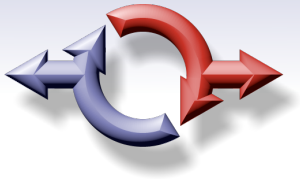
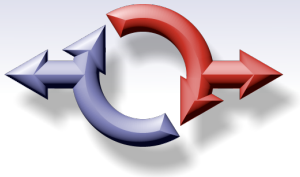


Hands-on DWR

Joe Walker
DWR Developer
directwebremoting.org



Agenda

Overview of DWR

Comet and Reverse Ajax

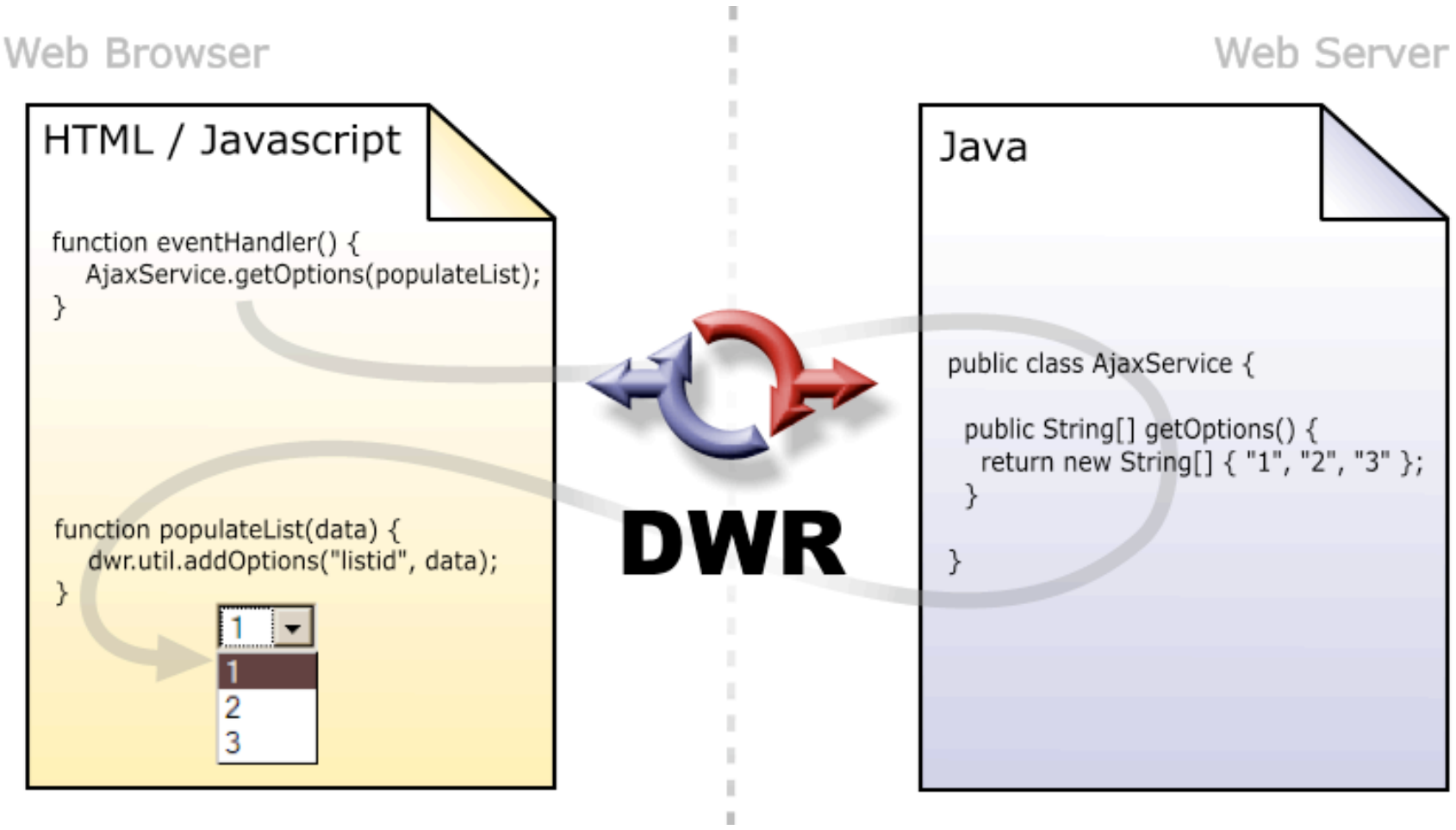
First 2 pages with DWR

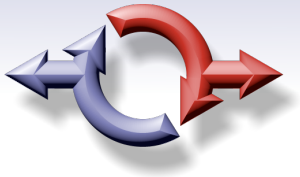
Integration with other projects

Security and Accessibility

Call Center Demo

Summary


Web Browser
Web Server
HTML / Javascript
function eventHandler() {
AjaxService.getOptions(populateList);
}
function populateList(data) {
dwr.util.addOptions("listid", data);
}
1
1
2
3
DWR
Java
public class AjaxService {
public String[] getOptions() {
return new String[] { "1", "2", "3" };
}
}

