Ajax-Push for Revolutionary Enterprise Applications

Micha Kiener, mimacom ag

Head of Research and Innovation

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Agenda

- **Web2.0™**
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Web 2.0

A Web by the people, for the people.

- Documents on the web increasingly generated by users

- Out of the Information Age, into the Participation Age
- As a whole, the World Wide Web is a collaborative environment, but individual pages are only weakly so
- Are web user interfaces becoming more powerful?
- Is the user an HTTP client?

Ajax

Ajax is a state of mind.

- It was AJAX (Asynchronous JavaScript™ Technology with XML)
  - or Asynchronous JavaScript technology with XMLHttpRequest
  - now it's Ajax (not an acronym) because many different techniques satisfied the same goals
  - coined by Jesse James Garrett in 2005 to sell an insurance company on re-writing all their software
- Is the web defined by the W3C or by browser implementers? (Ajax does not exist in W3C universe yet.)
- Ajax decouples user interface from network protocol
- Ajax is the leading edge of the user interface possible with current popular browsers
- The user experience is important

The Asynchronous Web Revolution

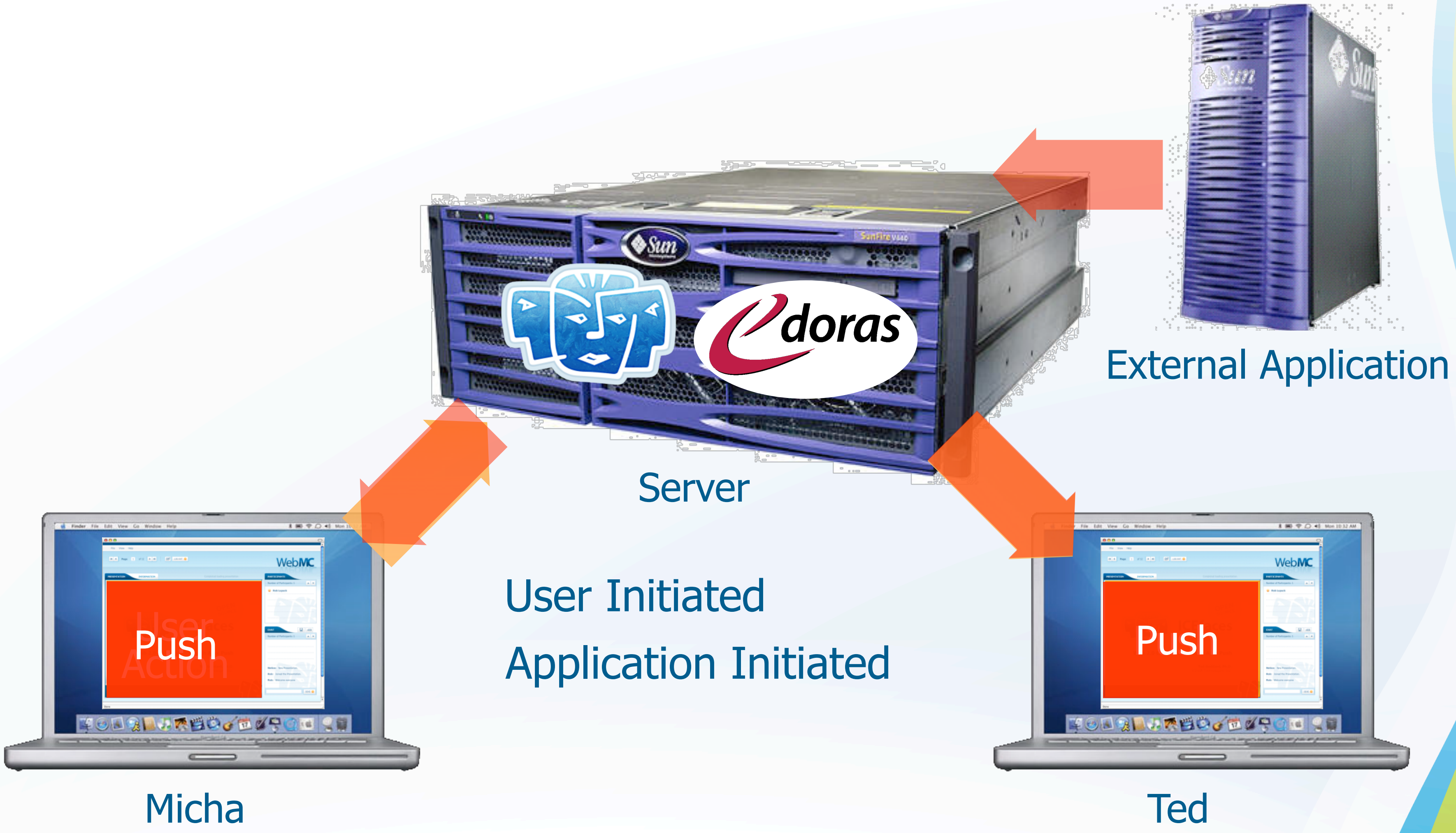
The Web enters the Participation Age.

