Eclipse 4 Application Platform

Tom Schindl - BestSolution Systemhaus GmbH

EclipseCon March 2012

About Tom

- CEO BestSolution Systemhaus GmbH
- Eclipse Committer
  - e4
  - Platform UI
  - EMF
- Projectlead: UFaceKit, Nebula
- Member of the Architectual Council

A bit of history (and future)

A bit of history (and future)

- EclipseCon `08: Announcement of e4-project

A bit of history (and future)

- EclipseCon `08: Announcement of e4-project
- End of May `08: Boris Bokowski and Tom Schindl publish a prototype of a new Eclipse 4 Platform in an mail named „A radical approach to explore new paths for e4“

A bit of history (and future)

- EclipseCon `08: Announcement of e4-project
- End of May `08: Boris Bokowski and Tom Schindl publish a prototype of a new Eclipse 4 Platform in an mail named „A radical approach to explore new paths for e4“
- Summer `09: e4 0.9 published as a tech preview

A bit of history (and future)

- EclipseCon `08: Announcement of e4-project
- End of May `08: Boris Bokowski and Tom Schindl publish a prototype of a new Eclipse 4 Platform in an mail named „A radical approach to explore new paths for e4“
- Summer `09: e4 0.9 published as a tech preview
- Summer `10: Eclipse 4.0 SDK published as a developer preview

A bit of history (and future)

- EclipseCon `08: Announcement of e4-project
- End of May `08: Boris Bokowski and Tom Schindl publish a prototype of a new Eclipse 4 Platform in an mail named „A radical approach to explore new paths for e4“
- Summer `09: e4 0.9 published as a tech preview
- Summer `10: Eclipse 4.0 SDK published as a developer preview
- Summer `11 (as part of Indigo): Eclipse 4.1 SDK published

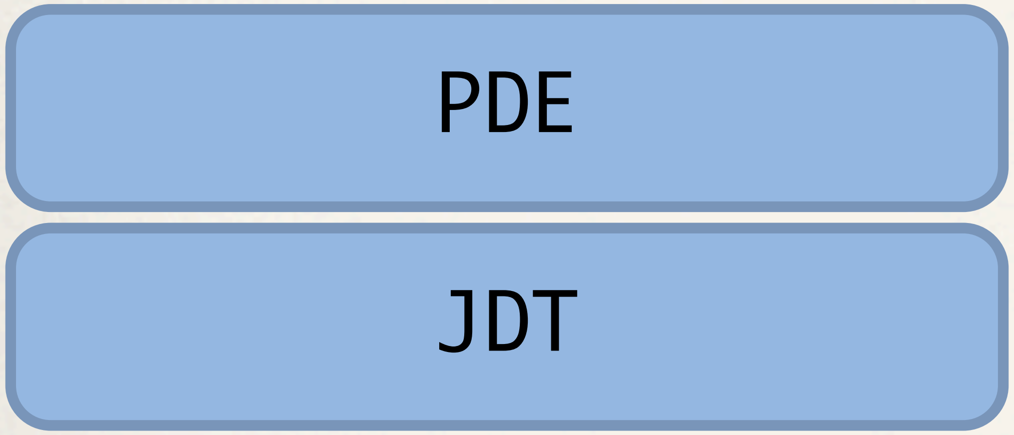
A bit of history (and future)

- EclipseCon `08: Announcement of e4-project
- End of May `08: Boris Bokowski and Tom Schindl publish a prototype of a new Eclipse 4 Platform in an mail named „A radical approach to explore new paths for e4“
- Summer `09: e4 0.9 published as a tech preview
- Summer `10: Eclipse 4.0 SDK published as a developer preview
- Summer `11 (as part of Indigo): Eclipse 4.1 SDK published
- Summer `12: Juno release train will ship on Eclipse 4.2 SDK

