Java is a trademark of Sun Microsystems, Inc.

Speaker logo centered below image

Java

JavaOne℠

Developing Revolutionary Web Applications, Using Ajax Push or Comet

Doris Chen Ph.D., Carol McDonald, Justin Bolter

Sun Microsystems, Inc.

Technology Evangelists

Instructor-Led Hands-on Labs

- Instructor(s) will provide background for exercises
- Exercises are self-paced
- Hard-copy and online lab guides are available
- Use suggested durations as a guide
- Raise your hand at any time for assistance
- To get the most out of the lab...
- **Read** the lab guide, especially the background

Getting Started

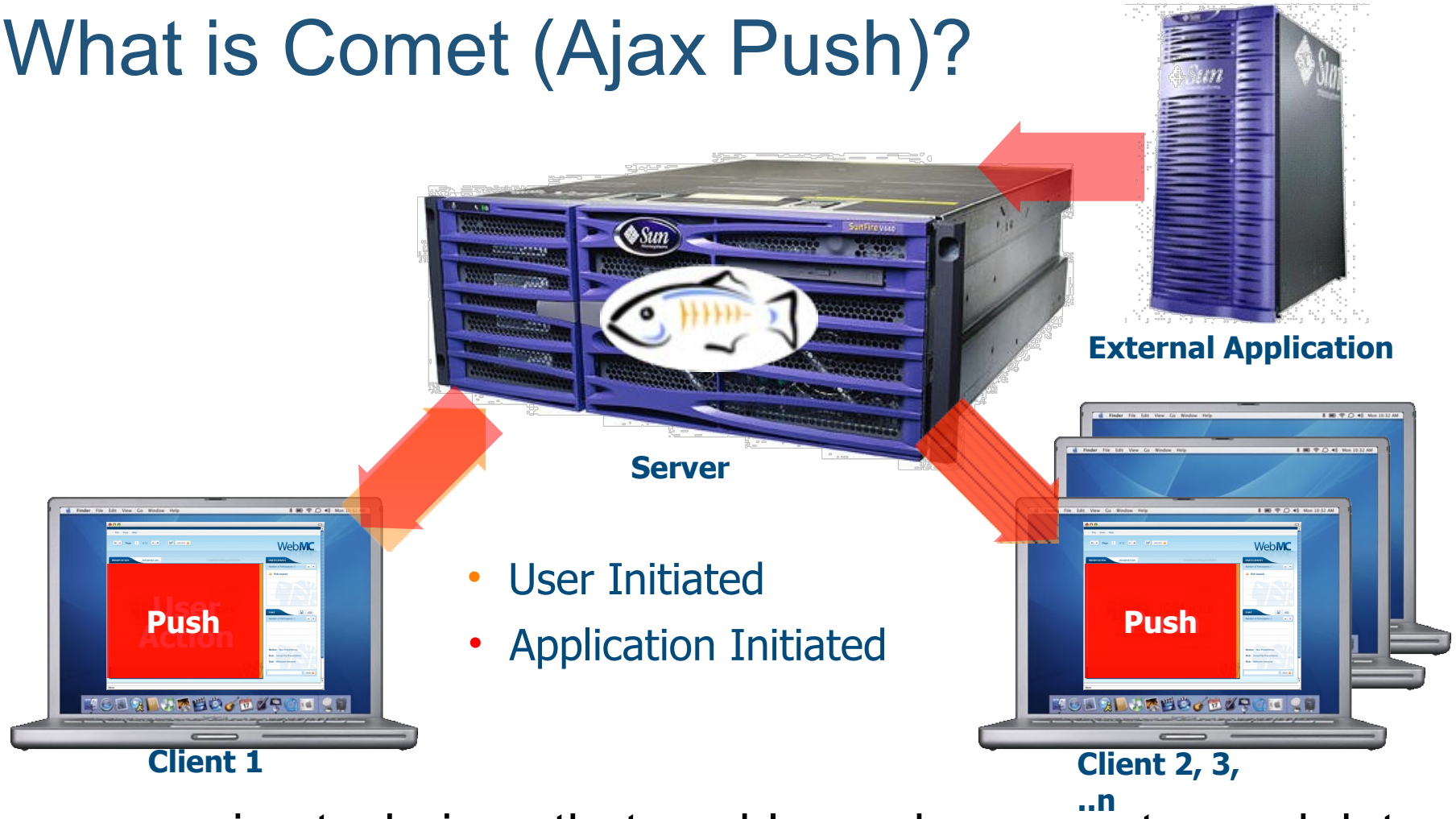
- If you have not logged in, log in with
  - username: lab5558
    - password: hol009
- Online lab guide will open in a browser window
- All necessary software and lab files are already installed on your lab machine

Applications in the Participation Age

Application-mediated communication.

- GMail and GTalk
- Meebo
- 4homemedia.com
(using GlassFish project's Comet)
- Distance learning
- Hybrid chat/email/discussion forums
- Games
- Auctions
- Blogging and reader comments
- SIP-coordinated mobile applications
- Shared trip planner or restaurant selector with maps
- Shared calendar, “to do” list, project plan
-

What is Comet (Ajax Push)?

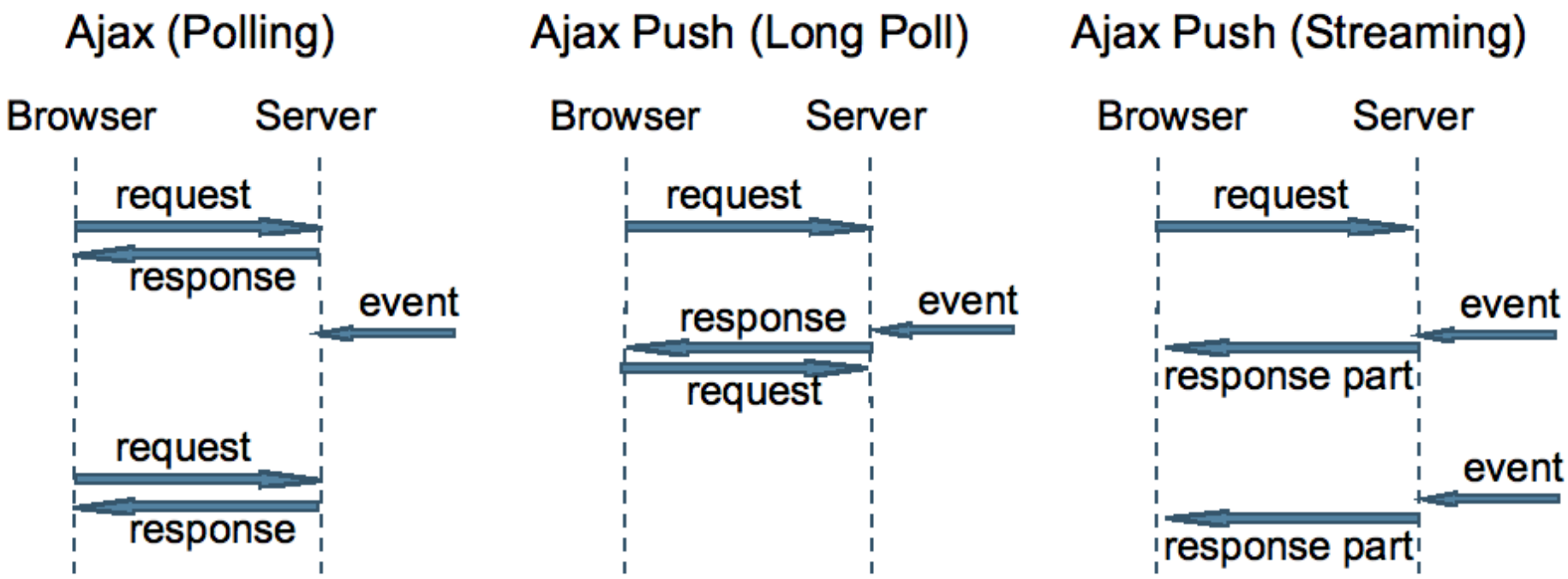


- User Initiated
- Application Initiated

A programming technique that enables web servers to send data to the client without having any need for the client to request for it

Ajax Poll vs Ajax Push

Bending the rules of HTTP.

