

Apex: Core Concepts

Toon Koppelaars
www.RuleGen.com



Oracle gebruikersclub Holland

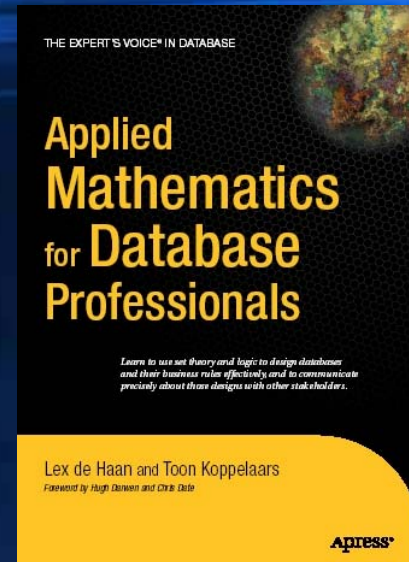
dinsdag 30 maart 2010 APEX DAG

Who am I?



De dikke database

TheHelsinkiDeclaration.blogspot.com



RULEGEN
Delivering the
Relational Promise

Apex: Core Concepts

- Session will deal with:
 - How to install it, how to configure it, how to administer it
- In three slides
- Rest of session will be about:
 - “I’m new to Apex, tell me how it works”
 - “Give me a headstart”

Only one hour...

- Web application: a simple model
- Apex core concepts
 1. Page
 2. Page rendering
 3. Page processing
 4. Branching
- Demo: building an application

Apex

- Apex is a DBMS hosted web application framework
 - The DBMS generates the html for the browser
 - With embedded IMG tags (pictures), Javascript references

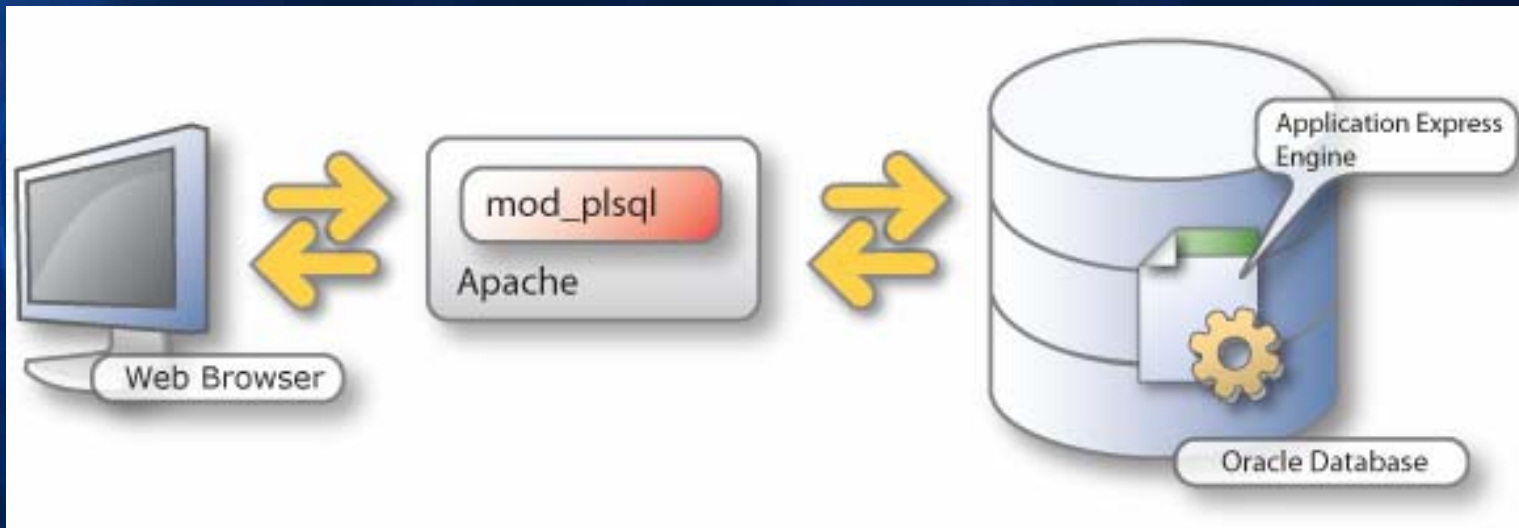
Function F generates html

- Every call to Apex is a call to (pl/sql) function F
- 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

