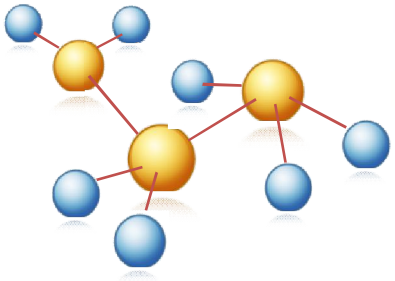
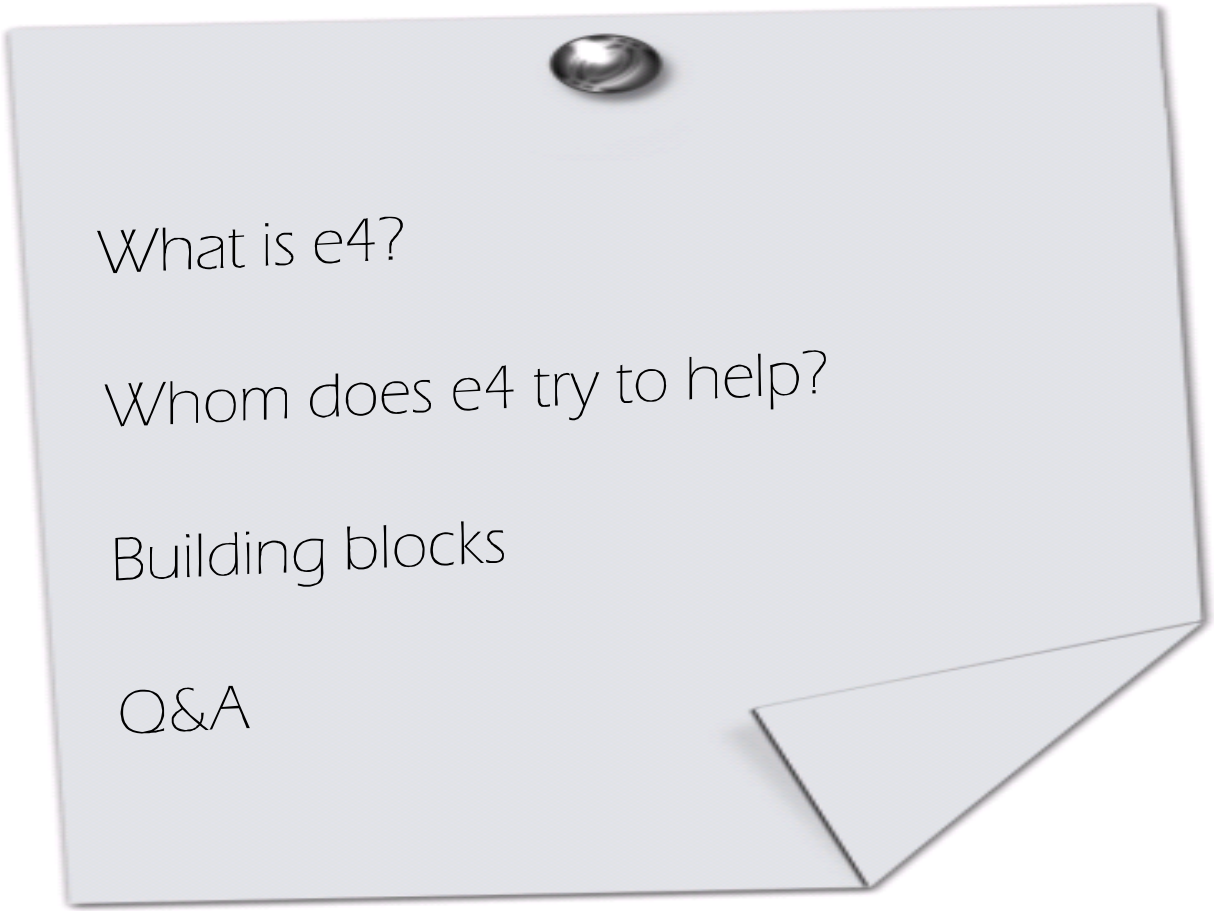
Eclipse 4.0

....also known as
Eclipse e4



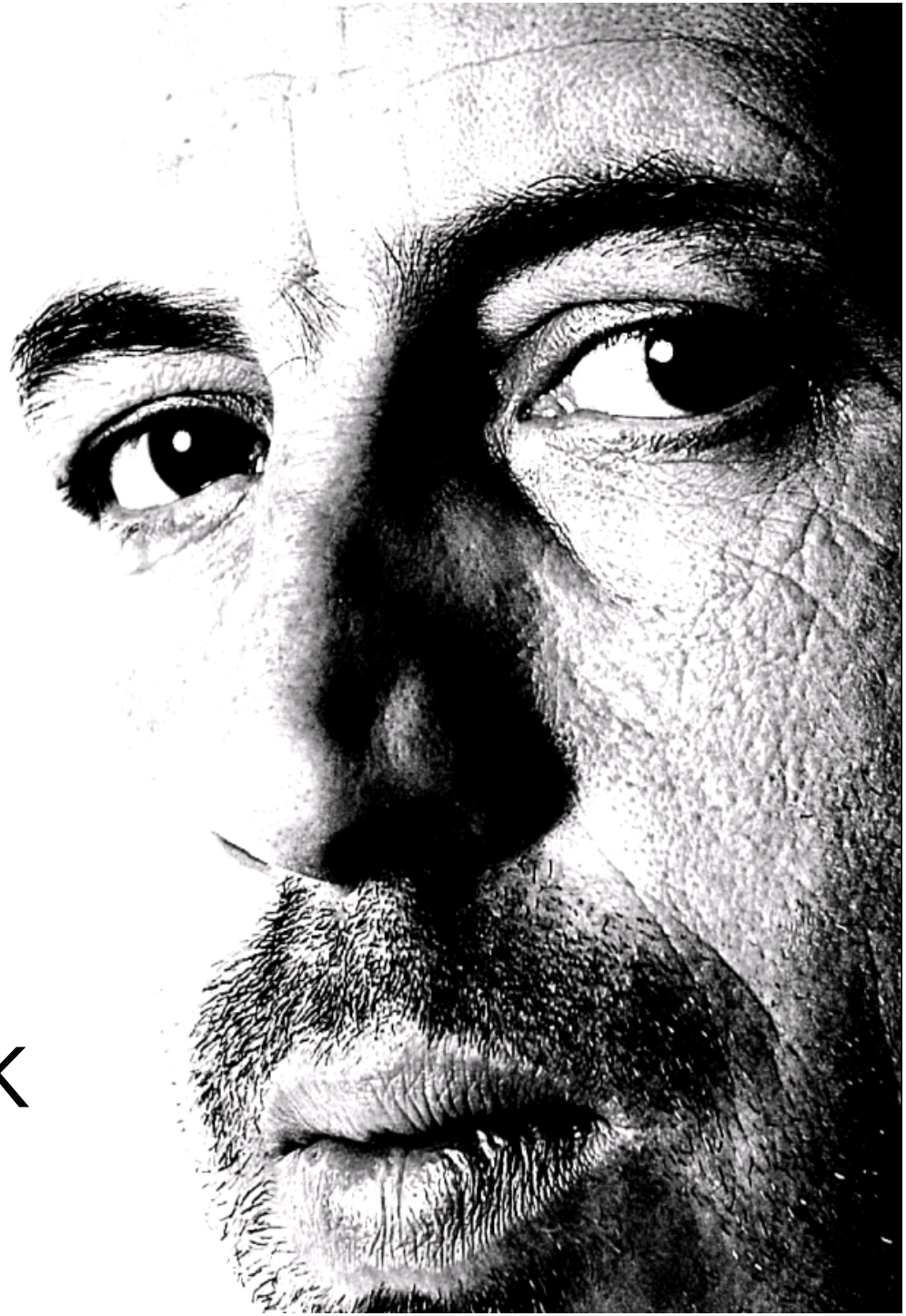
Lars Vogel
http://www.vogella.de
http://www.twitter.com/vogella

