

Apex: Core Concepts

Toon Koppelaars

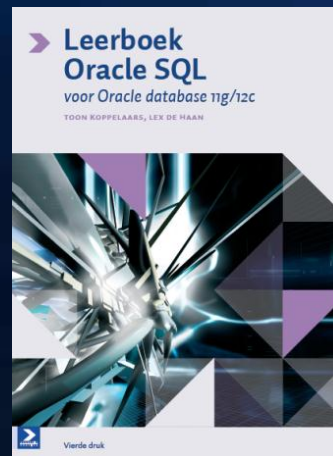
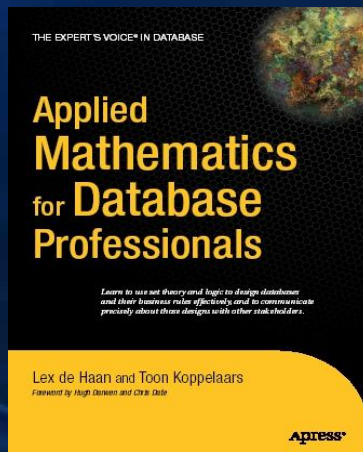


Oracle gebruikersclub Holland

dinsdag 9 april 2012 APEX DAG

Who am I?

- Oracle technology since 1987
- Twitter: @ToonKoppelaars
- Blogs
 - TheHelsinkiDeclaration.blogspot.com
 - HarmfulTriggers.blogspot.com



Apex: Core Concepts

- Session will deal with:
 - High level architecture
 - How to install it, how to configure it, →
In seven slides
- Rest of session will be about:
 - “I’m new to Apex, explain how it works”
 - “Give me a head start”

Only one hour...

- Web application: a simple model
 - Http protocol
- Apex core concepts
 1. Workspace/application/parsing-schema
 2. Page (regions, items, buttons)
 3. Page rendering
 4. Page processing
 5. Branching

[session context, pooling, Ajax, more...]
- Demo: building an application

Apex

- Apex is a DBMS hosted web application framework
 - The DBMS generates the html for the browser
 - Sounds like PFM, how does it do it?

Function F generates html

- Every call to Apex is a call to (pl/sql) function(?) F
 - F uses HTP and HTF packages to generate html (pl/sql Web Toolkit)
- F has parameter P
 - And a few others...
- General web syntax:
 - <url>/F?par1=val1&par2=val2&etc...

Show F in sqlplus and on URL

Deployment Modes

- Article by Dimitri Gielis
 - <http://www.ogh.nl/artikel.aspx?id=244>

Moving to the APEX Listener - *Dimitri Gielis*

Moving to the APEX Listener

In order to run Oracle Application Express (APEX) you need to make a choice which web server you want to use. With APEX 4.1 you have three choices:

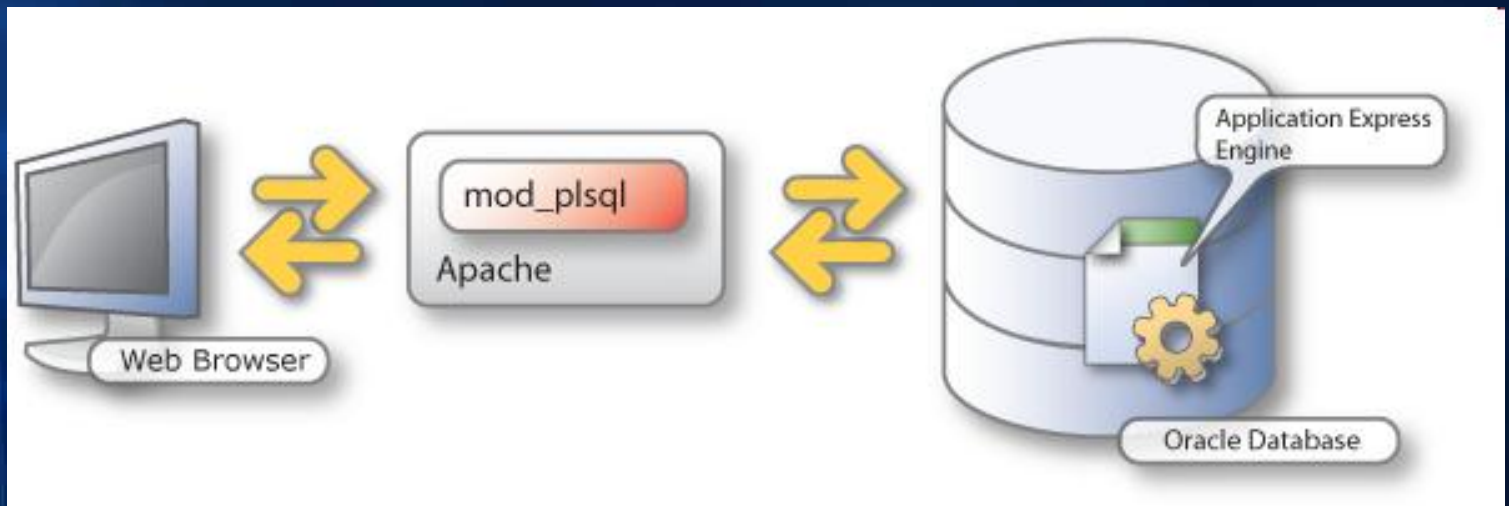
- **The Oracle HTTP Server with mod_plsql (OHS)**
- **The Embedded PL/SQL Gateway (EPG)**
- **The APEX Listener (standalone or on top of a Java Application Server)**

