Building Ajax-Based Web Applications with ArcGIS Server and Java (Deep Dive)

Antony Jayaprakash

Jayant Sai

Presentation outline

- Ajax integration in the Java Web ADF
  - Web ADF controls
  - Javascript library
  - Serverside objects
- Common use cases
  - Listen to control updates
  - Update controls on client
  - Updating custom components
  - Timer based requests
- Customizing applications
  - Demo: Trip Planner
  - Additional customizations
- Tips & Tricks
- Q&A

Our assumptions

- Prerequisites
  - Java
  - Working knowledge of
    - JSF
    - Javascript
    - XML
    - Ajax

Terminology

- Phase Listener
  - A class registered in the JSF lifecycle to be notified at the beginning and ending of a specific JSF phase
- JSF Components & Web ADF Controls
  - They are same, controls & components used interchangeably
- Postback
  - Action by webpage, to submit page to server for processing
- Clientside PostBack
  - Making clientside script calls (using XMLHttpRequest) to serverside code without causing a postback

Web ADF controls

- All Web ADF controls are Ajax enabled

```
<%@taglib uri="http://www.esri.com/adf/web" prefix="a"%>
  <a:map … />
  <a:toc …/>
  <a:overview … />
  <a:toolbar …>
    <a:tool … />
  </a:toolbar>
  <a:task …/>
```

Javascript library

- Extensive clientside Javascript library
  - Handles user interaction & communication with server
  - Framework for Ajax
  - Browser independent functions
  - Object oriented and name spaced with 1 Global (EsriControls)
  - Does not conflict with 3rd party Ajax toolkits (prototype, dojo, DWR, MS Ajax, etc)

Serverside objects

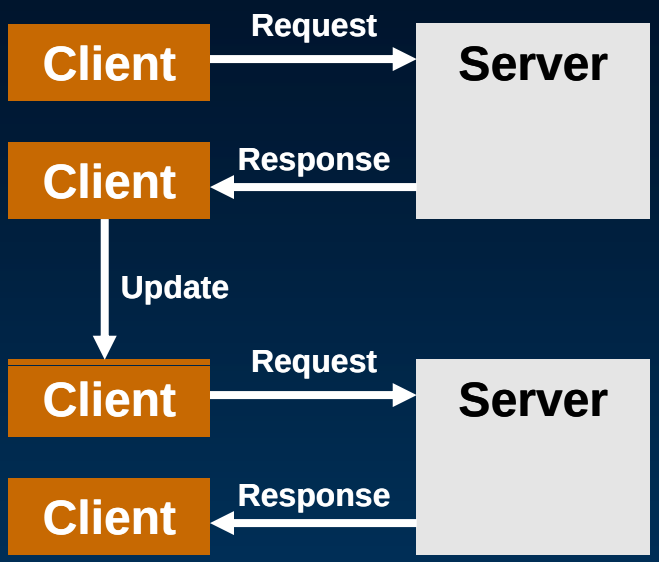
- Java object to support Ajax
  - JSF Phase Listeners
  - XML Ajax Renderers
    - MapRenderer
    - TocRenderer
    - ToolbarRenderer
    - OverviewRenderer
    - TaskRenderer
  - AJAXResponseRenderer class
- Custom renderer hook-ins
  - AJAXRenderer interface
  - WEB-INF/ajax-renderers.xml

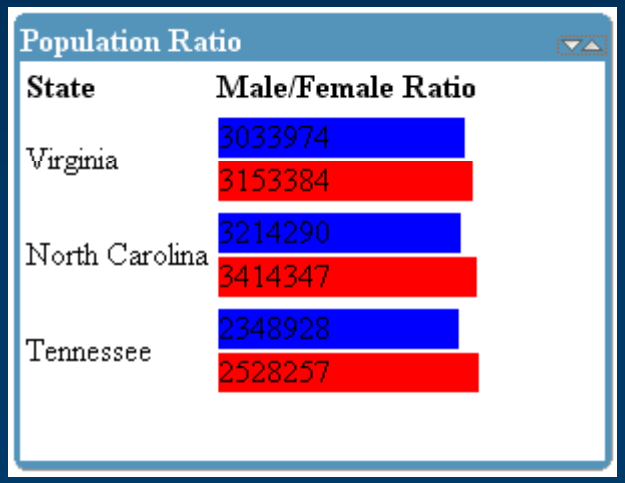
Common use cases

- #1: Listen to control updates to perform custom operations
- #2: Update controls on client based on server operation
- #3: Update custom components by integrating into request- response cycle
- #4: Timer based requests to display data on client

