



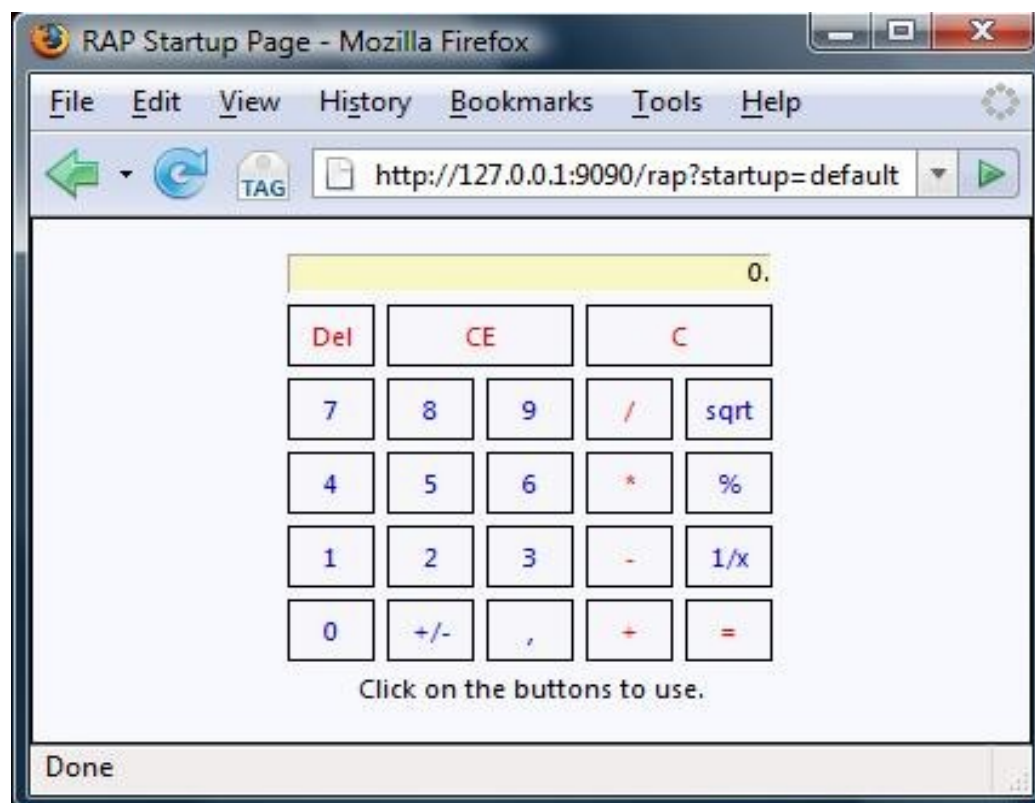
Eclipse Rich Ajax Platform (RAP) Writing Webapps in Java

Salem Java User Group, March 4th 2008

Elias Volanakis

evolanakis@innoopract.com

Our Goal: Write a Calculator



Eclipse Rich Ajax Platform Project (RAP)