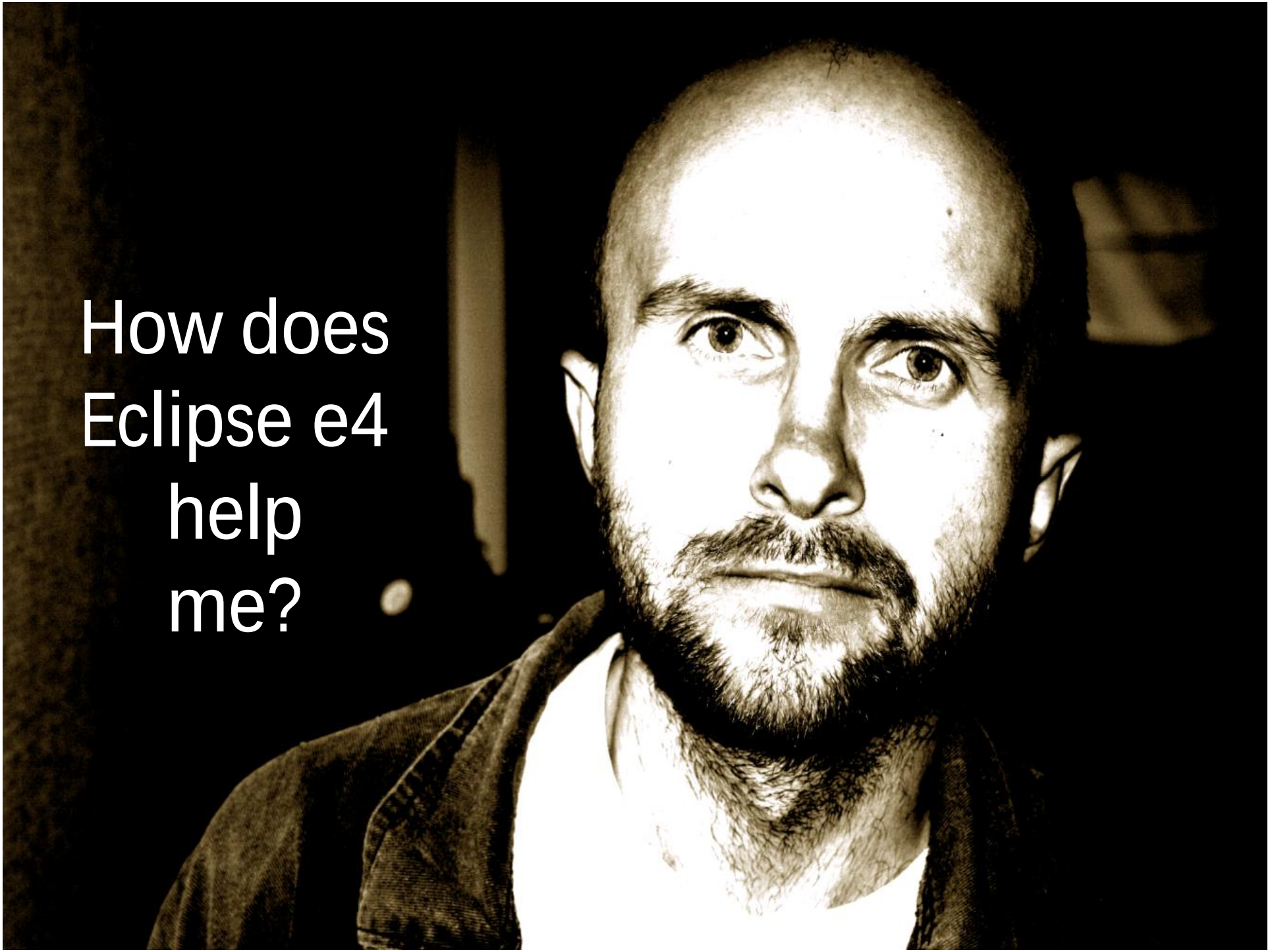
Eclipse e4 -Agenda



Eclipse e4

Eclipse e4 is the incubator project which will produce the Eclipse 4.0 SDK




How does
Eclipse e4
help
me?

Java SE

Java ME

Java EE

Android, Spring, GWT,….

Eclipse e4 Scope

Make development for Eclipse easier

from Eclipse Developer

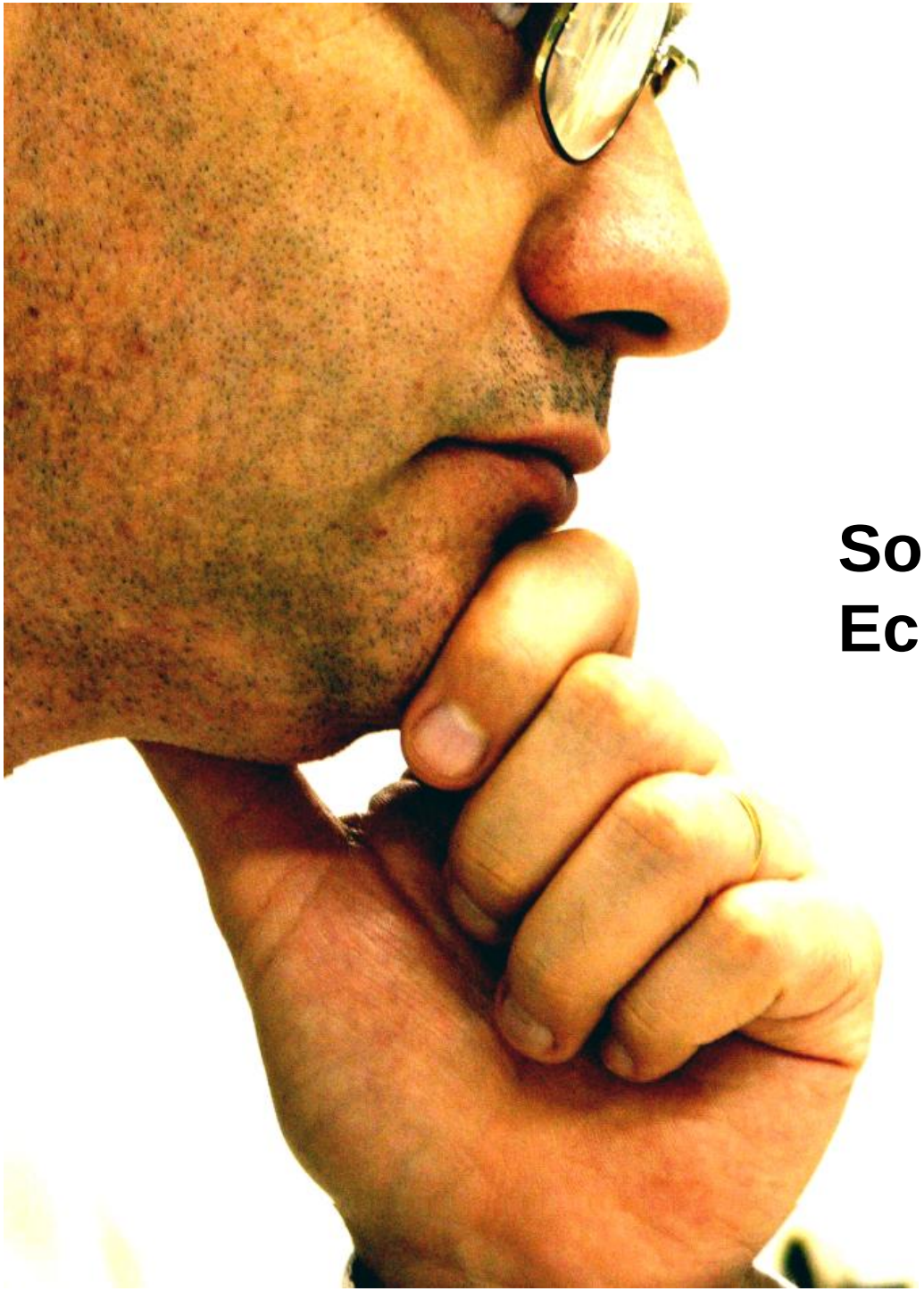
Eclipse E4

to Eclipse Developer

At the beginning Eclipse e4 will not produce valuable functionality for the user of the Eclipse IDE.....

Mid- and long term …much better tooling

By modernizing the Eclipse programming model the user will benefit from the increased development speed

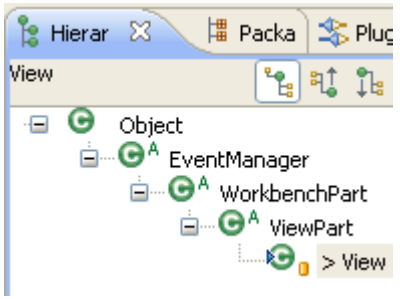


So, what’s wrong with Eclipse 3.X?