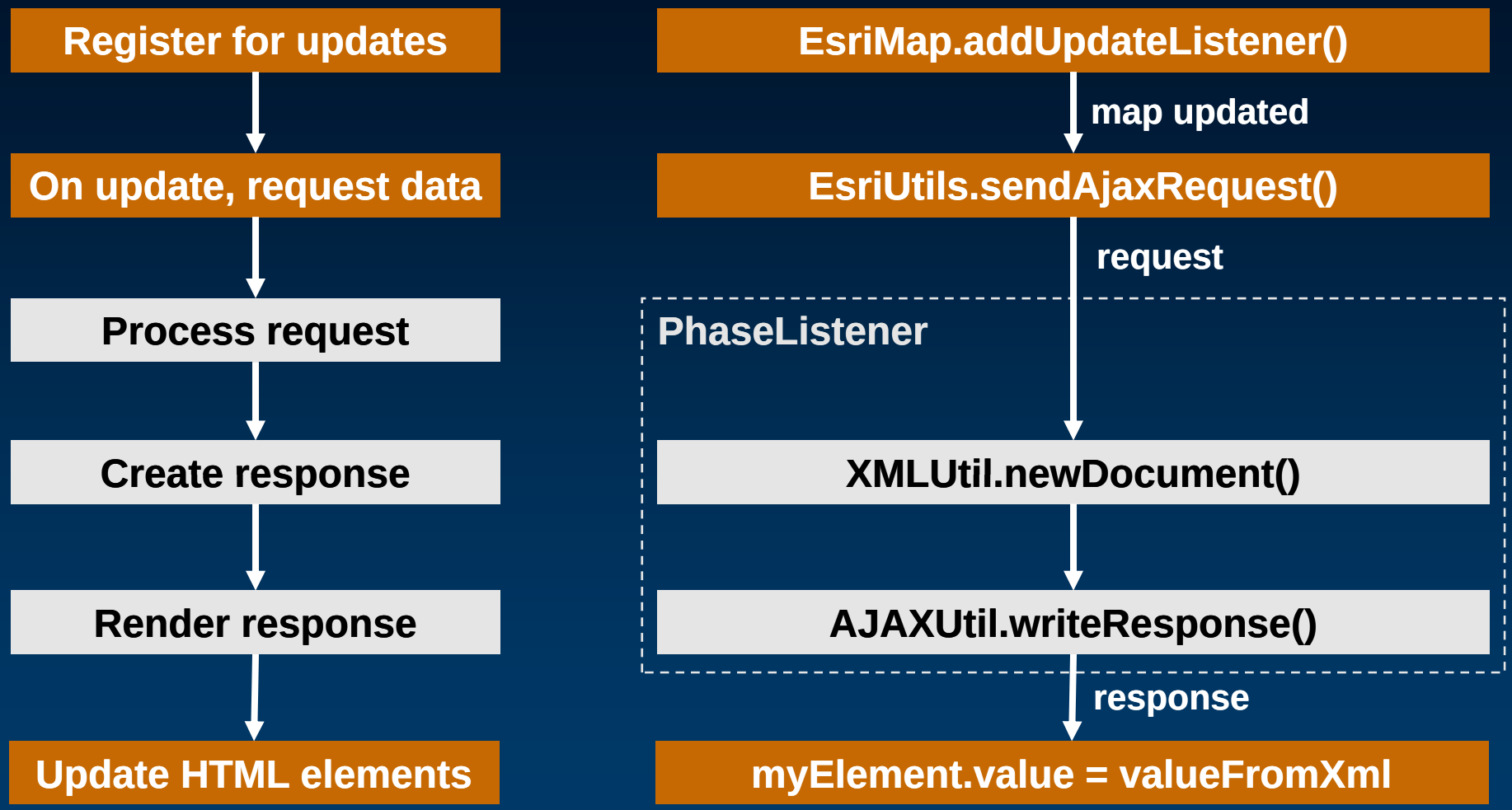
#1: Listen to control updates

- When to Use?
  - Perform asynchronous updates based on updates to controls
  - Request-response should not interfere with user interaction
- Usage in Web ADF?
  - Retrieving images for continuous pan
- Demo
  - On map update
  - Request census data
  - Display male-to-female ratio for states within visible extent