By Dimitri Gielis

This article will go briefly about the advantages and disadvantages of each option, but will dive deeper into the APEX Listener. The article tries to answer two questions in particular:

Deployment Modes

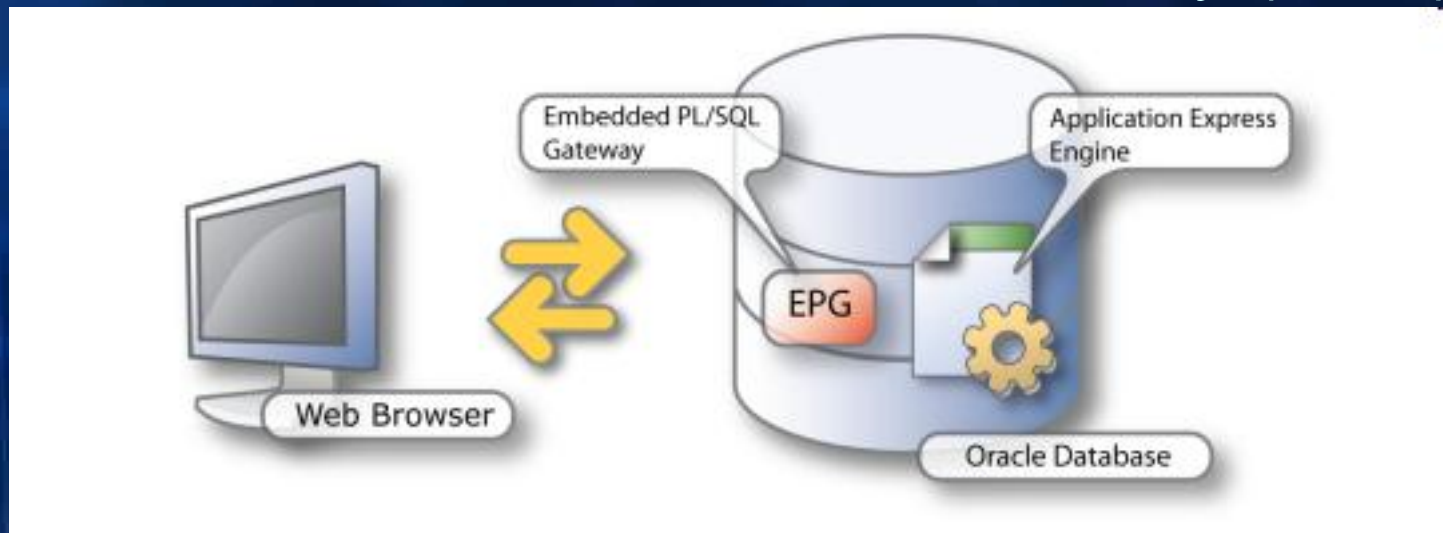
- Browser connected to Apache (OHS)
 - With mod_plsql to get into the DBMS



Mod_plsql receives html from F.
Hands over to Apache.
Streams to browser.

