

ORACLE[®]

Oracle ADF Faces - Active Data Services **Get your answers before asking!**

Frank Nimphius
Senior Principal Product Manager

“There are very little things in this life I cannot afford and patience is one of them”

Larry Hagman

Problem Statement

- Rich Internet applications and Web 2.0 are members of the same disconnected Web
 - Traditional web applications use the same interaction model
- If the Web wants to replace the desktop, it needs to behave like a desktop
- Private use of the Internet drives business application requirements
- Tighter integration of client with server side changes
 - collaborative tooling
 - trading systems
 - live reporting
 - monitoring

Overcoming **the Disconnected Web**

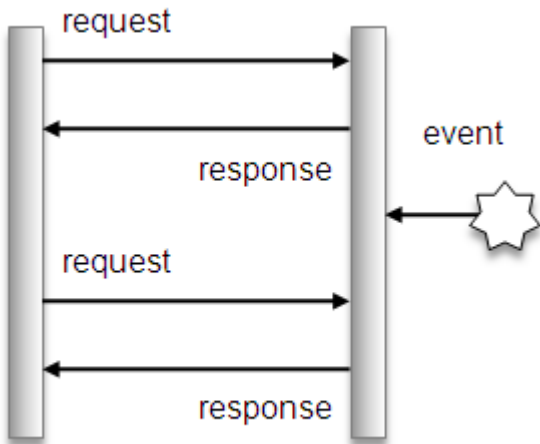


Web Interaction

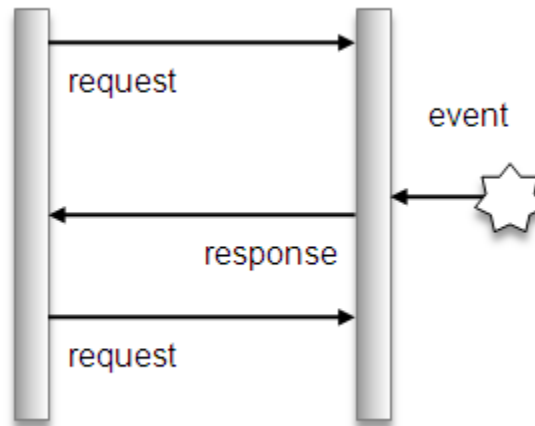
- **Poll**
 - Frequently sends a request to the server
 - Response may come back empty
- **Long Poll**
 - Client sends a request to the server
 - Response is not returned without event
 - Leader in popularity
- **Streaming**
 - Request opens connection
 - Response provides only parts of an answer
 - Event content added to response when ready

Server Side Push Mechanism

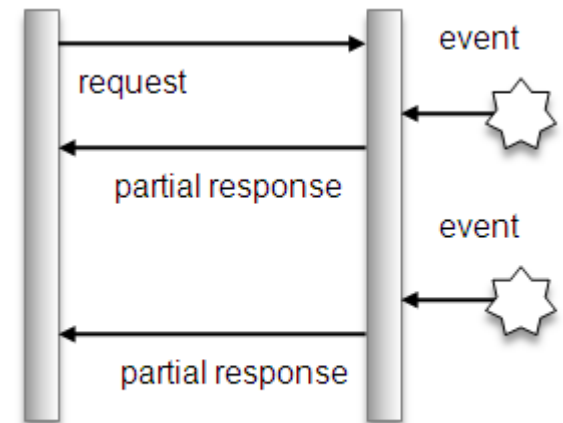
Polling



Long Polling



Push



Let's Understand the Mechanisms...

- polling **VS** long-polling **VS** streaming

	polling	long-poll	push
latency	-*	+	+ +
proxy servers	OK	OK	some issues**
live connection	++	-	-

* → depends on the interval...

** → fallback to long-polling



Product Demonstration

Twitter

Push Challenges

- Select the implementation that meets your environment the best
- Avoid close to the Http protocol coding
 - Find a framework or library
- Ensure multiple opened windows don't cause new client/server connections
- Look ahead for what is future safe

Choice ...