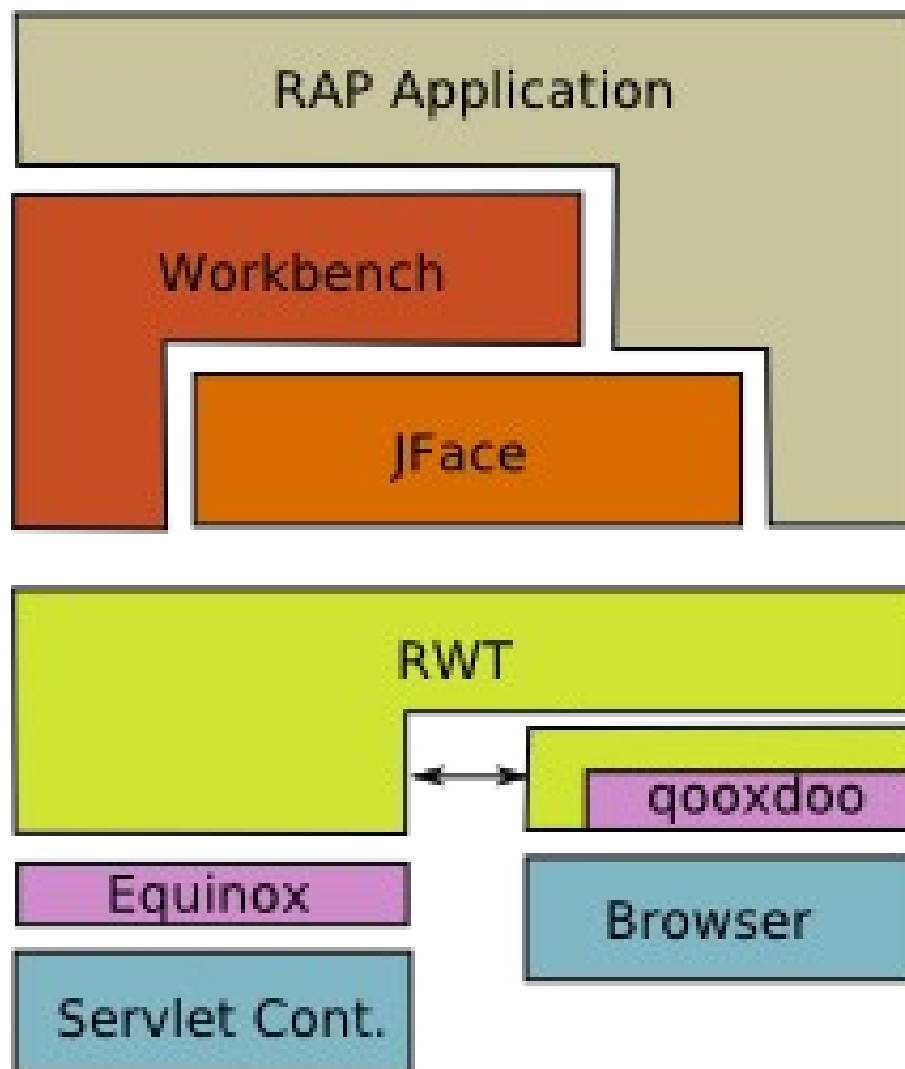
Goal: develop rich Ajax-enabled web applications by:

- coding in Java
 - using the full Java libraries
 - using Java-based Widget Toolkit
 - component-oriented development model (plug-ins / OSGi)
 - extensible (xml-meta info + java-code)
 - without hand-coding Ajax
-
- **skill preservation** – leverage existing Java / Eclipse know-how
 - **single sourcing** of rich client and rich internet applications

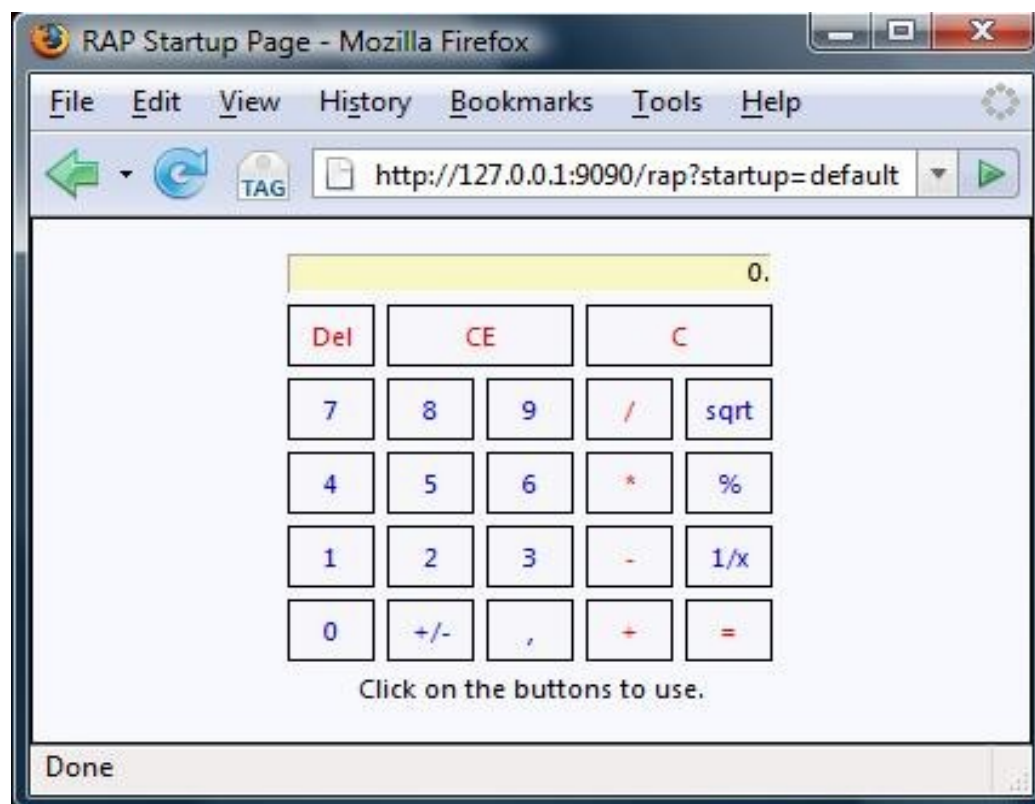
How does it work?

