)
  - Blackberry 7.0
  - Jetty Client 7,8,9
- **Java Servers**
  - Jetty 7, Jetty 8, Jetty 9
  - Glassfish
  - Resin 4
  - Tomcat 7
    - *not final yet*

59.03% by StatCounter GlobalStats

Simone Bordet
sbordet@intalio.com

WebSocket Conclusions

- **WebSocket is Here NOW!**
  - In production
  - Big benefits
  - Use Jetty!

- **WebSocket Limitations**
  - Does not provide HTTP semantic
    - *New applications need to be written*
  - Intermediaries may need to be WebSocket aware
    - *There are some problems*
  - No connection limit
    - *Multiplexing coming*
  - Very Low level
    - *Use CometD!*

Simone Bordet
sbordet@intalio.com

SPDY

Simone Bordet
sbordet@intalio.com

SPDY is LIVE!

- **SPDY is a live experiment to improve HTTP**
  - Addresses HTTP 1.1 limits
  - Designed to be faster and better than HTTP
- **Already widely deployed!**
  - Google, Twitter, Webtide, etc.
  - Chrome & Firefox
- **The SPDY protocol replaces HTTP on the wire**
  - But it's transparent for applications

Simone Bordet
sbordet@intalio.com

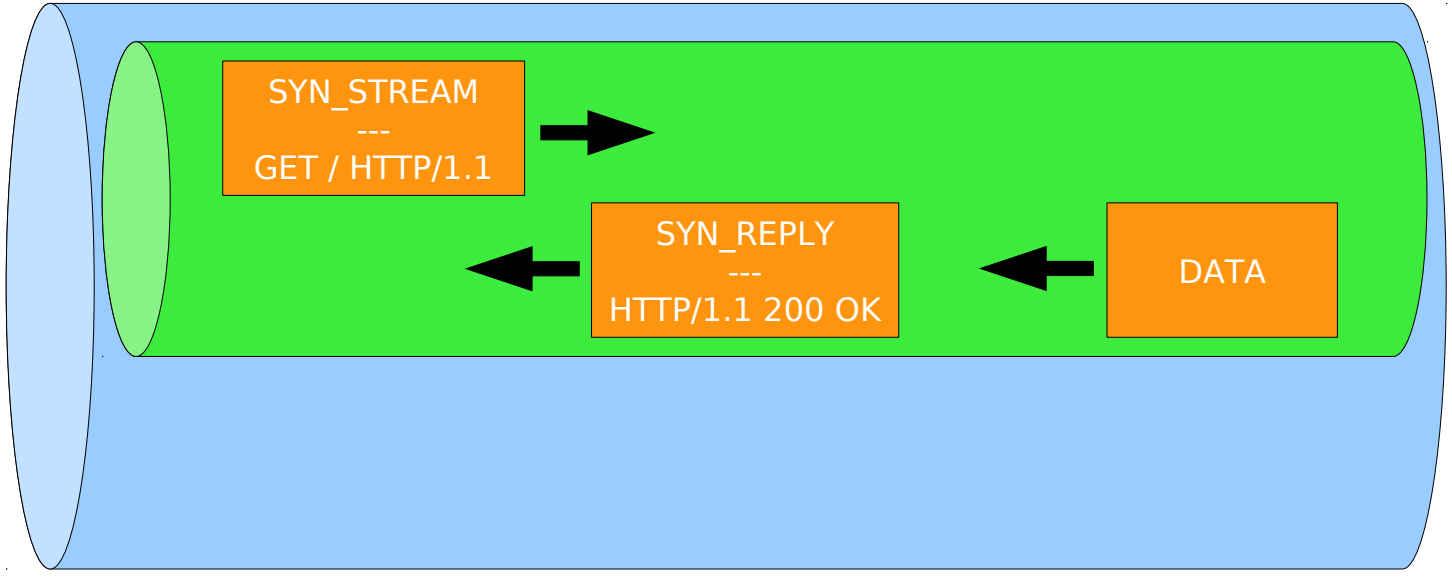
SPDY Protocol Initiation

- **SPDY uses TLS (aka SSL) connection on port 443**
  - TLS extended with Next Protocol Negotiation
- **New framing protocol over TLS**
  - Intermediaries don't know it is not HTTP wrapped in TLS
- **Transports existing HTTP Semantics (GET, POST, HEAD etc)**
  - Client and Server applications don't know it is not HTTP

