Java™

ORACLE

Building WebSocket Apps in Java using JSR 356

Arun Gupta
blogs.oracle.com/arungupta, @arungupta

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Primer on WebSocket
- JSR 356: Java API for WebSocket

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Interactive Web Sites

- HTTP is half-duplex
- HTTP is verbose
- Hacks for Server Push
  - Polling
  - Long Polling
  - Comet/Ajax
- Complex, Inefficient, Wasteful

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

WebSocket to the Rescue

- TCP based, bi-directional, full-duplex messaging
- Originally proposed as part of HTML5
- IETF-defined **Protocol**: RFC 6455
  - Handshake
  - Data Transfer
- W3C defined **JavaScript API**
  - Candidate Recommendation

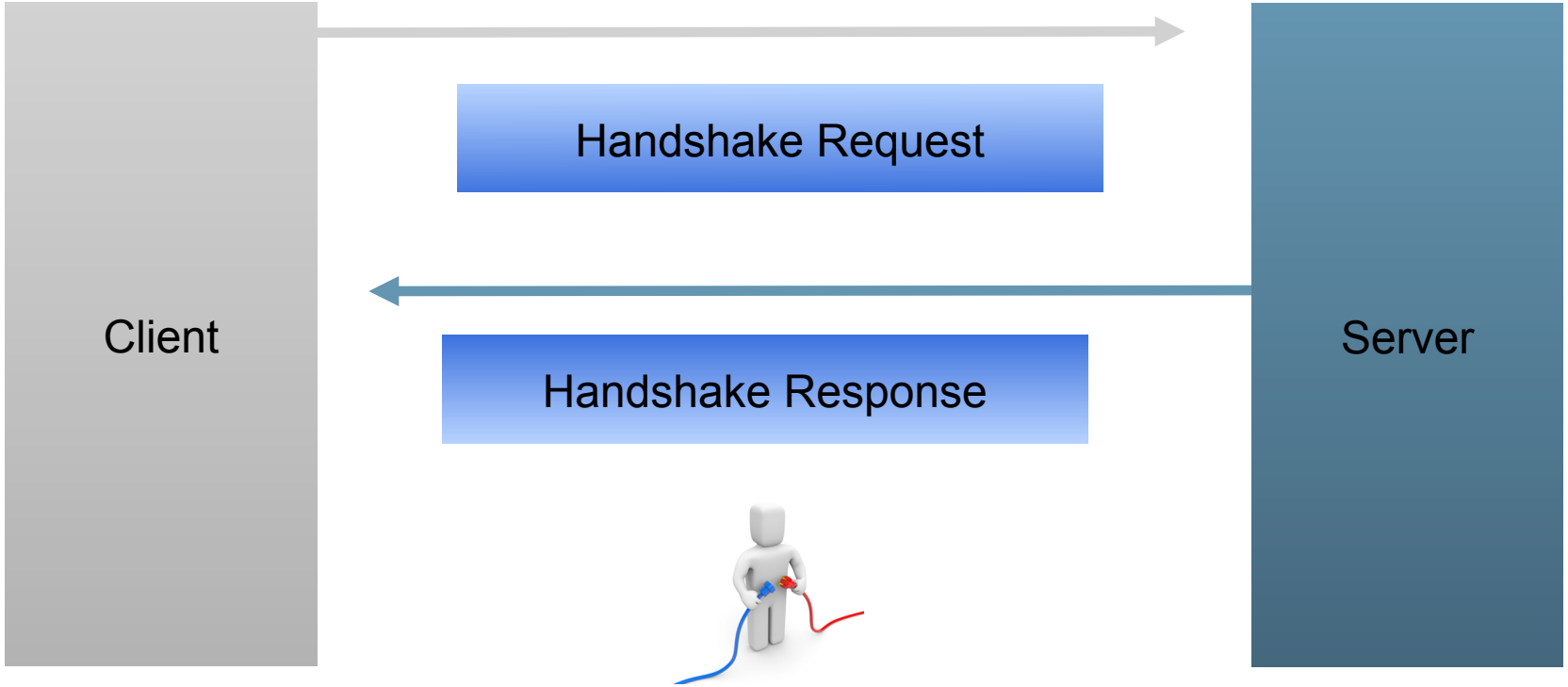
 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

What's the basic idea ?