#1: Application flow

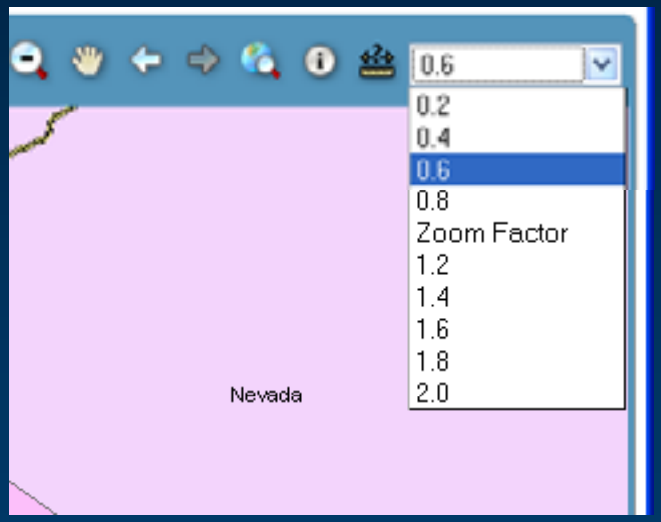


#1: Demo details

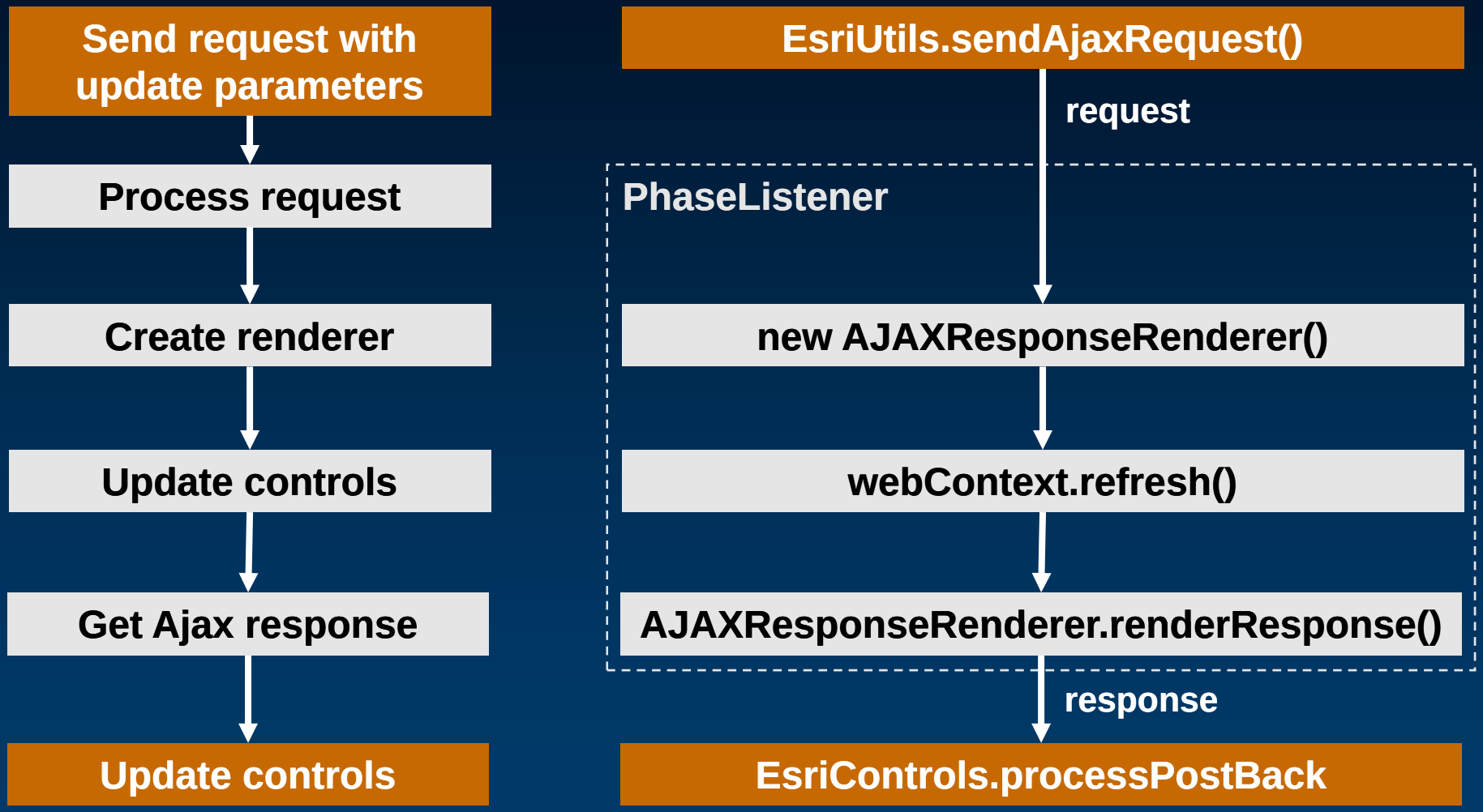
- **Javascript** (mapviewer.js)
  - Register for map updates
  - Send request for data (updatePopRatioRequest())
- **Implement PhaseListener** (PopulationRatioPhaseListener)
  - Process request (afterPhase())
  - Get extent
  - Render & return XML response
- **Register PhaseListener** (fc-ajaxdemo.xml)
  - Add ‘<lifecycle>’ declaration
- **Javascript** (mapviewer.js)
  - Process response to update HTML (updatePopRatioResponse())