Eclipse 3.x programming model

- Complex
- Lots of API
- Platform functionality via singletons
- Not easy to test
- Not a consistent way to define the UI

Inheritance in Eclipse 3.X



OBJECT

EVENTMANAGER

WORKBENCHPART

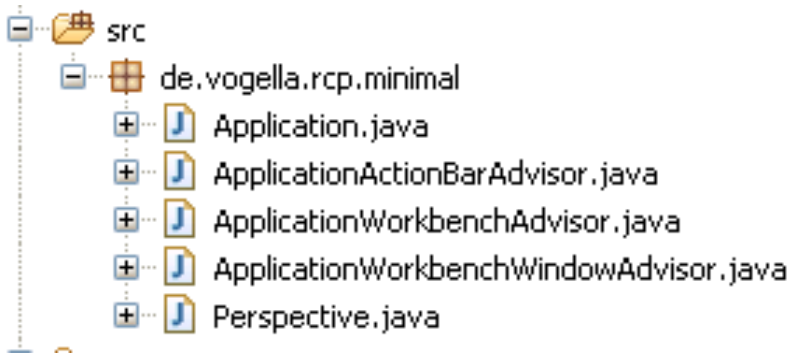
VIEWPART

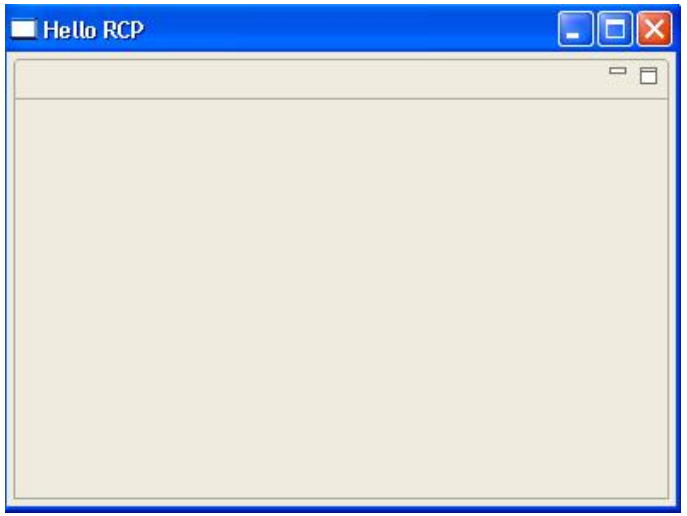
VIEW

The smallest Eclipse RCP app in Eclipse 3.X

Already looks like the Eclipse IDE

Lots of boilerplate code

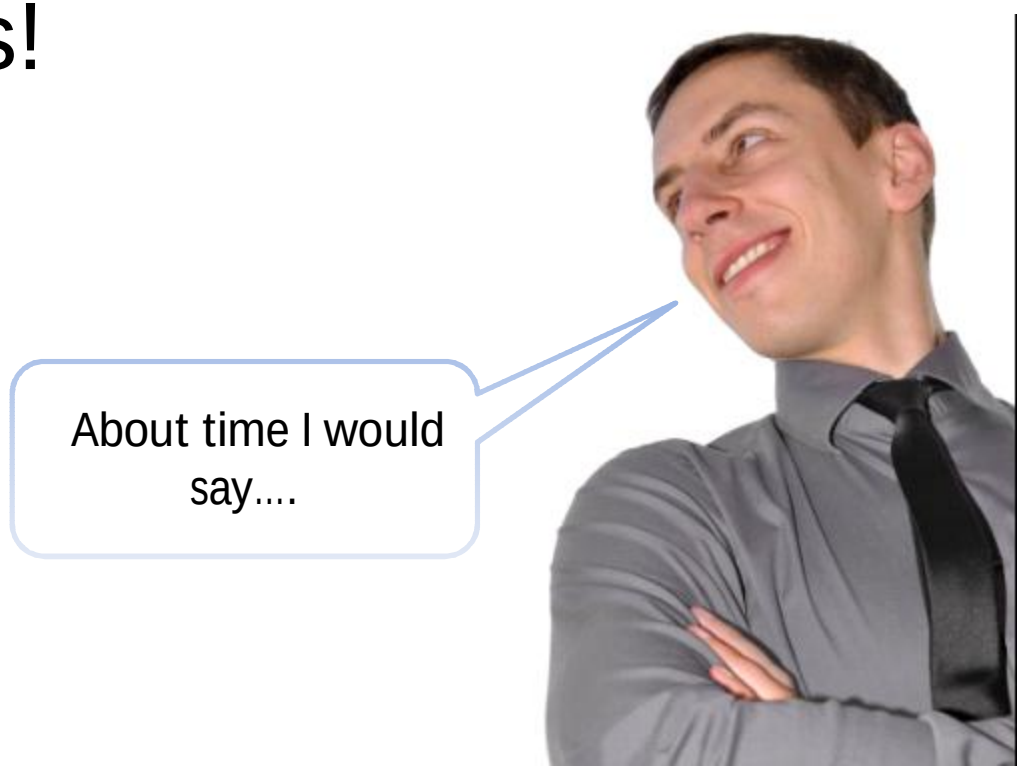




If only Eclipse development would be easier and more visual appealing.

The first amazing change

Eclipse e4 uses Java 1.5 language features!