- RAP Widget Toolkit (RWT) - port of the [Standard Widget Toolkit](#) (SWT)
- server-centric - logic resides on the server
- UI is rendered by emitting browser-side JavaScript at run-time
- user action triggers a state-update (example button was clicked)
- state is recomputed on server
- only the “delta” is transmitted (very small)

RAP Architecture Overview



Show me the code!



Get RAP - <http://www.eclipse.org/rap>



Demos

See some demos here



Downloads

Get the latest RAP release

The RAP project enables developers to build rich, Ajax-enabled Web applications by using the Eclipse development model, plug-ins with the well known Eclipse workbench extension points, JFace, and a widget toolkit with SWT API (using **qooxdoo** for the client-side presentation). The project has graduated from incubation and released its 1.0 release.

[Learn more ...](#)

Thanks for attending!

- Contact Info:

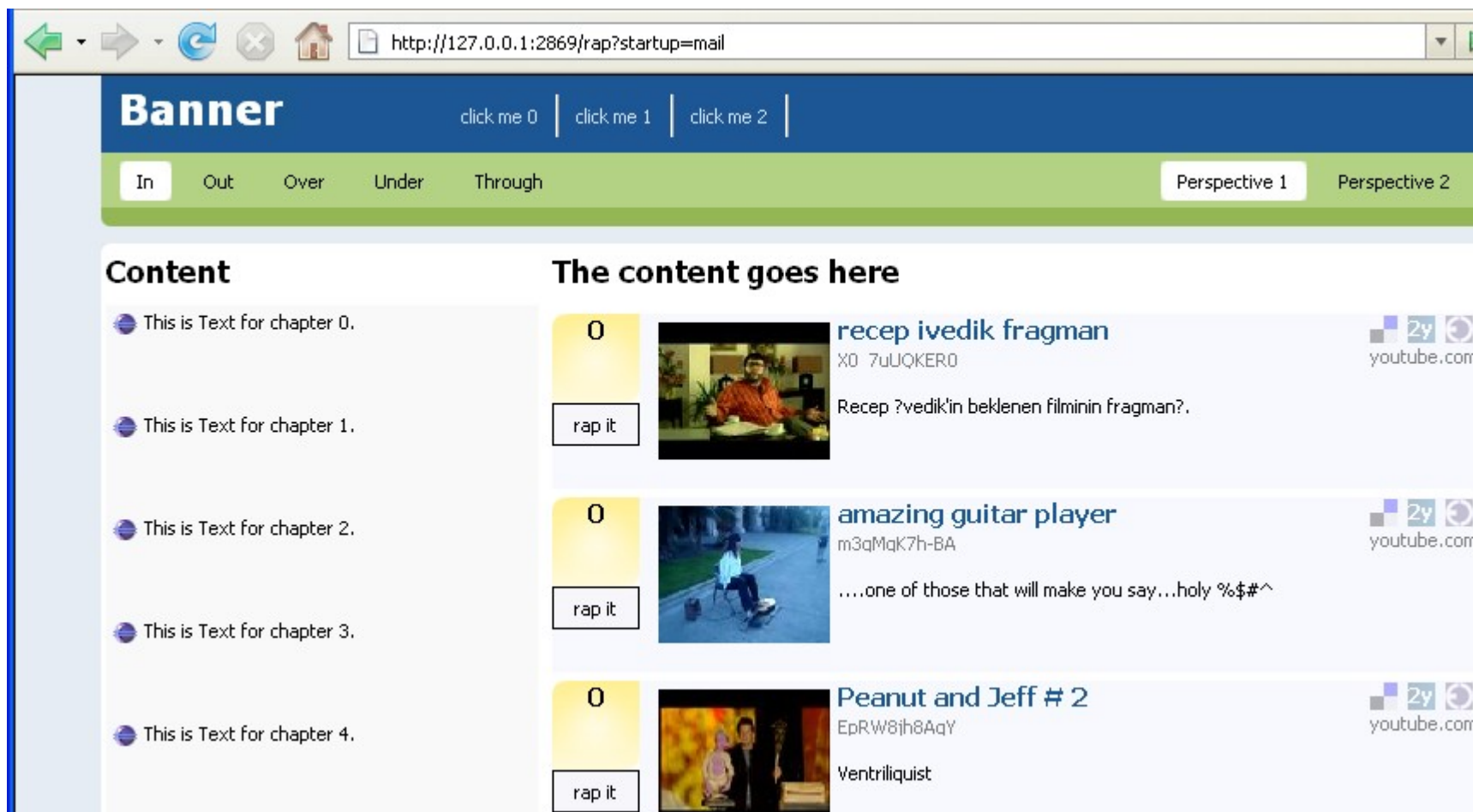
Elias Volanakis
evolanakis@innoopract.com



