# E4 - Modeled Workbench

Tom Schindl - BestSolution.at / Platform UI

# E4 - About me

- CSO BestSolution.at
- Eclipse Ecosystem

**Committership on**
- E4
- Platform UI
- EMF

**ProjectLead on**
- Nebula
- UFaceKit

# E4 – Some (historic) facts

- 2008
  - Short before EclipseCon 2008: Annoucement of a new project idea
  - First mock up presentations at Eclipse 2008
  - E4-Summit in Ottawa (22nd /23rd May)
    - Project groups formed
    - The modeled workbench was born

# E4 – May 20th 2008

- It started with a mail

  „A radical approach to explore new paths for e4“

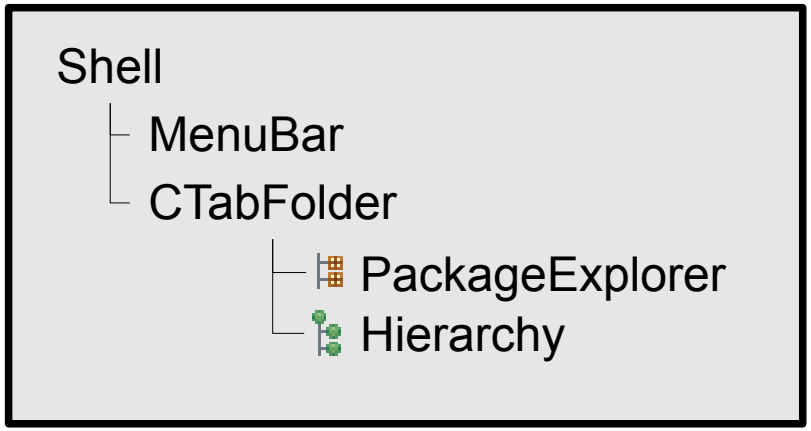
  - Published a Platform designed from Scratch
  - Key points
    - Completely backed up by an live Ecore-Model
    - No statics and singleton
    - Usage of ServiceLocator and Dependency-Injection

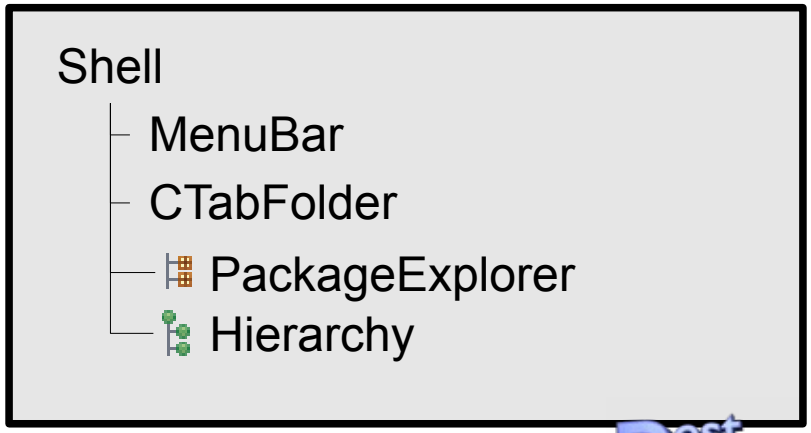
# E4 - Some (historic) facts

- 2009
  - Release of E4 0.9 in Summer (tech-preview)
  - Evaluation of overlapping technologies and draw a decision

# E4 - Why? Why now?

- The current code is hard and is getting harder to maintain
  - Different MVC implementations
  - Legacy code because of historic platform limitations



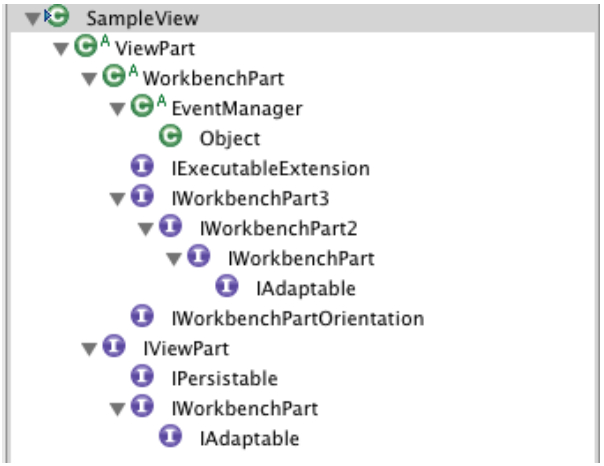



# E4 - Why? Why now?

- The platform code uses outed patterns
  - Usage of Statics and Singletons

```
PlatformUI.getWorkbench()
```

  - Too much usage of inheritance






BestSolution.@t

# E4 – Why? Why now?

- The core platform suffers from a vast number of API and API that overlaps each other
  - Preference APIs
  - Commands vs. Actions
  - Different notification systems

# E4 - Why? Why now?

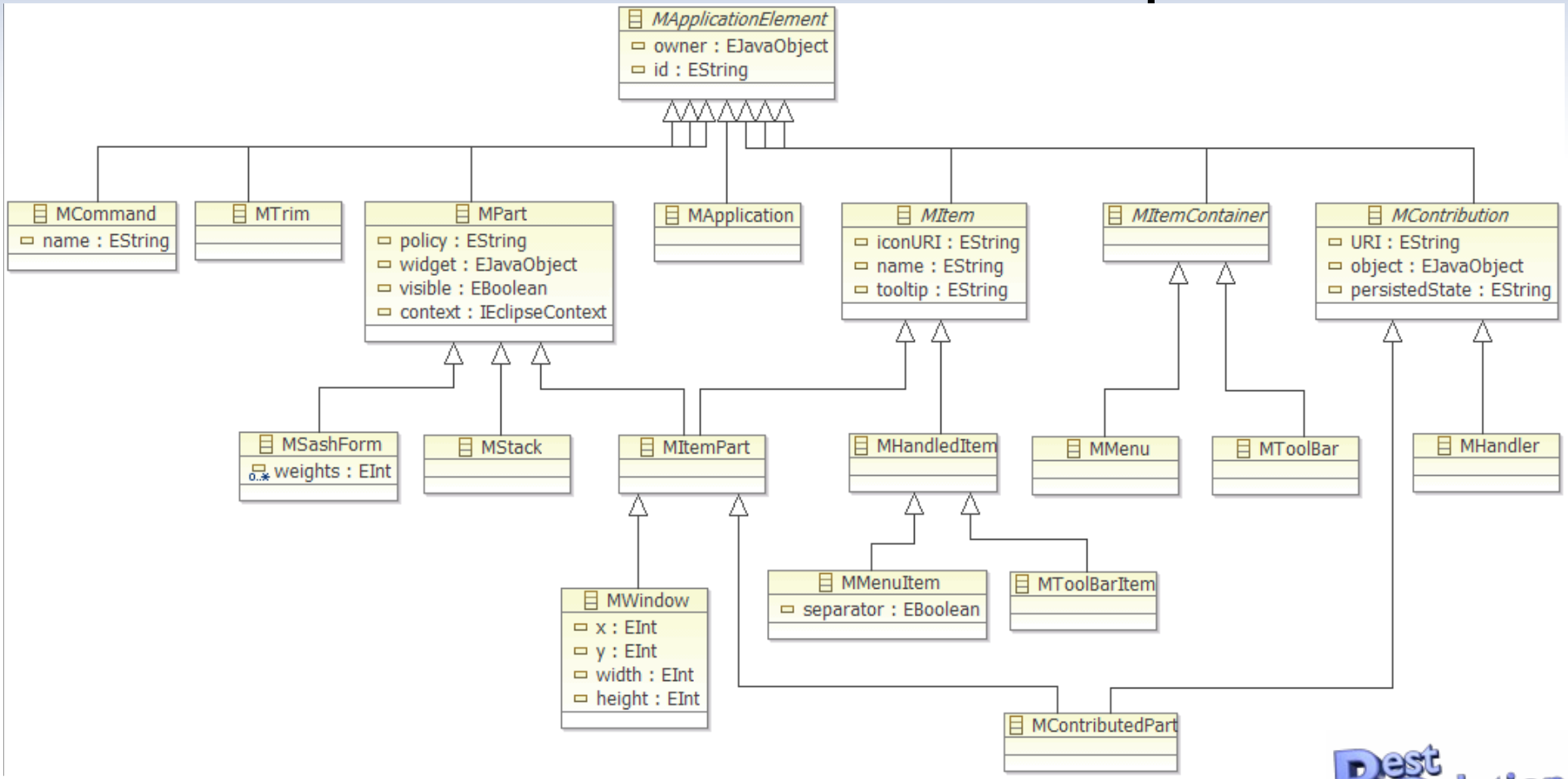
- New competitors in RCP
  - RIA-Frameworks like Flex, Silverlight, JavaFX
  - GWT, Ajax-Frameworks (Qooxdoo, Ext-Js, ...)
- New UI-Philosophies
  - Shift away from native looking UIs