- Upgrade HTTP to upgrade to WebSocket
  - Single TCP connection
  - Transparent to proxies, firewalls, and routers
- Send data frames in both direction (Bi-directional)
  - No headers, cookies, authentication
  - No security overhead
  - “ping”/”pong” frames for keep-alive
- Send message independent of each other (Full Duplex)
- End the connection

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Establish a connection



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Handshake Request

```
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: 13
```

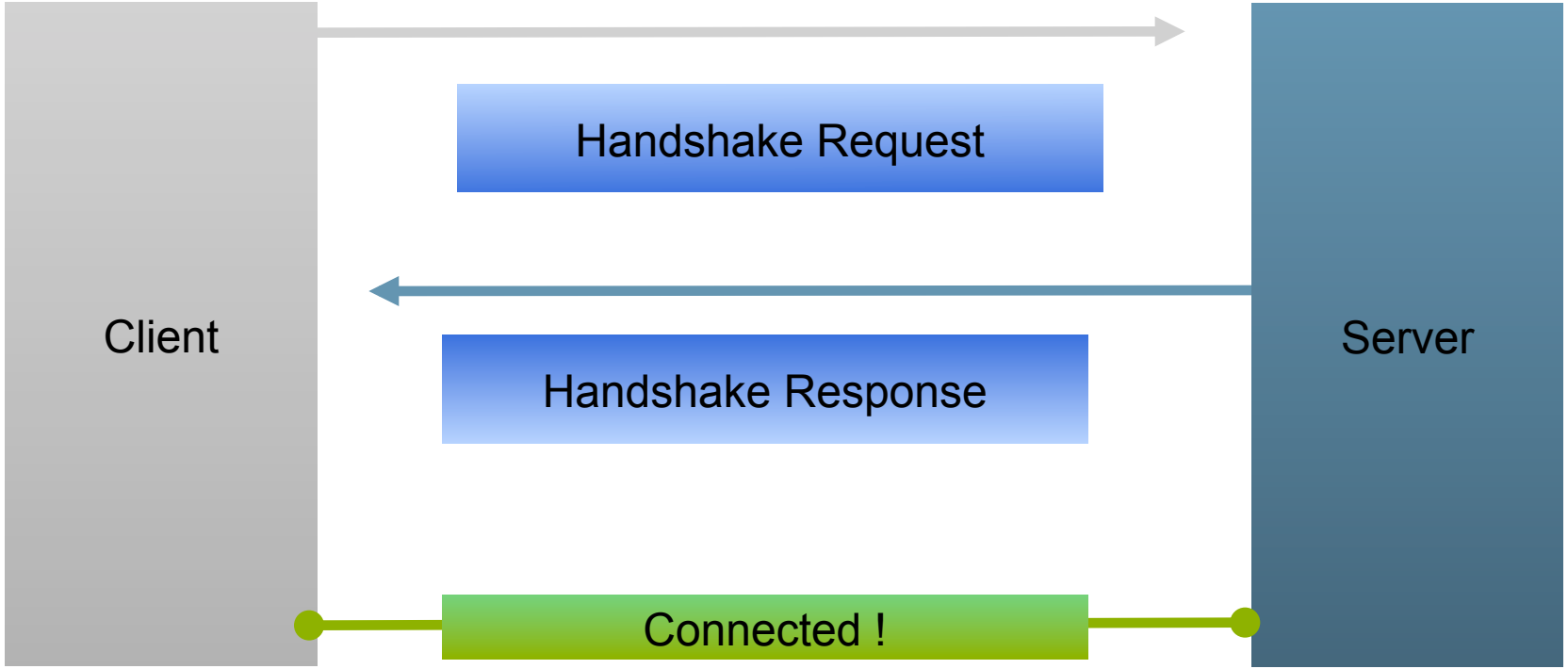
 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Handshake Response