Deployment Modes

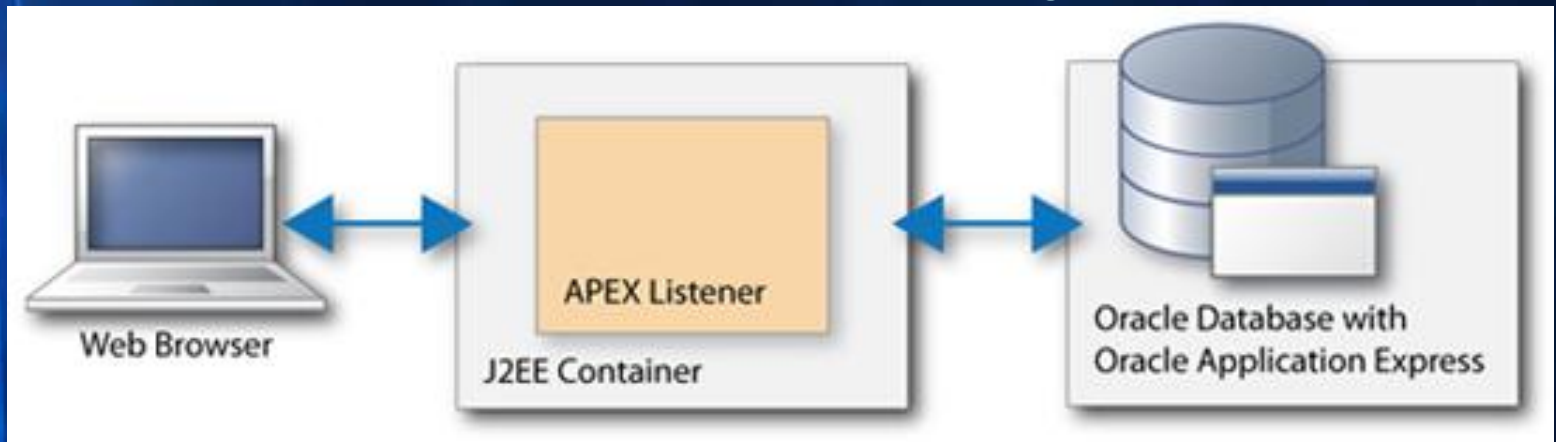
- Browser connected to sqlnet listener
 - Using http at XDB port
 - Embedded Procedural Gateway (EPG)



EPG receives html from F.
Hands over to listener.
Streams to browser.

Deployment Modes

- Browser is connected to Apex listener
 - Servlet running in JEE container
 - Tomcat, Glassfish, Weblogic, ...



Servlet receives html from F.
(Hands over to Apache.)
Streams to browser.

Installation & Configuration

- Install F + many more plsql objects in APEX_04xxxx schema
- Install images + javascript in Apache/EPG/JEE environment
- Then configure:
 - DAD for MOD_PLSQL (apex_public_user)
 - Embedded PLSQL Gateway (anonymous)
 - Apex listener (apex_public_user)

http://docs.oracle.com/cd/E37097_01/doc/install.42/e35123/toc.htm

Installation & Configuration

- After installation:
 - INTERNAL workspace + ADMIN user/pass
- Use ADMIN@INTERNAL to create *your* workspaces
 - With “Parsing schemas”
 - New workspaces have their own ADMIN (and *developers* and *users*)
 - Inside these workspaces you create your applications
 - And choose a parsing schema for them