Eclipse e4 – Building blocks

Modeled Workbench

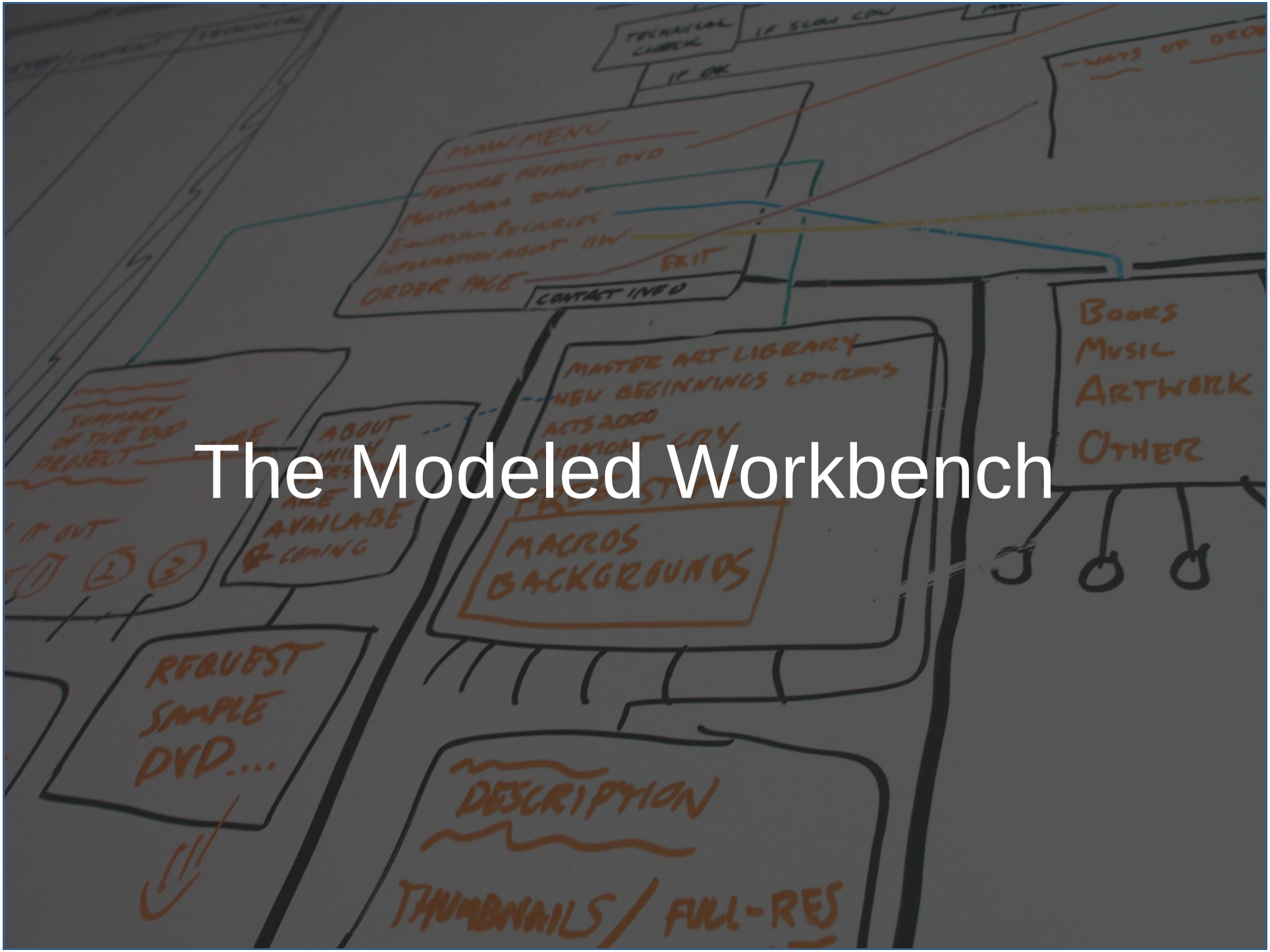
Rendering Engine

Declarative Styling

Dependecy Injection

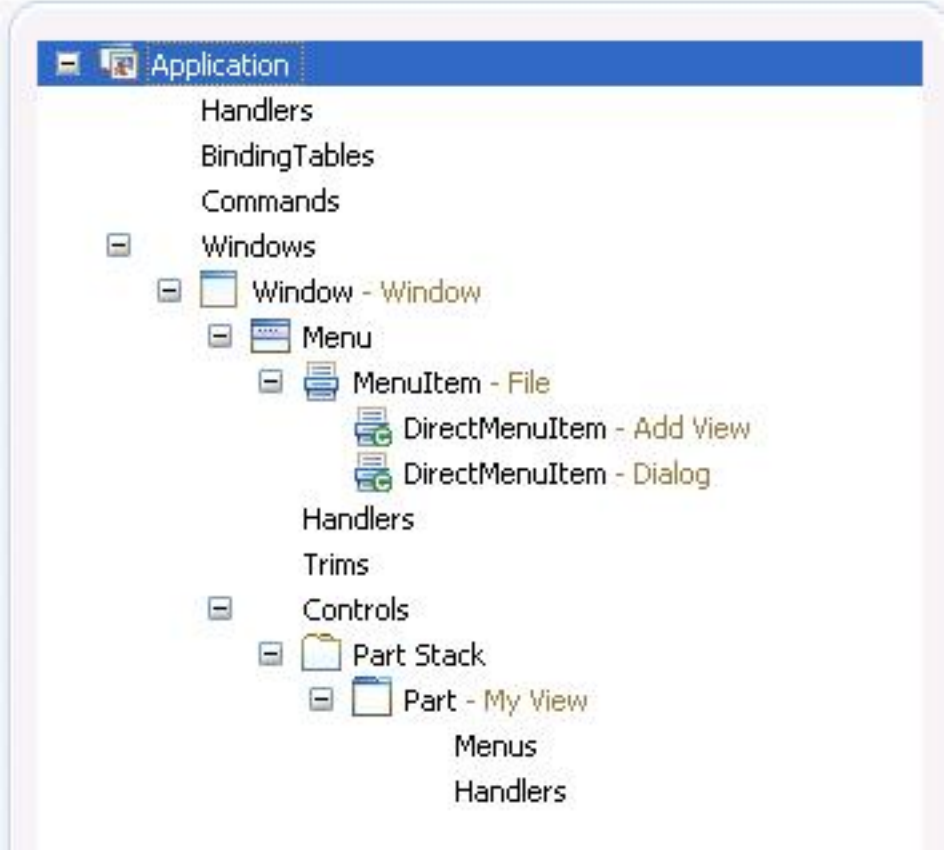
Context

Core Services


The Modeled Workbench
MAIN MENU
EXIT
CONTACT INFO
ORDER PAGE
MASTER ART LIBRARY
NEW BEGINNINGS CD-ROMS
MACROS
BACKGROUNDS
BOOKS
MUSIC
ARTWORK
OTHER
REQUEST
SAMPLE
DVD...
DESCRIPTION
THUMBNAILS / FULL-RES

The e4 Workbench Model

The e4 Workbench Model



- Workbench window
  - Menu with menu items
  - Window Trim, e.g. toolbar with toolbar items
  - Parts Sash Container
    - Parts
  - Part Stack (CTabFolder)
    - Parts
  - Handlers
  - Key Bindings
  - Commands

Model is Flexible

No distinction between View and Editor

Perspectives are optional

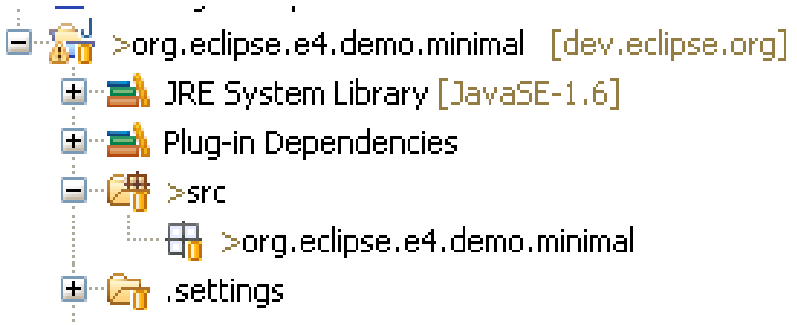
Stack / Sash are optional

Several windows possible

Flexible Toolbars

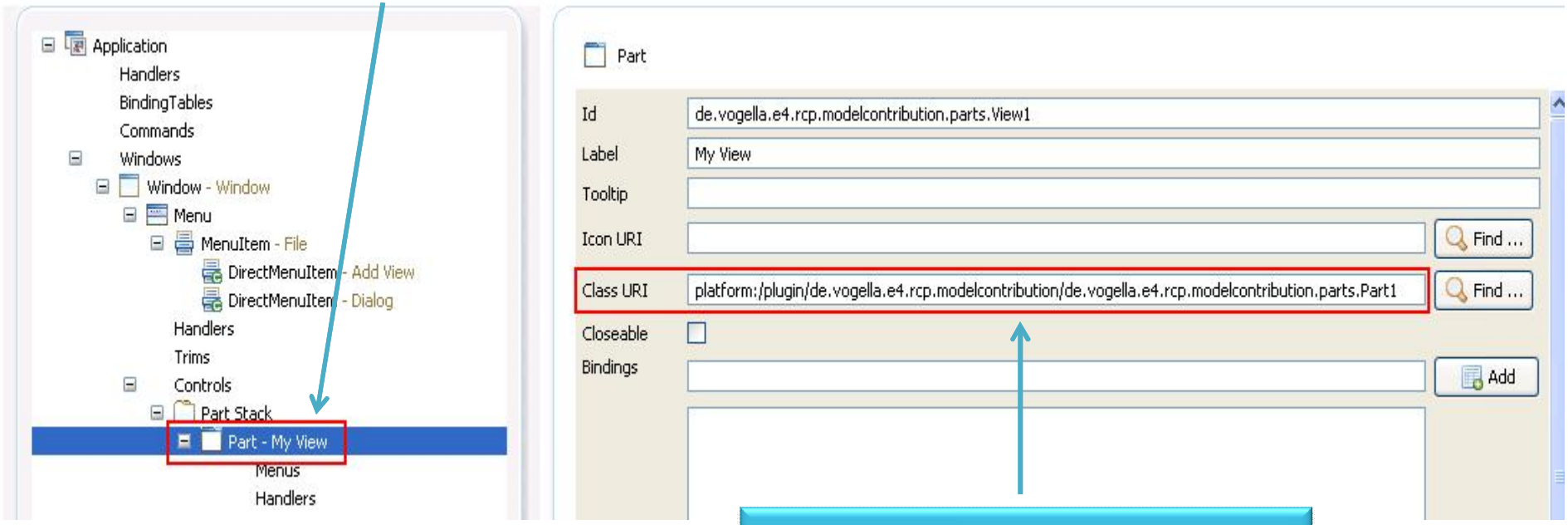
Model removes the need for boilerplate code

Minimal Eclipse e4 app: zero classes



Model URI's

The Part in the Model



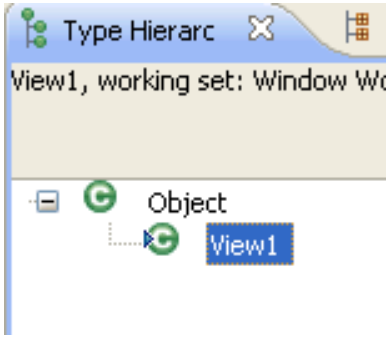
The Part's URI

The POJO* has won!