Eclipse SDK 3.x vs 4.x

Eclipse 3.x **Eclipse 4.x**

Eclipse SDK 3.x vs 4.x



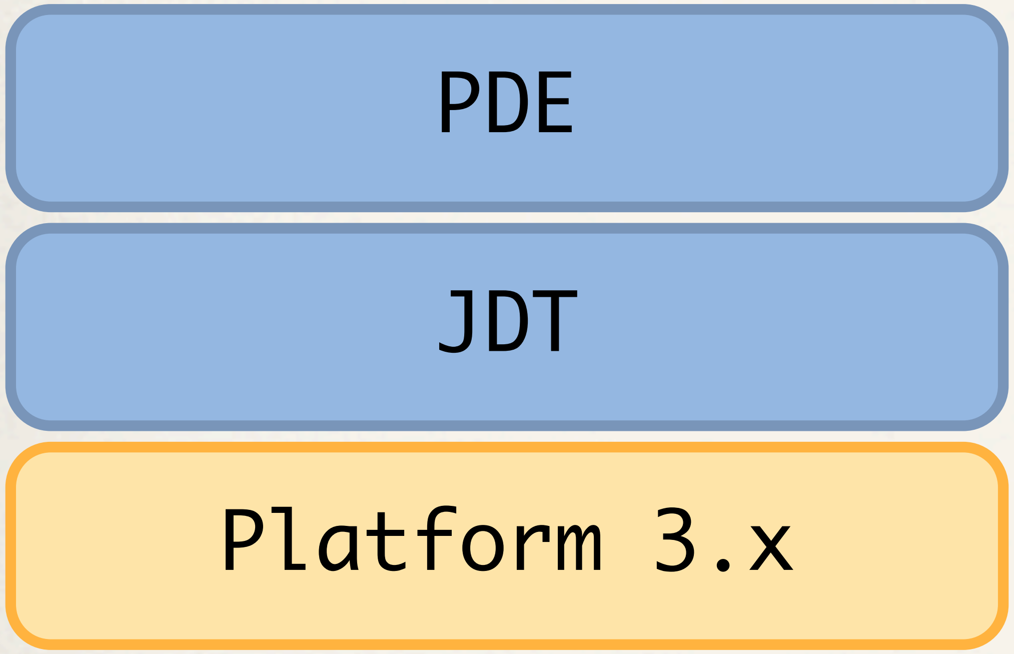
PDE

JDT

Eclipse 3.x

Eclipse 4.x

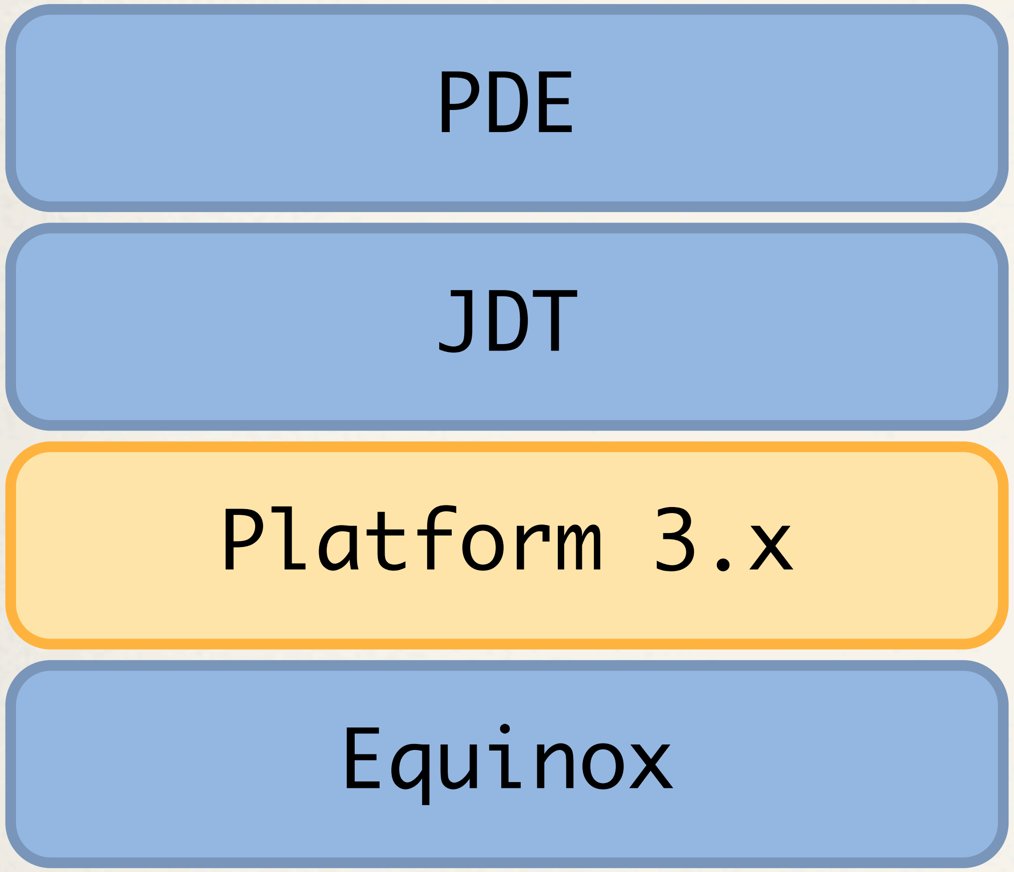