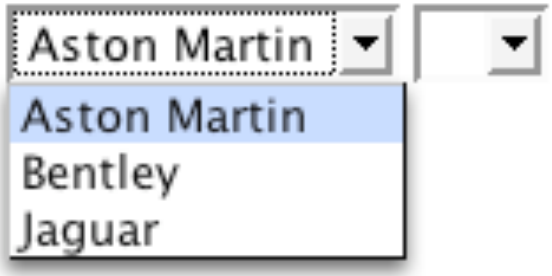


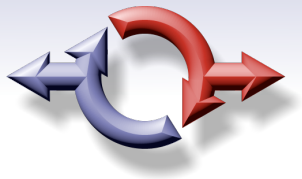
Web Page:

```
<script src='dwr/interface/CarService.js'>

function getModels() {
  var make = dwr.util.getValue("makes");
  CarService.getModels(make, populateMakes);
}

<select id="makes" onchange="getModels"> ...
```



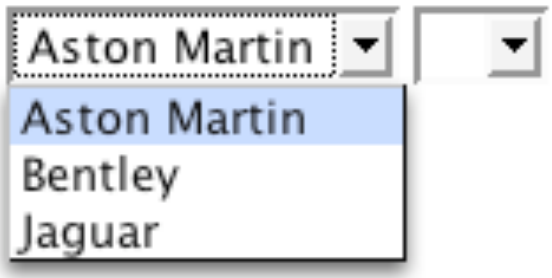


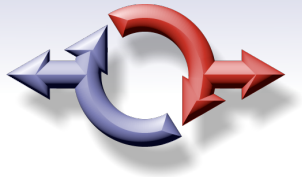
Web Page:

```
<script src='dwr/interface/CarService.js'>

function getModels() {
  var make = dwr.util.getValue("makes");
  CarService.getModels(make, populateMakes);
}

<select id="makes" onchange="getModels"> ...
```





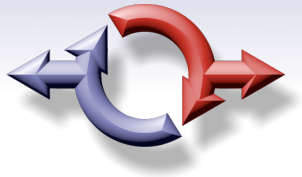
On the server: dwr.xml

```
<create creator="new" javascript="CarService">
  <param name="class" value="com.example.CarService"/>
</create>
```

On the server CarService.java

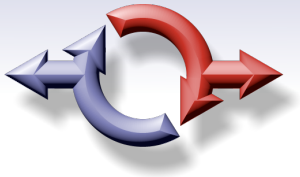
```
public class CarService {

    public String[] getModels(String make) {
        return new String[] { "DB9", "DB7", "DB6" };
    }
}
```



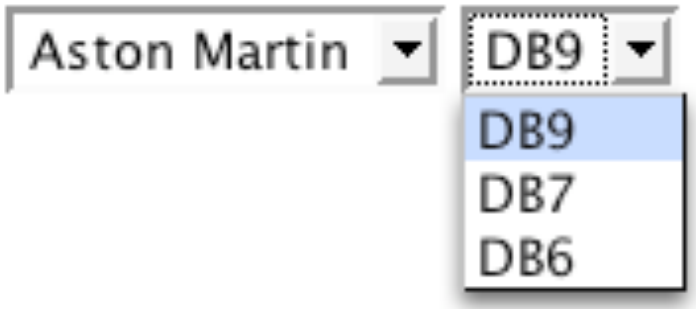
On the server CarService.java

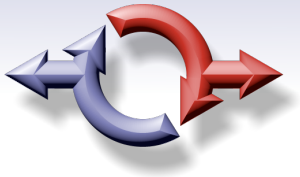
```
@RemoteProxy
public class CarService {
    @RemoteMethod
    public String[] getModels(String make) {
        return new String[] { "DB9", "DB7", "DB6" };
    }
}
```



Web Page:

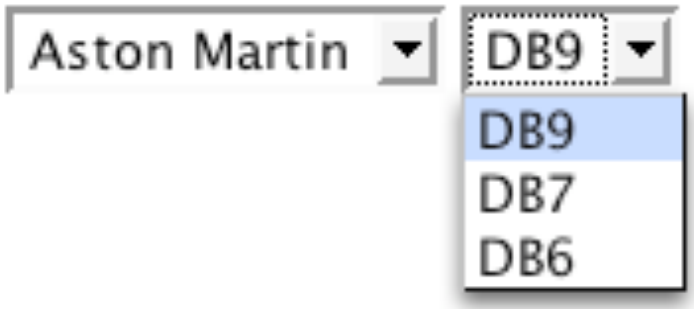
```
function getModels() {
  var make = dwr.util.getValue("makes");
  CarService.getModels(make, function(reply) {
    dwr.util.setValue("models", reply);
  });
}
```