#2: Update controls on client

- When to Use?
  - Update controls based on changes to GIS business objects
- Usage in Web ADF?
  - Update on zoom level change
- Demo
  - User zoom in/out of map using drop down
  - Update map's extent based on zoom factor
  - Use Web ADF Ajax classes to update controls on client



#2: Application flow

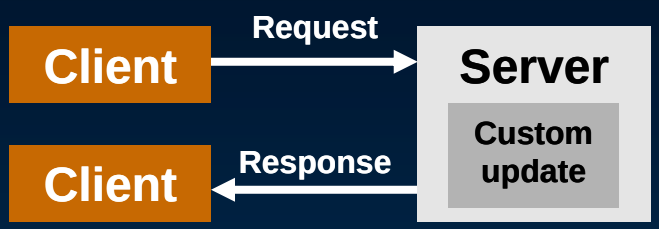


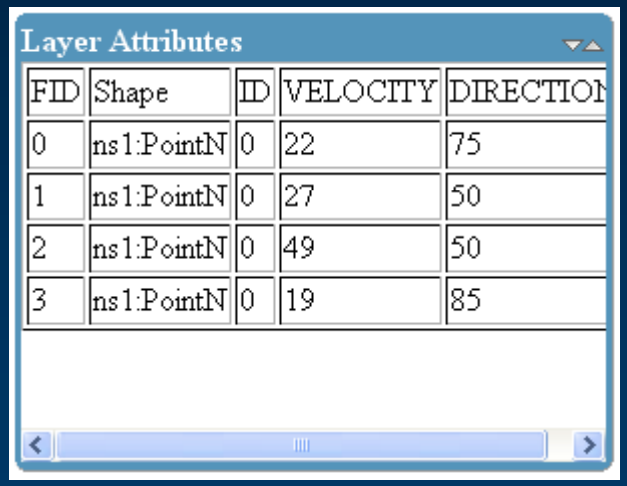
#2: Demo details

- **Javascript** (mapviewer.js)
  - Send ajax request with zoom factor (sendMapZoomRequest())
  - Set callback function (EsriControls.processPostBack)
- **Implement PhaseListener** (MapZoomPhaseListener)
  - Process request (afterPhase())
  - Create AJAXResponseRenderer
  - Update map & context
  - Render & return XML response