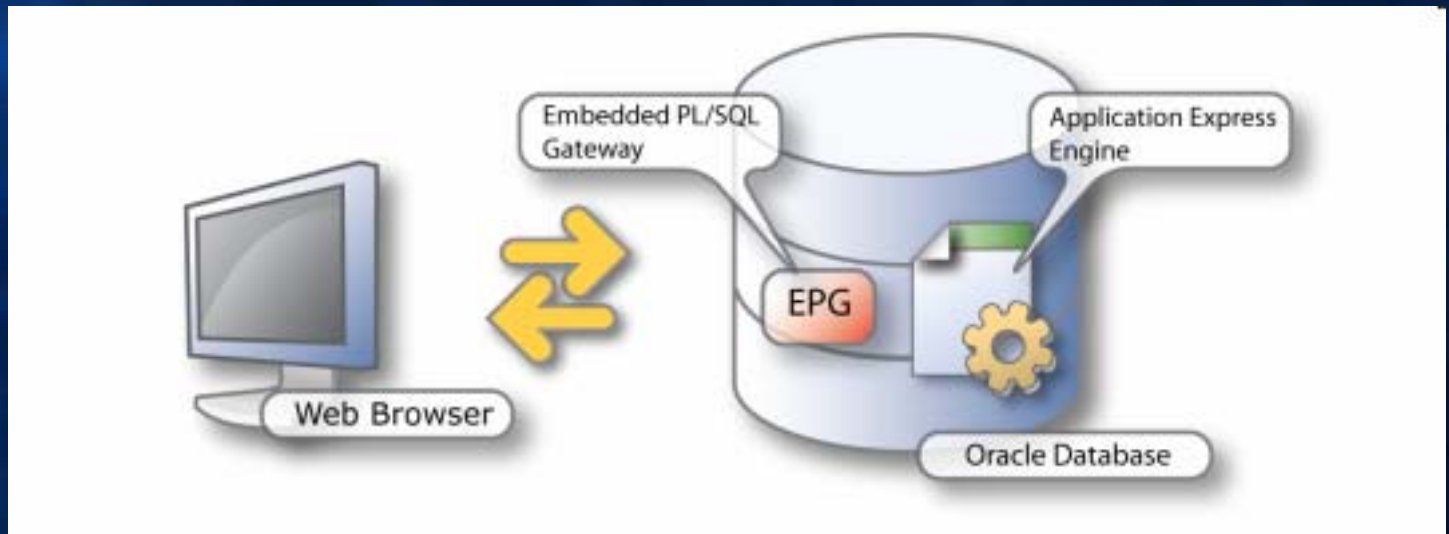
Installation & Configuration

- Browser is connected to Apache
 - With mod_plsql to get into the DBMS



Installation & Configuration

- Or, as of 11G, browser can go straight to Oracle listener
 - Embedded Procedural Gateway (EPG)



Installation & Configuration

- Install objects in FLOWSxxx schema
- Install images/javascript in Apache environment
- Configure DAD for MOD_PLSQL

Or,

- Configure embedded plsql gateway (EPG)

http://download.oracle.com/docs/cd/B32472_01/doc/install.300/b32468/toc.htm

http://www.oracle.com/technology/obe/11gr1_db/install/apexinst/apexinst.htm

Or, use Google

Installation & Configuration

- After installation:
 - INTERNAL workspace + ADMIN user/pass
- Use ADMIN@INTERNAL to create other new 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 it

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

Core Concepts

- Page
- Page processing
- Page rendering
- Branching

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

- Header
- Region(s)
- Footer

- All can contain:
 - Items, forms, reports, buttons

Show: new region + positioning (chosen template)

Page 1 View Definition Go

Run **Copy** **Delete** **Create >**

Page Rendering



Page

Page Name: [Page 1](#) Template: [Application default](#)

Title: [Page 1](#) Header Text:

URL Header: Footer Text:

HTML Body: Build Option:

Help Text: [No help is available for this](#) Authorization: [No](#)

Page Group: Cached: [No](#)

Regions



Display Point: Page Template Body (1)

Zoek HTML

Sessions Report

Display Point: Region Position 01

[Breadcrumbs](#) [Breadcrumb Entry](#)

Buttons



Label: Zoek Submit as "ZOEK"

Label: Sessions Submit as "KILL_SESSION"