# E4 - Why? Why now?

- Programming model too complex for many RCP-Applications
  - Perspectives are often useless in RCP-Applications so why do I need one?
  - We do I have to contribute my views through those extension points?
  - Why can I not define the menu-item order?
  - ...

# E4 - The design ideas

- One coherant model below the platform



# E4 - The design ideas

- Why is the model an EMF-Model?
  - It's here, it's mature and has a vibrant community and it's an Eclipse-Project
  - It has tooling which will make it easy to create models
  - It has fantastic code generation facilities
  - Foundation for interesting stuff like CDO

# E4 - The design ideas

- Why is the model an EMF-Model
  - Isn't EMF bloat?

    Installation vs Runtime bloat
    - EMF adds **(in its current packageing)** 1.5 MB to the download size
    - Runtime size of EMF-Models highly optimized and (storage of booleans using bitmasks)
    - Benefits of upstream changes (EMF Ultra Slim Diet save 108 bytes / instance) in 3.5

# E4 - The design ideas

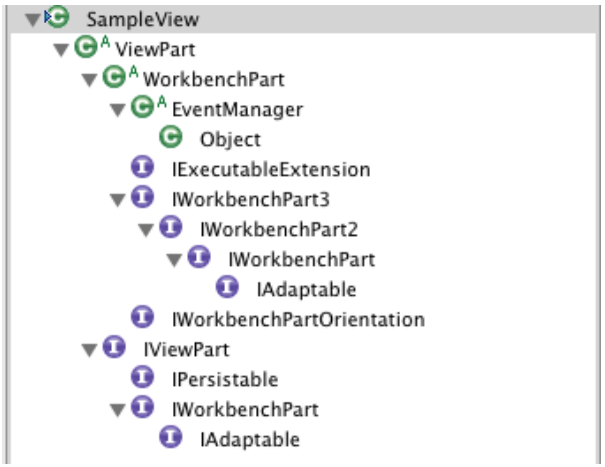
- The idea of an IEclipseContext
  - Sits between application and framework
  - Brokers interaction between like service registration and lookup
  - Hierarchical