- Ajax is still typically synchronous with user events
- Full asynchrony has updates pushed from server any time

- Update pages after they load
- Send users notifications
- Allow users to communicate and collaborate within the web application

- Called “Ajax Push”, “Comet”, or “Reverse Ajax”
  - This is the full realization of Ajax, now fully asynchronous

Server-mediated Collaboration



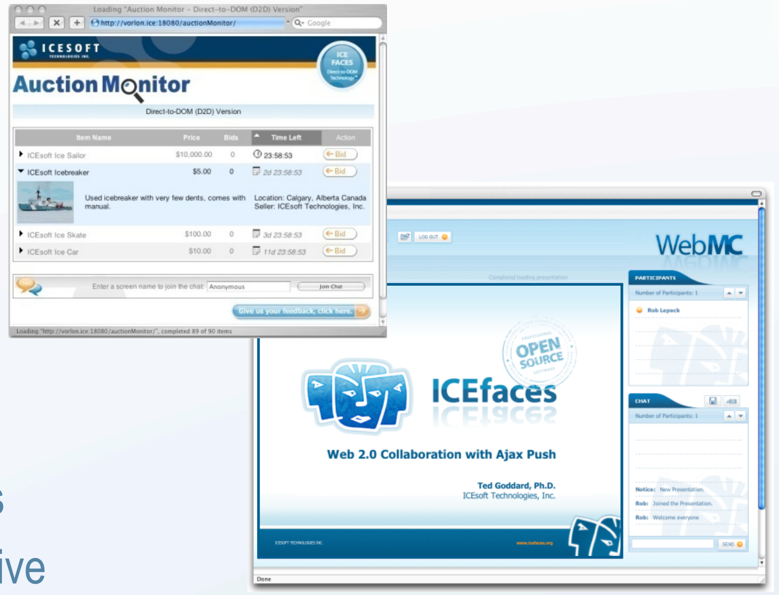
User Initiated

Application Initiated

Applications in the Participation Age

Application-mediated communication.

- Distance learning
- Collaborative authoring
- Auctions
- Shared WebDAV filesystem
- Blogging and reader comments
- SIP-coordinated mobile applications
- Hybrid chat/email/discussion forums
- Customer assistance on sales/support pages
- Multi-step business process made collaborative
- Shared trip planner or restaurant selector with maps
- Shared calendar, “to do” list, project plan
- Enterprise shared record locking and negotiation
- Games



Loading "Auction Monitor – Direct-to-DOM (D2D) Version"
http://vorlon.ice:18080/auctionMonitor/
Google
ICESOFT TECHNOLOGIES INC.
ICE FACES Direct-to-DOM Technology™
Auction Monitor
Direct-to-DOM (D2D) Version

Item Name	Price	Bids	Time Left	Action
ICEsoft Ice Sailor	$10,000.00	0	23:58:53	Bid
ICEsoft Icebreaker	$5.00	0	2d 23:58:53	Bid
ICEsoft Ice Skate	$100.00	0	3d 23:58:53	Bid
ICEsoft Ice Car	$10.00	0	11d 23:58:53	Bid

Used icebreaker with very few dents, comes with manual.