Items



Label: Zoek

1 WHERE Text Field

Computations



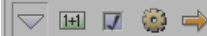
Processes



Page: nl

Space: HELSINKI User: ADMIN

Page Processing



Computations



Validations



Processes



After Submit

10 Kill_session PL/SQL anonymous block Conditional

Branches



After Processing

10 Go To Page_1 Conditional

9999 Go To Page_1 Unconditional

Shared Components



Tabs

Tab Set: TS1

Page 1

Lists of Values

Breadcrumbs

[Breadcrumb](#)

Lists

Templates

Page [One Level Tabs](#)

Region [Breadcrumb Region](#)

Region [Reports Region](#)

Label [Optional Label with Help](#)

Button [Button](#)

Breadcrumb [Breadcrumbs](#)

Report [Standard Report](#)

Theme

3. [Business](#)

Security

Navigation Bar

200. [Logout](#) Redirect To &LOGOUT_URL.

Page Rendering



Page Name: [Page 1](#) Template: [Application default](#)
 Title: [Page 1](#) Header Text:
 Footer Text:
 HTML Body: Build Option:
 Help Text: [No help is available for this](#) Authorization: [No](#)
 Page Group: Cached: [No](#)

Regions

Display Point: Page Template Body (1) [v] [^]
 Zoek HTML
 Sessions Report

Display Point: Region Position 01
 [v] [^] [v] [^] [v] [^] [Breadcrumbs](#) [Breadorumb Entry](#)

Buttons

Label: Zoek Submit as "ZOEK"
 Label: Sessions
 Label: Kill Session Submit as "KILL_SESSION"

Items

Label: Zoek [v] [^]
 Label: WHERE Text Field

Computations

Processes

Page: nl

Space: HELSINKI User: ADMIN

Page Processing



Computations

Validations

Processes

After Submit
 10 Kill session PL/SQL anonymous block Conditional

Branches

After Processing
 10 Go To Page 1 Conditional
 9999 Go To Page 1 Unconditional

Shared Components



Tabs

Tab Set: TS1 [v] [^]
 Page 1

Lists of Values

Breadcrumbs

[Breadorumb](#)

Lists

Templates

Page [One Level Tabs](#)
 Region [Breadorumb Region](#)
 Region [Reports Region](#)
 Label [Optional Label with Help](#)
 Button [Button](#)
 Breadorumb [Breadcrumbs](#)
 Report [Standard Report](#)

Theme

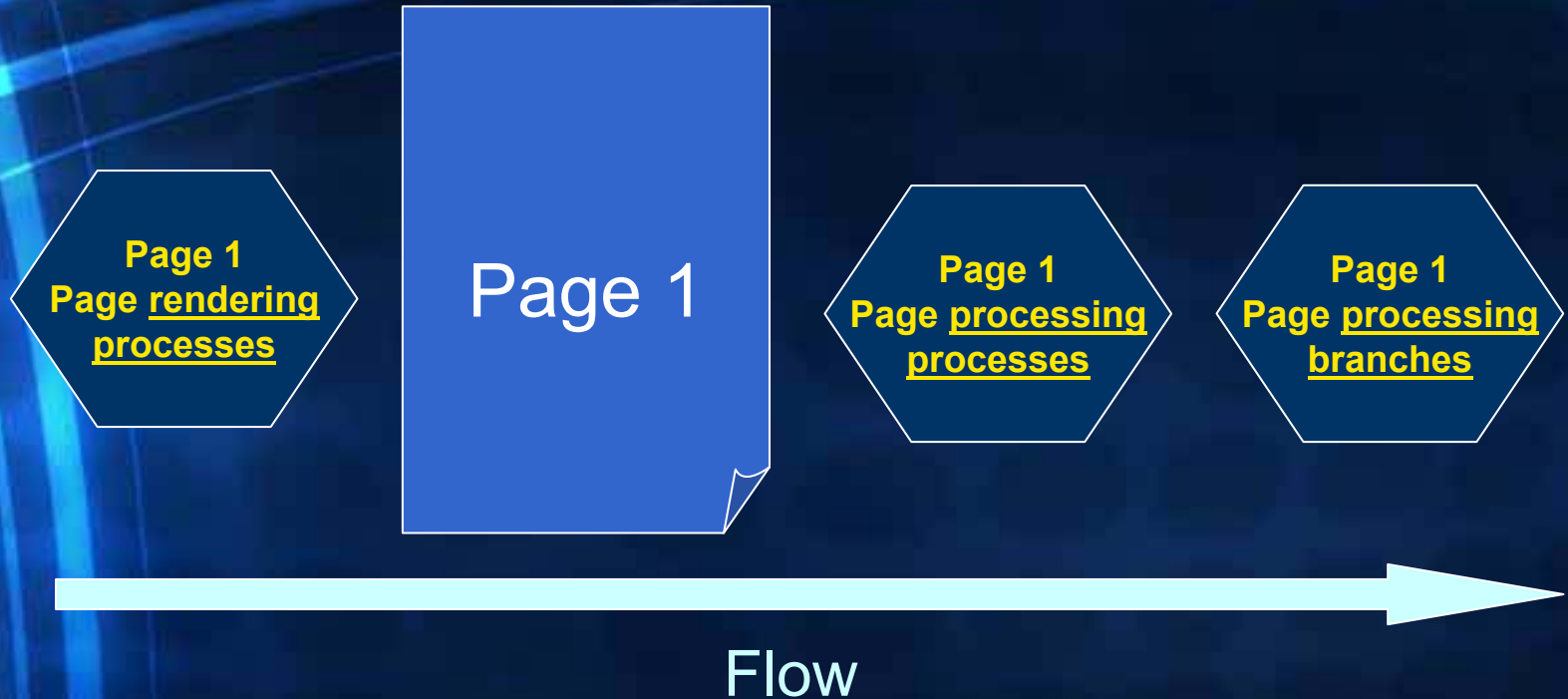
3. [Business](#)