Eclipse SDK 3.x vs 4.x



Eclipse 3.x

Eclipse 4.x

Eclipse SDK 3.x vs 4.x



PDE

JDT

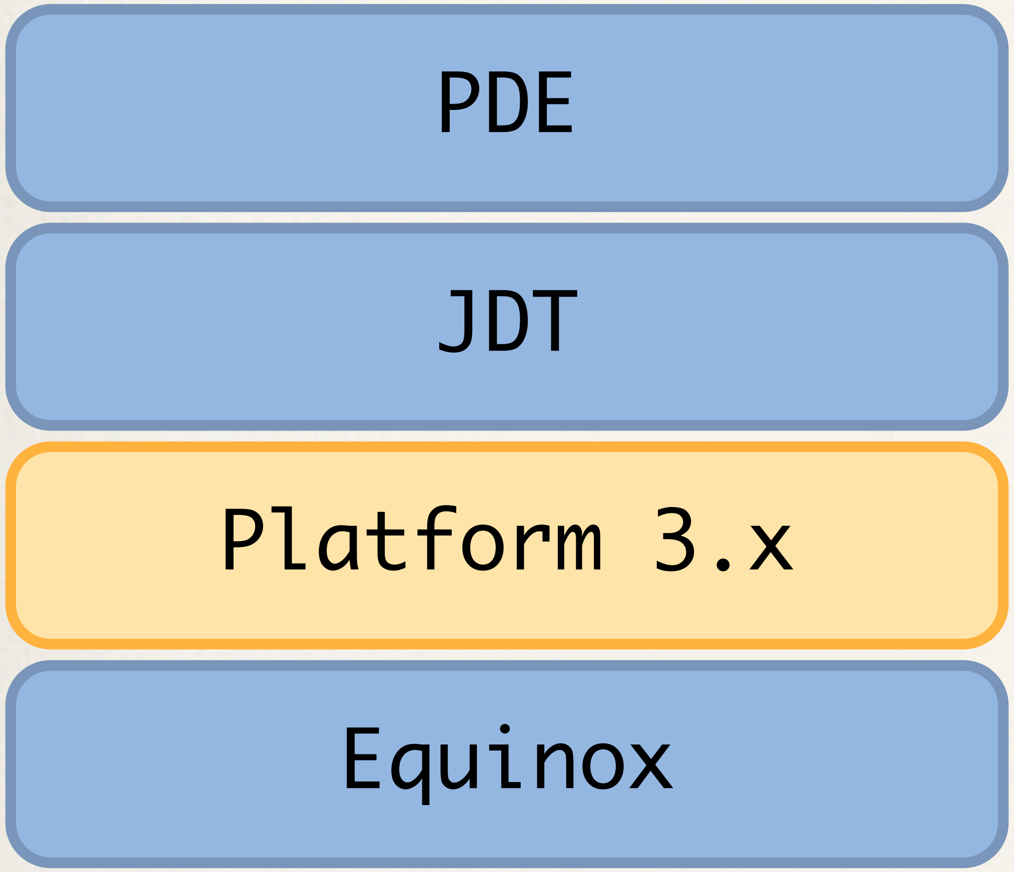
Platform 3.x

Equinox

Eclipse 3.x

Eclipse 4.x