* Plain Old Java Objects

Model Elements in e4

POJO‘s



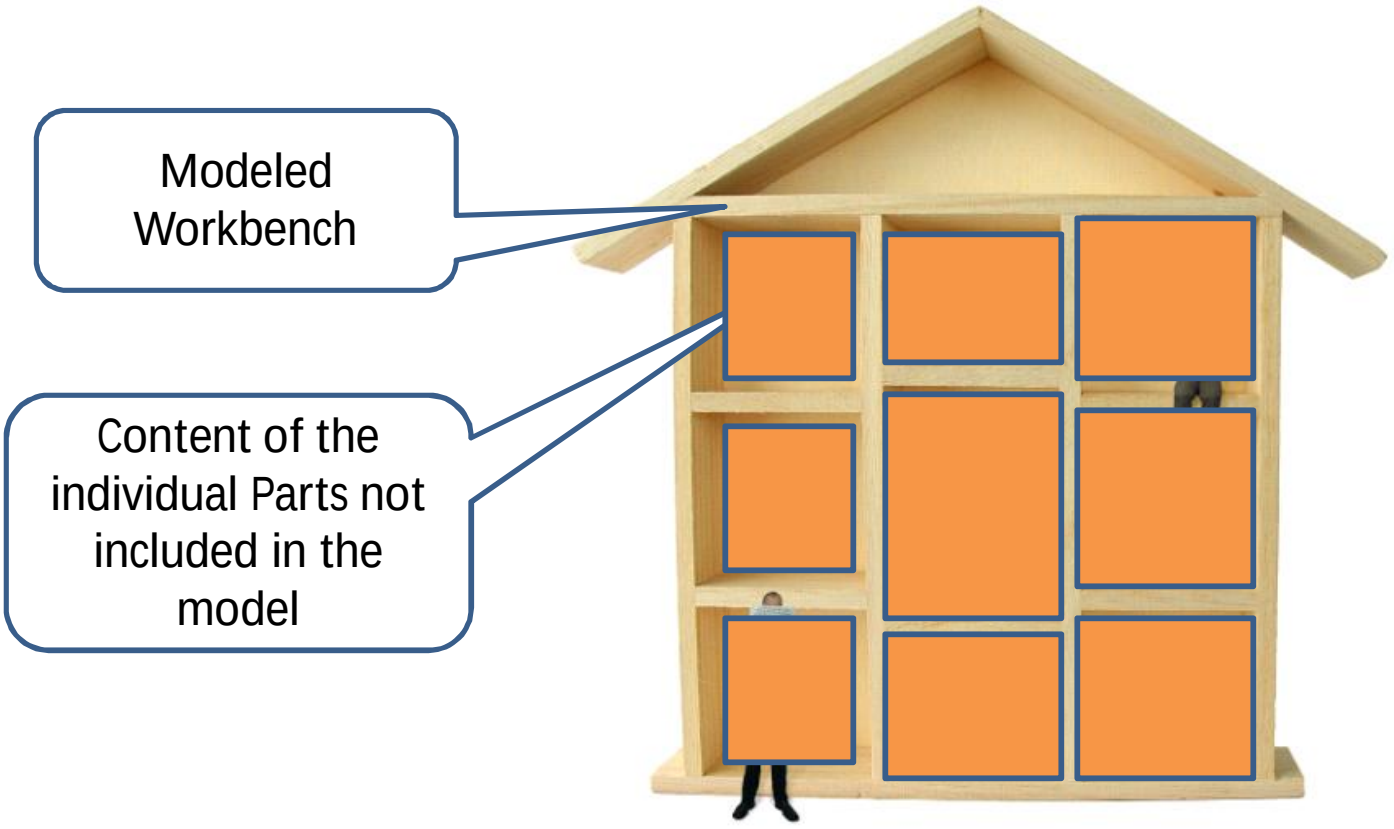
Usage of annotations

Annotations are used
To indicate which
methods are called

Model available at runtime

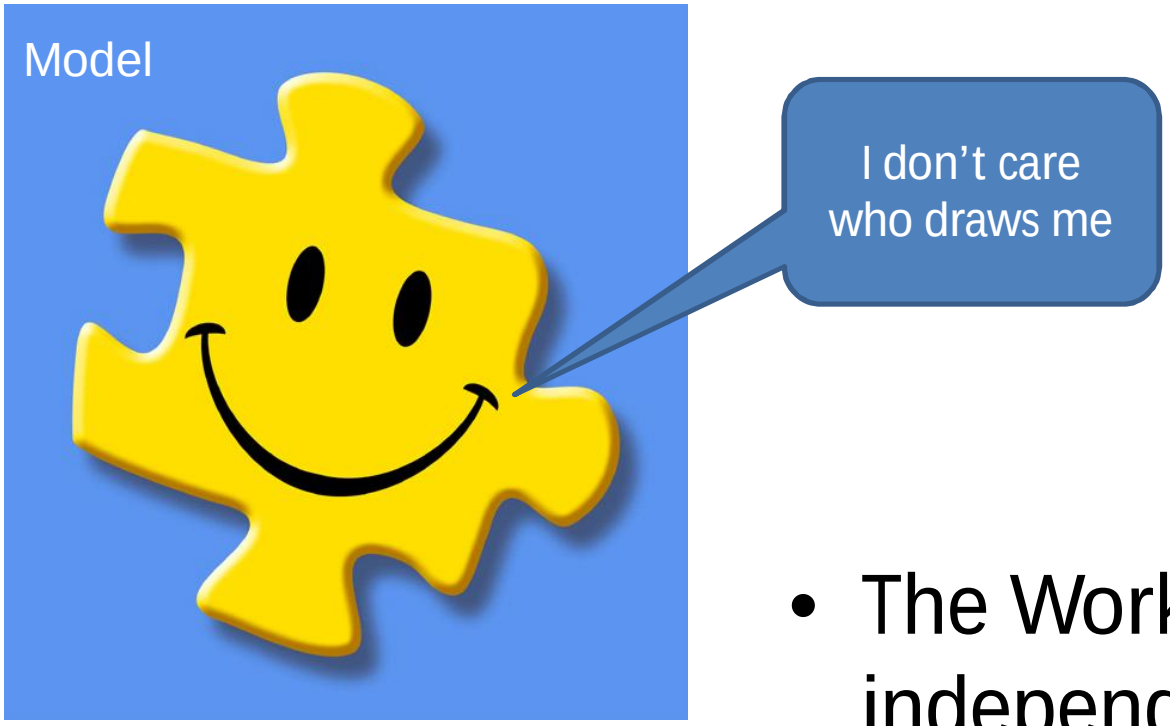
Limits of the e4 application model

- Only models the Application (frame)



How is this model translated into UI components?