Simone Bordet
sbordet@intalio.com

SPDY Framing

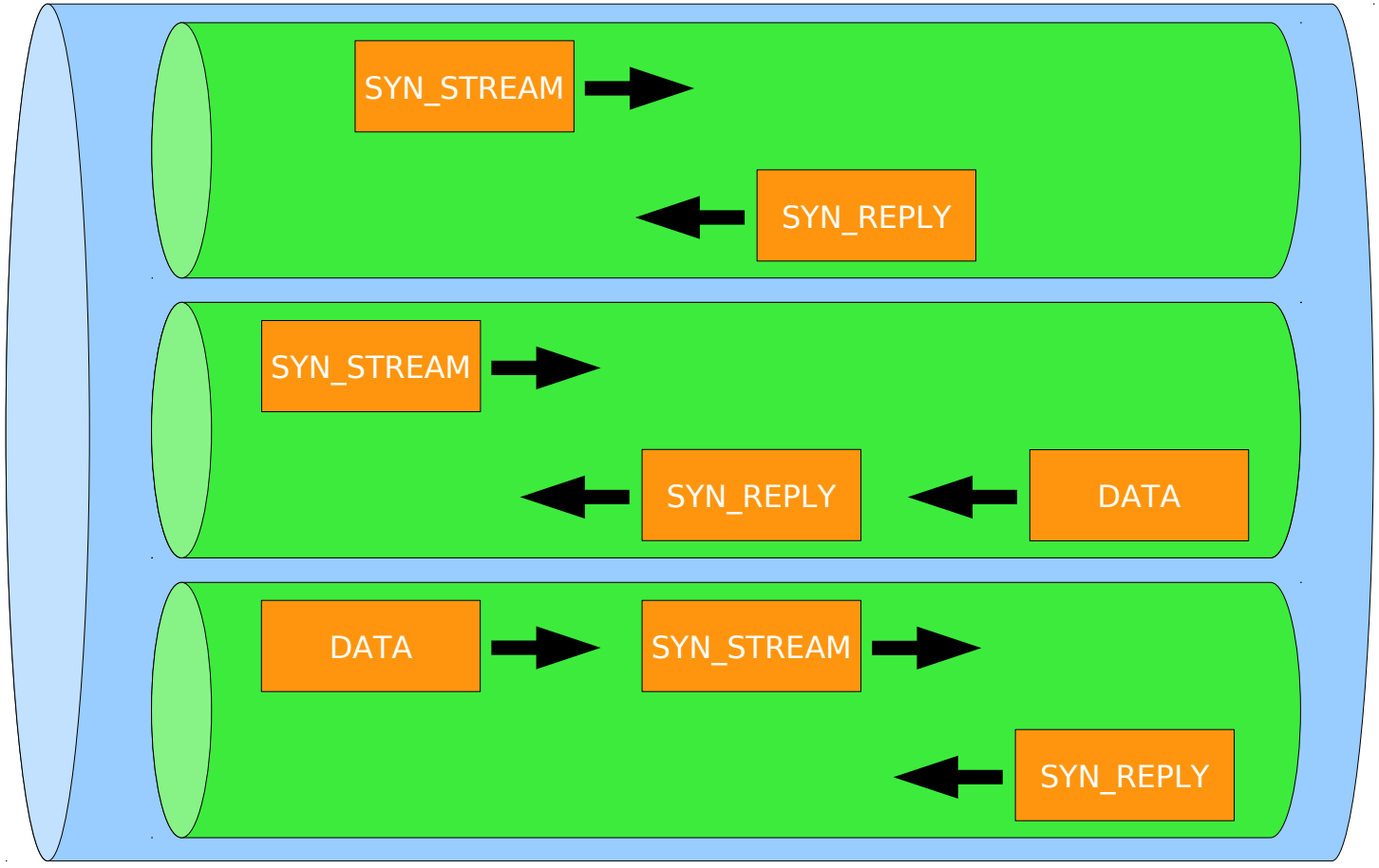
- **SPDY defines a new framing layer**
  - Binary, slightly more expensive than WebSocket's
  - Faster than HTTP to parse
  - Built to carry HTTP, but can carry other protocols