Location: Calgary, Alberta Canada
Seller: ICEsoft Technologies, Inc.

Enter a screen name to join the chat: Anonymous

Loading "http://vorlon.ice:18080/auctionMonitor/", completed 89 of 90 items

Join Chat
Give us your feedback, click here.
LOG OUT
WebMC
Completed loading presentation
OPEN SOURCE
ICEfaces
Web 2.0 Collaboration with Ajax Push
Ted Goddard, Ph.D.
ICEsoft Technologies, Inc.
ICESOFT TECHNOLOGIES INC.
www.icefaces.org
PARTICIPANTS
Number of Participants: 1
Rob Lepack
CHAT
Number of Participants: 1
SEND
Done

Notice: New Presentation.
Rob: Joined the Presentation.
Rob: Welcome everyone

irian
THE JAVA EXPERTS

JSFdays 2009

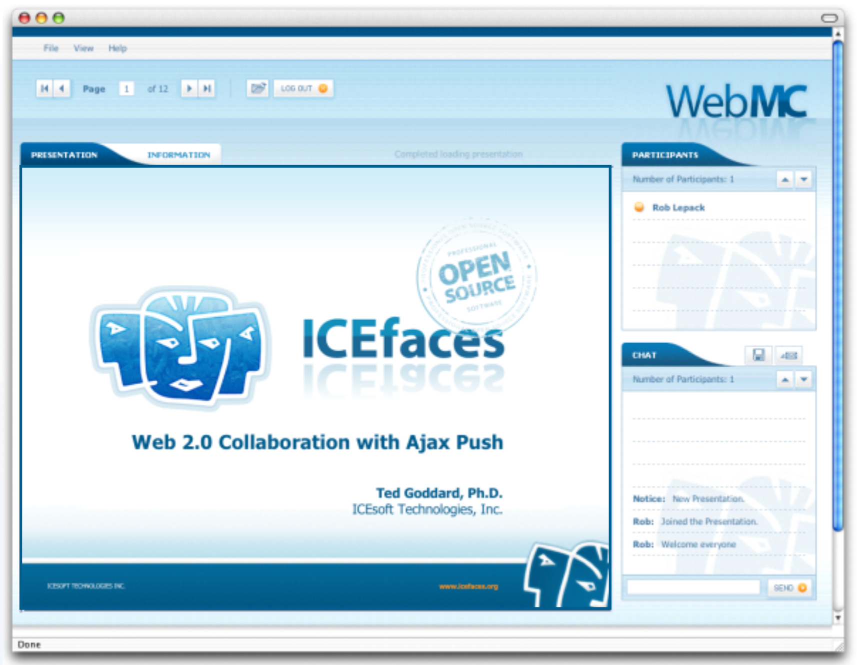
MIMACOM

Agenda

- Web2.0™
- **Multi-user Ajax Demo**
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Asynchronous Ajax Demo

GlassFish/Grizzly with ICEfaces WebMC


File
View
Help
Page
1
of 12
LOG OUT
WebMC
PRESENTATION
INFORMATION
Completed loading presentation
ICEfaces
OPEN SOURCE
Web 2.0 Collaboration with Ajax Push
Ted Goddard, Ph.D.
ICEsoft Technologies, Inc.
PARTICIPANTS
Number of Participants: 1
Rob Lepack
CHAT
Number of Participants: 1
Notice: New Presentation.
Rob: Joined the Presentation.
Rob: Welcome everyone
SEND
Done

http://webmc.icefaces.org

Agenda

- Web2.0™
- Multi-user Ajax Demo
- **Push for enterprise collaboration**
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Didn't we have that already?