Model and UI Renderers



- The Workbench model is independent of a specific UI toolkit

Renderers

Renderer Factory

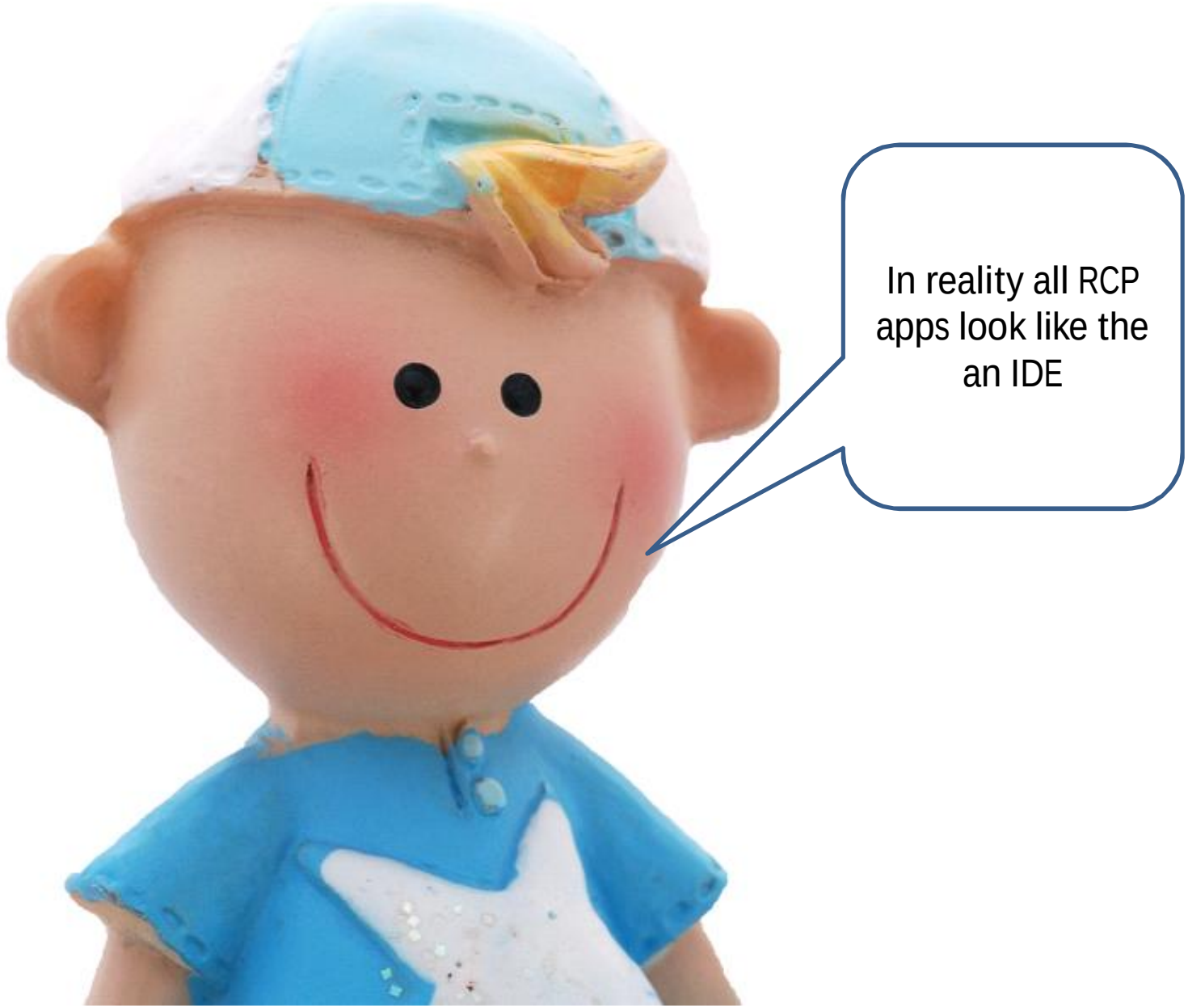
Widget Renderer

Returns for every model element

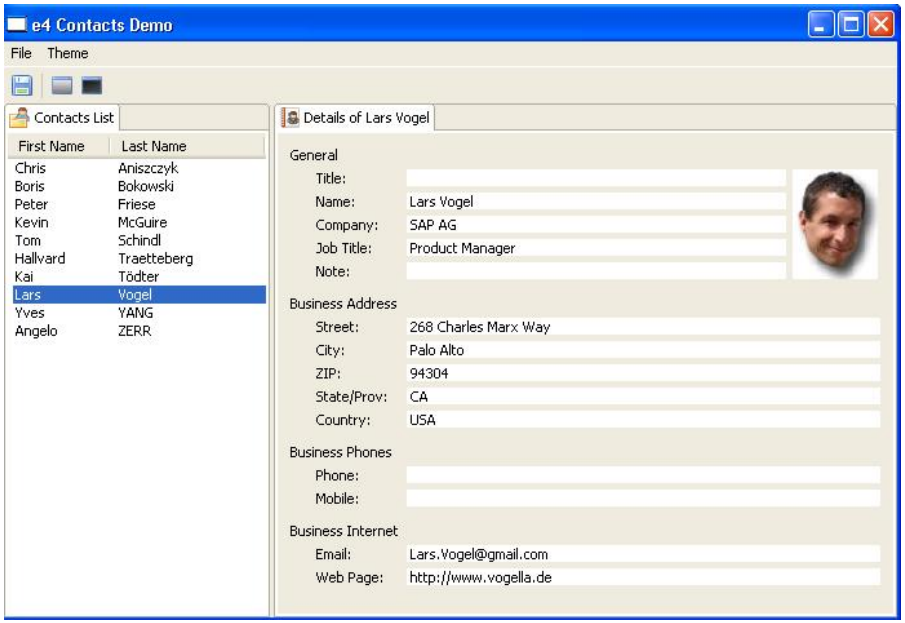
Standard SWT renderer can be exchanged

Renderer: flexible but complex

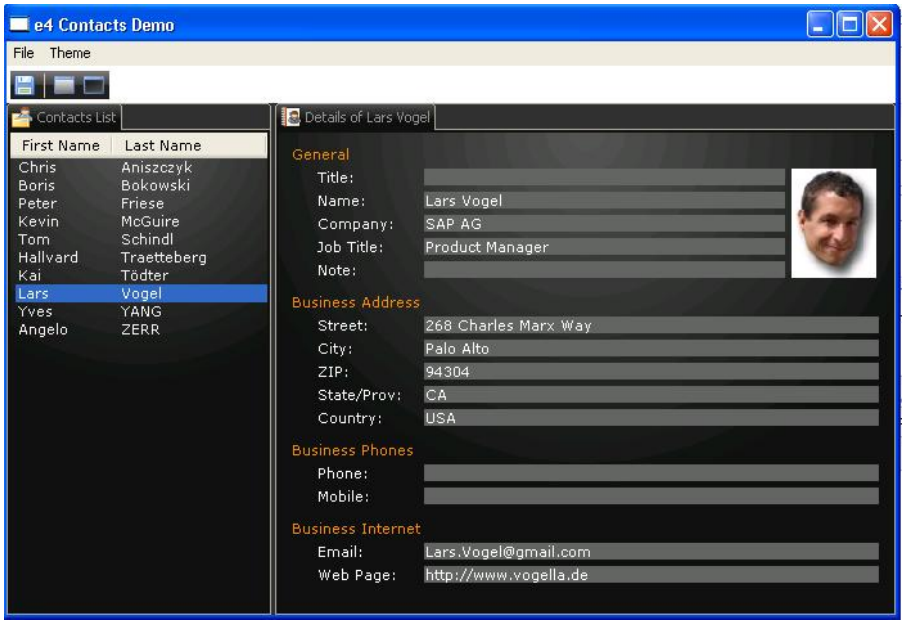
e4 CSS Styling


In reality all RCP apps look like the an IDE

Eclipse 3.X - IDE feeling



Eclipse e4 – CSS Styling



Example from Kai Toedter

Limitations for:

- Menu bar background
- Table headers

e4 supports theme switching during runtime

How to enable CSS Styling

Extension point **"org.eclipse.e4.ui.css.swt.theme“**
defines the style sheet

```
<extension
         point="org.eclipse.e4.ui.css.swt.theme">
      <theme
            basestylesheeturi="css/styles.css"
            id="org.eclipse.e4.minimal.theme.standard"
            label=“Standard">
      </theme>
</extension>
```

How to enable CSS Styling

Property **"cssTheme”** for extension point
"org.eclipse.core.runtime.products" selects the initial theme

```
<extension
        id="product"
        point="org.eclipse.core.runtime.products">
     <product
           application="org.eclipse.e4.ui.workbench.swt.E4Application"
           name="E4 Contacs Demo">
        ....
        <property
              name="cssTheme"
              value="org.eclipse.e4.demo.contacts.themes.darkgradient">
        </property>
         ....
```

Example CSS

```
Label {
    font: Verdana 8px;
    color: rgb(240, 240, 240);
}

Table {
    background-color: gradient radial #575757 #101010 100%;
    color: rgb(240, 240, 240);
    font: Verdana 8px;
}

ToolBar {
    background-color: #777777 #373737 #202020 50% 50%;
    color: white;
    font: Verdana 8px;
}
```