Innoopract Inc
351 NW 12th Avenue
Portland, OR 97209

Backup Slides

Example - Customized UI

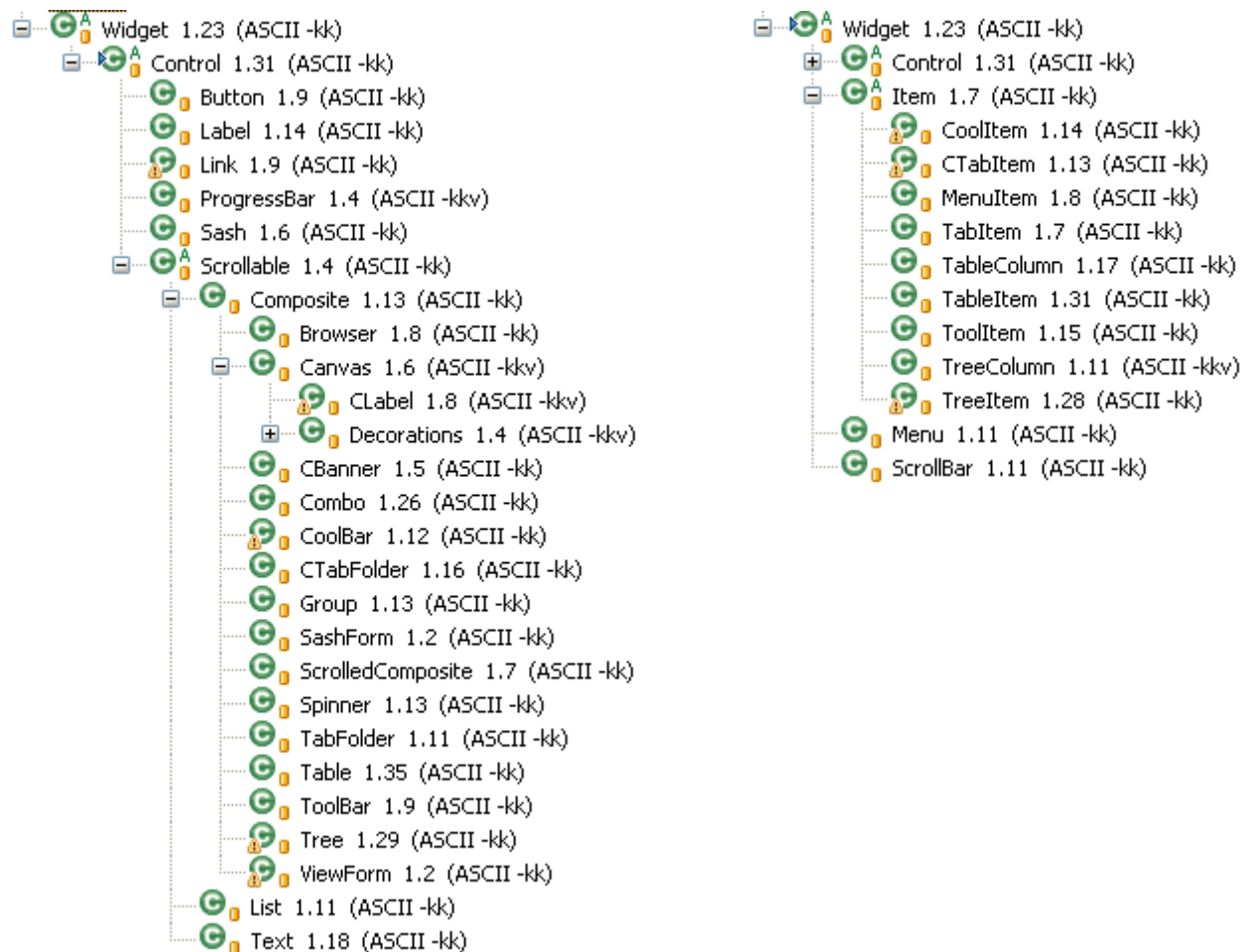


The screenshot shows a browser window with the URL `http://127.0.0.1:2869/rap?startup=mail`. The web application has a blue header with the word "Banner" and three buttons labeled "click me 0", "click me 1", and "click me 2". Below the header is a green bar with navigation tabs: "In", "Out", "Over", "Under", "Through", "Perspective 1", and "Perspective 2".

The main content area is divided into two sections:

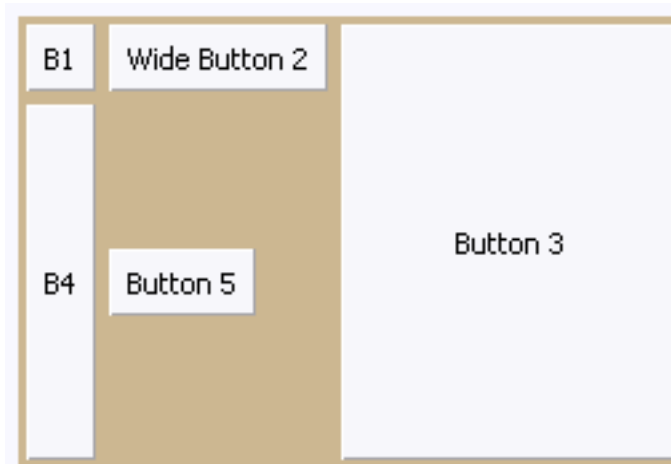
- Content:** A vertical list of five items, each with a blue circular icon and text: "This is Text for chapter 0.", "This is Text for chapter 1.", "This is Text for chapter 2.", "This is Text for chapter 3.", and "This is Text for chapter 4."
- The content goes here:** A vertical list of three video thumbnails, each with a yellow square icon containing the number "0" and a "rap it" button.
 - recep ivedik fragman:** Video ID `X0 7uUQKERO`. Description: "Recep ?vedik'in beklenen filminin fragman?".
 - amazing guitar player:** Video ID `m3qMqK7h-BA`. Description: "...one of those that will make you say...holy %\$#^".
 - Peanut and Jeff # 2:** Video ID `EpRW8jh8AqY`. Description: "Ventriliquist".

Example – RWT Widgets



Example – RWT Layouts

- All usual layouts:
 - GridLayout,
 - RowLayout
 - FillLayout
 - FormLayout
 - StackLayout
 - and a lot more ...



- mostly verbatim copies of SWT (OS independent)
- Layout algorithms work exactly as in SWT
- Layouts are computed on the server, e.g. after a Shell has been resized

Example – RWT Theming

- Objective: allow for a custom look of web applications
- Predefined properties of widgets can be customized
- Dimensions, Colors, Borders, Fonts, Images
- Simple Java .properties file
- Themeable custom widgets