Simone Bordet
sbordet@intalio.com

SPDY Multiplexing

- **Multiplexing is built-in**



Simone Bordet
sbordet@intalio.com

SPDY Multiplexing

- **Multiplexing**
  - Allows to make better use of TCP connections
    - *Reduces TCP slow start*
  - Uses less resources on the server
  - Responses may be sent out of order
    - *Without waiting for slow responses*

- **Key point: reducing round trip waits**
  - Caused by limited connections and lack of multiplexing

Simone Bordet
sbordet@intalio.com

SPDY Semantics

- **SPDY is built for HTTP**
  - But allows WebSocket to be carried too
- **HTTP header compression**
  - Typical HTTP requests contains ~1 KiB of headers
    - *Most of them are constant*
  - Saves ~25% on first request, 80% and more on subsequent requests
- **Headers for typical requests to webtide.com:**
  - 568 bytes → 389 bytes → 26 bytes
  - 100% → 68% → 4.5%

Simone Bordet
sbordet@intalio.com

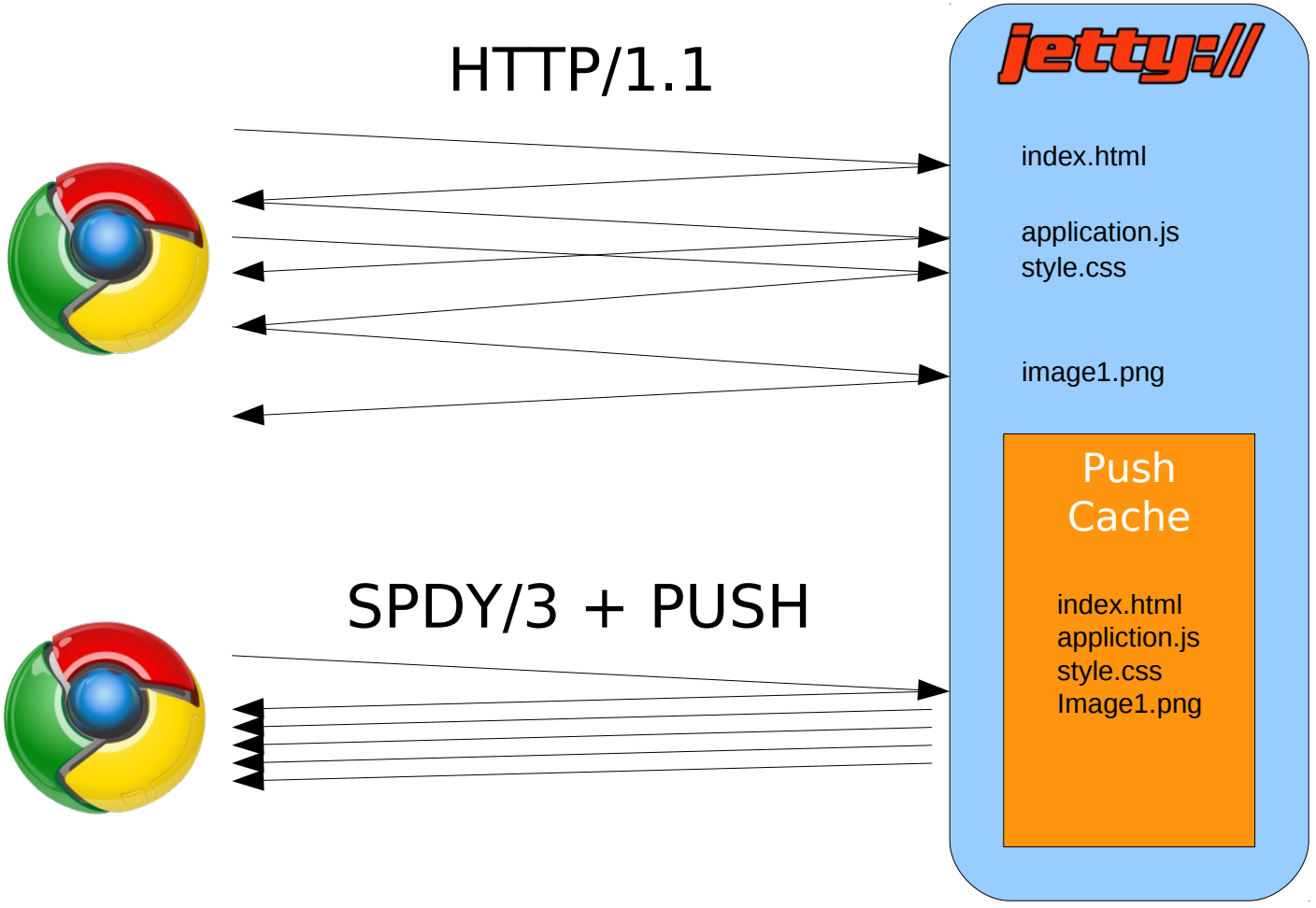
SPDY Push

- **SPDY Push**
  - Server pushes secondary resources that are associated to a primary resource
  - Works in collaboration with browser's cache for a better user experience

- **Jetty is one of the first SPDY servers that provides automated push functionalities**
  - Totally transparent for applications
  - Based on the “Referrer” header
    - *To associate primary and secondary resources*
  - Using “If-Modified-Since” header to avoid unnecessary pushes

Simone Bordet
sbordet@intalio.com

SPDY Push



Simone Bordet
sbordet@intalio.com

SPDY

- **How much faster than HTTP can SPDY be ?**
  - Early to say, but initial figures up to 25%
  - SPDY optimizations limited by HTTP “hacks”
    - *Domain sharding*
- **Can my JEE web application leverage SPDY ?**
  - Yes ! And without changes !

Simone Bordet
sbordet@intalio.com

SPDY & Jetty

- **Jetty 7 & Jetty 8**