Push-mechanisms in Rich-Clients

- Rich-Clients connected to the server in a keep-alive manner
- Full Java-API is available within the client for networking and event-handling
- Server can push an event to the client any time
  - Either by having the client polling for events (optionally combined with a heart-beat, ping-like request)
  - Or by callback from the server
- Since the technology behind is well-known and transparent, its easy to use push for collaborative features and updating mechanisms
- Rich-Clients were always claimed to support push-features

Showcase for Push

Features

- Collaboration through editing and pessimistic locking
- A list of Person objects which may be edited and created
- With pessimistic locking, a lock-object must be obtained before the object is editable
- If the lock is held by another user, it should be possible to notify him so he can release the lock
- When data is changed, all views should be automatically updated

Showcase for Push

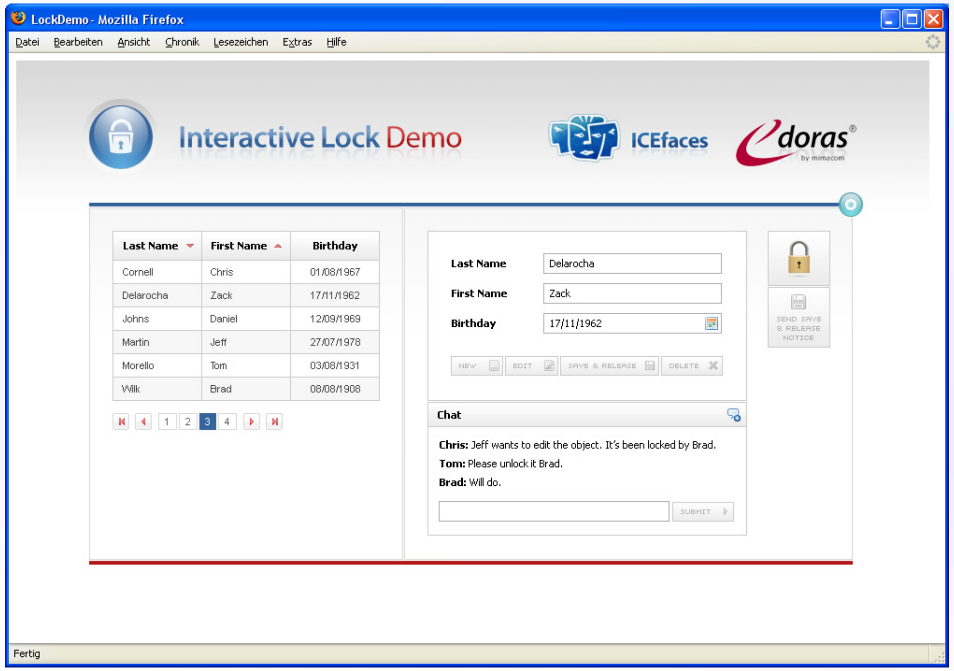
Features using Rich-Client Push or Ajax Push

- With Ajax, people stay longer on the same page, hence automatic page update is needed
- As the Web gets more and more social, collaborative tasks come in place
- Updating and collaboration are inherently asynchronous and need some push-mechanism to be fulfilled
- The next demo shows the collaborative features of the rich-client in an Ajax Push web-environment

Showcase for Push

Features using Rich-Client Push or Ajax Push

- Push in a Web-Client (Demo)
  - Automatically updating changed / added / removed data sets
  - Collaborative notifications in the context of pessimistic locking


LockDemo - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Interactive Lock Demo
ICEfaces
doras by mimacom
1 2 3 4
NEW EDIT SAVE & RELEASE DELETE
SEND SAVE & RELEASE NOTICE
Chat
SUBMIT
Fertig

Last Name	First Name	Birthday
Cornell	Chris	01/08/1967
Delarocha	Zack	17/11/1962
Johns	Daniel	12/09/1969
Martin	Jeff	27/07/1978
Morello	Tom	03/08/1931
Wilk	Brad	08/08/1908

Last Name	Delarocha
First Name	Zack
Birthday	17/11/1962

Chris: Jeff wants to edit the object. It's been locked by Brad.
Tom: Please unlock it Brad.
Brad: Will do.