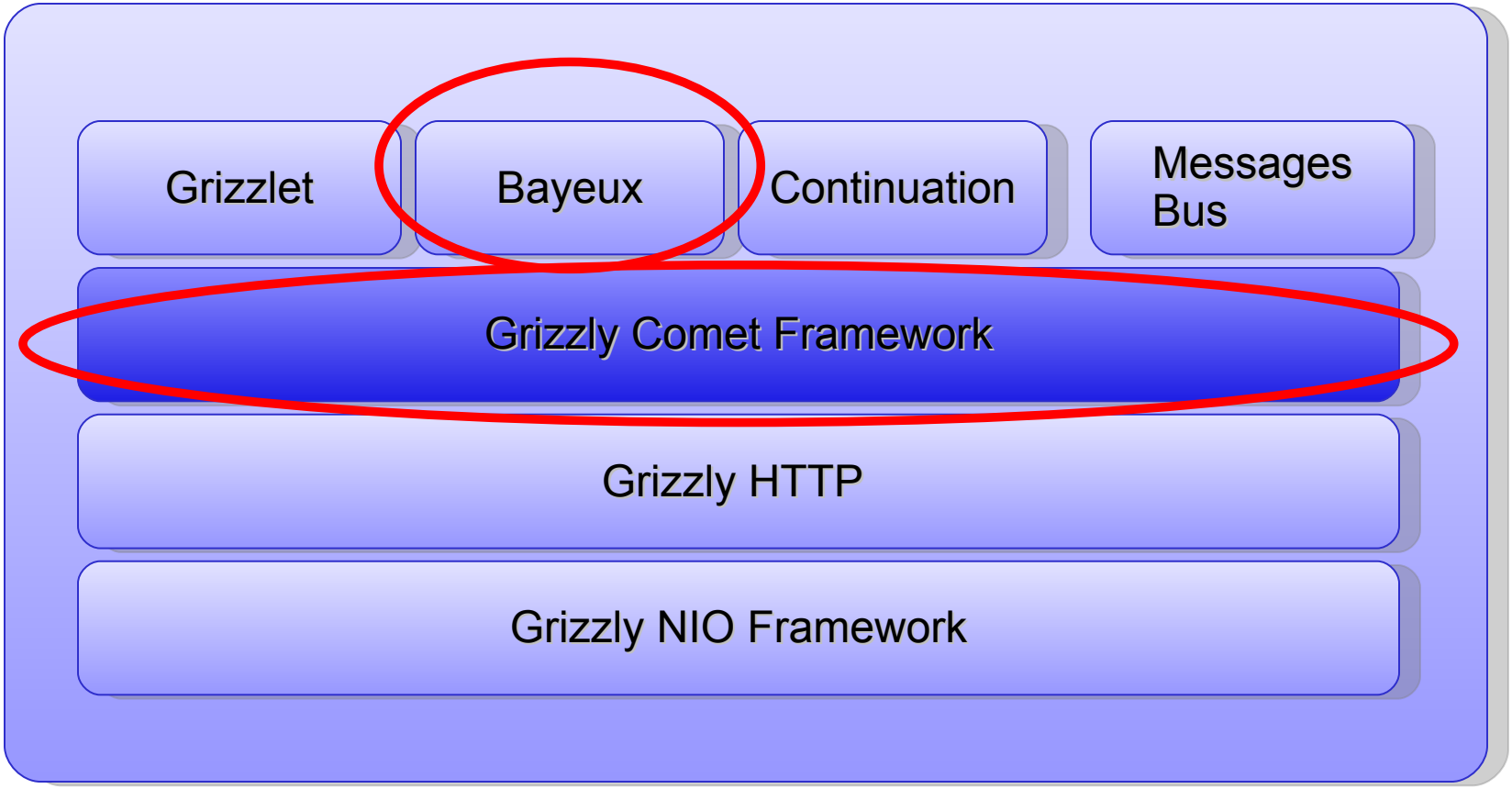


- **Comet** Refers to both the Long Polling and Streaming methods of web programming

Introduction to GlassFish Grizzly Comet

- Grizzly Comet is a framework that ship with GlassFish v1|2|3, and can also be embedded into any application using the Grizzly Embedded interface (no GlassFish involved)
- The Grizzly Comet Framework includes a set of components that can be used for building Comet based application:
  - Grizzly Comet Framework, Continuation, Grizzlet, Messages Bus, Bayeux support