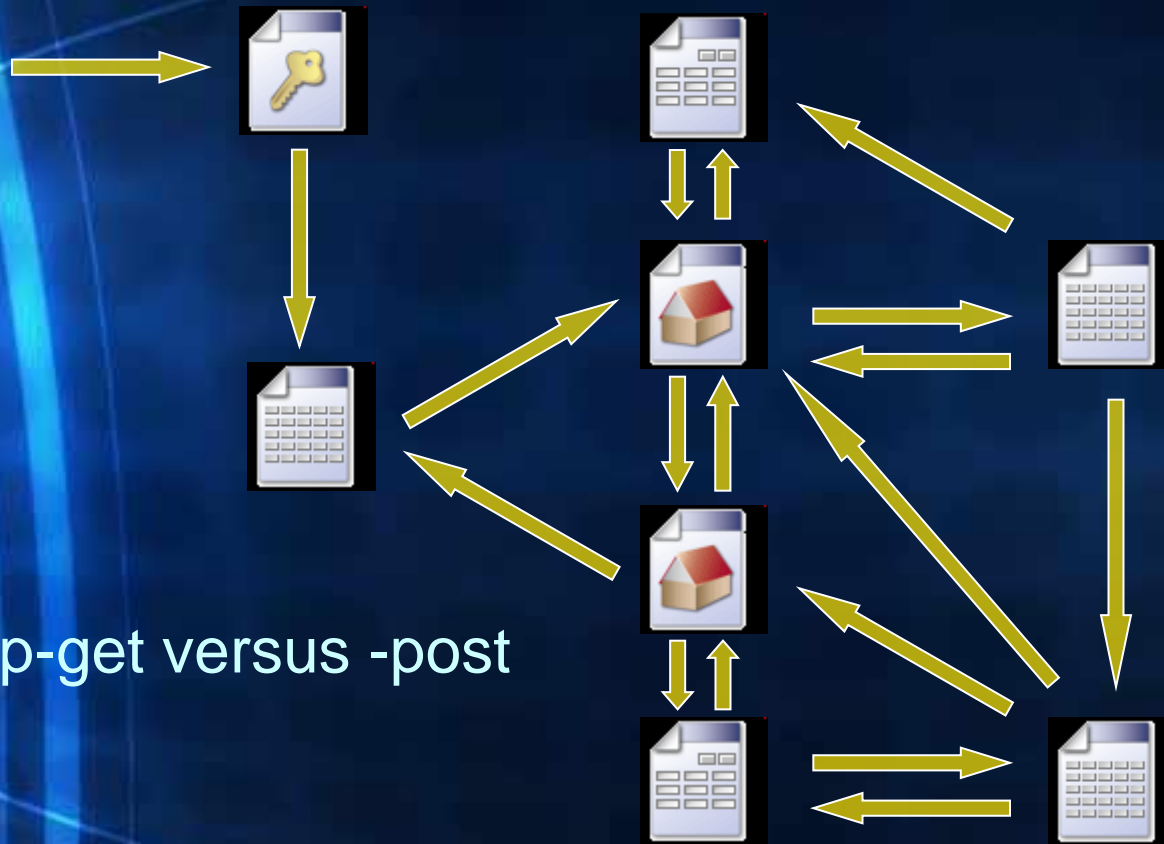
Demo: show internal/demo WS's + create empty application

Core Concepts

- Page
- Page processing
- Page rendering
- Branching

Web Application Model

- A network of pages



http-get versus -post

You navigate by clicking hyperlinks (urls) or pushing buttons

Apex Concepts: Page

- Every application has an id
- Every page has an id
 - URL shows what application and what page is currently displayed
 - `f?p=100:1`
- A page consists of:
 - Regions
 - Items / forms / reports
 - Buttons
 - ...

Page layout

- Standard html layout has 3 sections
 - Header
 - Body (Regions)
 - Footer
- In the IDE you're working on a page

Show: rendering sections
Show: region positioning (chosen template)

1 [Up] Go [Left] [Right] [List] [Table]

Run Utilities Create ADMIN 10 minutes ago

Page Rendering

- Home
 - Before Header
 - After Header
 - Before Regions
 - Regions
 - After Regions
 - Before Footer
 - After Footer
 - Dynamic Actions

Page Processing

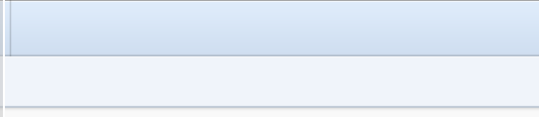
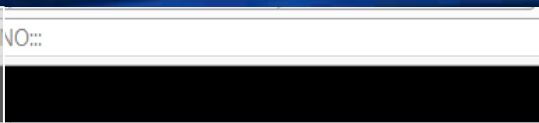
- After Submit
- Validating
- Processing
- After Processing
 - AJAX Callbacks

Shared Components

- Parent Tabs
- List of Values
- Breadcrumbs
- Lists
- Templates
- Security

Page Rendering +

- Home
 - Before Header
 - After Header
 - Before Regions
 - Regions
 - After Regions
 - Before Footer
 - After Footer
 - Dynamic Actions