irian THE JAVA EXPERTS

JSFdays 2009

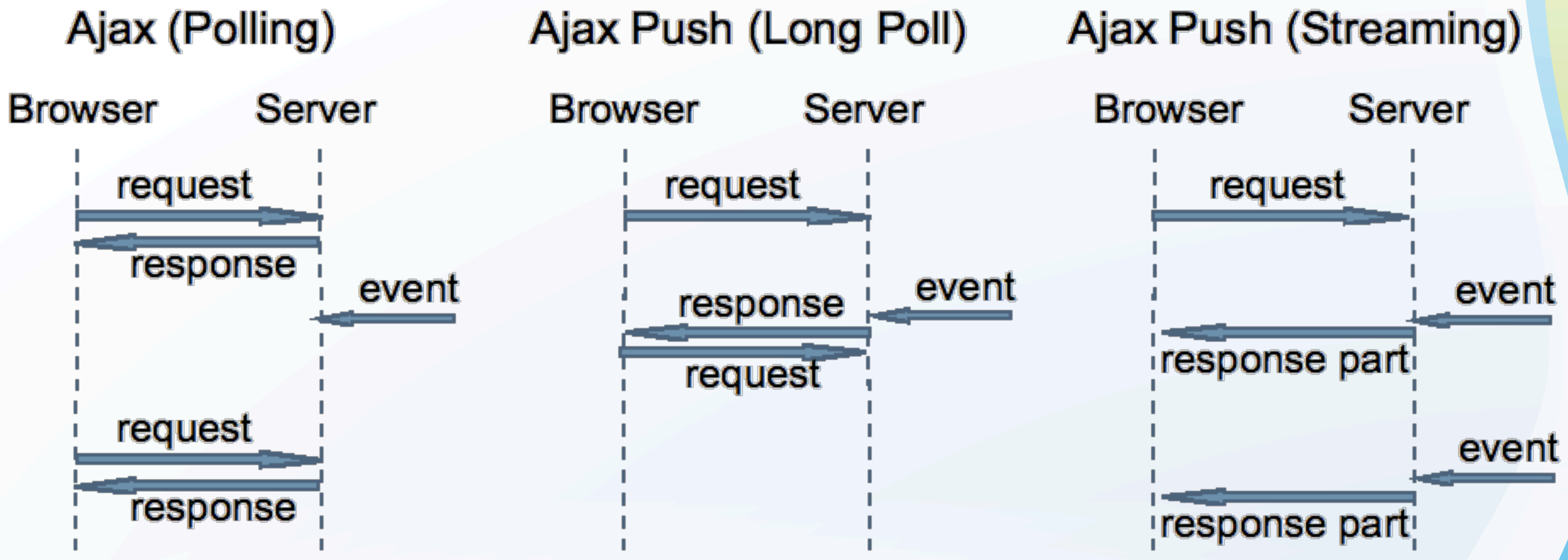
MIMACOM

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- **Asynchronous HTTP on the Wire**
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- Conclusion

Ajax Poll vs Ajax Push

Bending the rules of HTTP.



Bayeux / Cometd
JSON Pub/Sub.

```
[
  {
    "channel": "/some/name",
    "clientId": "83js73jsh29sjd92",
    "data": { "myapp" : "specific data", value: 100 }
  }
]
```

- JSON Messages are published on specified channels
- Channel operations: connect, subscribe, unsubscribe, etc.
- Multiple transports: polling, long-polling, iframe, flash
- Server implementations: Perl, Python, Java™ programming language
- Server-side reflector with no server-side application possible

Ajax Push

HTTP message flow inversion.

```
GET /auctionMonitor/block/receive-updates?icefacesID=1209765435 HTTP/1.1
Accept: */*
Cookie: JSESSIONID=75CF2BF3E03F0F9C6D2E8EFE1A6884F4
Connection: keep-alive
Host: vorlon.ice:18080

HTTP/1.1 200 OK
Content-Type: text/xml;charset=UTF-8
Content-Length: 180
Date: Thu, 27 Apr 2006 16:45:25 GMT
Server: Apache-Coyote/1.1

<updates>
  <update address="_id0:_id5:0:chatText">
    <span id="_id0:_id5:0:chatText">Howdy</span>
  </update>
</updates>
```

Chat message “Howdy”

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- **Asynchronous HTTP and the Server**
- Developing Asynchronous Applications
- Conclusion

Servlet Thread Catastrophe

Strangled by a thread for every client.