Eclipse SDK 3.x vs 4.x



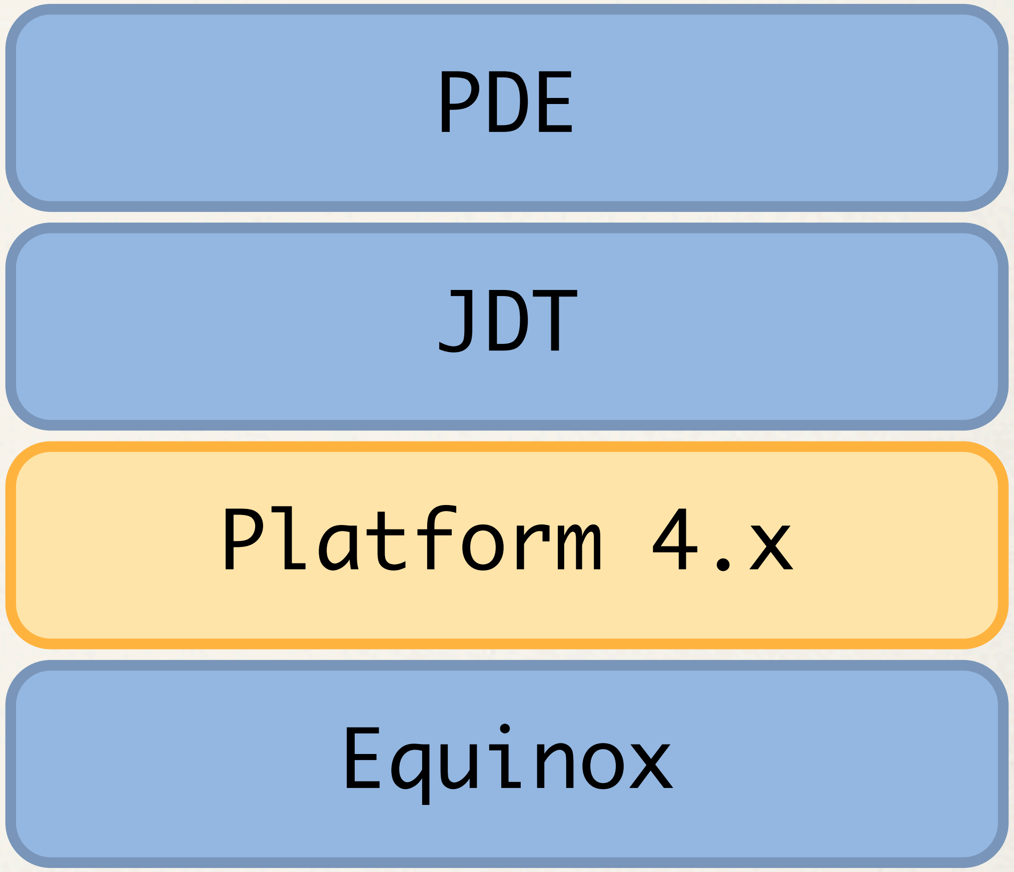
PDE

JDT

Platform 3.x

Equinox

Eclipse 3.x



PDE

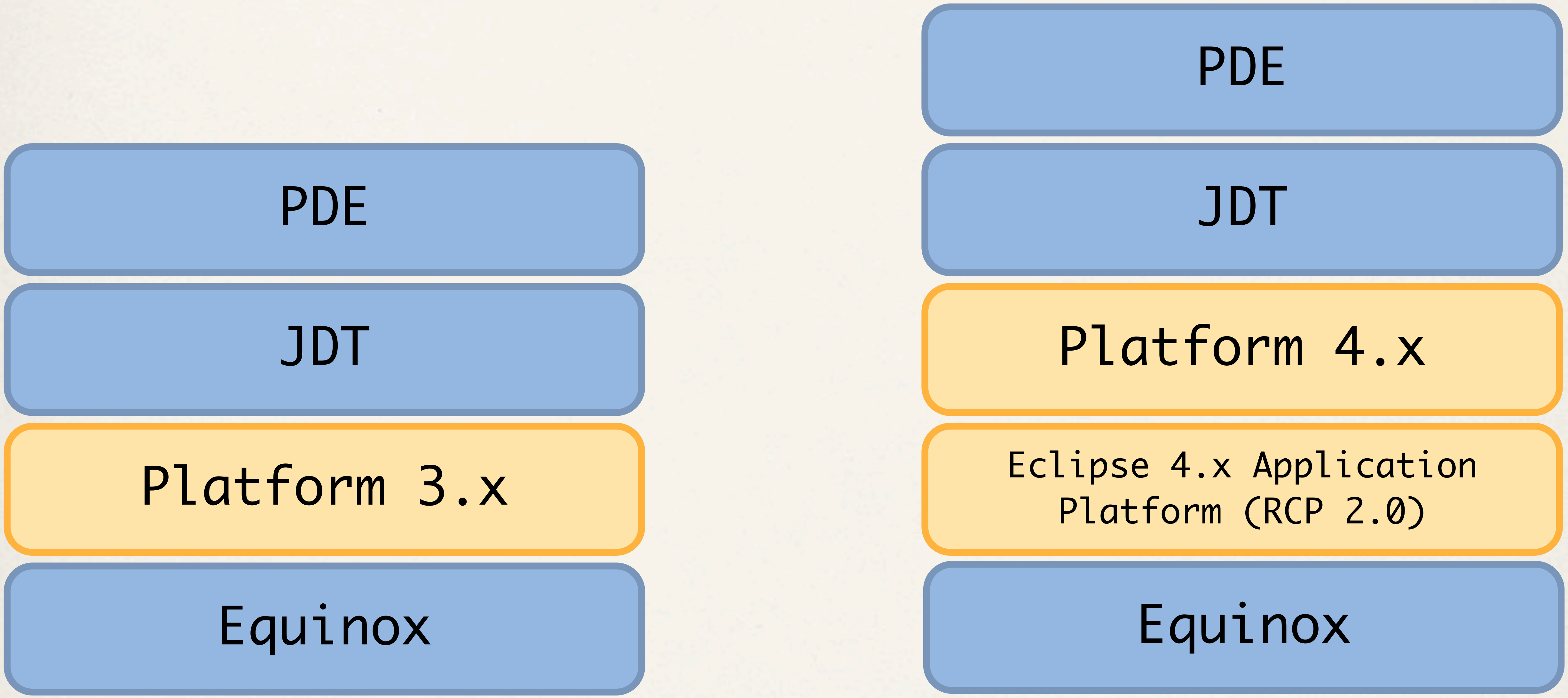
JDT

Platform 4.x

Equinox