Grizzly Comet Components



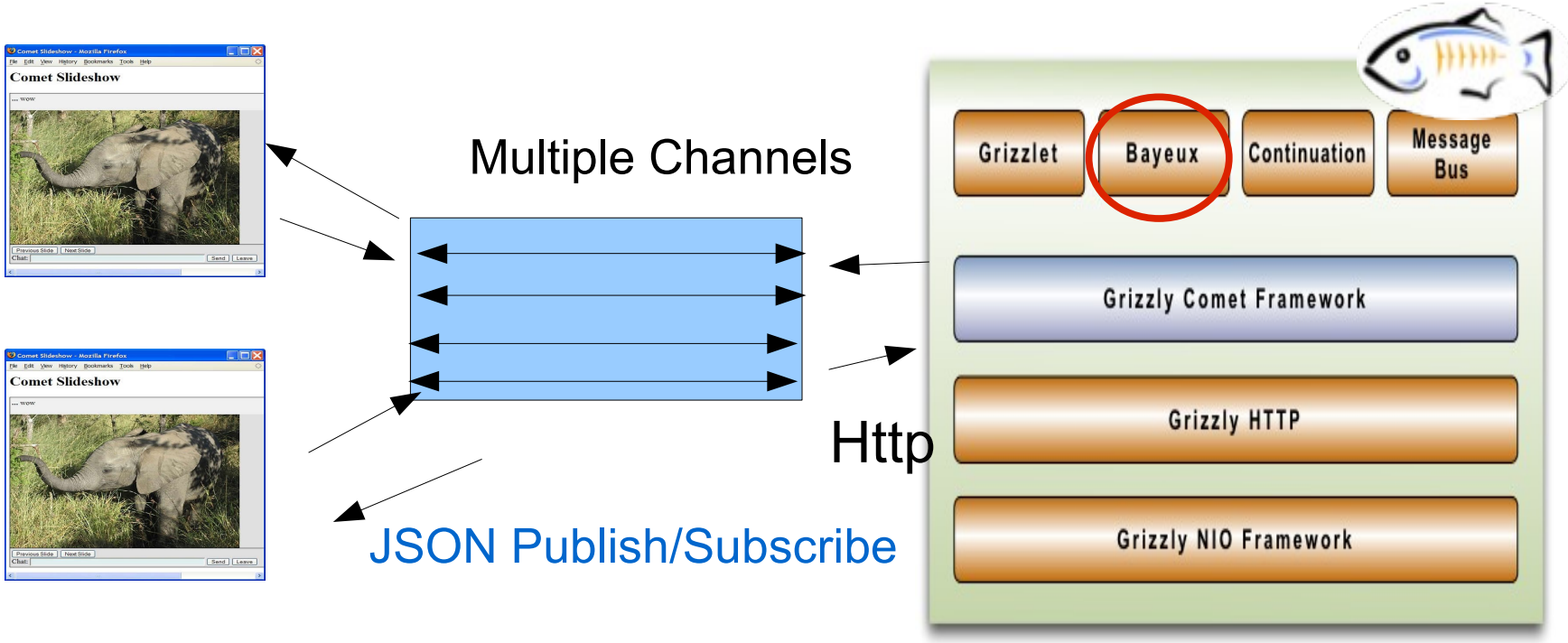
Exercises

- Exercise 1: Build a Comet Chat Application
  - Expected duration: 25 minutes
- Exercise 2: Build a Comet Slideshow Application
  - Expected duration: 20 minutes
- Exercise 3: Server Interact with Chat Application
  - Expected duration: 20 minutes
- Exercise 4: Build a Two Way Tic-tac-toe Game
  - Expected duration: 40 minutes

Exercise 1 Build a chat applicaton using cometd

Bayeux is a JSON-based protocol for clients to register interest in events and for servers to deliver them in a **publish-subscribe** model

Cometd Dojo JavaScript client Implementation of Bayeux Protocol

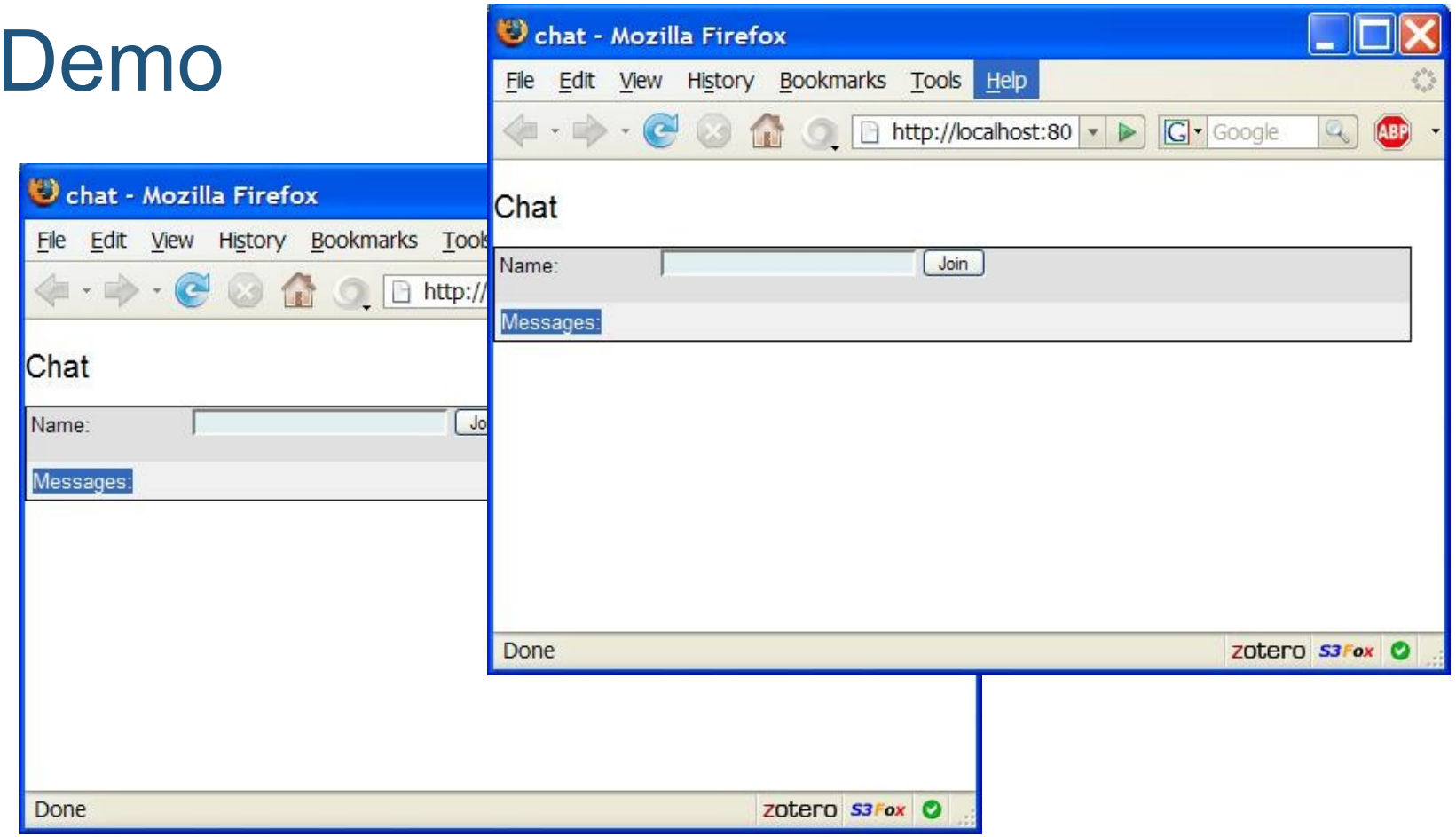


JSON Messages are published on specified channels
Channel operations: connect, subscribe, unsubscribe

Exercise 1: Build a Comet Chat Application (25 minutes)

Build a chat applicaton using cometd and Bayeux

Demo



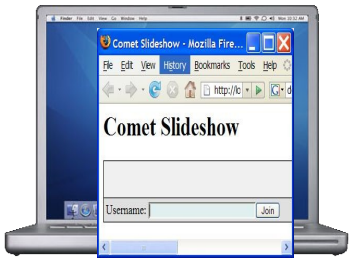
Cometd Step1 : Initialize

```
dojo.require("dojox.cometd");
```

```
//Initialize a connection to the given Comet server:
the GlassFish Grizzly Bayeux servlet

dojox.cometd.init("serverURL");
```