#3: Updating custom components

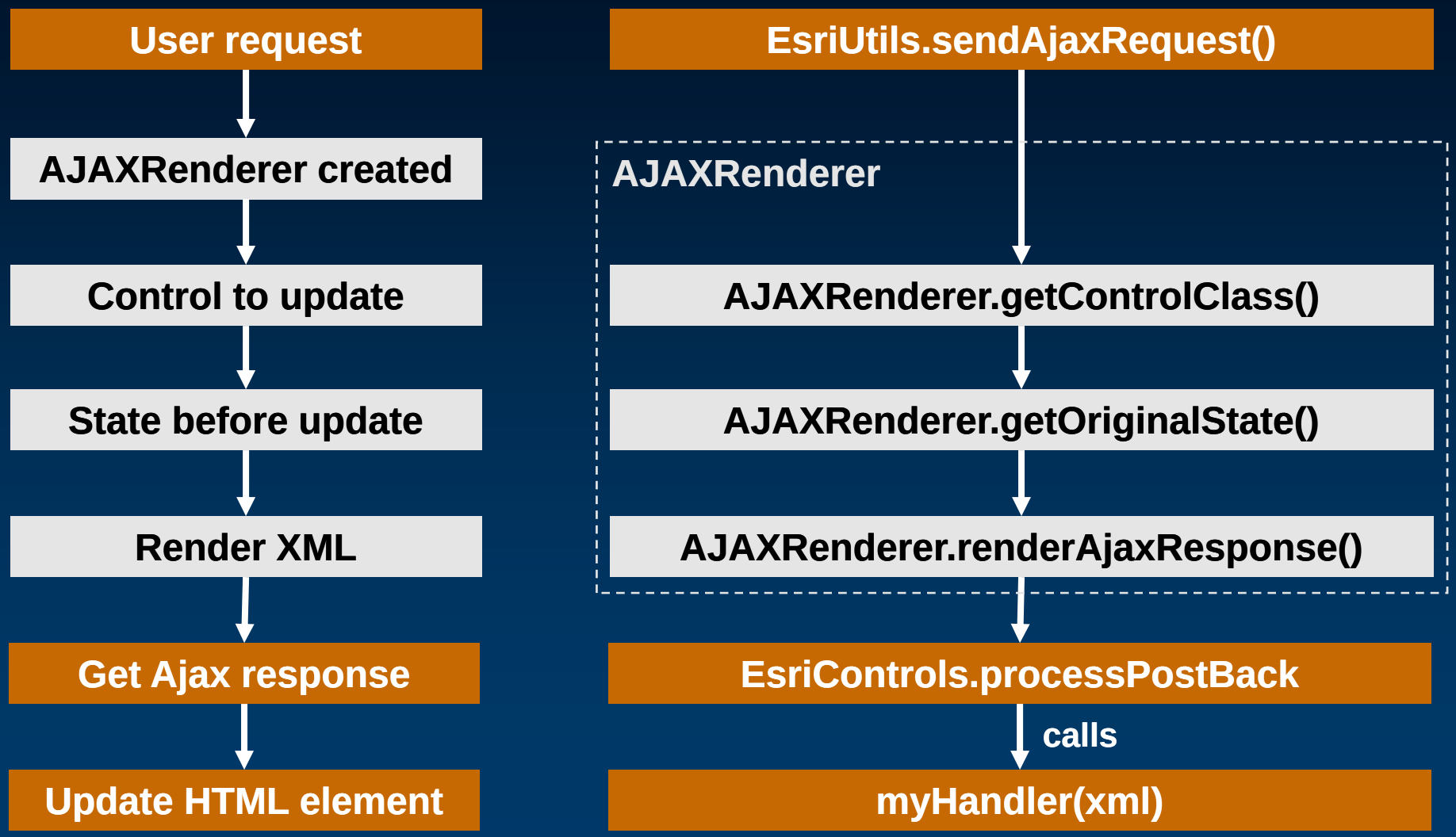
- When to Use?
  - Update custom components
  - Done as part of request-response cycle
- Usage in Web ADF?
  - All web control updates
  - Web mapping application result details
- Demo
  - Render XML with attributes of features in visible extent
  - Update table to display attributes data





FID	Shape	ID	VELOCITY	DIRECTION
0	ns1:PointN	0	22	75
1	ns1:PointN	0	27	50
2	ns1:PointN	0	49	50
3	ns1:PointN	0	19	85

#3: Application flow

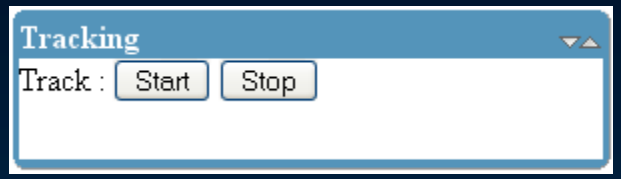


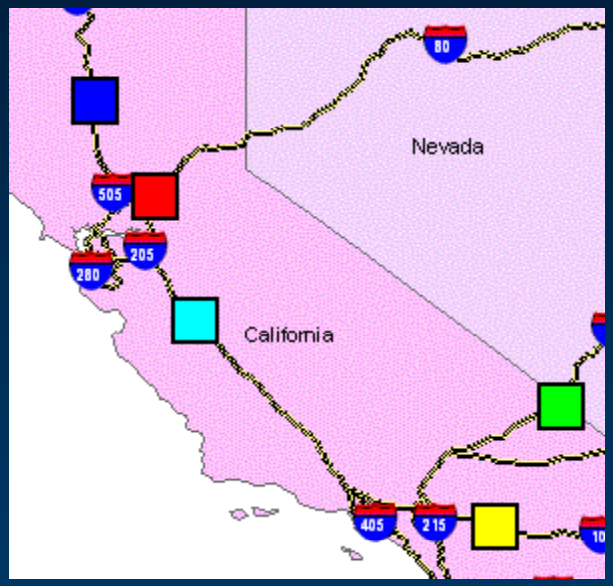
#3: Demo details

- Implement AJAXRenderer (LayerAttributesRenderer)
  - Check if updates required
  - Render XML tag with data
- Register AJAXRenderer
  - Add to list of AJAXRenderers (WEB-INF/ajax-renderers.xml)
- Javascript
  - Function to parse XML and update HTML elements (layerAttributesHandler())
- Register XML tag handler
  - Add function to list of xml tag handlers to EsriControls Javascript global object (EsriControls.addPostBackTagHandler)