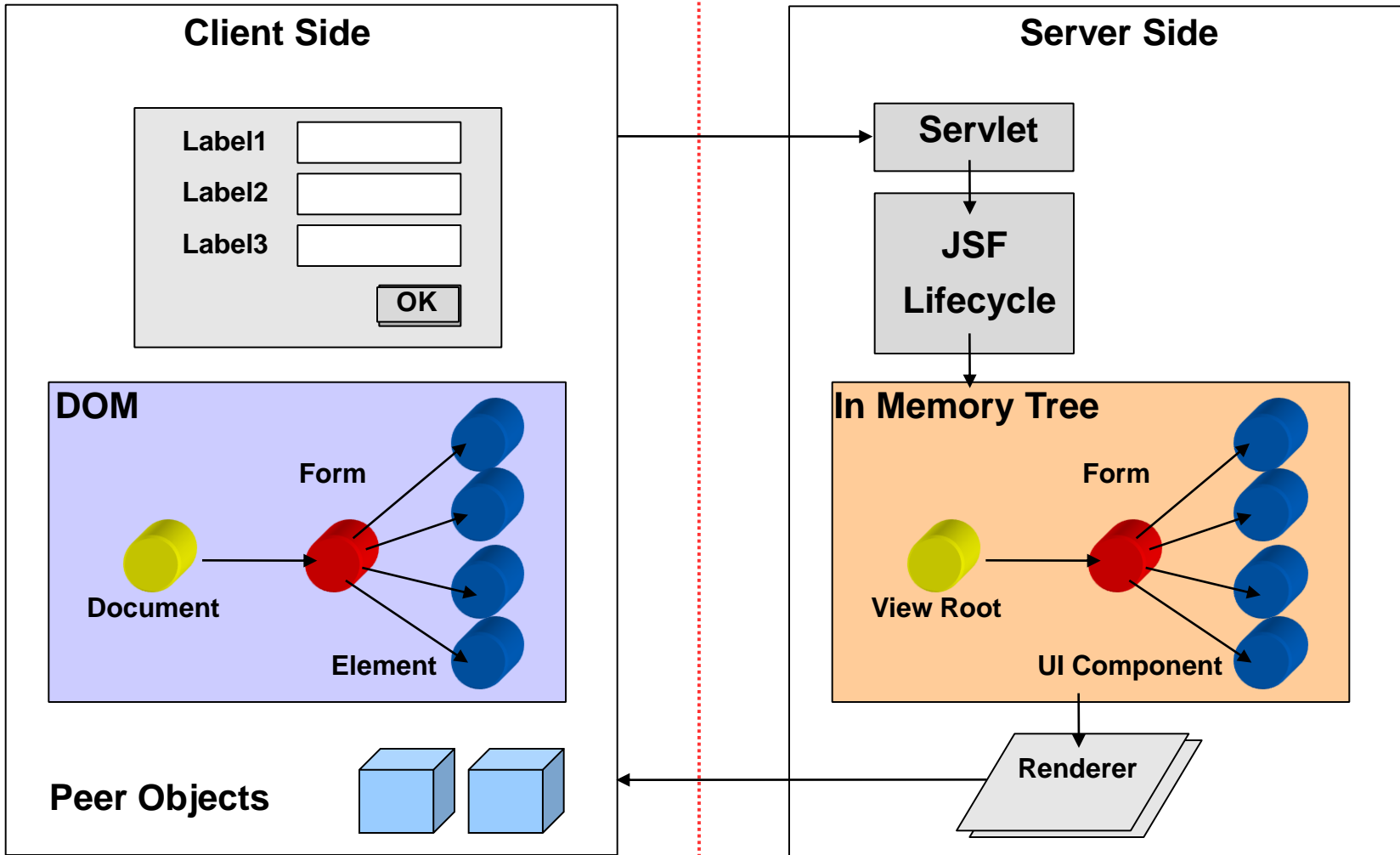


**... is not good if you don't know
what's good for you**

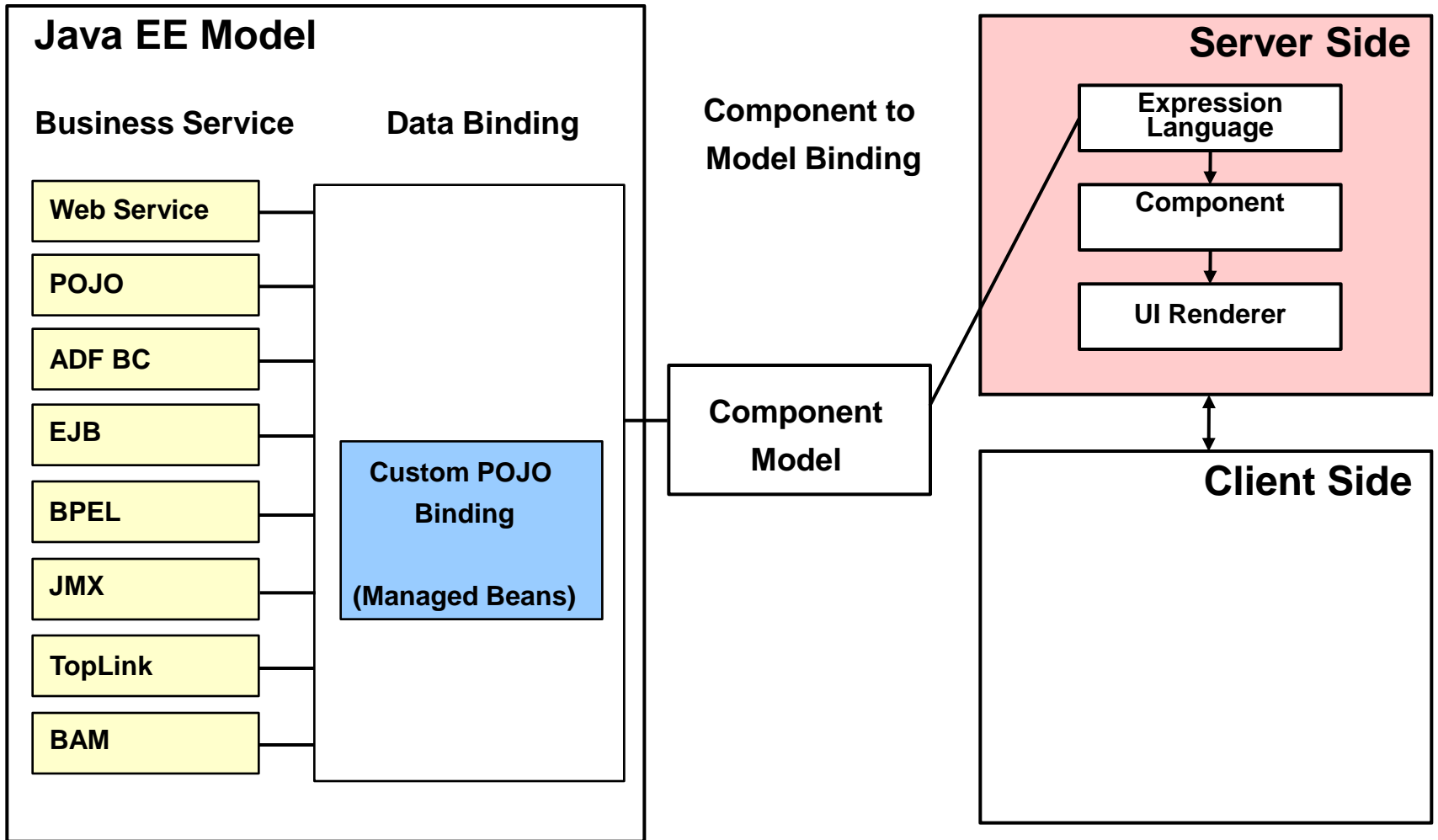
Real Time with **ADF Faces**



ADF Faces - Architecture



JavaServer Faces Architecture



ADF Faces **Client Side Poll**



af:poll

- Client side timer
- May access managed bean method or be used to partially refresh ADF Faces components
- "Pseudo Real Time" update for
 - ADF Business Components
 - WS/POJO Data Controls

```
<af:document>
  <af:form>
    ...
    <af:poll interval="5000"