Comet Server



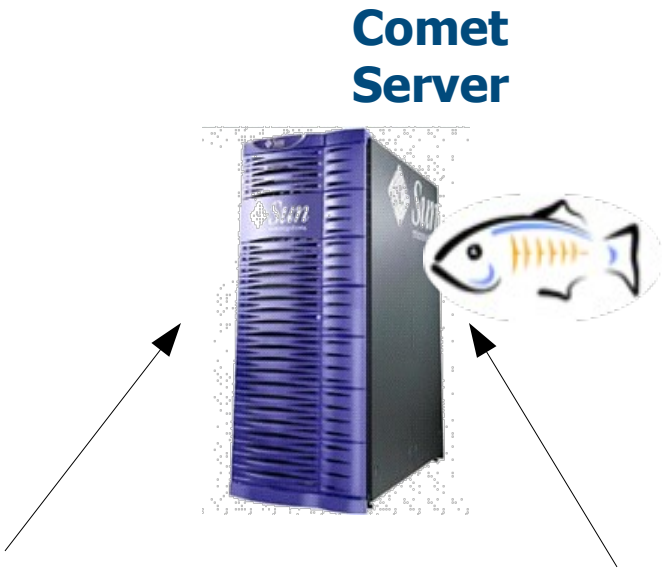
Client 1

Client 2

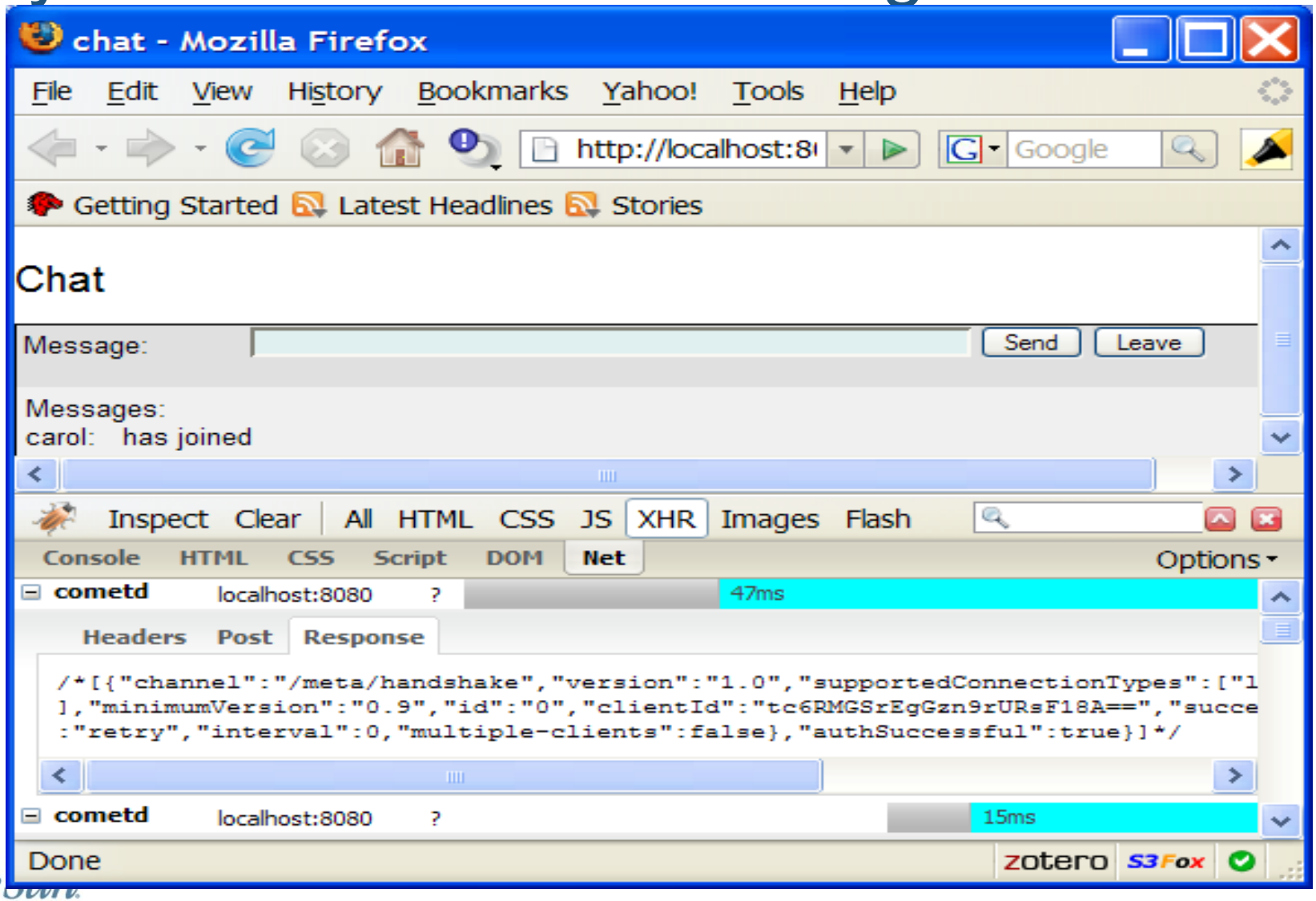
Cometd Step 2: Subscribe

```
dojox.cometd.subscribe(room.channel, room,
                       "chatCallback");
```

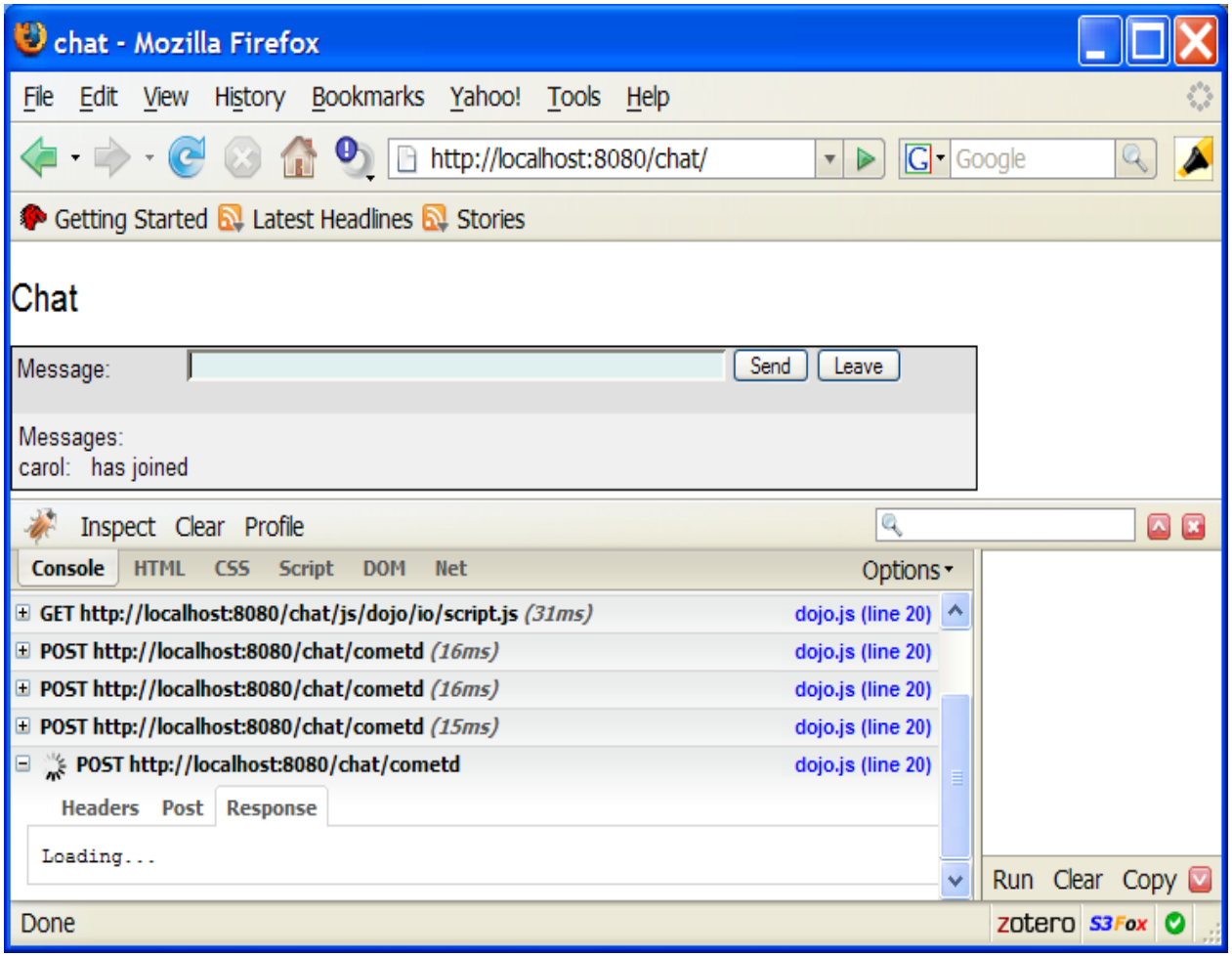
- Subscribes the client to the topic channel
- Any time a message is sent to the topic channel
- the object's callback will be invoked