GET /updates HTTP/1.1

Connection: keep-alive

GET /updates HTTP/1.1

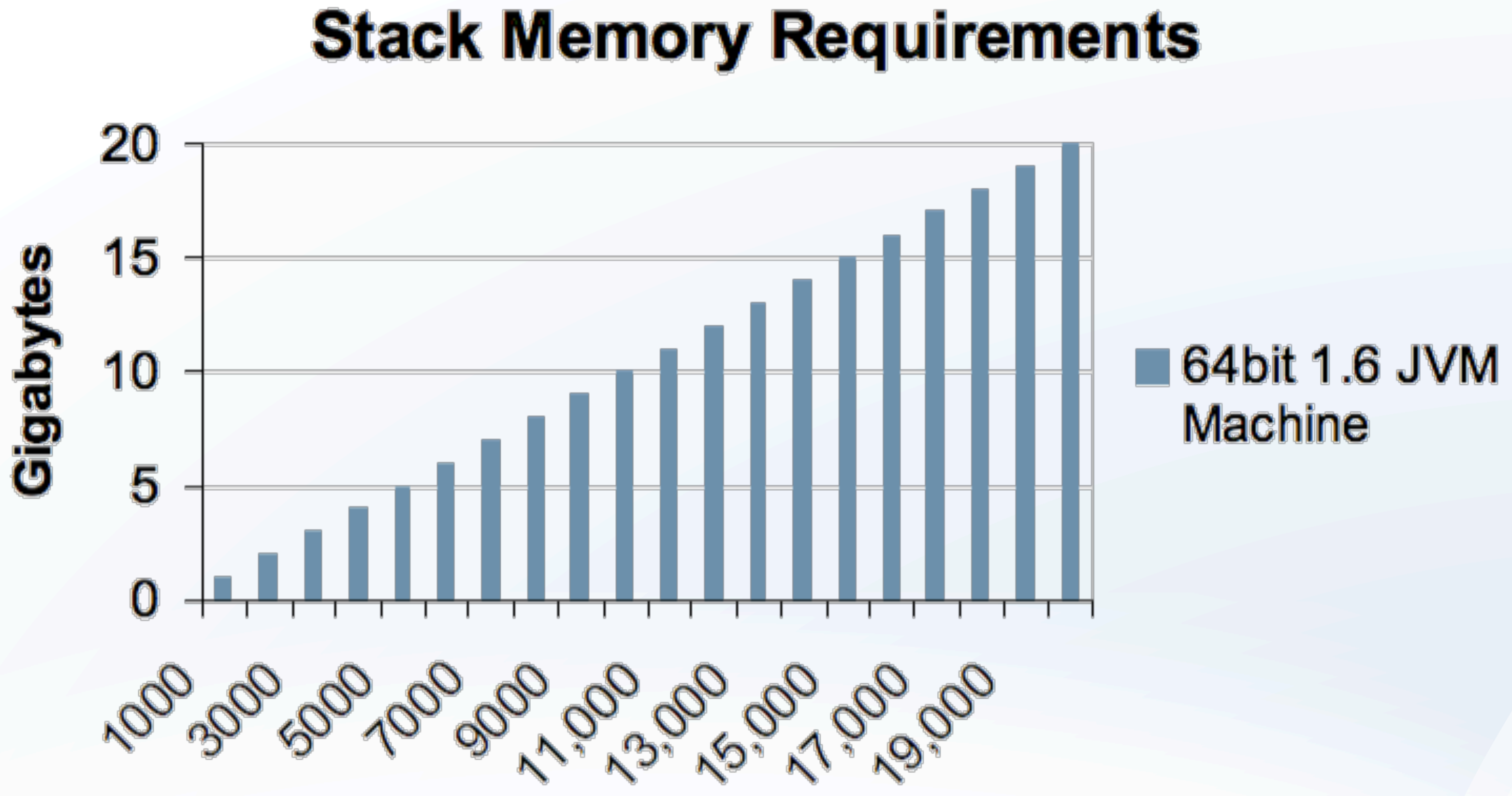
Connection: keep-alive

GET /updates HTTP/1.1

Connection: keep-alive

Architecture Challenges

The serious effect of blocking threads.