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

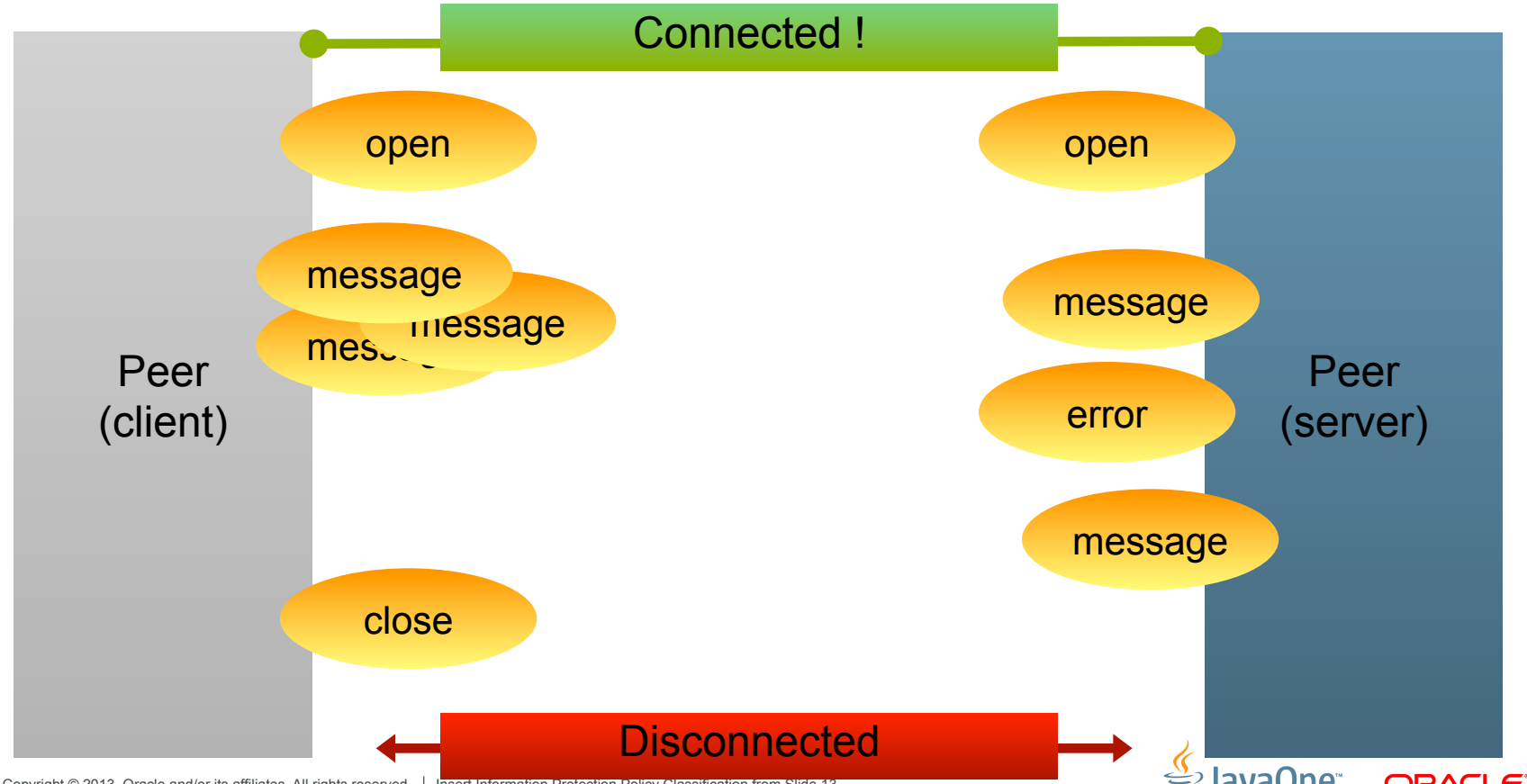
 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Establishing a Connection



 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

WebSocket Lifecycle



 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

WebSocket API

www.w3.org/TR/websockets/

```
[Constructor(DOMString url, optional (DOMString or DOMString[]) protocols)]
interface WebSocket : EventTarget {
  readonly attribute DOMString url;

  // ready state
  const unsigned short CONNECTING = 0;
  const unsigned short OPEN = 1;
  const unsigned short CLOSING = 2;
  const unsigned short CLOSED = 3;
  readonly attribute unsigned short readyState;
  readonly attribute unsigned long bufferedAmount;

  // networking
           attribute EventHandler onopen;
           attribute EventHandler onerror;
           attribute EventHandler onclose;
  readonly attribute DOMString extensions;
  readonly attribute DOMString protocol;
  void close([Clamp] optional unsigned short code, optional DOMString reason);

  // messaging
           attribute EventHandler onmessage;
           attribute DOMString binaryType;
  void send(DOMString data);
  void send(Blob data);
  void send(ArrayBuffer data);
  void send(ArrayBufferView data);
};
```