# E4 - The design ideas

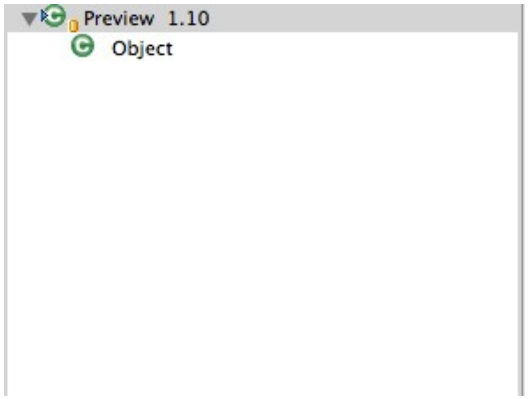
- Useage of Dependency Injection

  Flatten the level of inheritance

Eclipse 3.x




E4



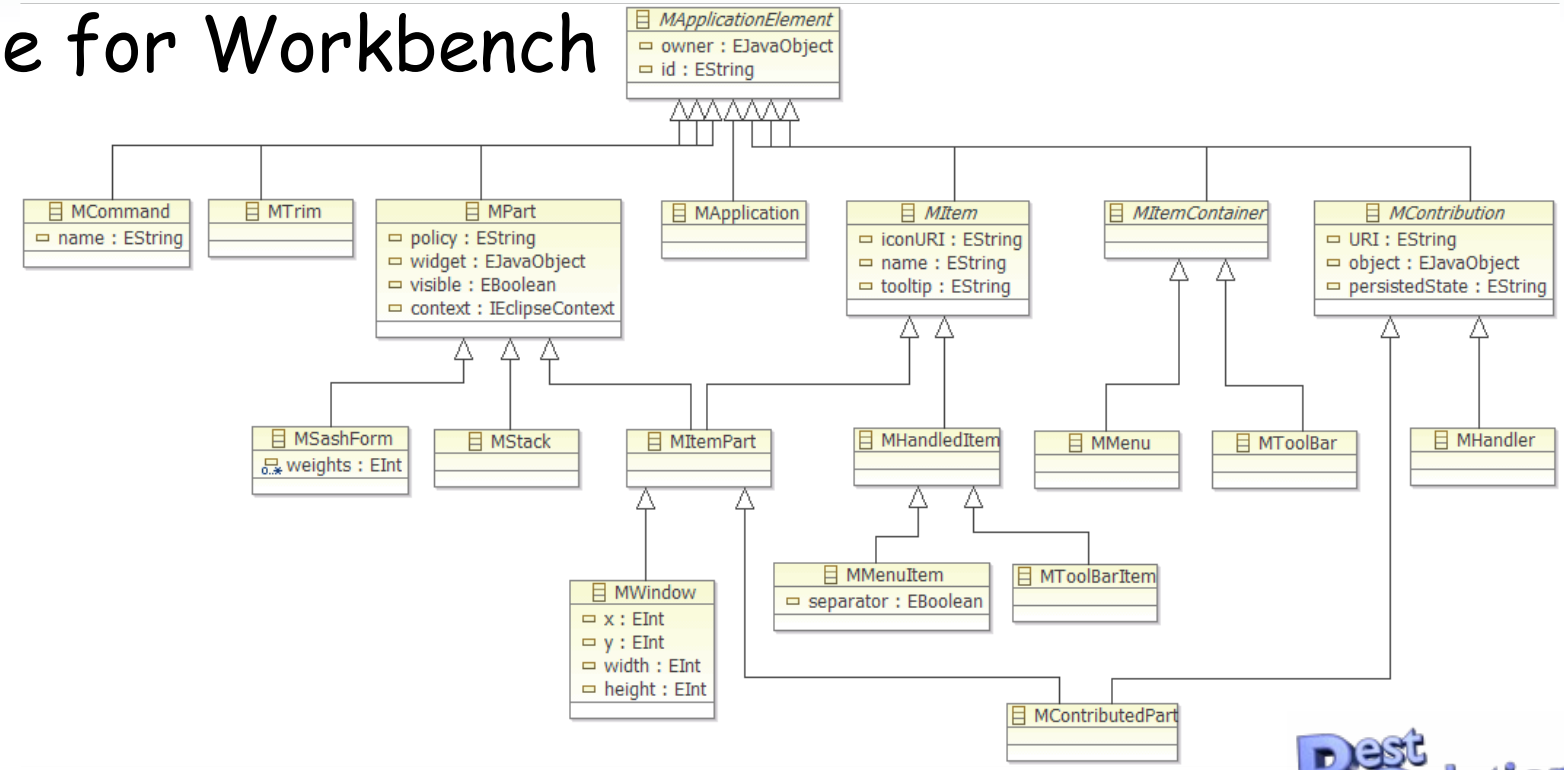
# E4 - The design ideas

- A clean split between the platform code and the presentation/rendering
  - Unlimited possibilities to customize the UI
  - Programming happens against the model not the rendered UI – everything in the UI is accessible
- Support other programming languages to write your UI-Plugins, Handlers, …
- Declarative Styling