Bayeux handshake in firebug



http long poll in firebug



Cometd Step 3: Publish

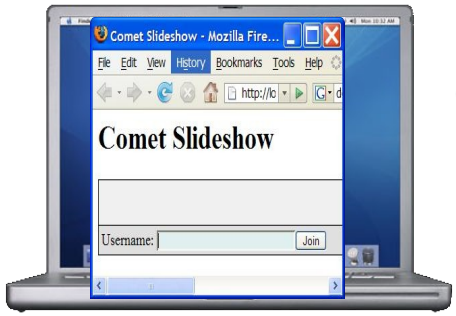
```
//a client publishes the username
and message to the topic channel


dojox.cometd.publish(room.channel, {
        user: room.username,
        text: text
    });
```

JSON message

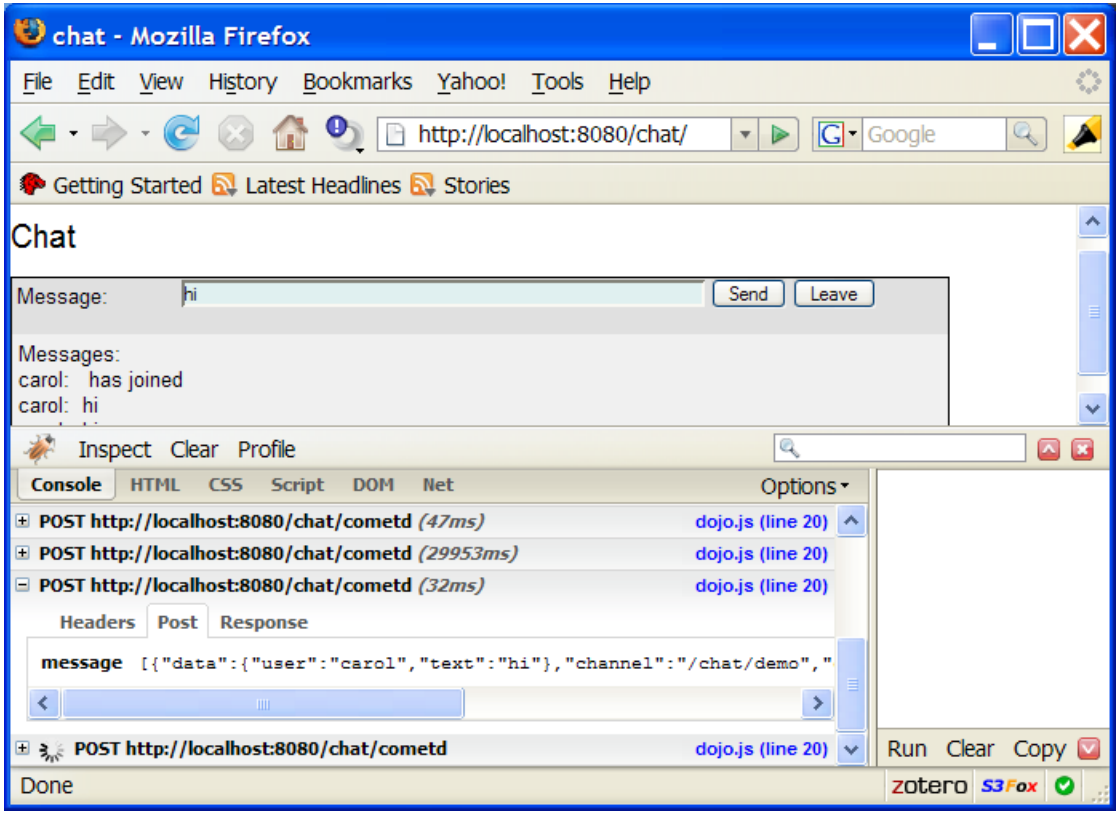
Comet Server

User and Text Msg



client 1

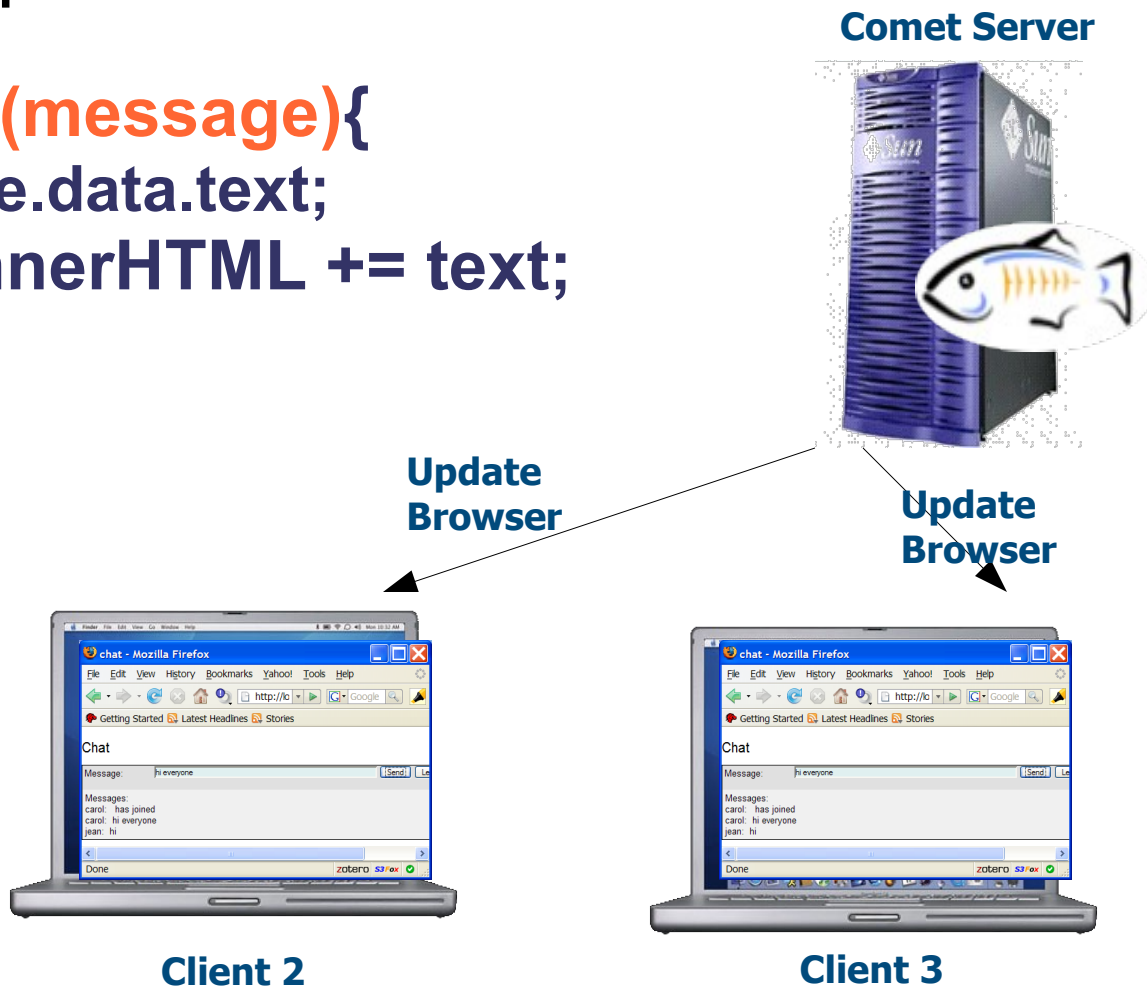
Publish in firebug



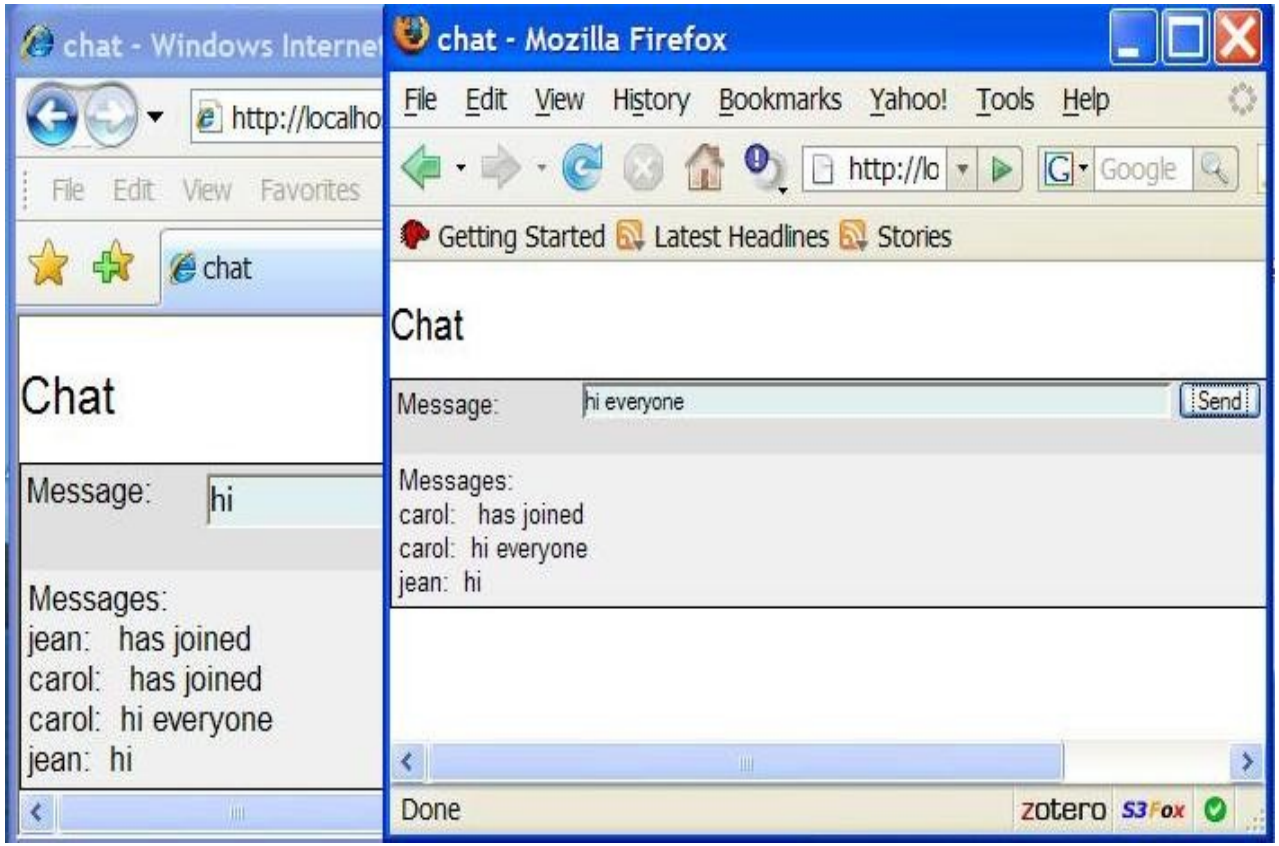