              immediate="true" id="poll11"/>
  </af:form>
</af:document>
```

About the URL DataControl

- Sends GET request to provided URL
 - Endpoint URL
 - Base URL to access
 - May have Basic or Digest authentication specified
 - Source
 - Query parameters
 - Line must start with "?"
 - May contain variables (?attr1=##myval##&attr2...)
- Payload
 - Characters Delimited
 - XML
 - Requires XSD at Designtime



Product Demonstration

Twitter Reloaded – URL Data Control

ADF Faces **Auto PPR**



Partial Page Refresh

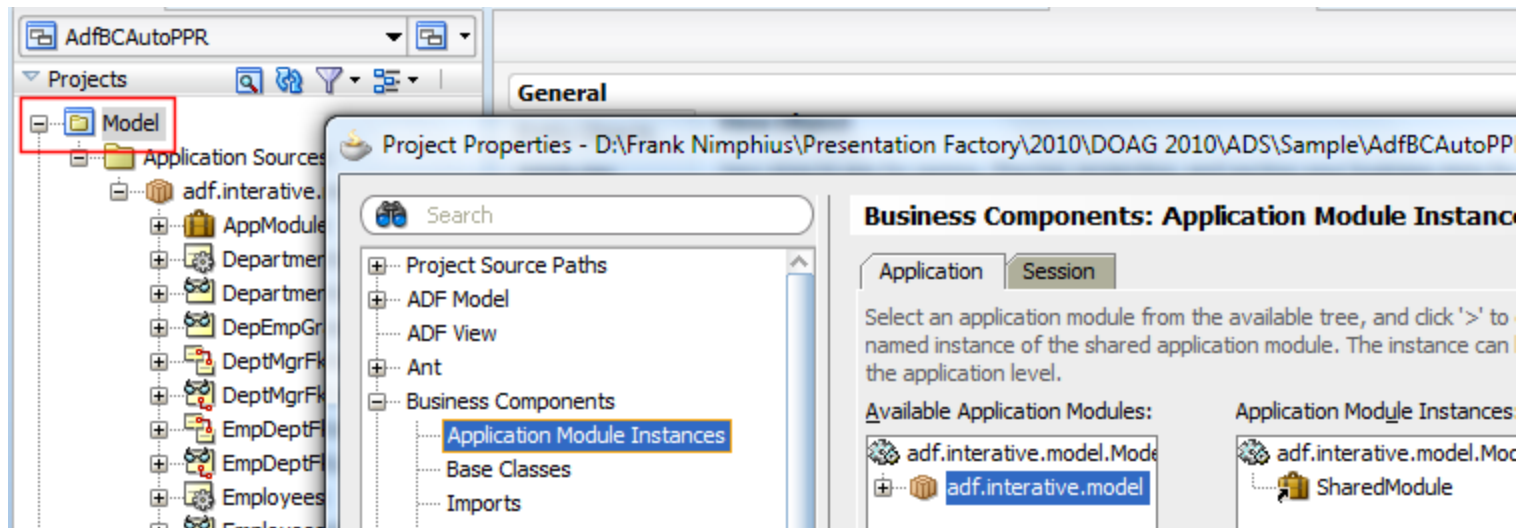
- The ability to invoke partial page rendering on a component
 - In Java using `addPartialTarget()` method on `AdfFacesContext`
 - Declaratively by configuring the component `PartialTriggers` property
 - Declaratively in ADF using the `ChangeEventPolicy` binding property, with a value of PPR
- Partial Refresh requires partial request/submit
 - `af:commandButton`, ...
 - `af:dialog`
 - `af:selectOneChoice` auto submit
- PPR
 - ... executes within the ADF Faces lifecycle
 - ... works on rendered components
 - ... Requires client side component to exist