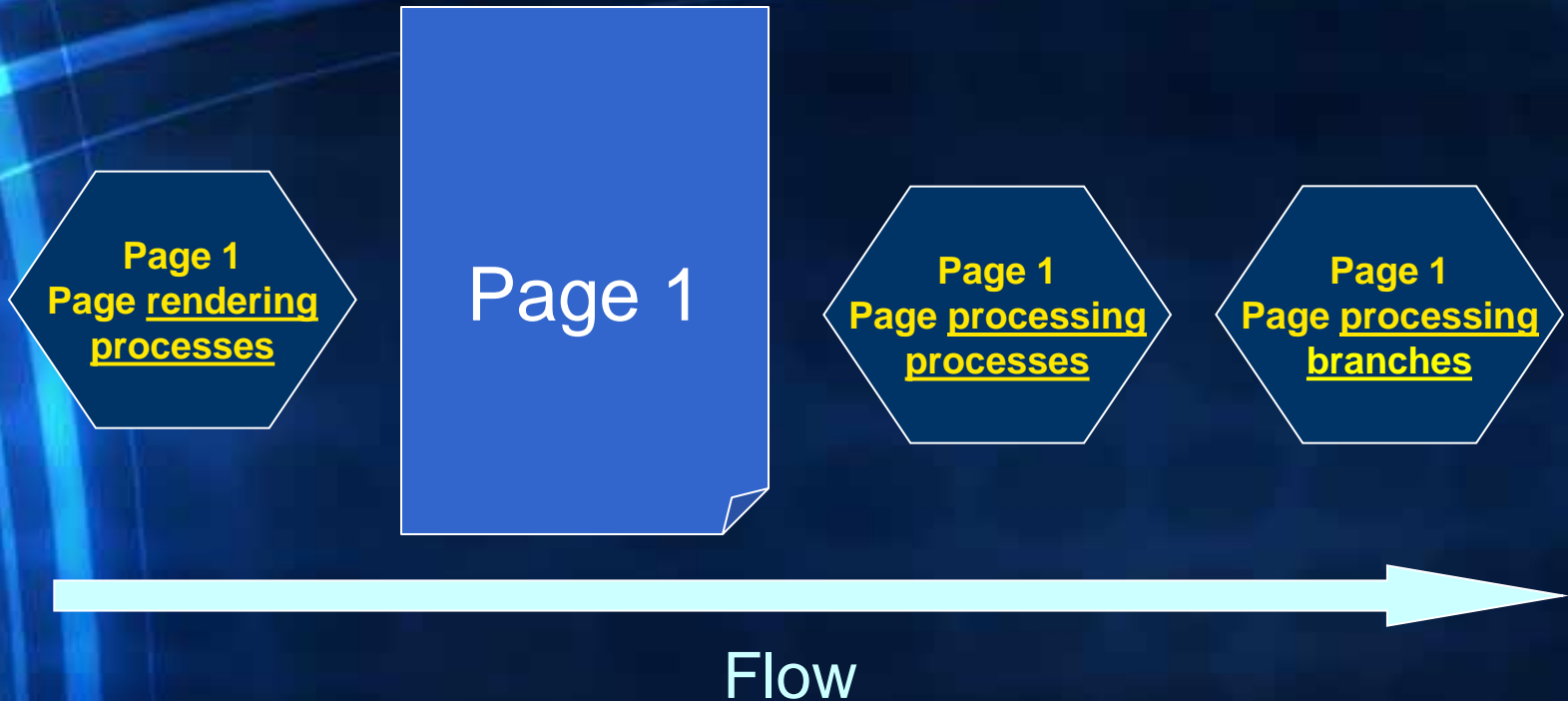
Page Processing +

- After Submit
- Validating
- Processing
- After Processing
 - AJAX Callbacks

Shared Components +

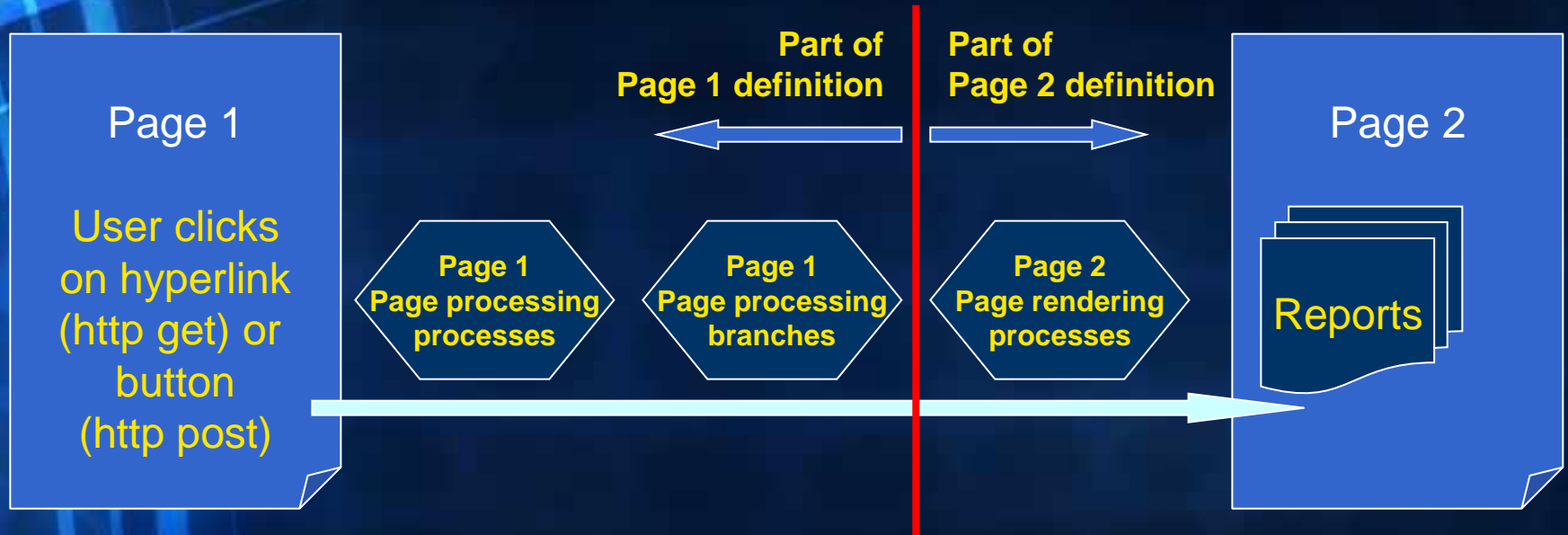
- Parent Tabs
- List of Values
- Breadcrumbs
- Lists
- Templates
- Security

One page definition



Show: rendering + processing process types
Show: branch types

From Page 1 to Page2



On hyperlink: flow goes directly to Page 2 rendering.
I.e Page 1 processing and branches are skipped.

Apex Concepts: Order of Execution

- Processes (rendering and processing)
and,
- Branches

all have a sequence number and optional condition

For rendering: Items etc. all have sequence numbers too

In Summary:

Page rendering section

Regions (reports, ...)

Page processing section

The screenshot displays the Oracle APEX Application Express interface. The top navigation bar includes 'ORACLE Application Express' and 'Workspace DEMO (Logout)'. Below the navigation, there are tabs for 'Application Builder', 'SQL Workshop', 'Team Development', and 'Administration'. The main content area is divided into three sections: 'Page Rendering', 'Page Processing', and 'Shared Components'. The 'Page Rendering' section shows a tree view with 'Regions' and 'Processes' highlighted. The 'Page Processing' section shows a tree view with 'After Submit', 'Validating', 'Processing', 'After Processing', 'Branches', and 'AJAX Callbacks' listed. The 'Shared Components' section shows a tree view with 'Tabs', 'List of Values', 'Breadcrumb', 'Lists', 'Templates', and 'Security'. Red circles and arrows highlight specific elements: a red circle around 'Regions' in the Page Rendering section, a red circle around 'Processing' and 'Branches' in the Page Processing section, and a yellow bracket around the 'Shared Components' section. Red arrows point from the text labels to these elements.

(Page rendering) processes

(Page processing) processes

Branches (to other pages)

Page definition

Demo 1

- EMP – DEPT database design
 - Show tables and view
- Create report based on view
V_EMP_DEPT