Eclipse 4.x

Eclipse SDK 3.x vs 4.x



PDE

JDT

Platform 3.x

Equinox

Eclipse 3.x

PDE

JDT

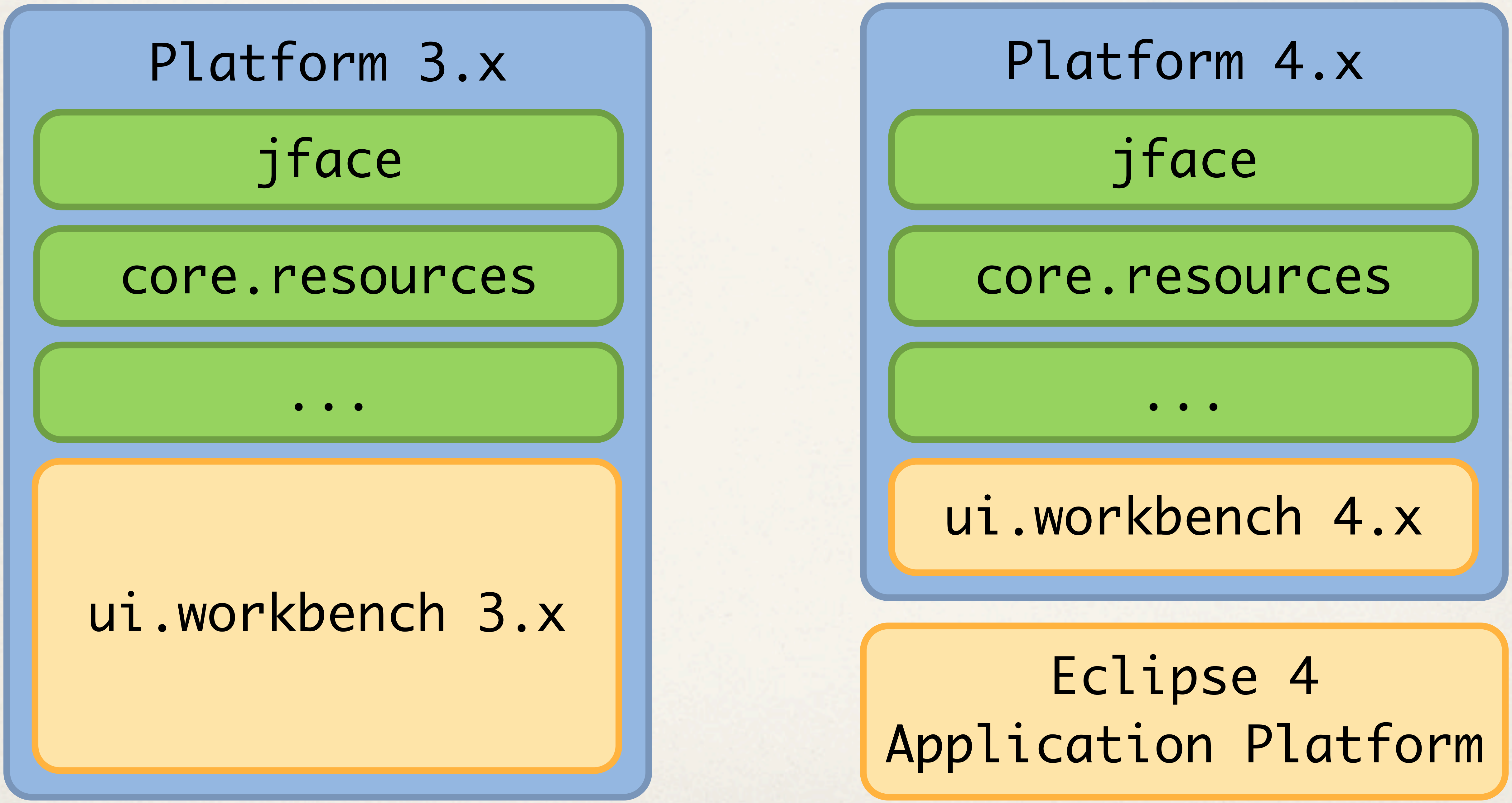
Platform 4.x

Eclipse 4.x Application Platform (RCP 2.0)