Cometd Step 4: Update Clients: CallBack

```
callBackFunction(message){
    var text=message.data.text;
     messageLog.innerHTML += text;
}
```

- The **callback function** is called with the **published message** when a new message is received
- This callback function updates the browser content
- with the text from the published message



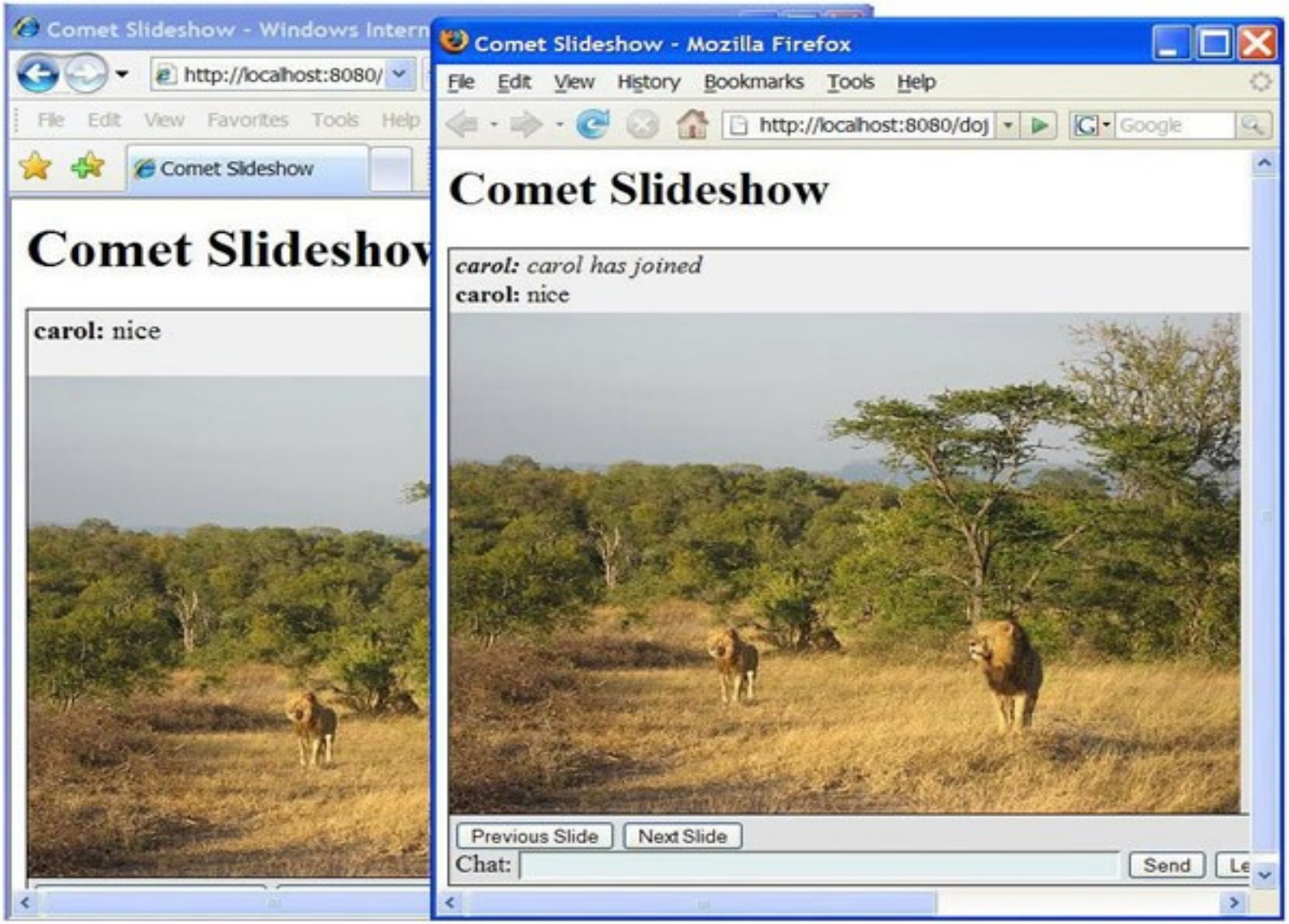
Update Clients: CallBack



Exercise 2: Build a Comet Slideshow Application (20 minutes)