# E4 - The implementation

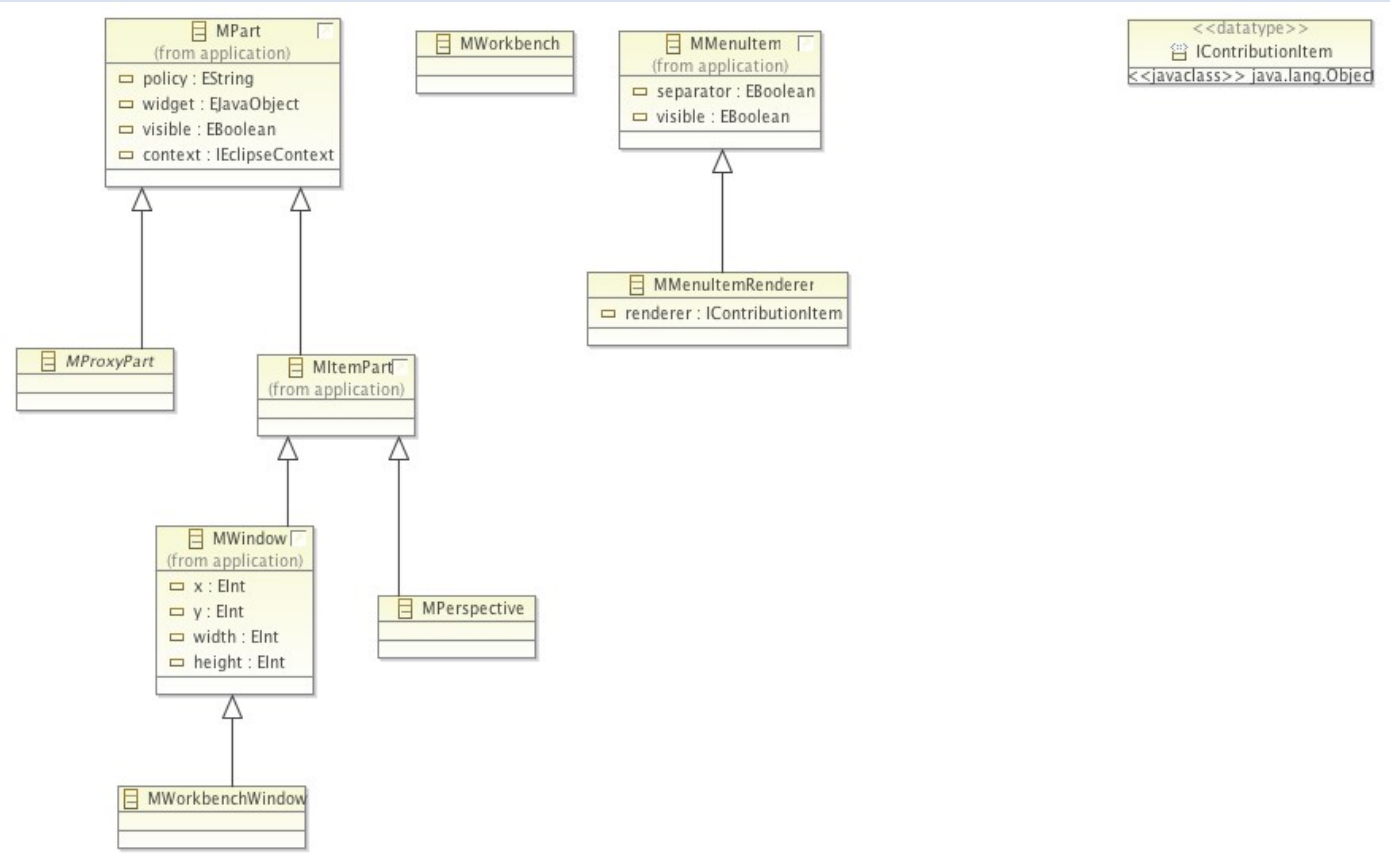
- The EMF-Model

  Split in two: 1 Core for RCP and 1 Extended one for Workbench



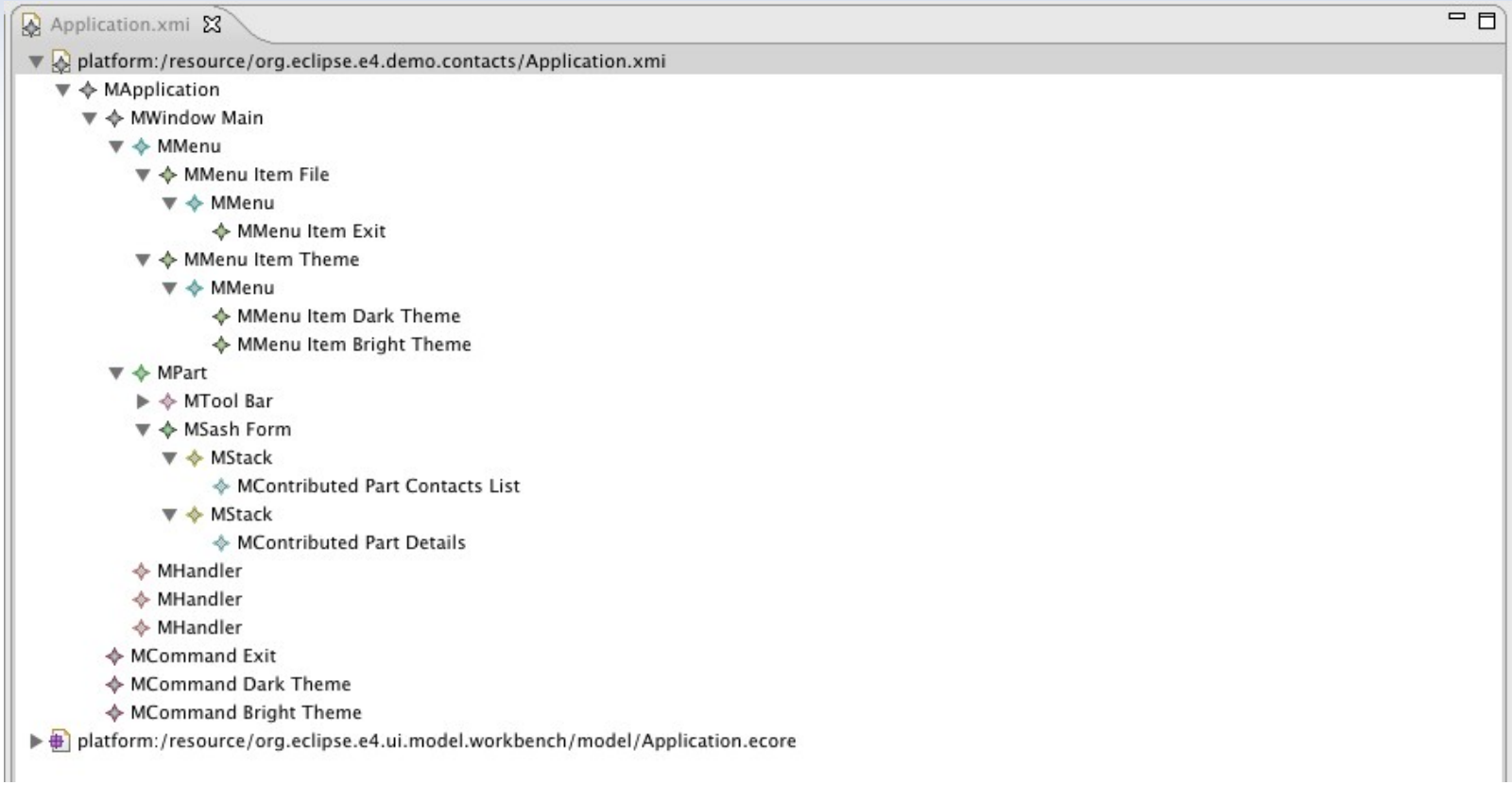
# E4 - The implementation

- The EMF-Model



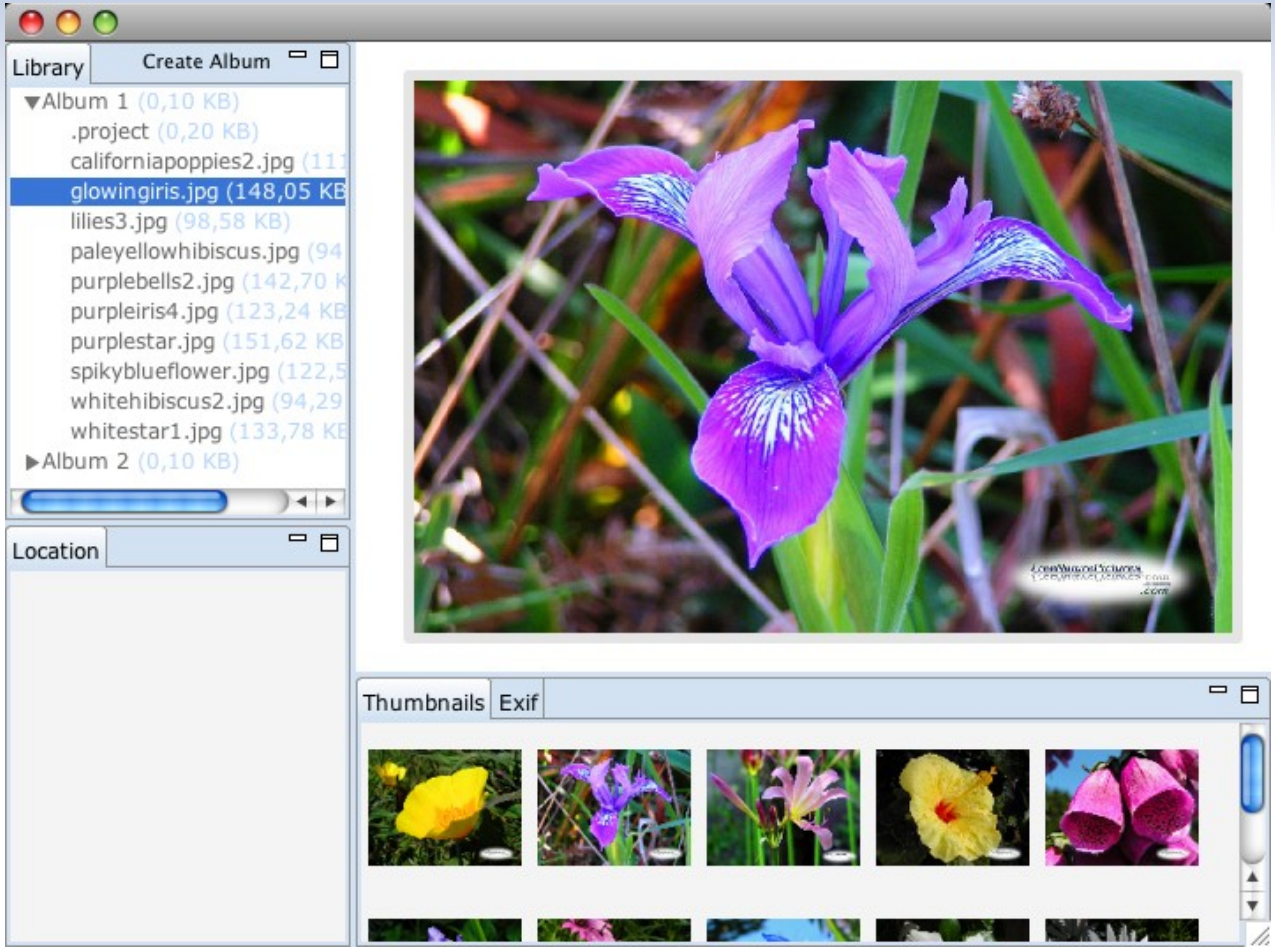
# E4 - The implementation

- Application Development with EMF-Tools



# E4 - The implementation

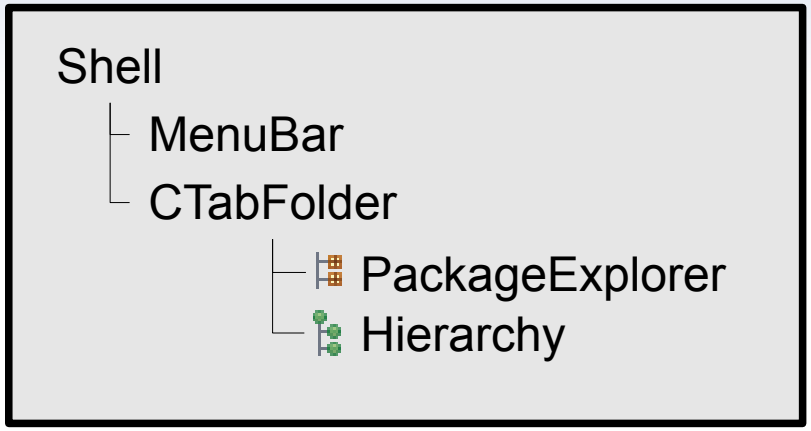
- Photo Demo



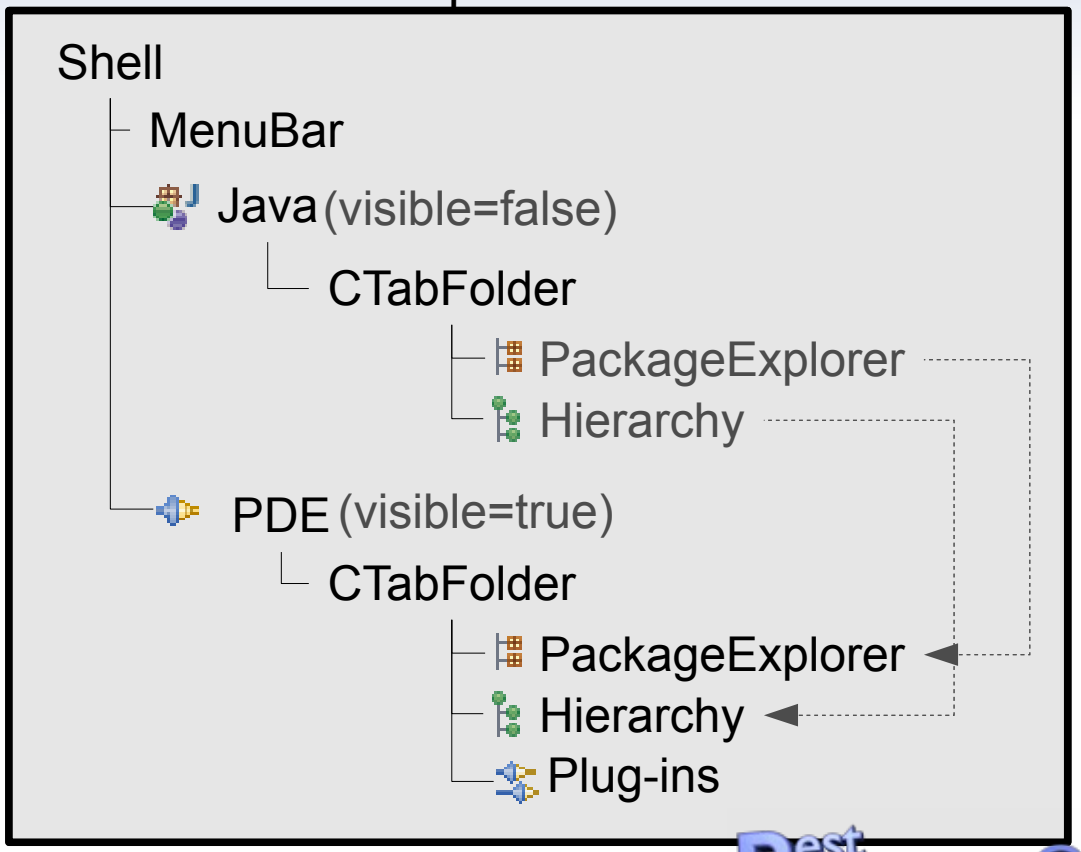
# E4 - The implementation

- The Perspective Problem

A standard UI is a Tree




An UI with Perspectives is a Tree with Links



# E4 - The implementation

- The usage of DI and IEclipseContext

  Example the Workbench-Selection

  - Reacting on selection changes

3.x

```
public class SampleView extends ViewPart {
  public void createPartControl(Composite parent) {
    // ...
    getSite().getWorkbenchWindow().getSelectionService().addSelectionListener(new ISelectionListener() {
      @Override
      public void selectionChanged(IWorkbenchPart part, ISelection selection) {
        // Do something
      }
    });
  }
}
```

E4

```
public class Preview {
  @In
  public void setInput(final IFile input) {
    // Do something
  }
}
```

# E4 - The implementation

- The usage of DI and IEclipseContext

  Example the Workbench-Selection

  - Setting the workbench selection

3.x

```
public class SampleView extends ViewPart {
  public void createPartControl(Composite parent) {
    TreeViewer viewer = ...;
    getSite().setSelectionProvider(viewer);
  }
}
```

E4

```
public class Library {
  public Library(final Composite parent, final IWorkspace workspace, final IEclipseContext outputContext){
    TreeViewer viewer = ...;
    viewer.addSelectionChangedListener(new ISelectionChangedListener(){
      public void selectionChanged(SelectionChangedEvent event) {
        StructuredSelection selection = (StructuredSelection)event.getSelection();
        outputContext.set(
          IServiceConstants.SELECTION,
          selection.size() == 1 ? selection.getFirstElement() : selection.toArray());
      }
    });
  }
}
```

# E4 - The implementation

- Declarative Styling with CSS
  - Keep UI-Code free of Styling informations
  - Apply styles on the fly without closing your application