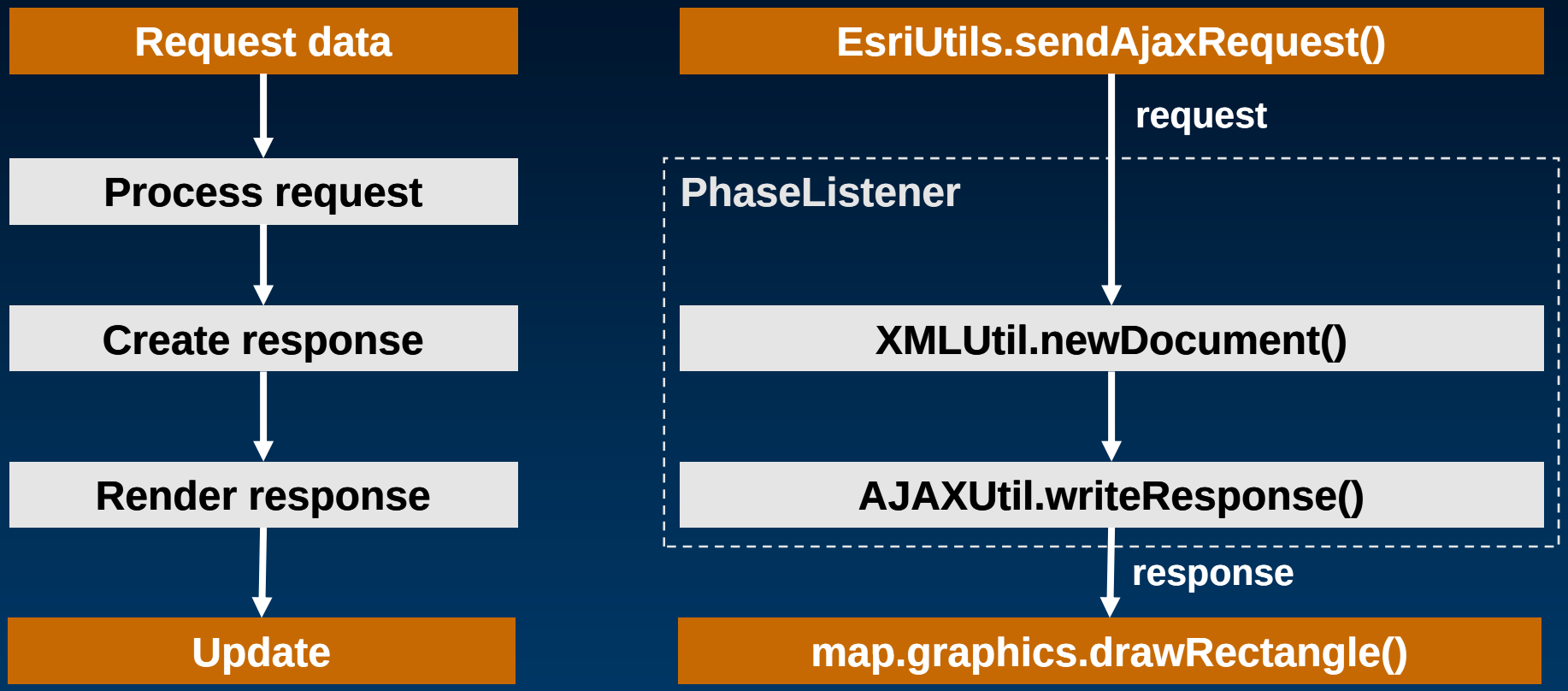
#4: Timer based updates

- When to Use?
  - Execute operation periodically
  - Display temporal data
- Usage in Web ADF?
  - Checking GP task results status
- Demo
  - Periodic request to get and display location of fleet
  - Start/Stop timer





#4: Application flow



#4: Demo details

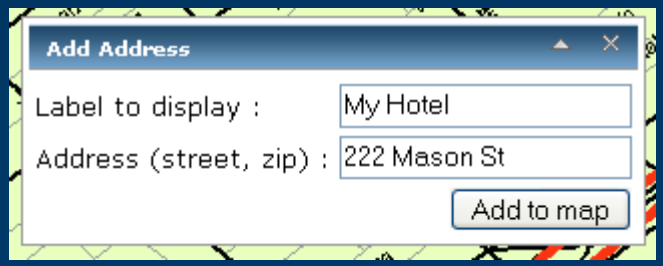
- **Javascript** (mapviewer.js)
  - Send request for data (trackingRequest())
- **Implement PhaseListener** (TrackingPhaseListener)
  - Process request (afterPhase())
  - Get data to return
  - Render & return XML response
- **Register PhaseListener** (fc-ajaxdemo.xml)
  - Add ‘<lifecycle>’ declaration
- **Javascript** (mapviewer.js)
  - Process response to update HTML (trackingResponse())

Customizing applications

- Web ADF
  - Base application
  - GIS functionality
- 3rd party Ajax toolkits
  - Clientside behavior & widgets
  - External data sources