Auto-PPR

- ChangeEvent Policy setting on ADF binding layer
 - Active Data Services uses FacesCtrl* bindings to
 - Detect changes in the binding layer
 - Identify control bindings that need to refresh its components
 - Add components to ADF Faces list of Partial Targets
- Works for UI component bindings
 - Attribute Binding, List Binding, Tree Binding ...
 - Does work also when UI component is bound to managed bean if managed bean accesses binding
 - Does not work for direct iterator access

How-to respond to RDBMS updates using ADF BC

- Use a mix of auto-PPR and af:poll
- In RDBMS: "grant change notification to <schema name>"
- Shared Application Module
 - Not all data change frequently
 - For better performance, this data could be cached and shared
 - Application, Session



How-to respond to RDBMS updates using ADF BC

- Configure View Object and set Auto Refresh to true

The screenshot displays the ADF BC Property Inspector for the 'DepartmentsView' View Object. The left pane shows the project structure with 'DepartmentsView' selected. The right pane shows the 'Tuning' section with the following properties:

Property	Value
Fetch Mode:	
Fetch Size:	
Max Fetch Size:	
Query Optimizer Hint:	
Retain Viewlink Accessors:	
Page Iterator:	Full
Passivate State:	<default> (NoTransient)
Auto Refresh:	true
Access Mode:	<default> (DEFAULT)

The 'Auto Refresh' property is highlighted with a red box, indicating it is set to 'true'.

How-to respond to RDBMS updates using ADF BC

- Only refresh the UI when data has changed
- Override View Object Impl
 - Listen for Database Change notification
 - Get and store timestamp
 - Expose method to get timestamp of last change notification
- Create managed beans
 - Reference from af:poll
 - Access operation binding to read time stamp on View Object Impl
 - Compare with time stamp saved on client
 - Refresh if time stamps differ

```
long lastRequery = 0;
...
public long getLastRequery() {
    return lastRequery;
}

//refresh query flag when View Object data is re-queried
@Override
protected void bindParametersForCollection(
    QueryCollection queryCollection,
    Object[] object,
    PreparedStatement preparedStatement) throws SQLException {
    super.bindParametersForCollection(
        queryCollection, object, preparedStatement);
    if (queryCollection != null) {
        lastRequery = System.currentTimeMillis();
    }
}

@Override
//Refresh query flag on database change event notification
protected void processDatabaseChangeNotification(
    QueryCollection queryCollection) {
    lastRequery = System.currentTimeMillis();
    super.processDatabaseChangeNotification(queryCollection);
}
```