Assign custom attributes

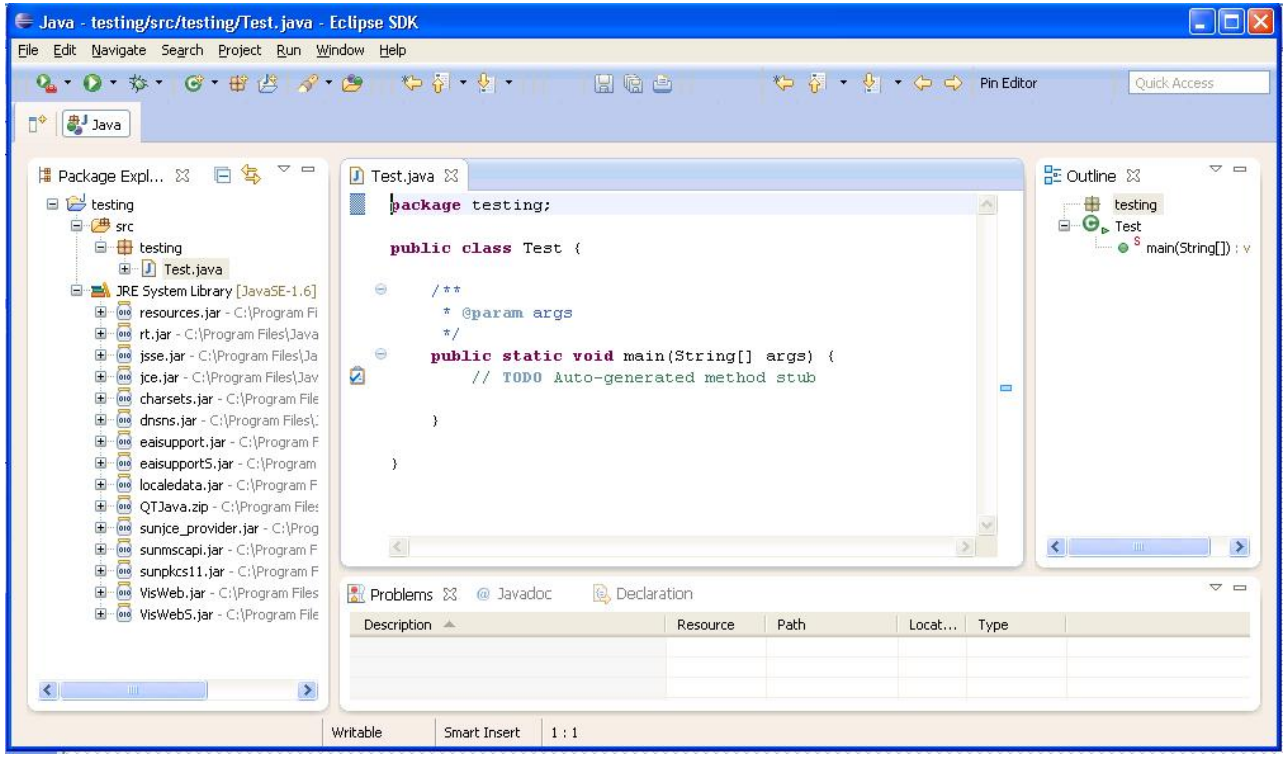
- Java

```
Label label =
    new Label(parent, SWT.NONE);
label.setData("org.eclipse.e4.ui.css.id",
                "SeparatorLabel");
```

- CSS

```
#SeparatorLabel {
    color: #f08d00;
}
```

New Look & Feel for Eclipse 4.0 – SDK



https://bugs.eclipse.org/bugs/show_bug.cgi?id=293481

The e4 Programming Model

Dependency Injection

- Inversion of control: The necessary functionality is injected into the class

Dependency Injection in e4

- JSR 330 compatible injection implementation
  - @javax.inject.Inject – Field, Constructor and Method injection
  - @javax.inject.Named – Specify a custom qualifier to context object (default is fully qualified classname of the injected type)
- e4 specific annotations, e.g. @Optional

Services are injected via the the e4 framework

```
public class ListView {

   @Inject
   private IEclipseContext context;
   @Inject
    private Logger logger;

   @Inject
   public ListView(Composite parent) {
         // ...
```

Lifecycle

- @PostConstruct: Called after Object created and Fields- and Methods-Injection finished
- @PreDestroy: Called before the object is destroyed (e.g. the Part shown in the UI is removed)


What can be injected?
18 AL 322
343 AL 37
613
8.82.071
830.840-3

IEclipseContext

- Stores information of possible Injection Values
- OSGi Services part of the Context
- Define your own services and use DI for them

Example: Writing Handlers

- Method identified via @Execute annotation
  - Can have any number of arguments
  - Use IServiceConstants for general context informations

```
public class AboutHandler {
@Execute
  public void execute(@Named(IServiceConstants.ACTIVE_SHELL) Shell shell){
    MessageDialog.openInformation(
      shell, "About", "e4 Application example.");
  }
}
```


Services

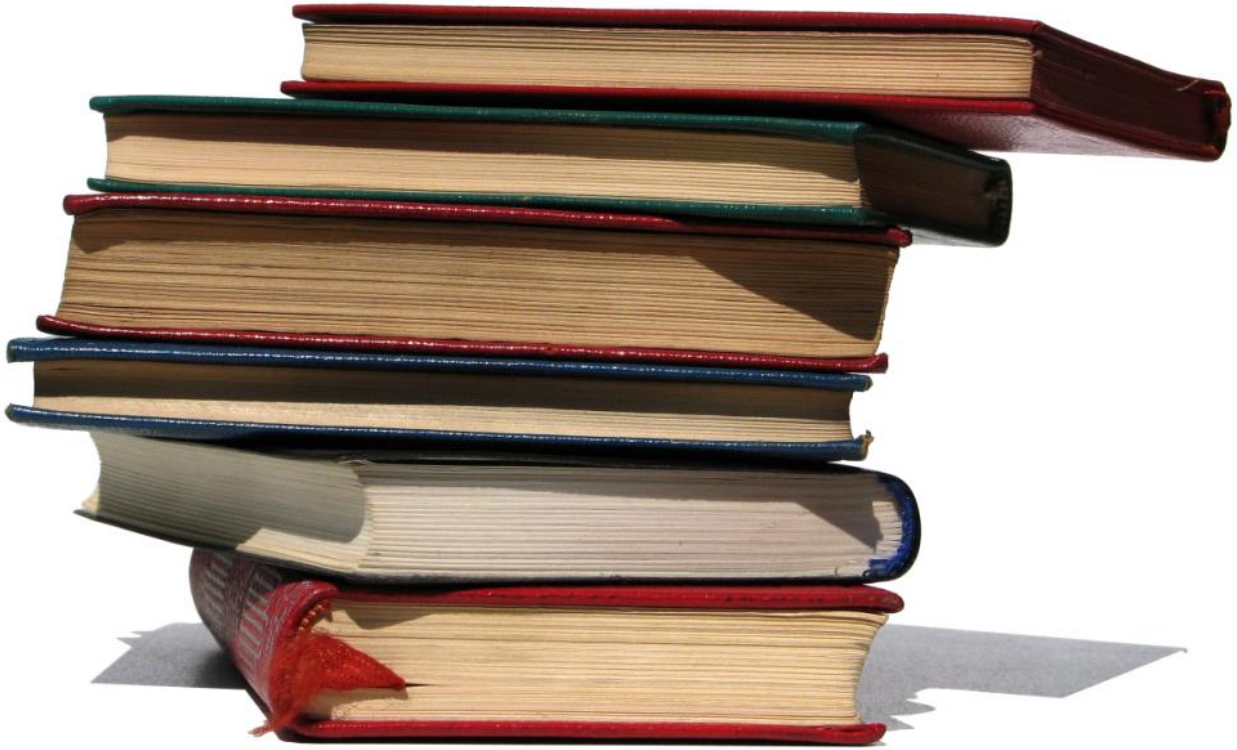
Eclipse Application Services (“Twenty Things”)

- Preferences
- Editor lifecycle
- Receiving input
- Producing selection
- Standard dialogs
- Persisting UI state
- Logging
- Interface to help system
- Menu contributions
- Authentication
- Authorization

- Long-running operations
- Progress reporting
- Error handling
- Navigation model
- Resource management
- Status line
- Drag and drop
- Undo/Redo
- Accessing preferences

- Don‘t forget: OSGi services are also available via dependency injection

Contributions



Contributions

Extension point “org.eclipse.e4.workbench.model”

Fragments – XMI snippets

Processor - Code

Context Communication

Change the context

```
context.modify("selection", selection.getFirstElement());
```

Get Notified

```
@Inject
public void setPerson(@Named("selection") @Optional
Person person) {
master.setValue(person);
}
```

Eclipse e4 – There is more

XWT

Open Social Gadgets

Toolkit Model

JDT ???