Security

Navigation Bar

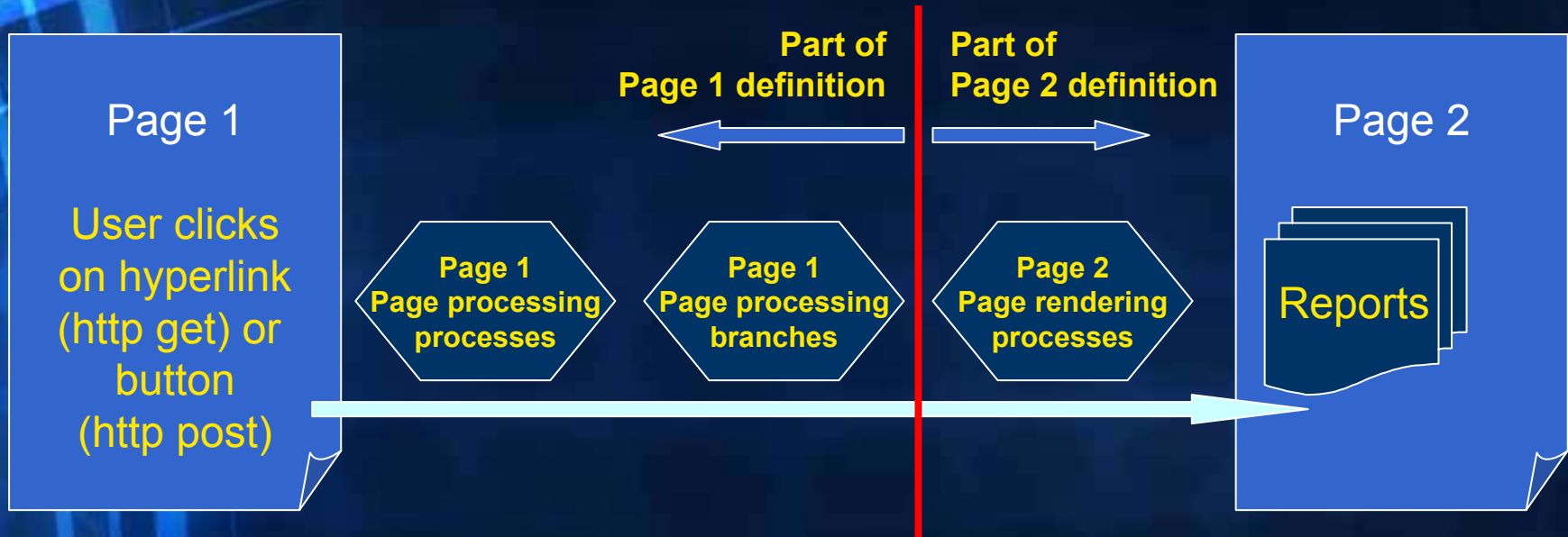
200. [Logout](#) Redirect To &LOGOUT_URL.