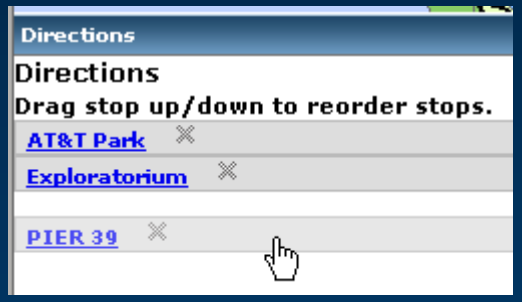
Demo: Trip Planner

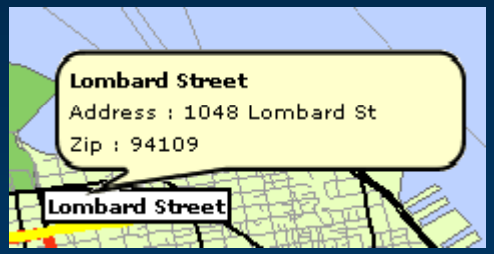
- Web mapping application
  - Custom left panel
  - Bottom directions panel
- Custom tool items
  - Send Ajax request
  - Toggle window visibility
- Custom window
  - To geocode address



Demo: Trip Planner (contd.)

- 3rd party toolkits
  - Prototype
    - DOM manipulation
  - Scriptaculous
    - Drag n' drop
    - Sorted lists
  - Overlib
    - Pop-ups





Tips & Tricks

- Start small and then build up application
  - ajaxdemo (mapviewer.html)
  - tripPlanner (tripPlanner0.html, mapviewer_orig.jsp)
- Browser tools
  - Firefox extensions
    - Firebug : Javascript debugging, Monitor XMLHttp
    - Web Developer : Page info, DOM info
  - IE debugging
    - Menu -> Tools -> Internet Options... -> Advanced :
    - Uncheck ‘Disable Script Debugging (Internet Explorer)
    - Uncheck ‘Disable Script Debugging (Other)’
    - Check ‘Display a notification about every script error’

Tips & Tricks (contd.)

- Issues with Clientside PostBack (edit jsp file)

```
<script type="text/javascript" language="Javascript">
  EsriUtils.doPostBack = false;
</script>
```

- Built in PhaseListeners and their request parameters
  - PostBackPhaseListener (doPostBack)
  - WebGraphicsPhaseListener (getWebGraphics)
  - ContinuousPanPhaseListener (doContinuousPan)
  - AGSPhaseListener (agsAjax)
  - GPAsyncTaskResultsPhaseListener (gpAsyncTaskResults)

Tips & Tricks (contd.)

- Render all responses within '<response>' tag
- Tags rendered by built-in AJAXRenderers
  - MapRenderer (<map>)
  - TocRenderer (<toc>)
  - ToolbarRenderer (<toolbar>)
  - OverviewRenderer (<overview>)
  - TaskRenderer (<task>)
  - GPAsyncTaskResultsRenderer (<gp-task-results-async>)
- To see XML output (edit web.xml)

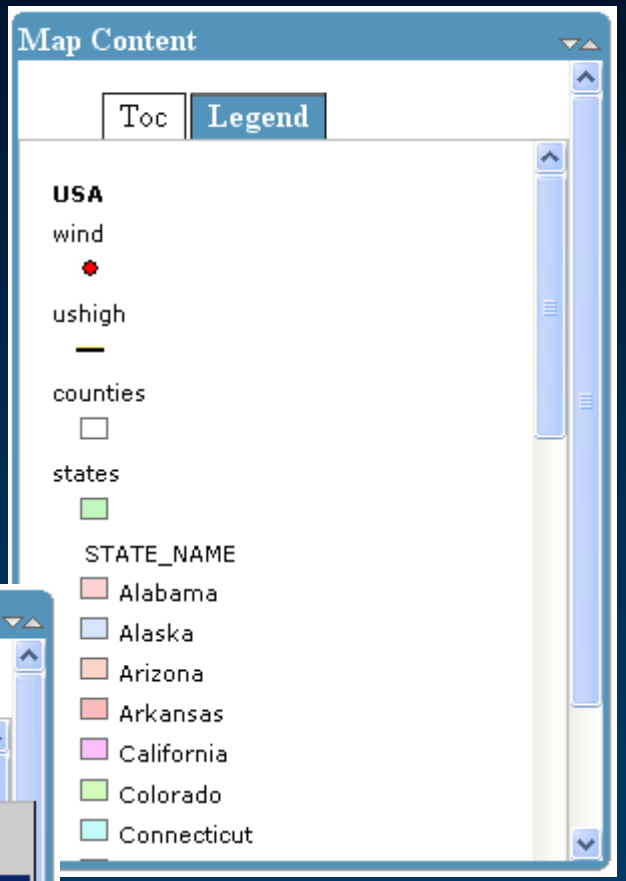
```
<context-param>
  <param-name>com.esri.adf.LOG_LEVEL</param-name>
  <param-value>CONFIG</param-value>
</context-param>
```