Equinox

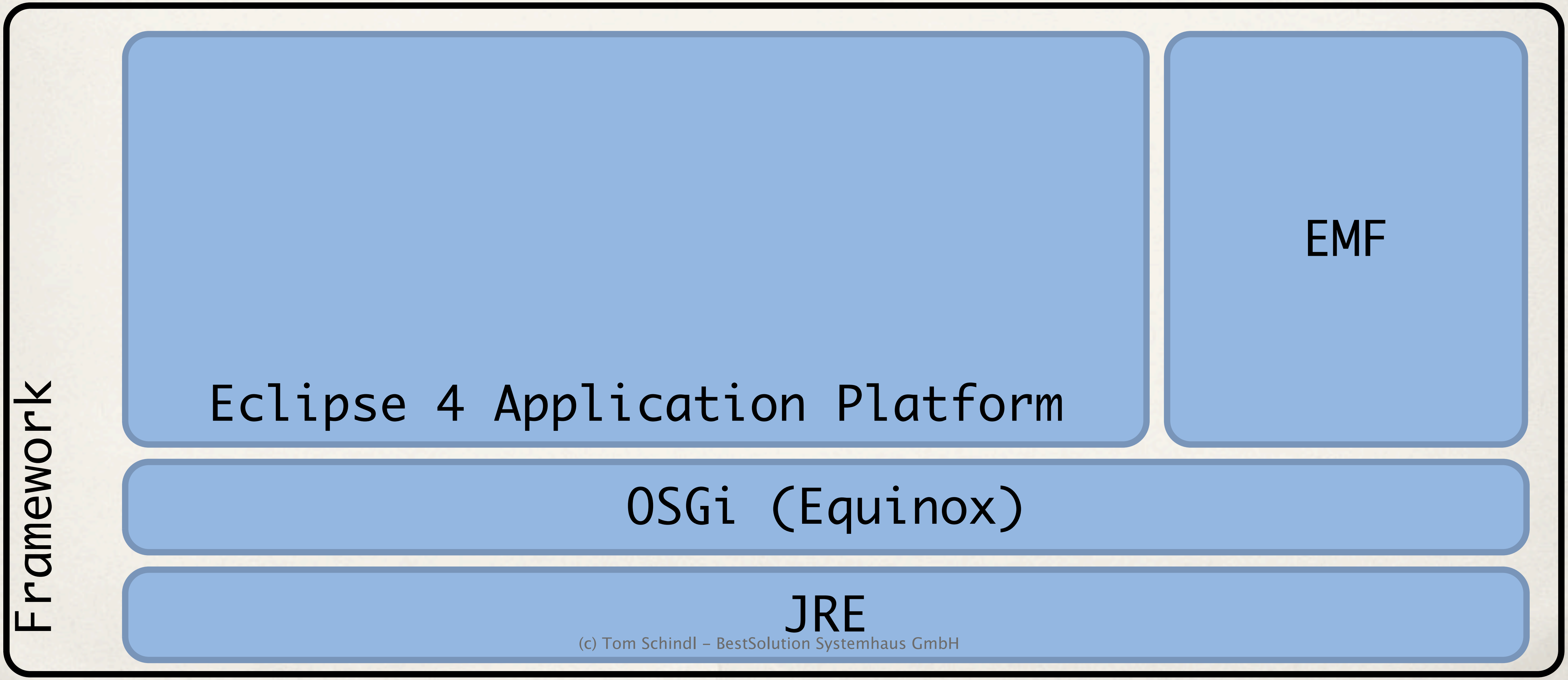
Eclipse 4.x

Eclipse SDK 3.x vs 4.x