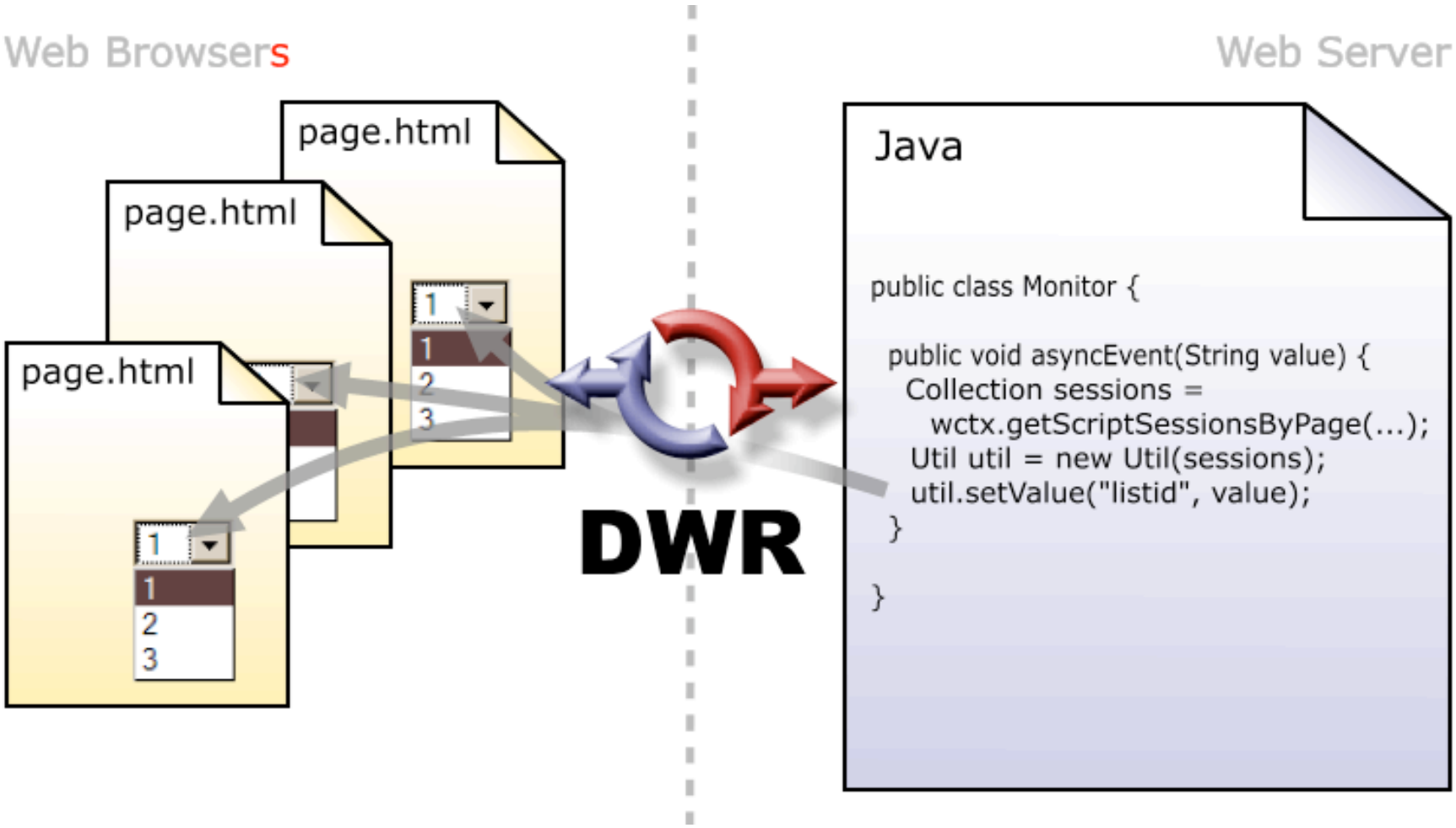


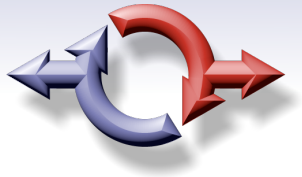


Web Page:

```
function getModels() {
  var make = dwr.util.getValue("makes");
  CarService.getModels(make, function(reply) {
    dwr.util.setValue("models", reply);
  });
}
```




Web Browsers
Web Server
page.html
page.html
page.html
1
1
2
3
1
1
2
3
DWR
Java
public class Monitor {
public void asyncEvent(String value) {
Collection sessions =
wctx.getScriptSessionsByPage(...);
Util util = new Util(sessions);
util.setValue("listid", value);
}
}



Web Page:

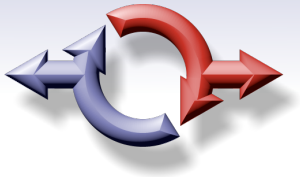
```
Stock Price: <span id="stock"></span>
```

On the Server:

```
Collection<ScriptSession> sessions
     = sctx.getScriptSessionsByPage("stock.html");

Util pages = new Util(sessions);

pages.setValue("stock", 42);
```



Web Page:

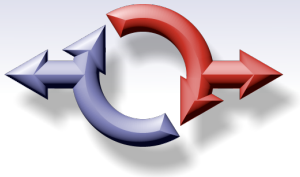
```
Stock Price: <span id="stock"></span>
```