Photo credits

- Open Door http://www.sxc.hu/photo/1228296
- Guy http://www.sxc.hu/photo/423354
- Corn http://www.sxc.hu/photo/939151
- About time guy http://www.sxc.hu/photo/1173019
- Thinking man http://www.sxc.hu/photo/ 324541
- Cool but hurts http://www.sxc.hu/photo/ 906072
- WWW http://www.sxc.hu/photo/987822
- Present http://www.sxc.hu/photo/1140205
- Happy figure http://www.sxc.hu/photo/125901
- Chess http://www.sxc.hu/photo/ 958410
- Pill box http://www.sxc.hu/photo/ 510413
- Syringe / Injection: http://www.sxc.hu/photo/ 468493
- Smiley http://www.sxc.hu/photo/ 1211480
- Lock http://www.sxc.hu/photo/ 352344
- Life Cycle http://www.sxc.hu/photo/1265027
- Powerful but complex http://www.sxc.hu/photo/607988
- Baby photo not yet published
- Excellent http://www.sxc.hu/photo/866529
- Thank you picture http://www.sxc.hu/photo/ 986313
- Runtime model http://www.sxc.hu/photo/ 765733
- Renderer http://www.sxc.hu/photo/976984
- Binder: http://www.sxc.hu/photo/443042
- Praying Girl http://www.sxc.hu/photo/646227
- Box: http://www.sxc.hu/photo/502457
- Screws http://www.sxc.hu/photo/1148064
- House with compartments http://www.sxc.hu/photo/494103
- Stacked stones http://www.sxc.hu/photo/998524
- Thinking Guy http://www.sxc.hu/photo/130484
- Drawing Hand http://www.sxc.hu/photo/264208
- Waiter http://www.sxc.hu/photo/157966
- Dancing Girt http://www.sxc.hu/photo/1187376
- Books http://www.sxc.hu/photo/1184809

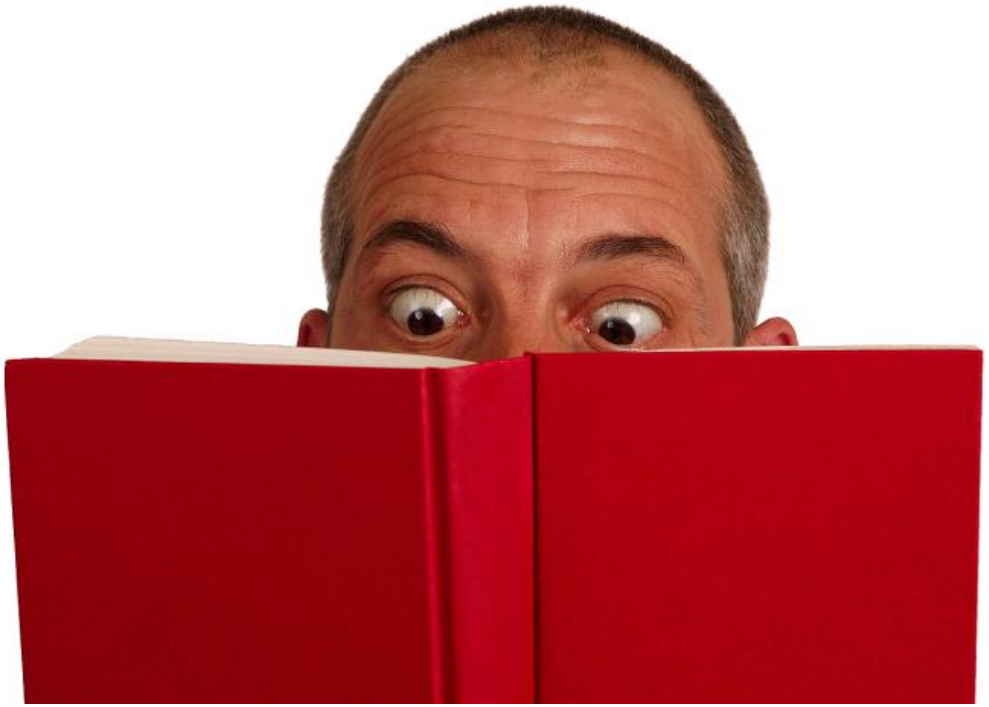
e4: Where to go from here:

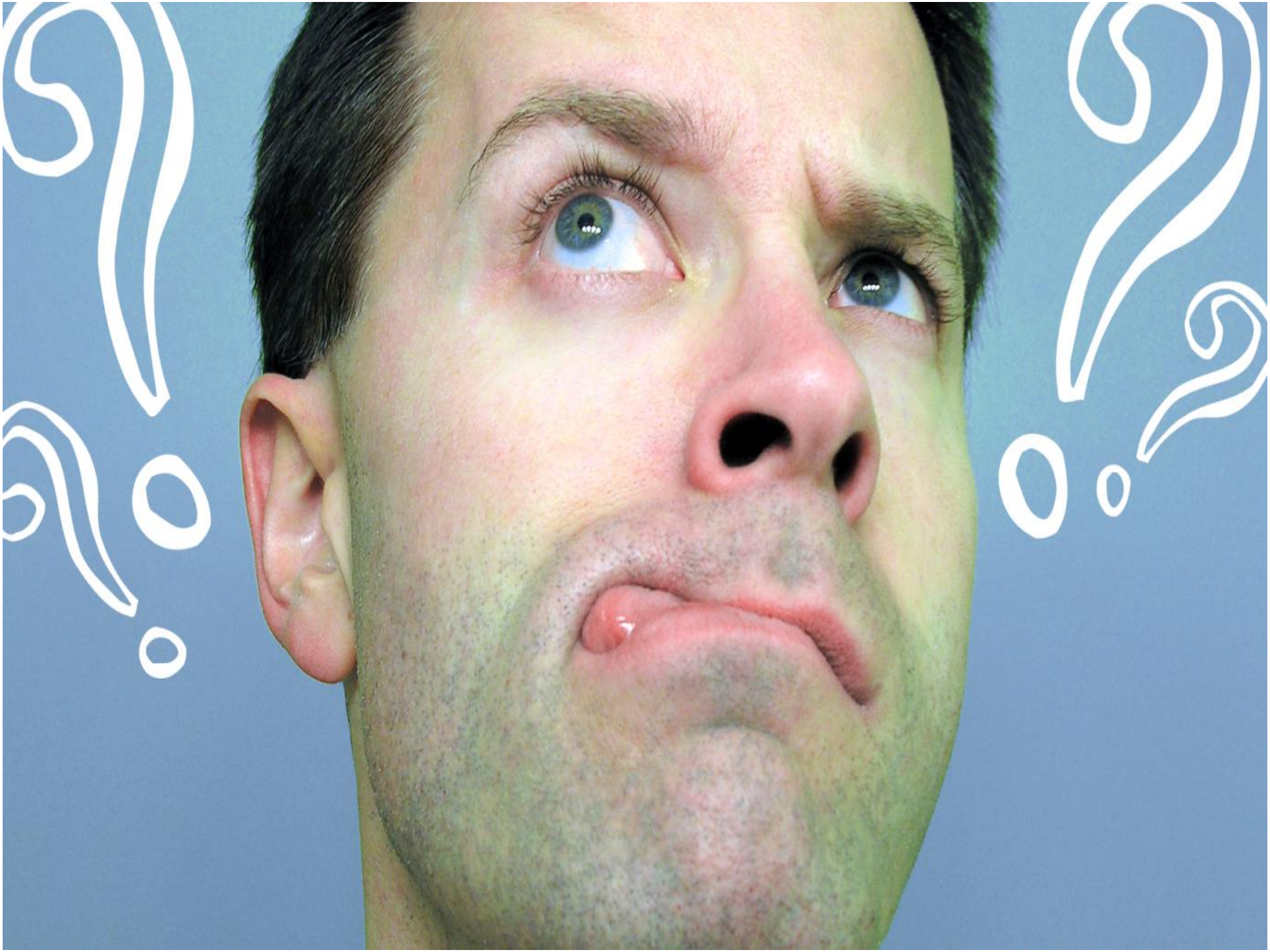
Eclipse e4 Wiki

http://wiki.eclipse.org/E4

Eclipse e4 Tutorial

http://www.vogella.de/articles/EclipseE4/article.html





Thank you

For further questions:

Lars.Vogel@gmail.com
http://www.vogella.de
http://www.twitter.com/vogella

License & Acknowledgements

- This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Germany License

  - See http://creativecommons.org/licenses/by-nc-nd/3.0/de/deed.en_US