Product Demonstration

RDBMS Change Notification

Active Data Services **Push**



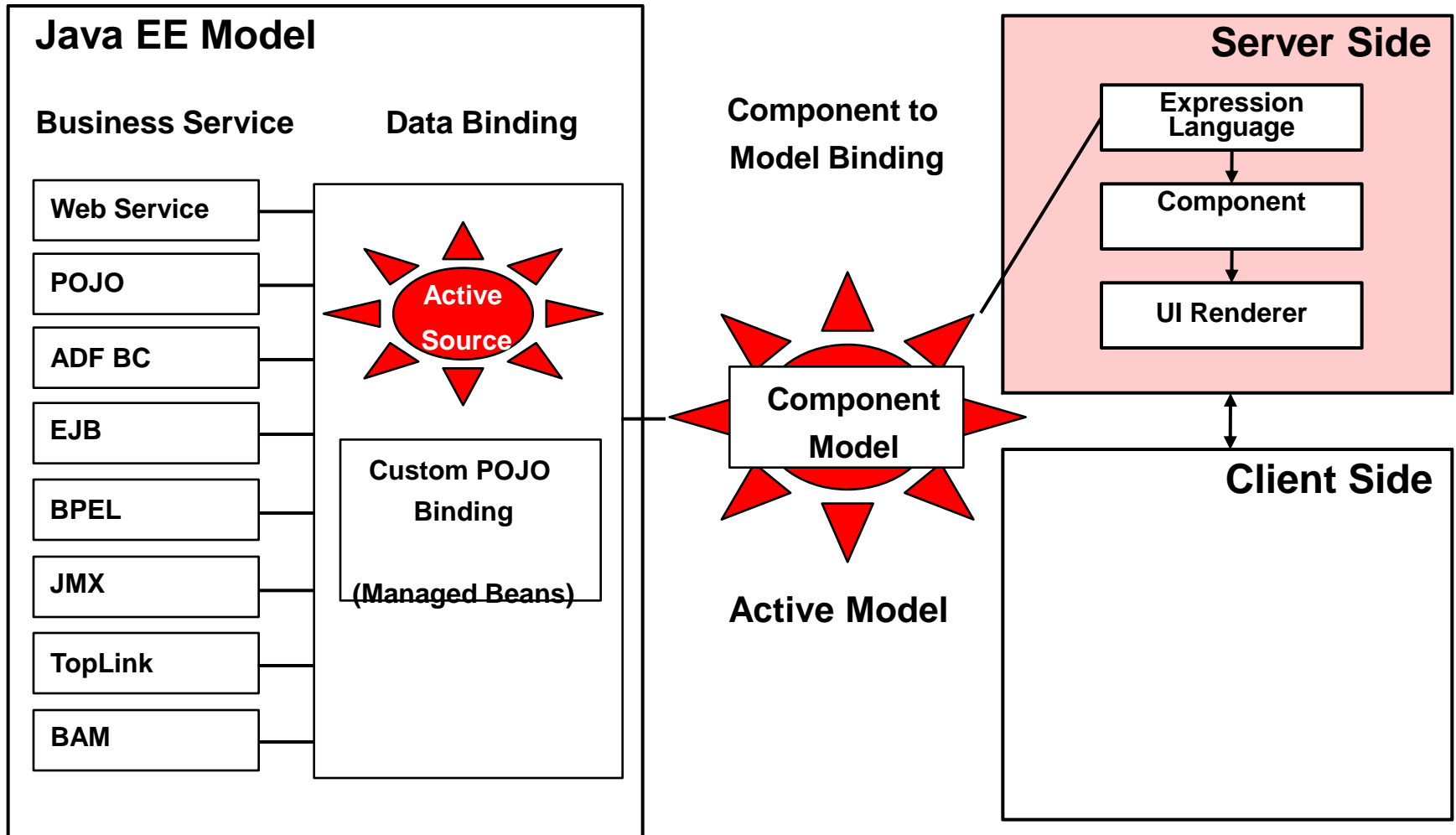
Implementing ADS in Web Applications

- Use for Dashboard views
- Active Data Support is integrated into ADF
 - BAM
- Important interface(s):
 - ActiveDataModel
 - ActiveDataListener
- Active Data Services Proxy
- No solution exists that is as mature