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Browser Support

Web Sockets - Working Draft

Bidirectional communication technology for web apps

Resources: Wikipedia Details on newer protocol WebSockets information

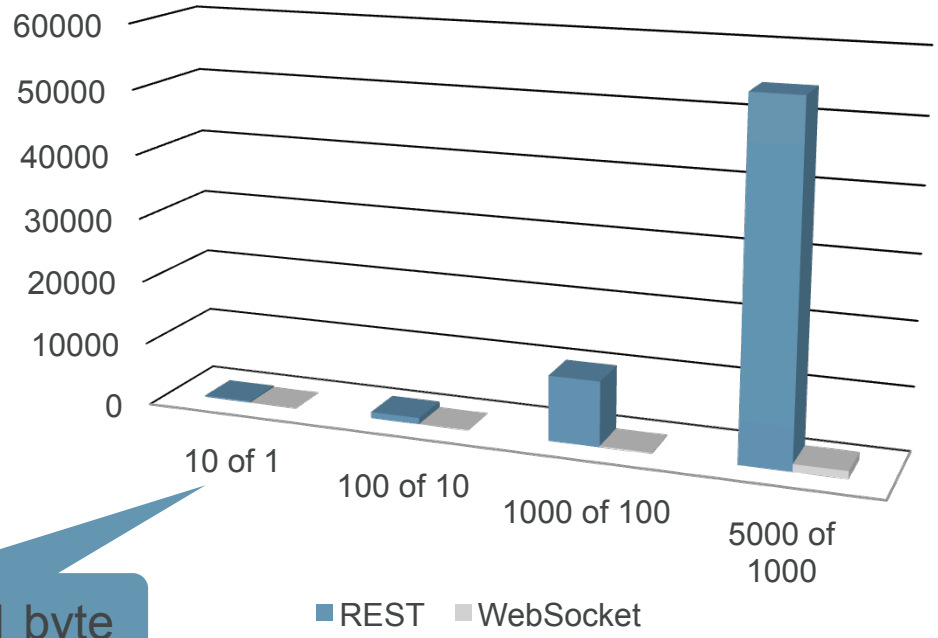
Global user stats*:	
Support:	57.1%
Partial support:	4.64%
Total:	61.74%

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android
20 versions back			4.0									
19 versions back			5.0									
18 versions back		2.0	6.0									
17 versions back		3.0	7.0									
16 versions back		3.5	8.0									
15 versions back		3.6	9.0									
14 versions back		4.0	10.0									
13 versions back		5.0	11.0									
12 versions back		6.0 Moz	12.0									
11 versions back		7.0 Moz	13.0									
10 versions back		8.0 Moz	14.0		9.0							
9 versions back		9.0 Moz	15.0		9.5-9.6							
8 versions back		10.0 Moz	16.0		10.0-10.1							
7 versions back		11.0	17.0		10.5							
6 versions back		12.0	18.0		10.6			2.1				
5 versions back	5.5	13.0	19.0	3.1	11.0			2.2		10.0		
4 versions back	6.0	14.0	20.0	3.2	11.1	3.2		2.3		11.0		
3 versions back	7.0	15.0	21.0	4.0	11.5	4.0-4.1		3.0		11.1		
2 versions back	8.0	16.0	22.0	5.0	11.6	4.2-4.3		4.0		11.5		
Previous version	9.0	17.0	23.0	5.1	12.0	5.0-5.1		4.1		12.0		
Current	10.0	18.0	24.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0	12.1	18.0	18.0
Near future		19.0	25.0		12.5				10.0			
Farther future		20.0	26.0									

http://caniuse.com/websockets

Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 44

REST vs WebSocket



10 messages of 1 byte

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

REST vs WebSocket

Payload size: 1000

How many times ?: 5000

Protocol: ☑ REST ☑ WebSocket

Echo Clear

REST Endpoint	WebSocket
Sending messages:	Sending messages:
Receiving messages:	Receiving messages:
Sending 10 messages of "1" byte(s) Total execution time: 220 ms	Sending 10 messages of "1" byte(s) Total execution time: 7 ms
Sending 100 messages of "10" byte(s) Total execution time: 986 ms	Sending 100 messages of "10" byte(s) Total execution time: 57 ms
Sending 1000 messages of "100" byte(s) Total execution time: 10210 ms	Sending 1000 messages of "100" byte(s) Total execution time: 179 ms
Sending 5000 messages of "1000" byte(s) Total execution time: 54449 ms	Sending 5000 messages of "1000" byte(s) Total execution time: 1202 ms