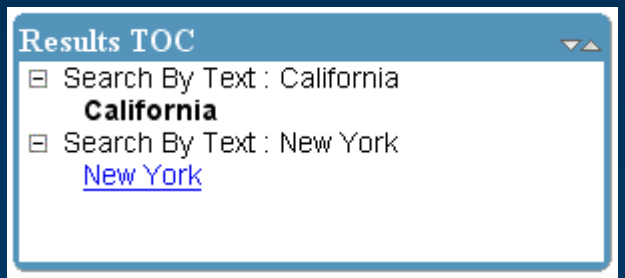
Tips & Tricks (contd.)

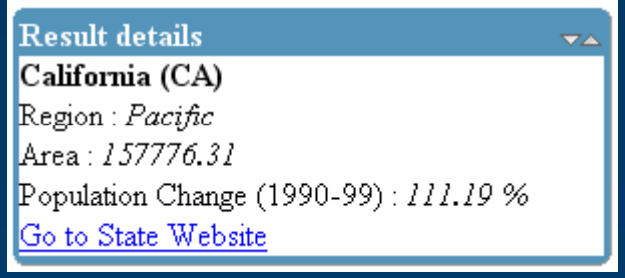
- Map Tools Task (faces-config.xml)
  - Pan tool to continuous pan
  - Identify tool to only identify states layer
- TOC visibility (context-attributes.xml)
  - TOC expanded to display all levels
- Optimize and remove unused objects
  - Managed beans & phase listeners (faces-config.xml)
  - ajaxGpAsyncTaskResultsRenderer (ajax-renderers.xml)
  - EsriTask_GPAsyncTaskResultsTimer javascript include & timer initialization (toc.xsl)
  - Removed functionality files (web.xml)

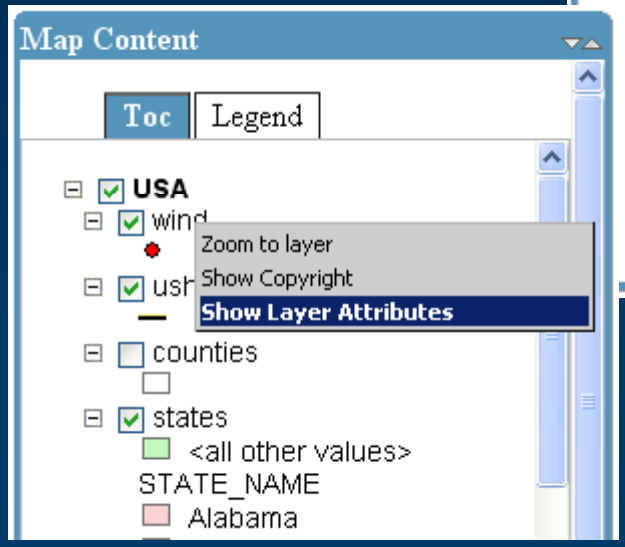
Additional customizations

- TOC rendering
  - Render custom TOC using EsriToc.nodes
- Result details rendering
  - Custom details formatting
- TOC context menu item
  - Functionality to display layer attributes









Q&A

- Our questions
  - What do you want to do using Ajax?
  - What do you want us to give you in terms of Ajax & Web ADF?
- Your questions

Presentation materials

- PowerPoint presentation and code are posted on the conference web site
  - http://www.esri.com/events/devsummit/index.html
- EDN – downloads and videos
- Java WebADF Docs
  - http://edn.esri.com/index.cfm?fa=java.gateway
  - http://edndoc.esri.com/arcobjects/9.2/Java/java/server/web_adf/ajax/ajax_integration.html
  - http://edndoc.esri.com/arcobjects/9.2/Java/java/server/web_adf/javascript/clientside_js.html
  - http://edndoc.esri.com/arcobjects/9.2/Java/server_samples/map-information/example.html

Further questions?

- TECH-TALK AREAS
  - What: Further opportunity to discuss questions and concerns with presenters and subject matter experts
  - Where: Tech Talk Area 3
  - When: during the next 30 minutes
- ESRI Showcase
- Meet the teams & SIG
  - Java Special Interest Group (Primrose C/D, Wednesday 12:00pm – 1:00pm)
  - Unix (Community Center, Wednesday 2:30pm)
- ESRI Developers Network (EDN) website
  - http://edn.esri.com