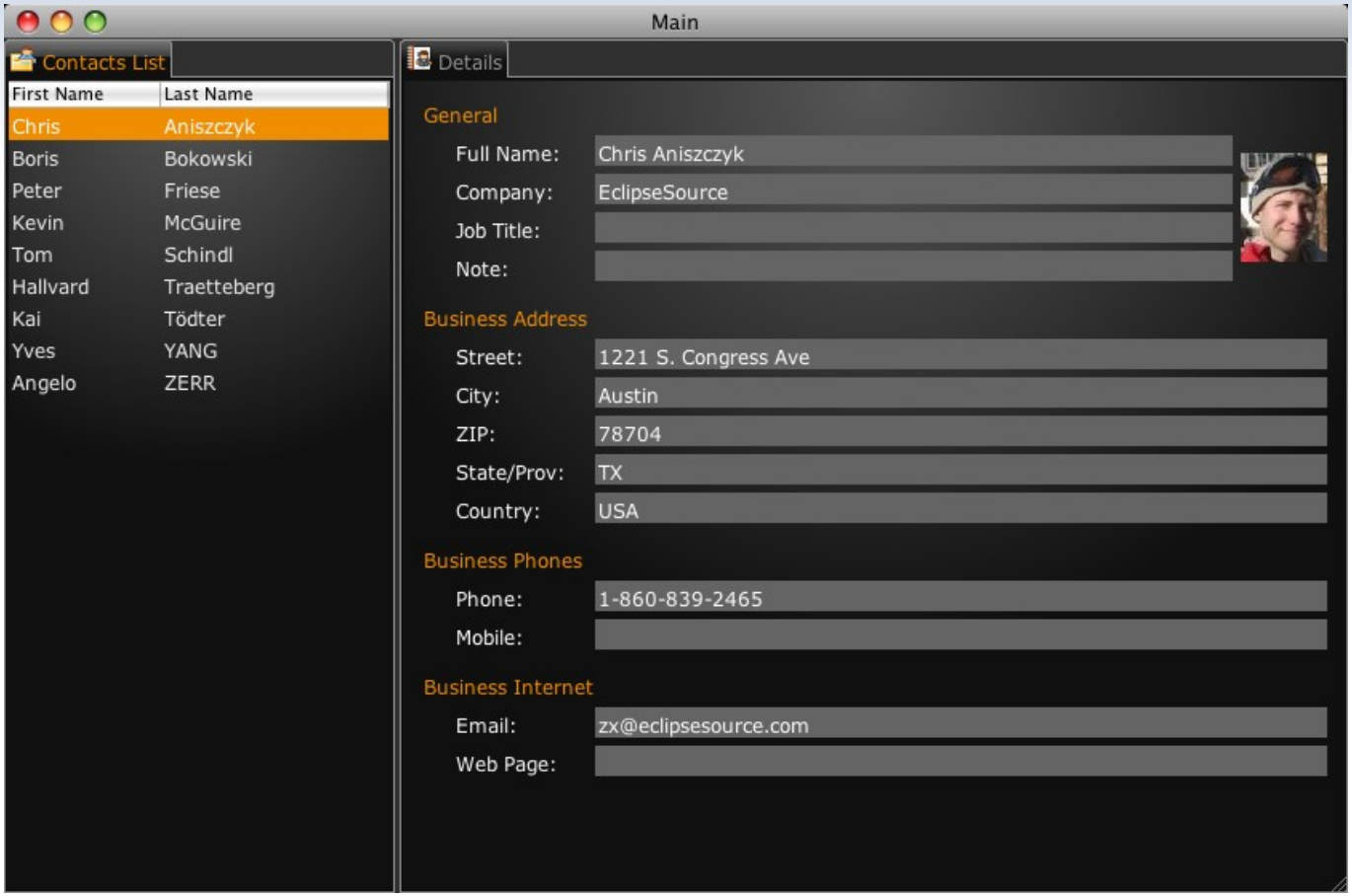
```
Shell shell = ...
URL url = FileLocator.resolve(new URL("file://my.css"));

InputStreamReader streamReader = new InputStreamReader(url.openStream());
engine.reset();
engine.parseStyleSheet(streamReader);
engine.applyStyles(shell, true, false);
shell.layout(true, true);
```

  - No E4 dependency = Useable in 3.x!