JSR 356 Specification

- Standard Java API for creating WebSocket Applications
- Transparent Expert Group
  - jcp.org/en/jsr/detail?id=356
  - java.net/projects/websocket-spec
- **FINAL**: Part of Java EE 7

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

JSR 356: Reference Implementation

- Tyrus: java.net/projects/tyrus
- Open source and transparent
- Integrated in GlassFish 4 Builds
  - download.java.net/glassfish/4.0/promoted

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Java API for WebSocket Features

- API for WebSocket Server and Client Endpoint
  - Annotated: `@ServerEndpoint`, `@ClientEndpoint`
  - Programmatic: `Endpoint`
    - WebSocket opening handshake negotiation
- Integration with Java EE Web container

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Hello World and Basics
POJO

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

JavaOne™ ORACLE®

Hello World

Annotated Endpoint

```
import javax.websocket.*;

@ServerEndpoint("/hello")
public class HelloBean {

    @OnMessage
    public String sayHello(String name) {
        return “Hello “ + name;
    }
}
```

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Annotations

Annotation	Level	Purpose
`@ServerEndpoint`	class	Turns a POJO into a Server Endpoint
`@ClientEndpoint`	class	Turns a POJO into a Client Endpoint
`@OnMessage`	method	Intercepts WebSocket Message events
`@PathParam`	method parameter	Flags a matched path segment of a URI-template
`@OnOpen`	method	Intercepts WebSocket Open events
`@OnClose`	method	Intercepts WebSocket Close events
`@OnError`	method	Intercepts errors during a conversation

Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

@ServerEndpoint attributes

`value`	Relative URI or URI template e.g. “/hello” or “/chat/{subscriber-level}”
`decoders`	list of message decoder classnames
`encoders`	list of message encoder classnames
`subprotocols`	list of the names of the supported subprotocols

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Custom Payloads

```
@ServerEndpoint(
    value="/hello",
    decoders={MyMessageDecoder.class},
    encoders={MyMessageEncoder.class}
)
public class MyEndpoint {
    . . .
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Custom Payloads – Text Decoder

```
public class MyMessageDecoder implements Decoder.Text<MyMessage> {

  public MyMessage decode(String s) {
    JsonObject jsonObject = Json.createReader(…).readObject();
    return new MyMessage(jsonObject);
  }

  public boolean willDecode(String string) {
    . . .
    return true;  // Only if can process the payload
  }
  . . .
}
```

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Java™ ORACLE®

Custom Payloads – Text Encoder

```
public class MyMessageEncoder implements Encoder.Text<MyMessage> {

  public String encode(MyMessage myMessage) {
    return myMessage.jsonObject.toString();
  }
  . . .
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Custom Payloads – Binary Decoder

```
public class MyMessageDecoder implements Decoder.Binary<MyMessage> {

  public MyMessage decode(byte[] bytes) {
    . . .
    return myMessage;
  }
  public boolean willDecode(byte[] bytes) {
    . . .
    return true;  // Only if can process the payload
  }
  . . .
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Which methods can be @OnMessage ?

- Exactly one of the following
  - Text: `String`, Java primitive or equivalent class, `String` and `boolean`, `Reader`, any type for which there is a decoder
  - Binary: `byte[]`, `ByteBuffer`, `byte[]` and `boolean`, `ByteBuffer` and `boolean`, `InptuStream`, any type for which there is a decoder
  - Pong messages: `PongMessage`
- An optional `Session` parameter
- 0..n `String` parameters annotated with `@PathParam`
- Return type: `String`, `byte[]`, `ByteBuffer`, Java primitive or class equivalent or any type for which there is a encoder

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Java™ ORACLE®

Sample Messages

- `void m(String s);`
- `void m(Float f, @PathParam("id")int id);`
- `Product m(Reader reader, Session s);`
- `void m(byte[] b);` or `void m(ByteBuffer b);`
- `Book m(int i, Session s, @PathParam("isbn")String isbn, @PathParam("store")String store);`

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Chat Server