Jetty

service() will resume shortly.

```
import org.mortbay.util.ajax.Continuation;

service(request, response) {
    Continuation continuation = ContinuationSupport
        .getContinuation(request, this);
    ...
    continuation.suspend();
    response.getWriter().write(message);
}
```

Asynchronously and elsewhere in the application ...

```
message.setValue(“Howdy”);
continuation.resume();
```

Tomcat 6

Eventful Comet.

```
import org.apache.catalina.CometProcessor;

public class Processor implements CometProcessor {

public void event(CometEvent event)   {
     request = event.getHttpServletRequest();
     response = event.getHttpServletResponse();

     if (event.getEventType() == EventType.BEGIN) { ...
     if (event.getEventType() == EventType.READ)  { ...
     if (event.getEventType() == EventType.END)   { ...
     if (event.getEventType() == EventType.ERROR) { ...
}
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
response.getWriter().write(message);
event.close();
```

GlassFish

Suspend with Grizzly

```
 CometContext context =
        CometEngine.getEngine().register(contextPath);
    context.setExpirationDelay(20 * 1000);

    SuspendableHandler handler = new SuspendableHandler();
    handler.attach(response);
    cometContext.addCometHandler(handler);

class SuspendableHandler implements CometHandler  {

    public void onEvent(CometEvent event) {
        response.getWriter().println(event.attachment());
        cometContext.resumeCometHandler(this);
   }
```

Asynchronously and elsewhere in the application ...

```
message.setValue("Howdy");
cometContext.notify(message);
```

Servlet 3.0

Future Asynchronous Standard.

- Defined by JSR-315 Expert Group
- DWR, Jetty, Tomcat, GlassFish project, and ICEfaces participants
- Standard asynchronous processing API being defined
  - Asynchronous I/O
  - Suspendible requests
  - Delivery guarantee not included

- Will improve portability of DWR, Cometd, and ICEfaces
- (But unless you write Servlets today, this API will be hidden by your chosen Ajax framework.)

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- **Developing Asynchronous Applications**
- Conclusion

JavaScript Polling

Are we there yet? Are we there yet? Are we there yet? ...

```
function poll()  {
  setTimeout('poll()', 10000);
  req = new XMLHttpRequest();
  req.onreadystatechange = update();
  req.open("POST", "http://server/getMessage.jsp");
}

function update() {
  chatLog.innerHTML = req.responseText;
}

poll();
```

Cometd

Distributed, loosely coupled, scripting

JavaScript

```
function update(message) {
  chatLog.innerHTML = message.data.value;
}
...
  cometd.subscribe(“chat”, remoteTopics, “update”)
  cometd.publish(“chat”, message)
```

Java

```
import dojox.cometd.*;

Channel channel = Bayeux.getChannel(“chat”, create);
channel.subscribe(client);
```

Asynchronously and elsewhere in the application ...

```
message.setValue(“Howdy”);
channel.publish(client, message, “chat text”);
```

DWR
JavaScript RPC

```
import org.directwebremoting.proxy.dwr.Util;

scriptSessions =
    webContext.getScriptSessionsByPage(currentPage);
    util = new Util(scriptSessions);
```

To “Reverse Ajax” and invoke arbitrary JavaScript:

```
util.addScript(ScriptBuffer script);
```

Asynchronously and elsewhere in the application ...

```
util.setValue(“form:chat:_id3”, “Howdy”);
```

ICEfaces

Preserve MVC with Transparent Ajax.

PageBean.java

```
public class PageBean {
   String message;

   public String getMessage() {
      return message;
   }

   public void setMessage(String message) {
      this.message = message;
   }
}
```

Presentation Model

Page.xhtml

```
<f:view
   xmlns:f="http://java.sun.com/jsf/core"
   xmlns:h="http://java.sun.com/jsf/html" >

  <html>
    <body>
      <h:form>
        <h:inputText
           value="#{pageBean.message}" />
      </h:form>
    </body>
  </html>

</f:view>
```

Declarative User Interface

A language for Ajax Push that preserves Designer and Developer roles

ICEfaces

High level push.

```
import org.icefaces.application.SessionRenderer;
```

To update all users in the application:

```
SessionRenderer.render(SessionRenderer.ALL_SESSIONS);
```

Or to keep track of groups of users:

```
SessionRenderer.addCurrentSession("chat");
```

Asynchronously and elsewhere in the application ...