On the Server:

```
Collection<ScriptSession> sessions
     = sctx.getScriptSessionsByPage("stock.html");

Util pages = new Util(sessions);

pages.setValue("stock", 42);
```



Web Page:

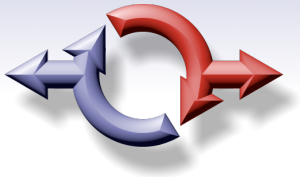
```
Stock Price: <span id="stock"></span>
```

On the Server:

```
Collection<ScriptSession> sessions
     = sctx.getScriptSessionsByPage("stock.html");

Util pages = new Util(sessions);

pages.setValue("stock", 42);
```



Web Page:

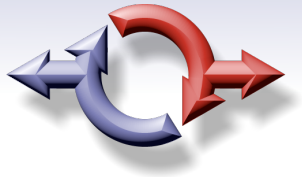
```
Stock Price: <span id="stock"></span>
```

On the Server:

```
Collection<ScriptSession> sessions
     = sctx.getScriptSessionsByPage("stock.html");

Util pages = new Util(sessions);

pages.setValue("stock", 42);
```



Web Page:

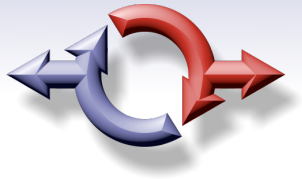
```
Stock Price: <span id="stock"></span>
```

On the Server:

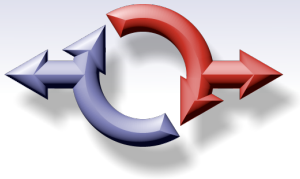
```
Collection<ScriptSession> sessions
     = sctx.getScriptSessionsByPage("stock.html");

Util pages = new Util(sessions);

pages.setValue("stock", 42);
```

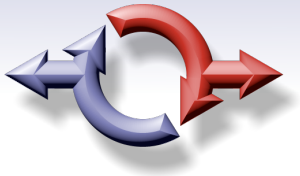


Unnecessary Complexity is a Bug



Installation in 4 steps:

1. Copy dwr.jar into WEB-INF/lib:
   http://directwebremoting.org/
2. Add the DWR servlet to WEB-INF/web.xml
3. Give DWR permission to call your code:
   Create dwr.xml or use Annotations
4. Browse the test pages:
   http://localhost:8080/WEBAPP/dwr/



DWR can marshall:

Primitive types, and their Object counterparts
int, boolean, long, float, double, etc

Obvious classes
(String, Date, BigDecimal, BigInteger, Enum, etc)

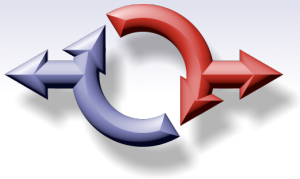
Arrays and Collections (Map, List, Set, Iterator, ...)

JavaBeans and Objects

XML objects (DOM, XOM, JDom, Dom4J)

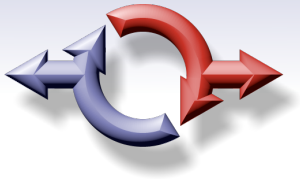
Images and Binary Files in version 2.1

(in short, basically anything ...)

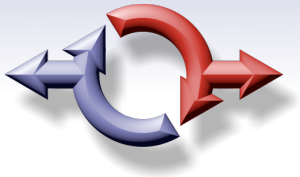


Demo

Your first 2 pages with DWR

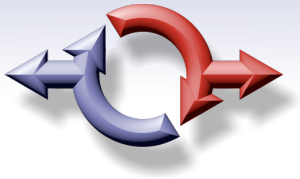


DWR is about remoting and not widgets



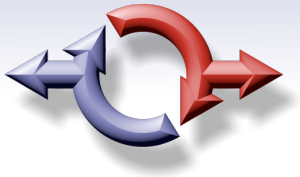
DWR Integrations

TIBCO GI	Struts	DOM4J
Scriptaculous	JSF	XmlBeans
Open Ajax Hub	Beehive	EJB
Guice	BeanShell	RIFE
Spring	Hibernate 2/3	IBDOM
Jetty	XOM	WingS
WebWork	JDOM	



Demo

GI + DWR + OAH



On the Web:

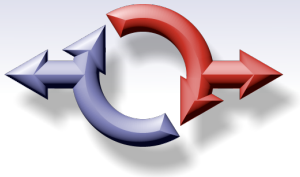
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



On the Web:

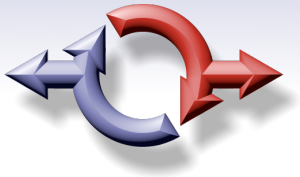
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



On the Web:

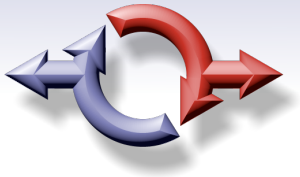
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



On the Web:

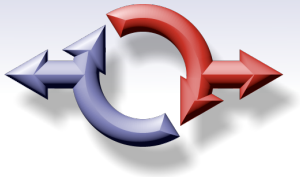
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



On the Web:

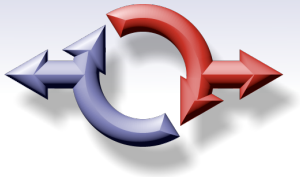
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



On the Web:

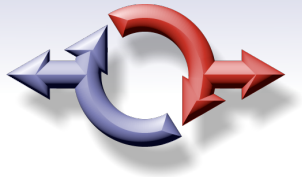
```
function giLoaded() {
  OpenAjax.hub.subscribe("gidemo.corp", objectPublished);

  dwr.engine.setActiveReverseAjax(true);
}

function objectPublished(prefix, name, hd, corporation) {
  var matrix = giApp.getJSXByName("matrix");

  var inserted = matrix.getRecordNode(corporation.jsxid);

  matrix.insertRecord(corporation, null, inserted == null);
  matrix.repaintData();
}
```



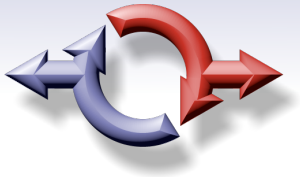
On the Server:

```
Corporation c = corporations.getNextChangedCorporation();

Collection<ScriptSession> sessions
    = serverContext.getScriptSessionsByPage("demo.html");

ScriptProxy proxy = new ScriptProxy(sessions);

proxy.addFunctionCall("OpenAjax.hub.publish",
                      "gidemo.corp", c);
```



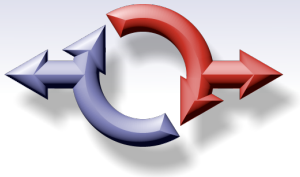
On the Server:

```
Corporation c = corporations.getNextChangedCorporation();
```

```
Collection<ScriptSession> sessions
    = serverContext.getScriptSessionsByPage("demo.html");

ScriptProxy proxy = new ScriptProxy(sessions);

proxy.addFunctionCall("OpenAjax.hub.publish",
                      "gidemo.corp", c);
```



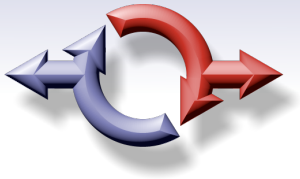
On the Server:

```
Corporation c = corporations.getNextChangedCorporation();

Collection<ScriptSession> sessions
    = serverContext.getScriptSessionsByPage("demo.html");

ScriptProxy proxy = new ScriptProxy(sessions);

proxy.addFunctionCall("OpenAjax.hub.publish",
                      "gidemo.corp", c);
```



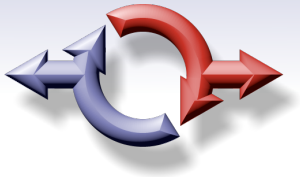
On the Server:

```
Corporation c = corporations.getNextChangedCorporation();

Collection<ScriptSession> sessions
    = serverContext.getScriptSessionsByPage("demo.html");

ScriptProxy proxy = new ScriptProxy(sessions);

proxy.addFunctionCall("OpenAjax.hub.publish",
                      "gidemo.corp", c);
```



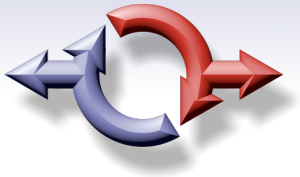
On the Server:

```
Corporation c = corporations.getNextChangedCorporation();

Collection<ScriptSession> sessions
    = serverContext.getScriptSessionsByPage("demo.html");

ScriptProxy proxy = new ScriptProxy(sessions);

proxy.addFunctionCall("OpenAjax.hub.publish",
                      "gidemo.corp", c);
```

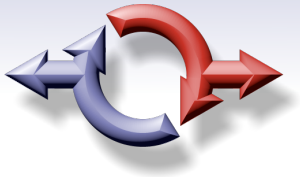


How it works

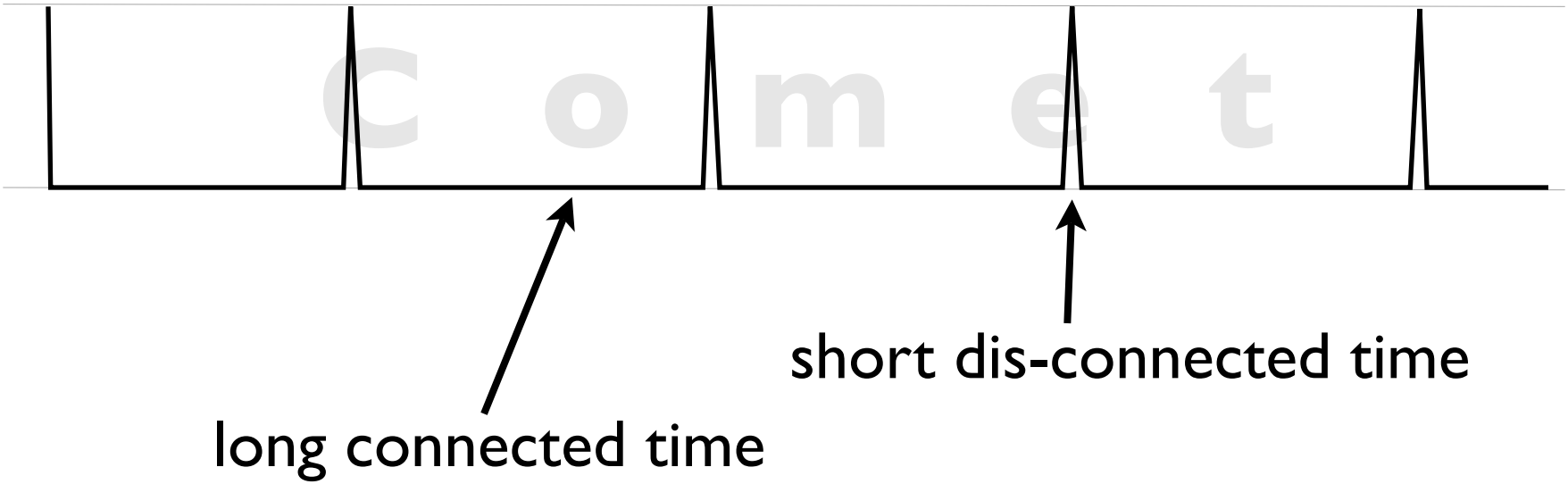
Reverse Ajax combines three methods:

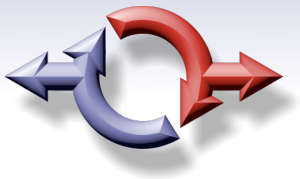
- Comet
- Polling
- Piggyback

The same thing
(depending on your perspective)

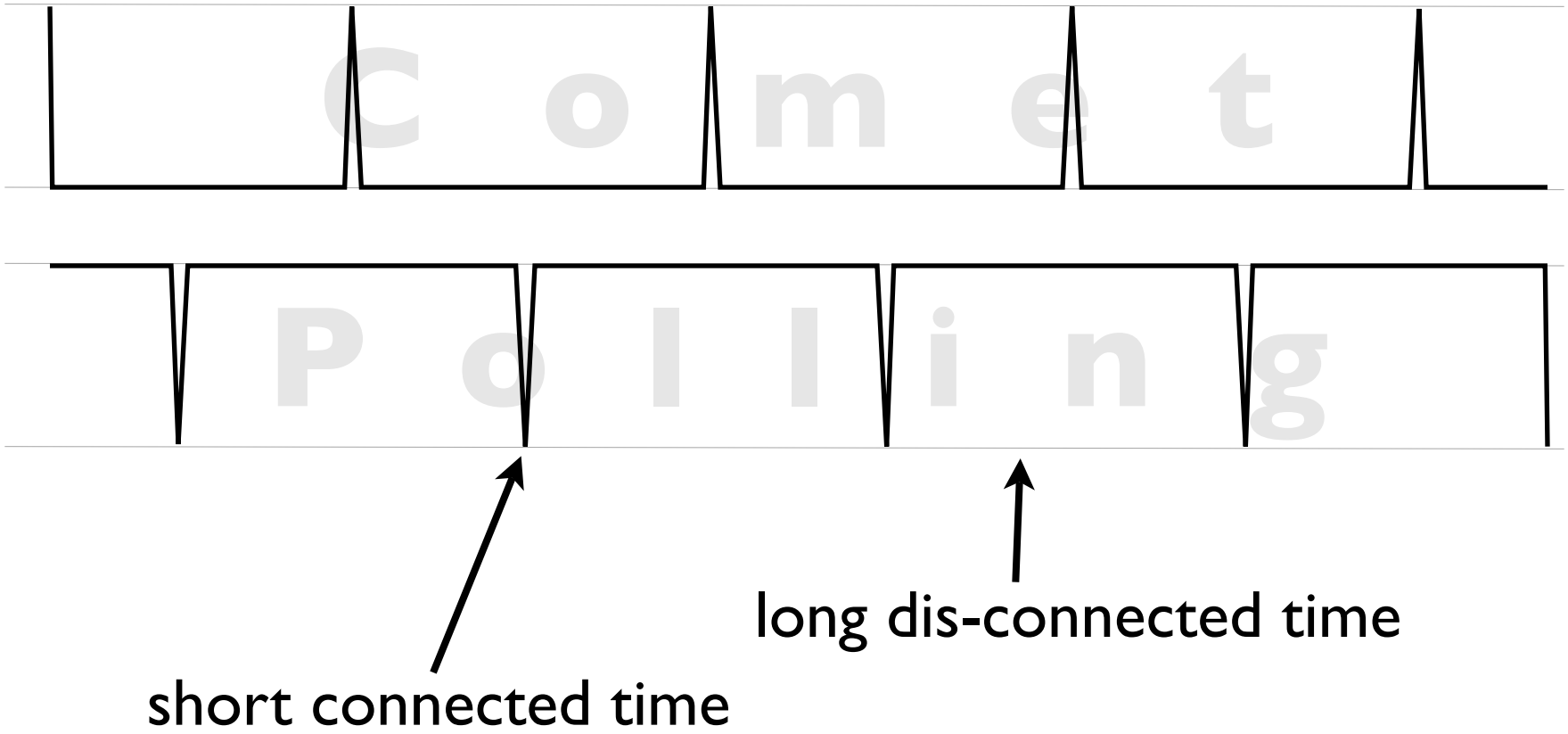


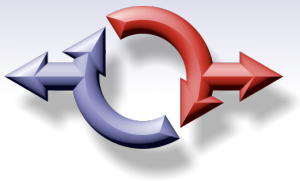
Comet vs. Polling



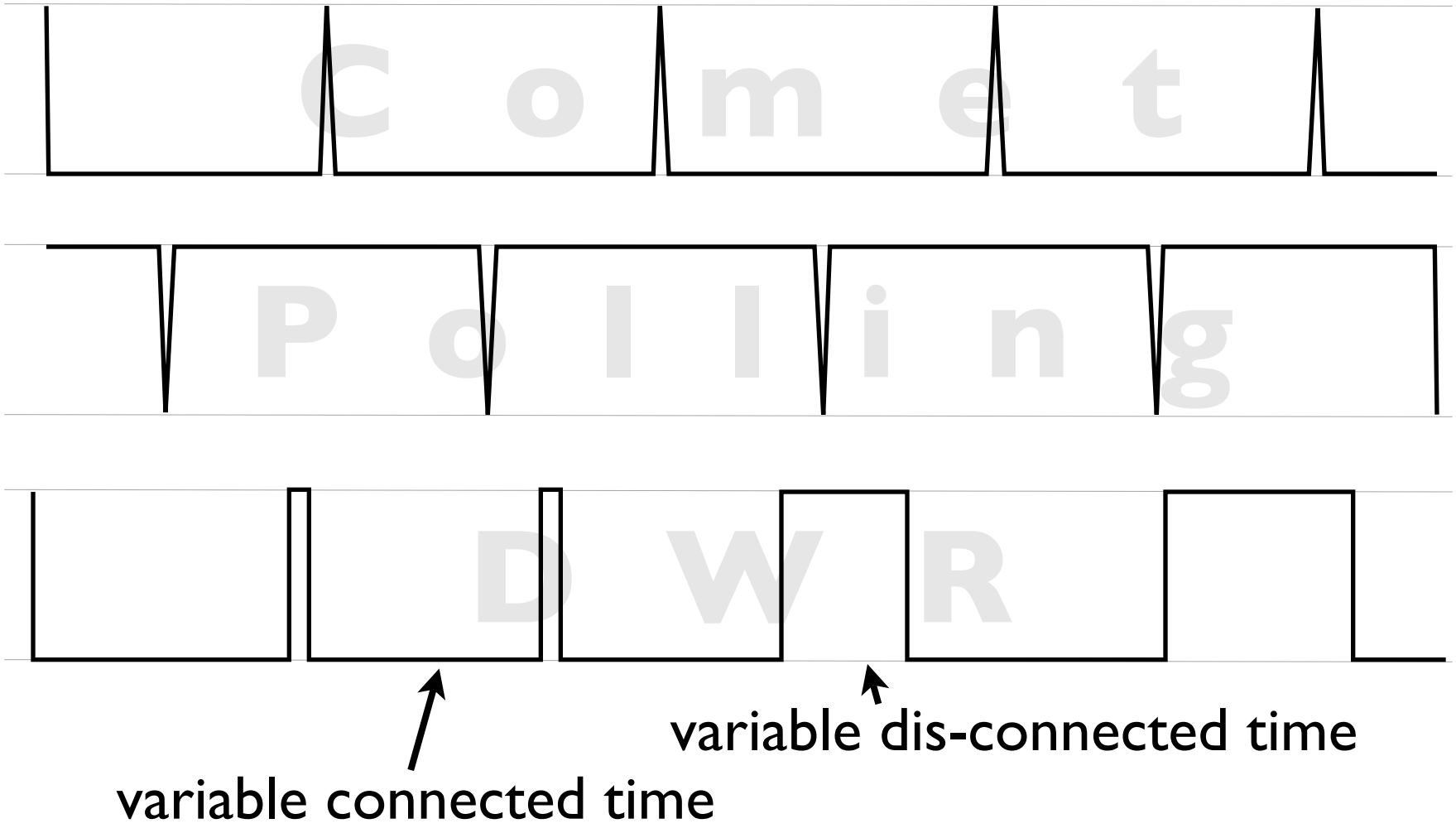


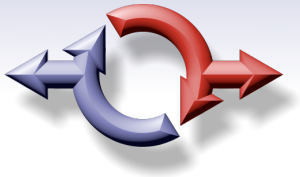
Comet vs. Polling





Comet vs. Polling





Issues

Some servers handle poorly in the presence of many simultaneous connections

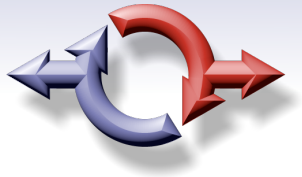
- Dynamically reconfigure towards polling