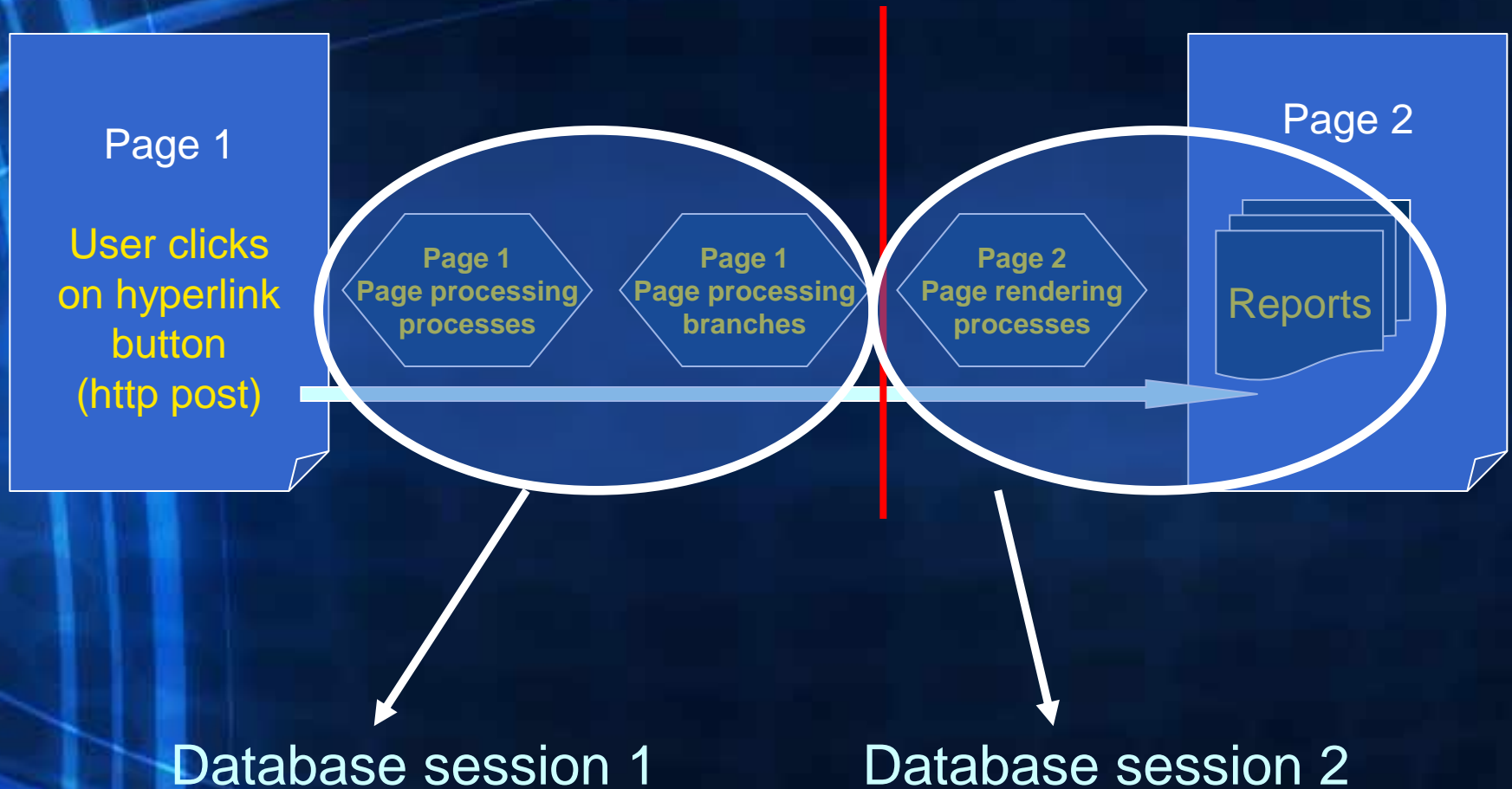
Demo 2

- Create page 2 for editing EMP details

Closing Remarks

- Sorely missing: a good concepts guide
- A clear transaction model
 - To be documented in missing concepts guide...
 - When does it start?
 - Are there internal commits?
 - Do we have to commit?
 - Can we use savepoints?
 - When is Apex context initialized, changed, saved?
 - And what about sessions / session pooling?
 - Beware of *your own* context

From Page 1 to Page2



Closing Remarks

- You will need build standards & guidelines
- Use it wisely: adopt a fat database architecture
- Don't fall in the same YAFET trap again:
→ design UI API layer
- You have a database project, not an Apex project

Yet Another FrontEnd Technology

The “Fat Database”

Outside

UI technology “du-jour”

Design
By
Contract

4. Business services layer

3. Business logic layer

2. Data services layer

1. Database design with
integrity constraints / triggers

Inside DBMS

Outside

Browser met Javascript

Application Express

Design
By
Contract

4. Business services layer

3. Business logic layer

2. Data services layer

1. Database design met
integrity constraints / triggers

Inside DBMS

Closing Remarks

- You *can* build robust enterprise applications with Apex
 - Use wizards wisely
 - Or don't use them like you didn't in Forms
 - Understand what they generate
 - Learn html(5), javascript and CSS
- Apex is here to stay for a long time
☺

Questions?