```
@ServerEndpoint("/chat")
public class ChatBean {
    static Set<Session> peers = Collections.synchronizedSet(…);

    @OnOpen
    public void onOpen(Session peer) {
        peers.add(peer);
    }

    @OnClose
    public void onClose(Session peer) {
        peers.remove(peer);
    }

    . . .
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Chat Server

```
    . . .

    @OnMessage
    public void message(String message, Session client) {
        for (Session peer : peers) {
            peer.getBasicRemote().sendObject(message);
        }
    }
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

URI Template Matching

- Level 1 only

```
@ServerEndpoint("/orders/{order-id}")
public class MyEndpoint {
  @OnMessage
  public void processOrder(
    @PathParam("order-id")String orderId) {
    . . .
  }
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

WebSocket Client

```
@ClientEndpoint
public class HelloClient {
  @OnMessage
  public void message(String message, Session session) {
    // process message from server
  }
}

WebSocketContainer c = ContainerProvider.getWebSocketContainer();
c.connectToServer(HelloClient.class, “hello”);
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Hello World and Basics
Non-POJO

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Programmatic Endpoint

```
public class MyEndpoint extends Endpoint {

  @Override
  public void onOpen(Session session) {
    session.addMessageHandler(new MessageHandler.Text() {
      public void onMessage(String name) {
        try {
          session.getBasicRemote().sendText("Hello " + name);
        } catch (IOException ex) {
        }
      }
    });
  }
}
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Interface-driven Endpoint

Server Packaging

```
ServerEndpointConfiguration config =
   ServerEndpointConfigurationBuilder
    .create(MyEndpoint.class, “/foo”)
    .build();
```

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Server and Client Configuration

- Server
  - URI matching algorithm
  - Subprotocol and extension negotiation
  - Message encoders and decoders
  - Origin check
  - Handshake response
- Client
  - Requested subprotocols and extensions
  - Message encoders and decoders
  - Request URI

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Relationship with Dependency Injection

- Full Dependency Injection support required in endpoints
  - Field, method, constructor injection
- Interceptors permitted too

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Relationship with Servlet 3.1

- Allows a portable way to upgrade HTTP request
- New API
  - `HttpServletRequest.upgrade(ProtocolHandler handler)`