# E4 - The implementation

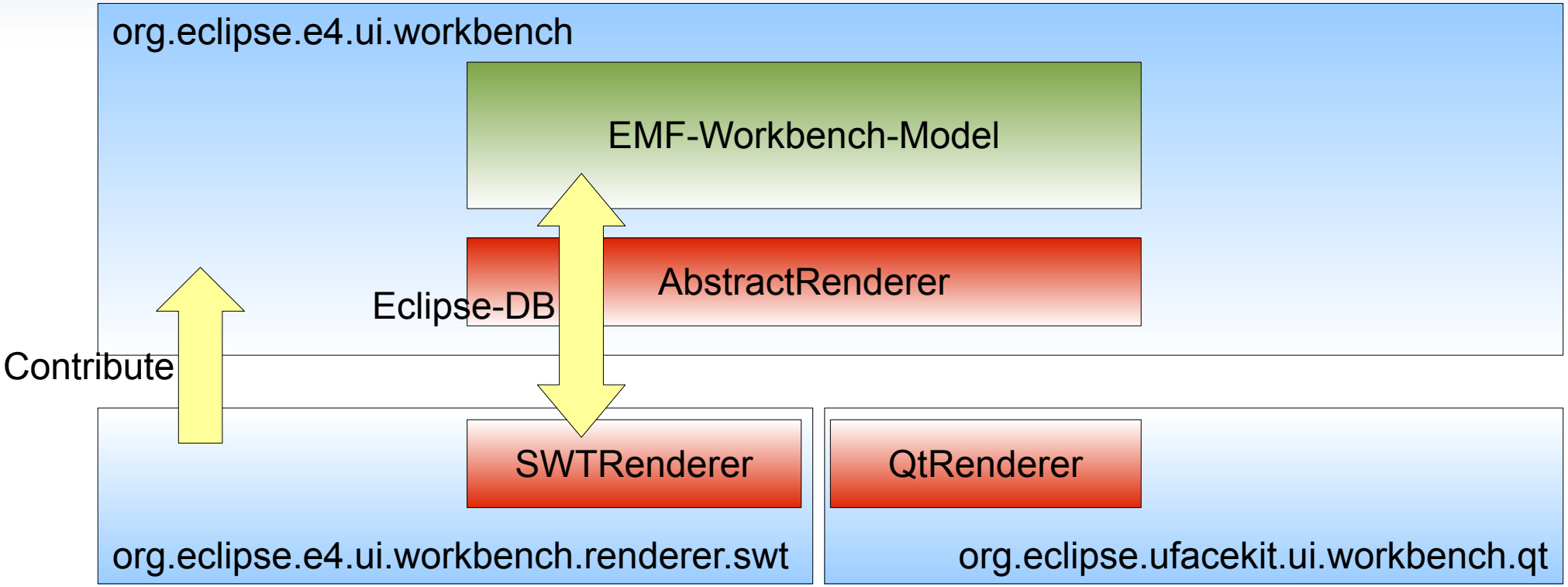
- Contacts Demo



# E4 - The implementation

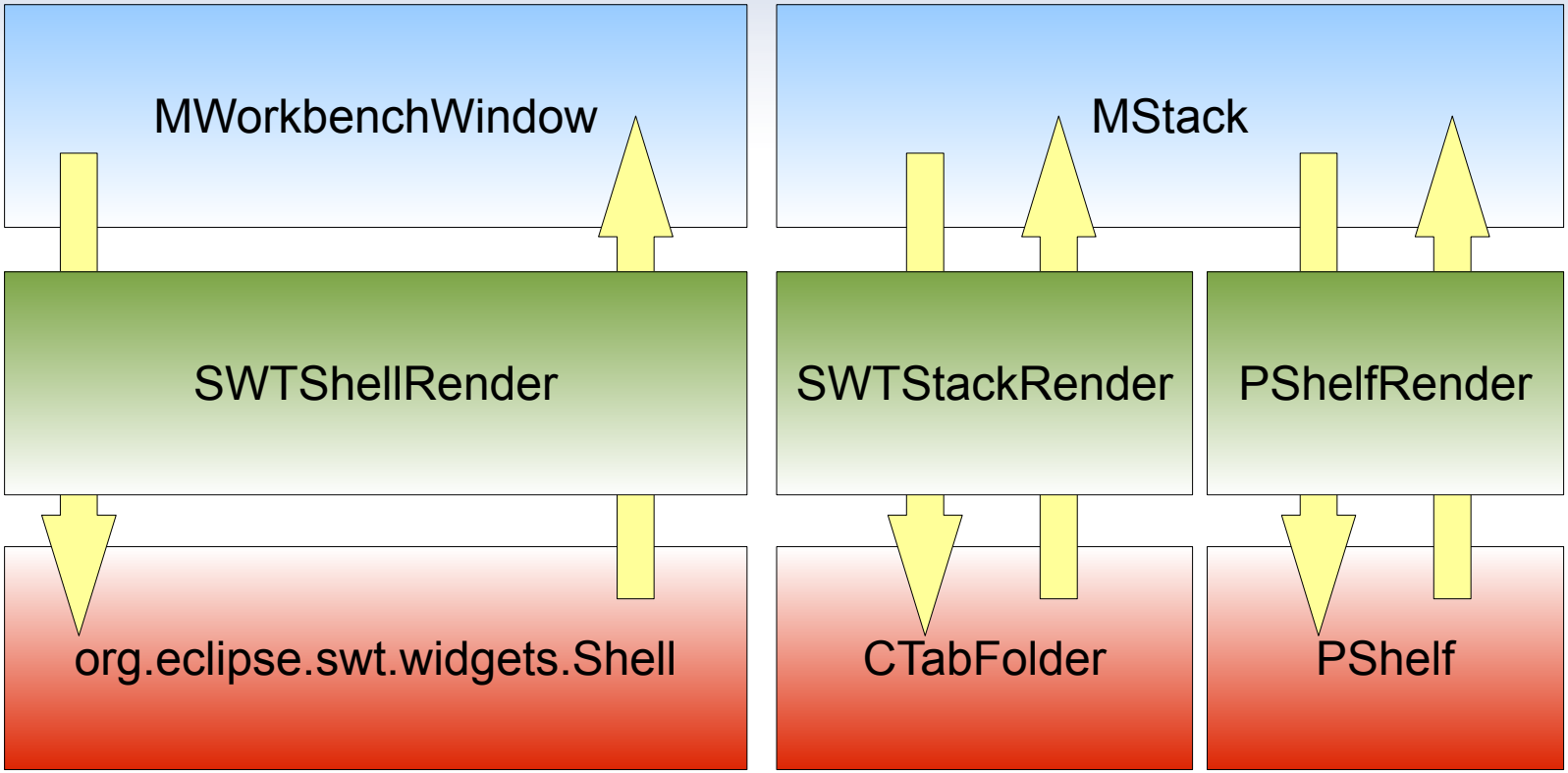
- Workbench-Renderes

  „Renderes start where Declarative Styling ends“



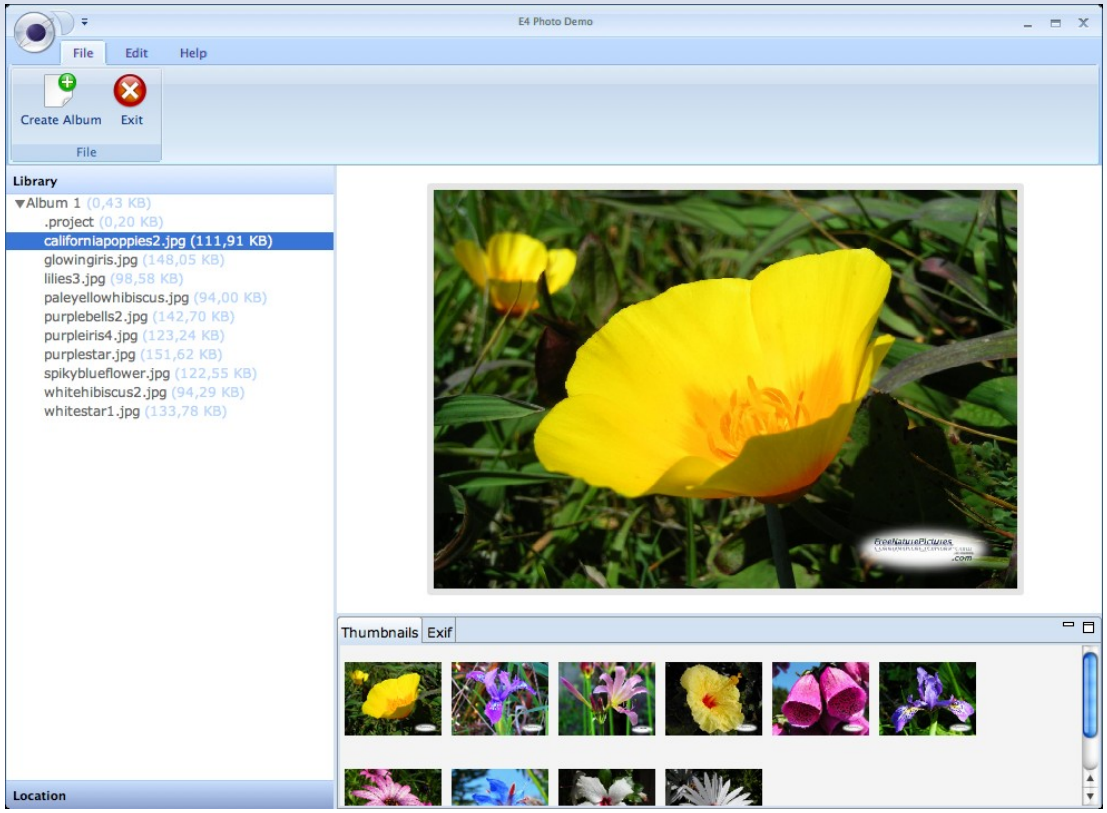
# E4 - The implementation

- 1:n relation between model and renderer



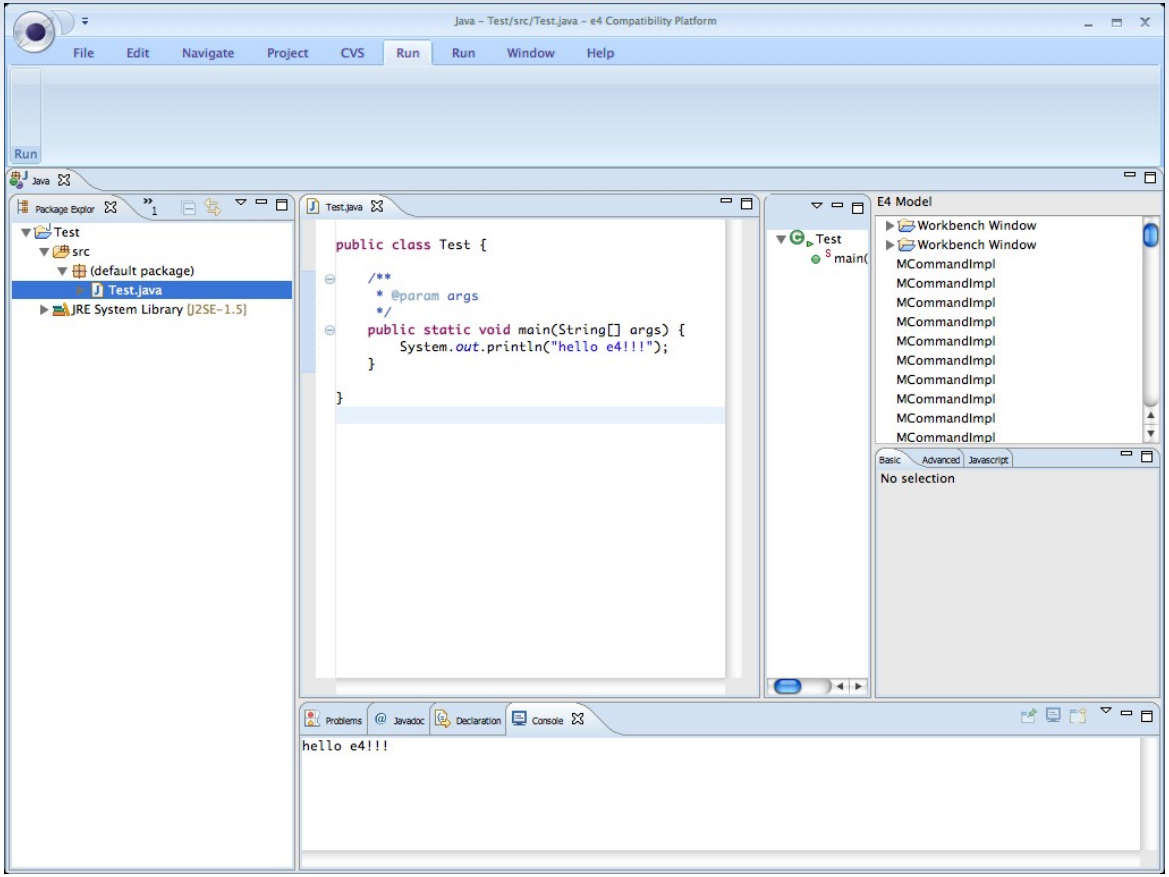