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 (for you to choose the value)

For rendering: Items etc. all have sequence numbers too

In Summary:

Page processing section

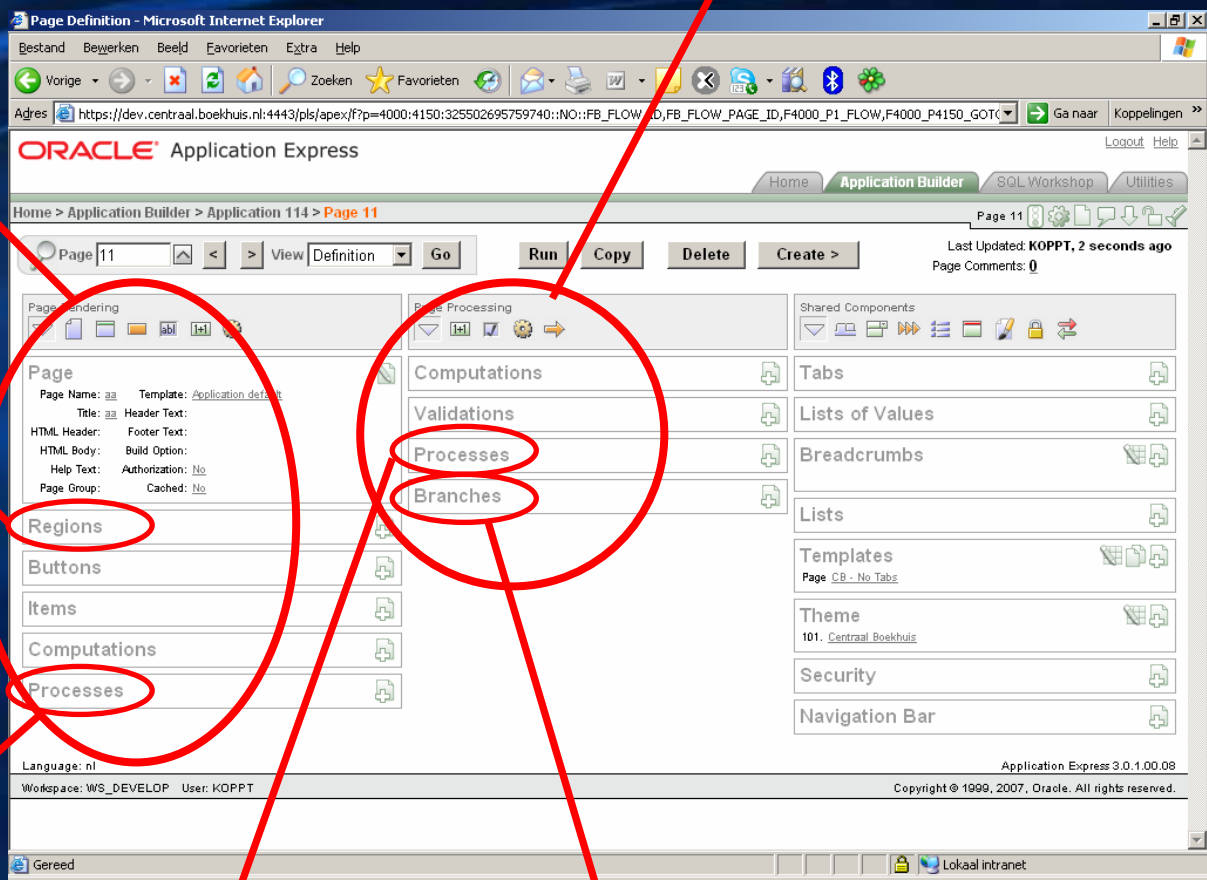
Page rendering section

Regions (reports, ...)

Page rendering processes

(Page processing) processes

Branches (to other pages)



Page definitio

Demo 1

- EMP – DEPT database design
 - Show acc.sql
- Create report based on view
V_EMP_DEPT