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

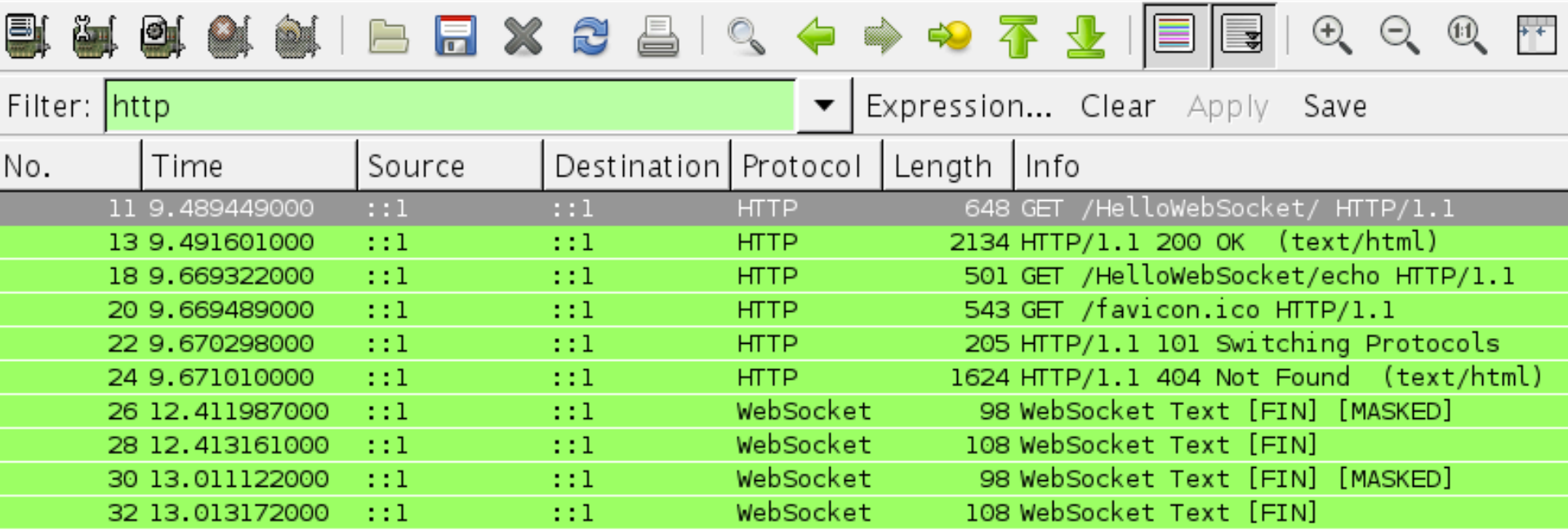
Security

- Authenticates using Servlet security mechanism during opening handshake
  - Endpoint mapped by `ws://` is protected using security model defined using the corresponding `http://` URI
- Authorization defined using `<security-constraint>`
  - TBD: Add/reuse security annotations
- Transport Confidentiality using `wss://`
  - Access allowed over encrypted connection only

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

How to view WebSocket messages ?

Capture traffic on loopback



No.	Time	Source	Destination	Protocol	Length	Info
11	9.489449000	::1	::1	HTTP	648	GET /HelloWebSocket/ HTTP/1.1
13	9.491601000	::1	::1	HTTP	2134	HTTP/1.1 200 OK (text/html)
18	9.669322000	::1	::1	HTTP	501	GET /HelloWebSocket/echo HTTP/1.1
20	9.669489000	::1	::1	HTTP	543	GET /favicon.ico HTTP/1.1
22	9.670298000	::1	::1	HTTP	205	HTTP/1.1 101 Switching Protocols
24	9.671010000	::1	::1	HTTP	1624	HTTP/1.1 404 Not Found (text/html)
26	12.411987000	::1	::1	WebSocket	98	WebSocket Text [FIN] [MASKED]
28	12.413161000	::1	::1	WebSocket	108	WebSocket Text [FIN]
30	13.011122000	::1	::1	WebSocket	98	WebSocket Text [FIN] [MASKED]
32	13.013172000	::1	::1	WebSocket	108	WebSocket Text [FIN]

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

How to view WebSocket messages ?

chrome://net-internals -> Sockets -> View live sockets

Capturing network events (5821) Stop Reset

Capture
Export
Import
Proxy
Events
Timeline
DNS
Sockets
SPDY
HTTP Pipelining
HTTP Cache

Filter: type:SOCKET is:active 3 of 108

	ID	Source	Description
	414219	SOCKET	linkhelp.clients.google.com:80
	414234	SOCKET	csi.gstatic.com:80
✓	414272	SOCKET	ws://localhost:8080/chat/chat

414272: SOCKET
ws://localhost:8080/chat/chat
Start Time: 2012-10-14 11:12:49.406

```
t=1350238369406 [st= 0] +SOCKET_ALIVE  [dt=?]
                         --> source_dependency = 4142
t=1350238369406 [st= 0]   +TCP_CONNECT  [dt=0]
                           --> address_list = ["[::1]
t=1350238369406 [st= 0]     TCP_CONNECT_ATTEMPT  [dt
                            --> address = "[::1]:808
t=1350238369406 [st= 0]   -TCP_CONNECT
                           --> source_address = "[::1
t=1350238369433 [st=27]    SOCKET_BYTES_SENT
                           --> byte_count = 415
```

Elements | Resources | Network | Sources | Timeline | Profiles | Audits | Console

Name
Path

websocket
/chat

Headers | Frames | Cookies

Data	Length	Time
Duke joined	11	4:46:59 PM
Duke2 joined	12	4:47:04 PM
Duke2 joined	12	4:47:04 PM

1 / 3 requests | 297 B / 3.8 KB tr

All | Documents | Stylesheets | Images | Scripts | XHR | Fonts | WebSockets | Other

 Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resources

- Specification
  - JSR: jcp.org/en/jsr/detail?id=356
  - Mailing Lists, JIRA, Archive: java.net/projects/websocket-spec
  - FINAL: Part of Java EE 7
- Reference Implementation
  - Tyrus: java.net/projects/tyrus
  - Integrated in GlassFish 4 builds

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Q & A

 | Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | Insert Information Protection Policy Classification from Slide 13

Java™