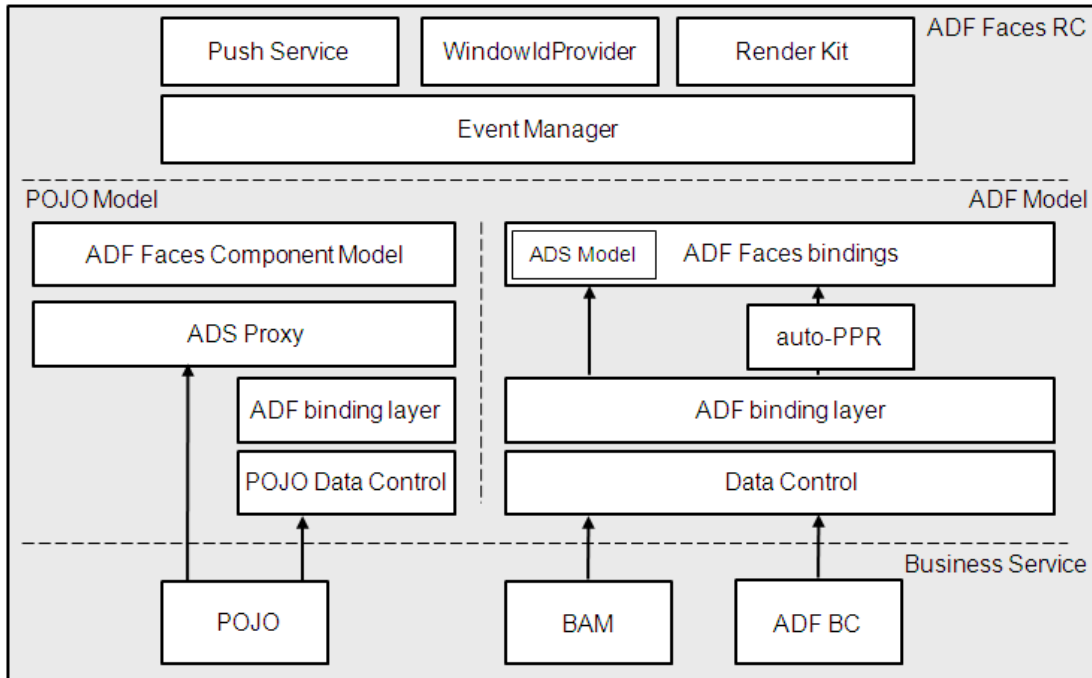
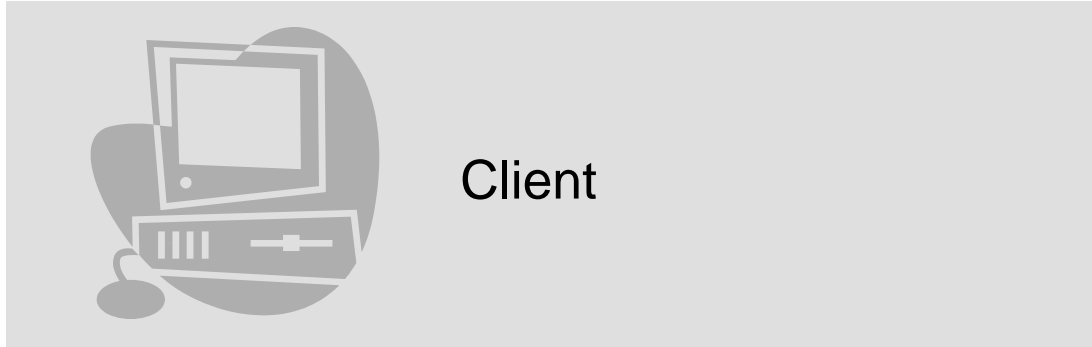
ADF Faces - Active Data Services (ADS) - Push

- Push server updates to the browser client
- Integrated framework for data streaming
- Based on JSF declarative model
- Externally configured
- Supports
 - Push
 - Long Polling
 - Polling
- Java APIs and ADF binding support
- ADF BC leverages auto-PPR