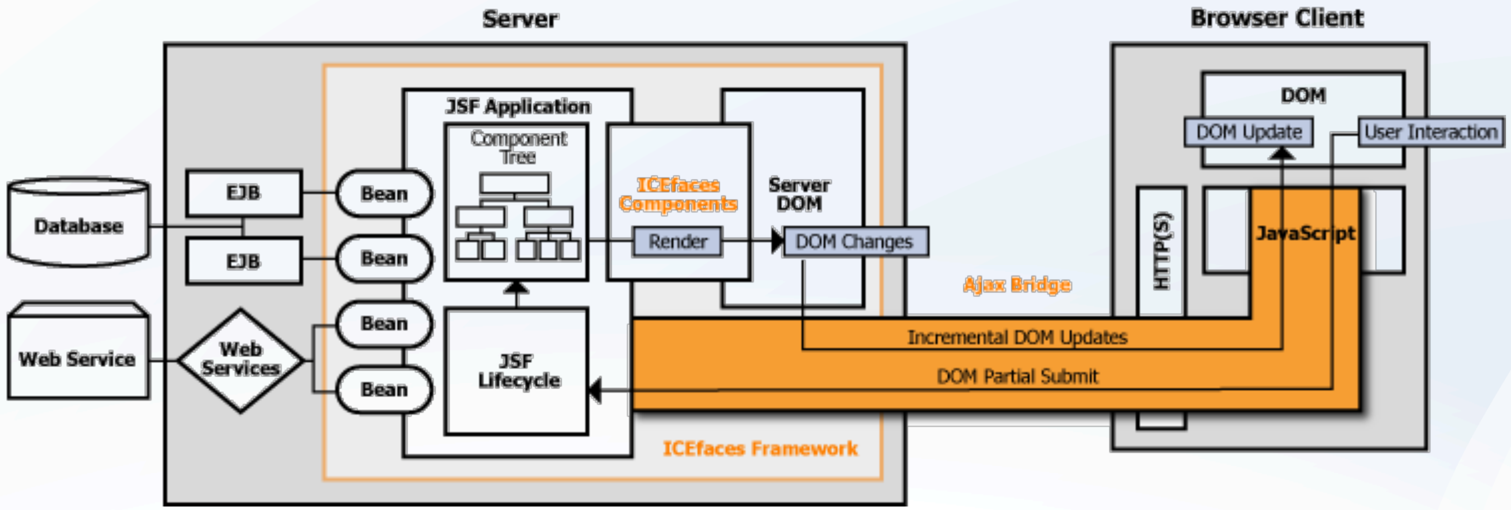
```
message.setValue("Howdy");
SessionRenderer.render("chat");
```

The JSF lifecycle runs and each user's page is updated from the component tree and current model state.

ICEfaces

Incremental updates

- Component-tree is maintained server-side
- Ajax or Ajax-Push (postback) invokes normal JSF lifecycle
- After Render Response phase, only incremental updates to the DOM-tree are sent back to client



irian THE JAVA EXPERTS

JSFdays 2009

MIMACOM

Agenda

- Web2.0™
- Multi-user Ajax Demo
- Push for enterprise collaboration
- Asynchronous HTTP on the Wire
- Asynchronous HTTP and the Server
- Developing Asynchronous Applications
- **Conclusion**

Summary
The Asynchronous Web Revolution is Now

- The Asynchronous Web will revolutionize human interaction
- Ajax Push is the key to enterprise collaboration for the Web
- Push can scale with Asynchronous Request Processing
- ICEfaces (www.icefaces.org) and edoras (www.edorasframework.org) provide the high-level capabilities for enterprise collaboration features in your application

Any Questions?

Micha Kiener, Head of Research and Innovation
micha.kiener@mimacom.ch

irian
THE JAVA EXPERTS

JSFdays 2009

MIMACOM