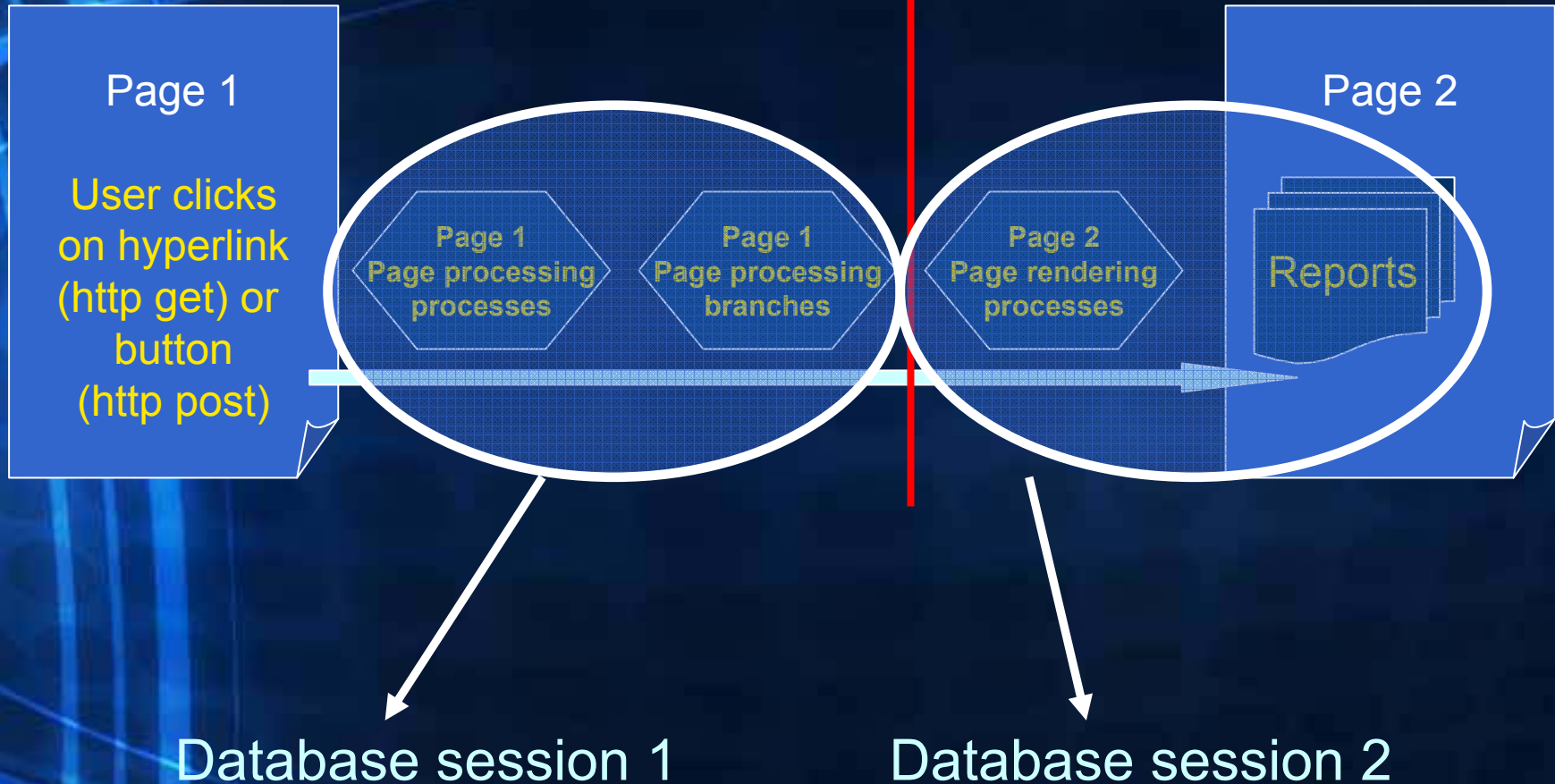
Demo 2

- Create page 2 for editing EMP details

Closing Remarks

- Sorely missing: a good concepts guide
 - Current docs → too much point-and-click like
- 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 Page 2



Closing Remarks

- No support for ref-cursors (yet)
 - Producer/consumer of query results
 - Design by contract
- Immature server-side error handling
 - Declarative constraints
 - Your `raise_application_errors`
 - In triggers, in stored procedures
- Poor DD-driven client-side constraint support
 - Need of meaningless sequence

Closing Remarks

- Apex 3.x = Forms 2.x
 - Implies: you'll need build standards & guidelines
- Apex 4.0 = Forms 3.0?
- Use it wisely: adopt fat database paradigm
- 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

Inside DBMS

4. Business services layer

3. Business logic layer

2. Data services layer

1. Database design met
integrity constraints / triggers

MVC
compliant

Closing Remarks

- You can build robust enterprise applications with Apex
 - Just don't use the wizards (like you didn't in Forms)
 - Learn html, javascript and CSS
- Apex is here to stay for a long time
😊

Questions?

Toon@RuleGen.com



TheHelsinkiDeclaration.blogspot.com