- Develop a photo slideshow that can be controlled by multiple users
  - Actions of one user affect the pages seen by other users
  - The application provides a chat feature that enables users to comment on photos and let other users see the comments

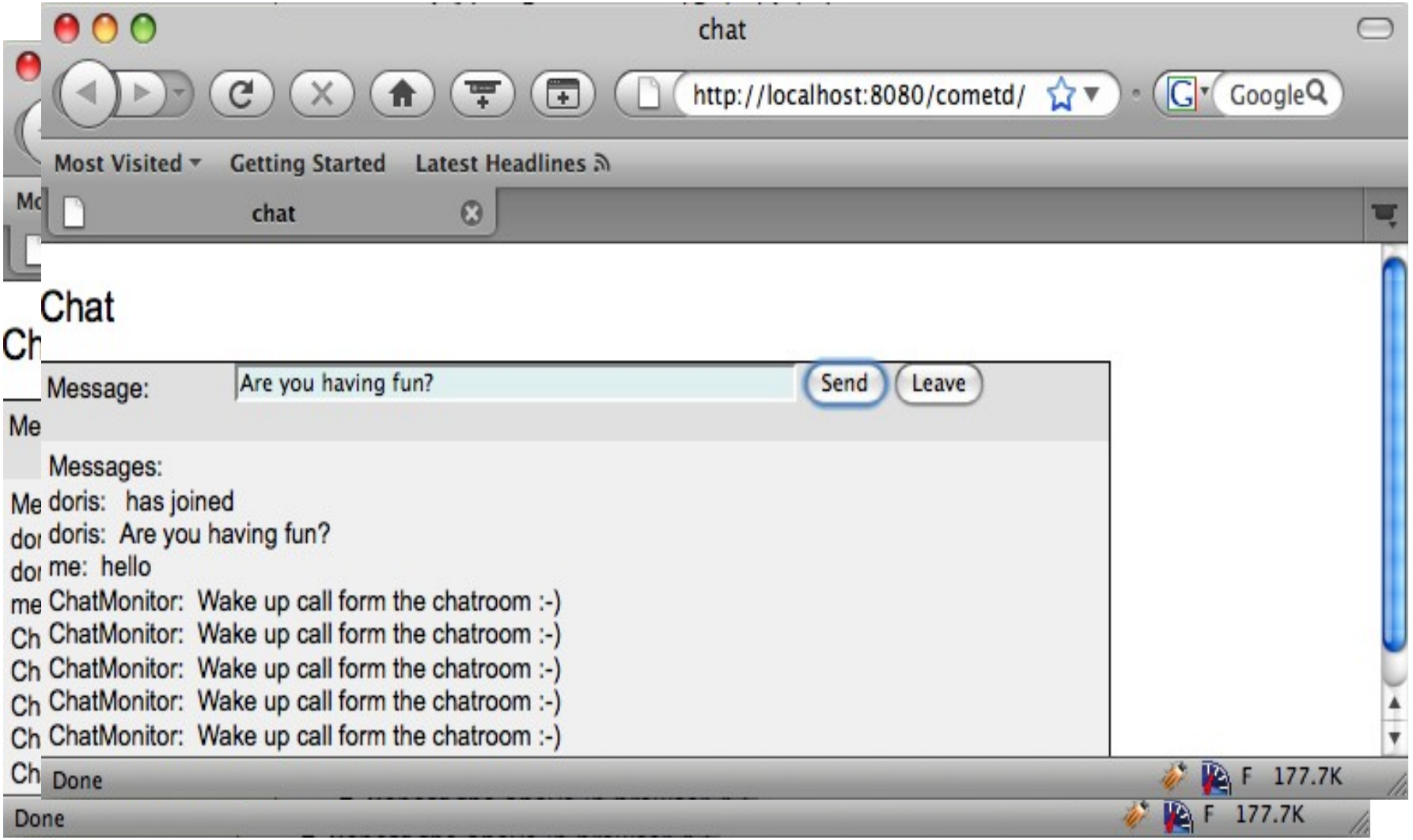
Demo


Comet Slideshow - Windows Intern
http://localhost:8080/
File Edit View Favorites Tools Help
Comet Slideshow
Comet Slideshow
carol: nice
Comet Slideshow - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/doj
Google
Comet Slideshow
carol: carol has joined
carol: nice
Previous Slide
Next Slide
Chat:
Send

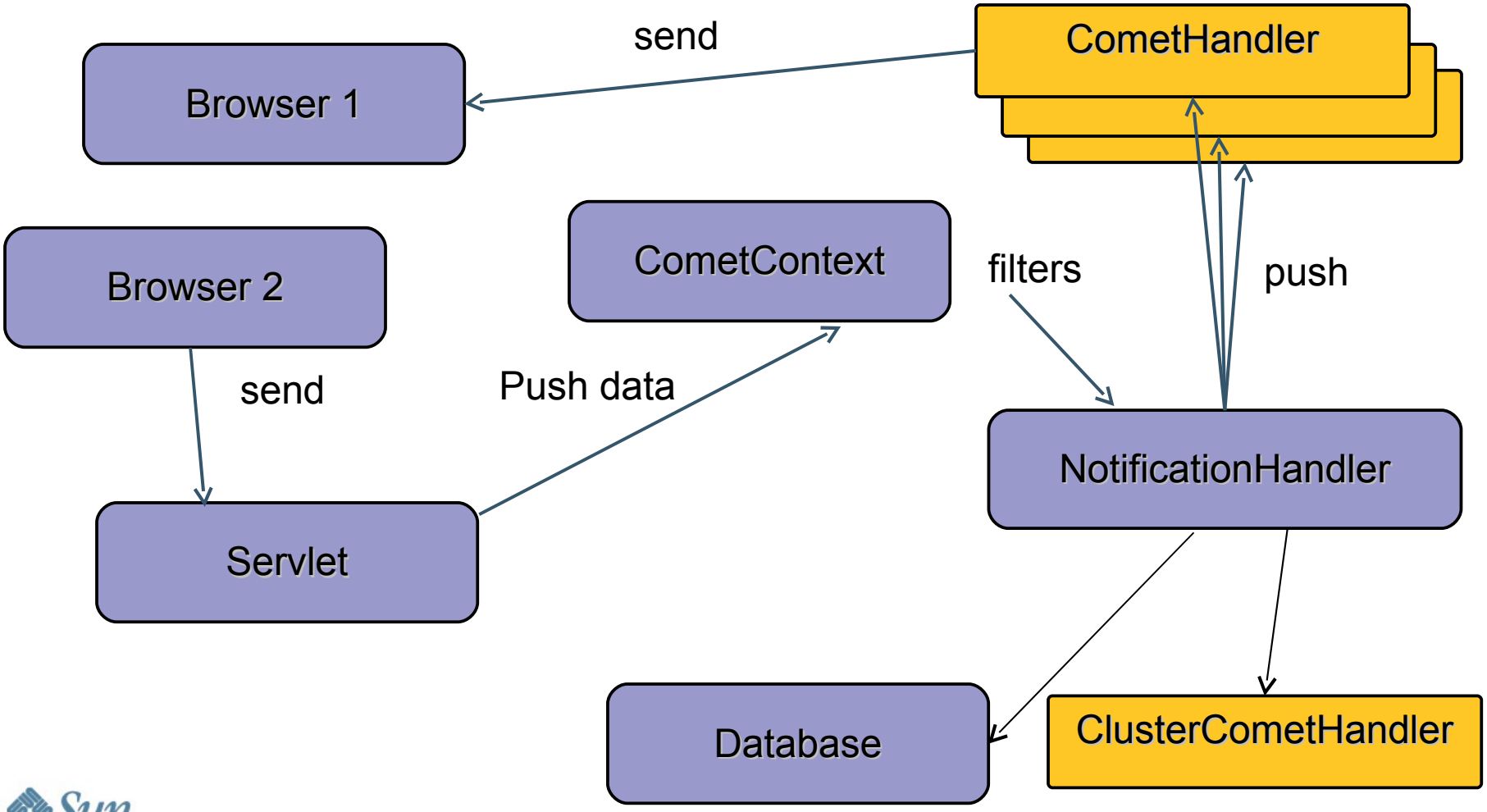
Exercise 3: Build a Comet Slideshow Application (20 minutes)