Some proxies don't do chunked mode

- Automatically close and re-open on output

Some people prefer slow response and lighter load

- Use polling with long disconnected time



Security

DWR does not call anything without permission

Method level access control

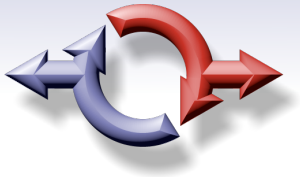
- For Creators and Converters

Java EE role based security:

- Declare roles that can access methods

Class leading CSRF protection

XSS filtering



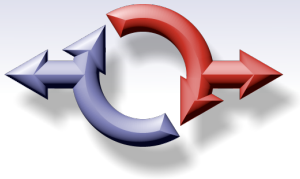
Accessibility

dwr.util supports pluggable notifiers

- .focus() for screen readers
- Yellow fade for partially sighted

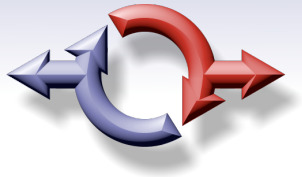
New notifiers can be easily added

- e.g. To make a sound on a change



Demo

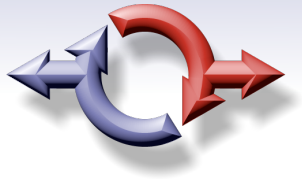
Ticket-Center



On the Server:

```
public class Call{
    private Date callStarted = new Date();
    private String notes = "";
    private boolean supervisorAlert = false;
    private boolean isHandled = false;
    private String name;
    private String address;
    private String phoneNumber;
    private int id;

    // Getters and setters
}
```



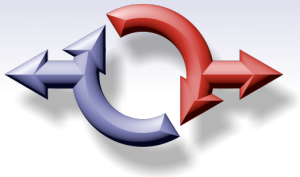
On the Server:

```
/** Mark this call as being handled */
public String beginHandling(int id) ...

/** Put the call back in queue with the supervisorAlert bit set */
public String alertSupervisor(int id, Call call) ...

/** Mark this call as being handled */
public String cancelHandling(int id) ...

/** Alert the back end systems and take this call off the queue */
public String completeHandling(int id, Call newCall) ...
```



On the Server:

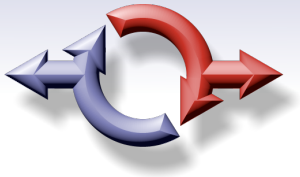
```
private void update() {

  Collection<ScriptSession> sessions
      = serverContext.getScriptSessionsByPage("tc.html");

  ScriptProxy proxy = new ScriptProxy(sessions);

  proxy.addFunctionCall("updateCallers", calls);

}
```



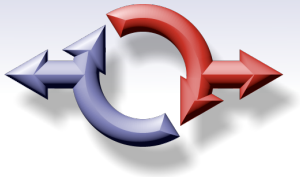
On the Server:

```
private void update() {
    Collection<ScriptSession> sessions
        = serverContext.getScriptSessionsByPage("tc.html");

    ScriptProxy proxy = new ScriptProxy(sessions);

    proxy.addFunctionCall("updateCallers", calls);

}
```



On the Server:

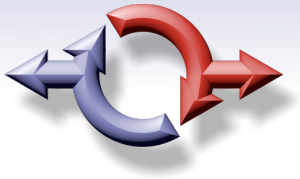
```
private void update() {

  Collection<ScriptSession> sessions
      = serverContext.getScriptSessionsByPage("tc.html");

  ScriptProxy proxy = new ScriptProxy(sessions);

  proxy.addFunctionCall("updateCallers", calls);

}
```



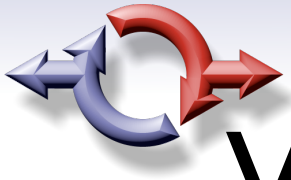
On the Server:

```
private void update() {

  Collection<ScriptSession> sessions
      = serverContext.getScriptSessionsByPage("tc.html");

  ScriptProxy proxy = new ScriptProxy(sessions);

  proxy.addFunctionCall("updateCallers", calls);
}
```



Where does DWR fit?

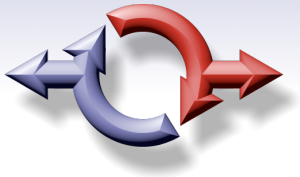
For some Ajax magic with Struts 1.0

With newer Spring, Webwork, JSF or RIFE code

By itself with dwr.util

As the service layer to a Scriptaculous or Dojo application

In the enterprise with TIBCO GI and a service oriented architecture



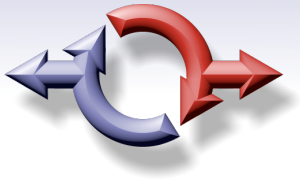
Summary

It's easy to get going with DWR

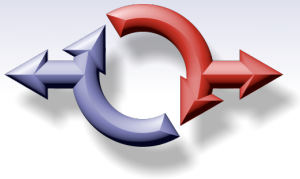
You can use it to create advanced applications without large amounts of code

Reverse Ajax allows your applications to have high levels of user interactivity

It fits in your current web application without requiring you to start from scratch



Questions?



directwebremoting.org

getahead.org/blog/joe