  - SPDY added by mocking HTTP wire protocol
  - Multiplexing requires mock protocol per channel
- **Jetty 9**
  - Re architected to separate wire protocol from semantics
  - HTTP semantics shared between HTTP, SPDY
  - Supports 1:n Protocol:Semantics
  - Same separation of Websocket protocol from semantics
  - HTTP, SPDY, WebSocket, TLS 1st class citizens
- **PLEASE USE SPDY NOW!**
  - Your use case can make Jetty and the SPDY better

Simone Bordet
sbordet@intalio.com

SPDY Clients

- **Desktop**
  - Firefox 13
  - Chrome 4
  - Opera 12.1
  - Jetty Client 7,8, 9
- **Mobile**
  - Android Browser 4.1
  - Chrome for Android 18.0
  - Firefox Android 15.0

45.68% by StatCounter GlobalStats

Simone Bordet
sbordet@intalio.com

DEMO

Simone Bordet
sbordet@intalio.com

HTTP 2.0

- **HTTP 2.0 work has started**
  - IETF HTTPbis working group rechartered
- **Several Proposals received:**
  - Google - SPDY as the basis
  - Microsoft – websocket framing with HTTP semantics
- **SPDY Adopted as the starting point**
  - Unlikely to require TLS or NPN
  - Will define precise interactions with proxies
  - PLEASE let us end up with one framing layer

Simone Bordet
sbordet@intalio.com

Conclusions

- **The Web is evolving**
  - Web Protocols must keep the pace new loads
  - Applications must keep pace with new semantics
- **Try WebSocket today**
  - Production ready
- **Try SPDY today**
  - Check your application is ready for the future
  - Check the future is ready for your application

Simone Bordet
sbordet@intalio.com

References

- **WebSocket Protocol**
  - http://www.ietf.org/rfc/rfc6455.txt
- **JSR 340 (Servlet 3.1)**
  - http://jcp.org/en/jsr/detail?id=340
- **SPDY**
  - http://www.chromium.org/spdy/
- **Jetty**
  - http://eclipse.org/jetty

Simone Bordet
sbordet@intalio.com

Questions
&
Answers

Simone Bordet
sbordet@intalio.com