- Implement a chat servlet to interact with the Chat application by using Grizzly Comet Framework
  - Have some control on server
    - Receiving messages from server
    - Interacting with server
- Key Steps
  - Continue from Chat application (in exercise 1)
  - Implement a Servlet for chat application
    - Implement doPost()
    - Implement init() for a timer

Demo


chat
http://localhost:8080/cometd/
Google
Most Visited
Getting Started
Latest Headlines
chat
Chat
Message:
Are you having fun?
Send
Leave
Messages:
doris: has joined
doris: Are you having fun?
me: hello
ChatMonitor: Wake up call form the chatroom :-)
ChatMonitor: Wake up call form the chatroom :-)
ChatMonitor: Wake up call form the chatroom :-)
ChatMonitor: Wake up call form the chatroom :-)
ChatMonitor: Wake up call form the chatroom :-)
Done
F 177.7K
Done
F 177.7K

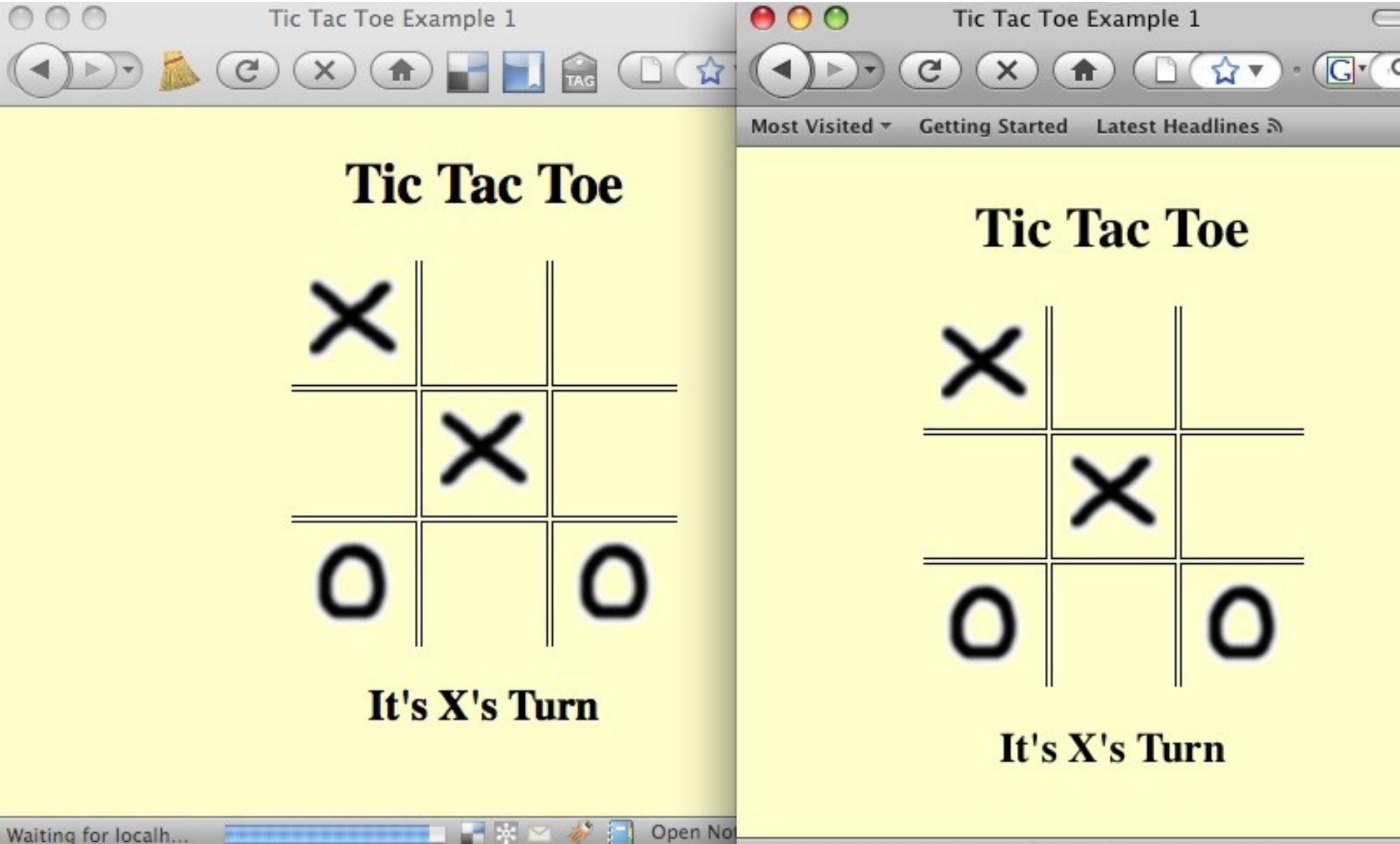
Grizzly Comet Framework
How it works



Exercise 4: Build a Two Way Tic-tac-toe Game (40 minutes)

- Develop a simple tic-tac-toe game in which two people play using Grizzly Comet Framework
- Key Steps
  - Step 1: Initialize Comet
  - Step 2: Define your CometHandler
  - Step 3: Connect to Comet Servlet: Add CometHandler to CometContext
  - Step 4: Advertise changes
  - Step 5: More details on Client

Demo


Tic Tac Toe Example 1
Tic Tac Toe
It's X's Turn
Tic Tac Toe Example 1
Most Visited
Getting Started
Latest Headlines
Tic Tac Toe
It's X's Turn
Waiting for localh...

Congratulations!

- You should now have completed this lab
- If you would like more time to continue working, please consider taking the lab exercises with you
  - Discs containing all of the labs offered this year are available for you to take home
- Please fill out a survey and hand it to someone before you leave
  - We really want to know what you think!
- Please log out of your machine when done
- Please look around to make sure you have all of your belongings
  - The hard copies of the lab guides are yours to keep
- Thank you for attending this hands-on lab!

Thank You

Doris Chen Ph.D., Carol McDonald, Justin Bolter

Sun Microsystems, Inc.

Technology Evangelists