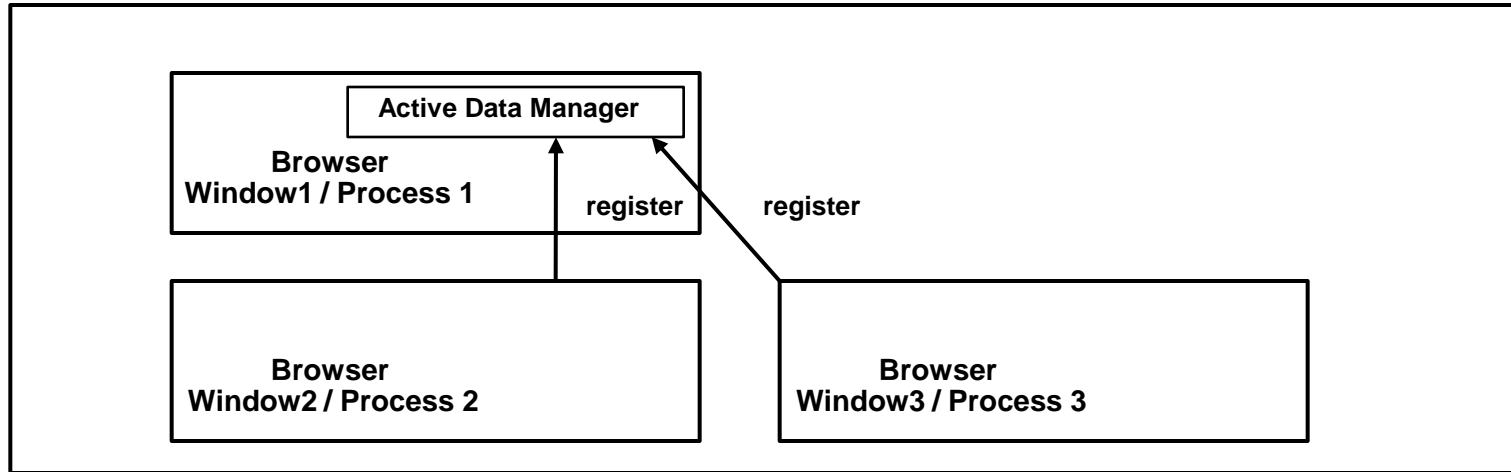
ADS in ADF Faces



ADS – Server Side Architecture



ADS – Client Side Architecture



- **Channel sharing**

- Multiple browser processes share the same ADS channel to address the limitation of browsers in the number of channels they can handle in parallel

- **Channel migration**

- If multiple browser windows are used and the parent window, the window that initialized the channel, is closed, then the channel parent window gets migrated to one of the child windows.

- **Multiplexing**

- A single ADS channel is used by multiple pages.

adf-config.xml

- <workspace_home>\.adf\META-INF
- In JDeveloper
 - Application Navigator | Application Resources | Descriptors | ADF META-INF
- What it is used for
 - **Active Data Services**
 - Enabling implicit save points
 - Persisting saved searches in MDS
 - Enabling Oracle ADF Security
 - Enabling seeded customizations
 - Configuring MDS

Active Data Services Simplest Configuration

- “streaming“, “polling” and “long-polling”

```
<adf-config xmlns="http://xmlns.oracle.com/adf/config"
  xmlns:adf="http://xmlns.oracle.com/adf/config/properties"
  xmlns:sec="http://xmlns.oracle.com/adf/security/config"
  xmlns:ads="http://xmlns.oracle.com/adf/activedata/config">
<?xml version="1.0" encoding="utf-8" ?>
  <adf-config xmlns="http://xmlns.oracle.com/adf/config"
    xmlns:ads="http://xmlns.oracle.com/adf/activedata/config">
    <ads:adf-activedata-config
      xmlns="http://xmlns.oracle.com/adf/activedata/config" >
      <transport>long-polling</transport>
    </ads:adf-activedata-config>
  </adf-config>
```




Product Demonstration

RDBMS Change Notification Reloaded

ORACLE®

Related Oracle Press Books



- Quick Start Guide to **Oracle Fusion Development**
- **Oracle JDeveloper 11g** Handbook
- **Oracle Fusion** Developer Guide