# E4 - The implementation

- Photo Demo



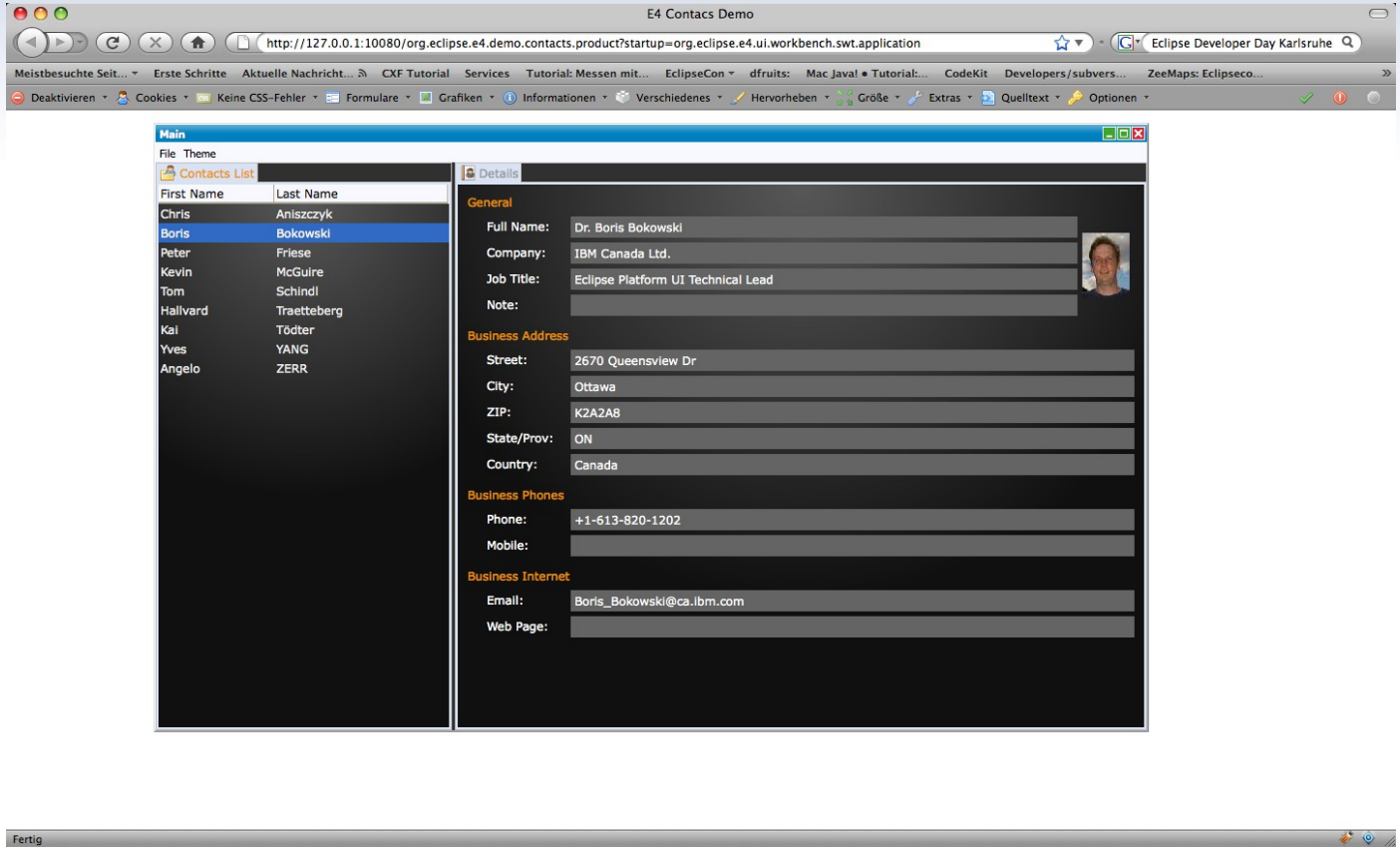
# E4 - You are cool but ...

- What about my current plugins?



# E4 - A „minor" side effect

- RAP doesn't need to patch the workbench



# E4 - Resources about E4

- EclipseCon 09 Presentations
- Wiki: http://wiki.eclipse.org/E4
- My Blog: http://tomsondev.bestsolution.at/