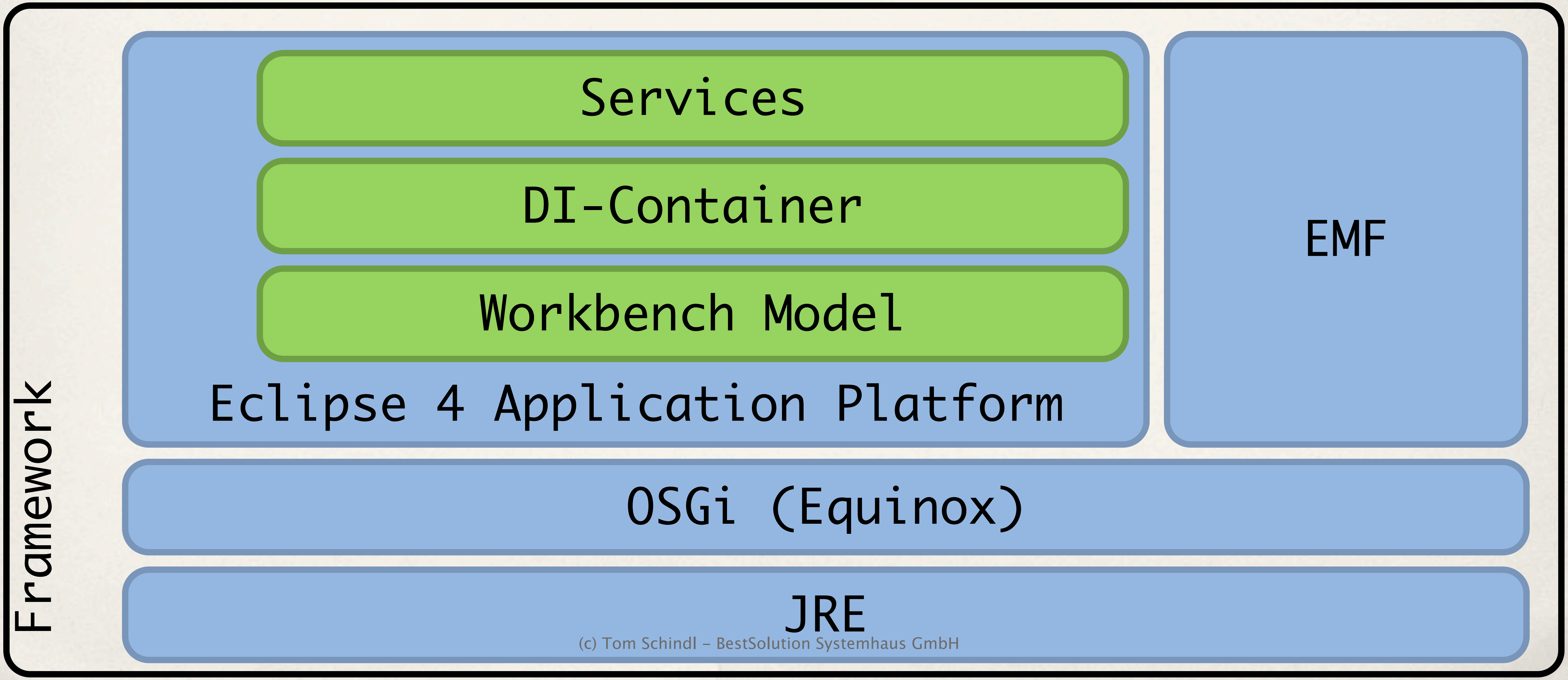
Eclipse 4.1 Application Platform

Eclipse 4.1 Application Platform



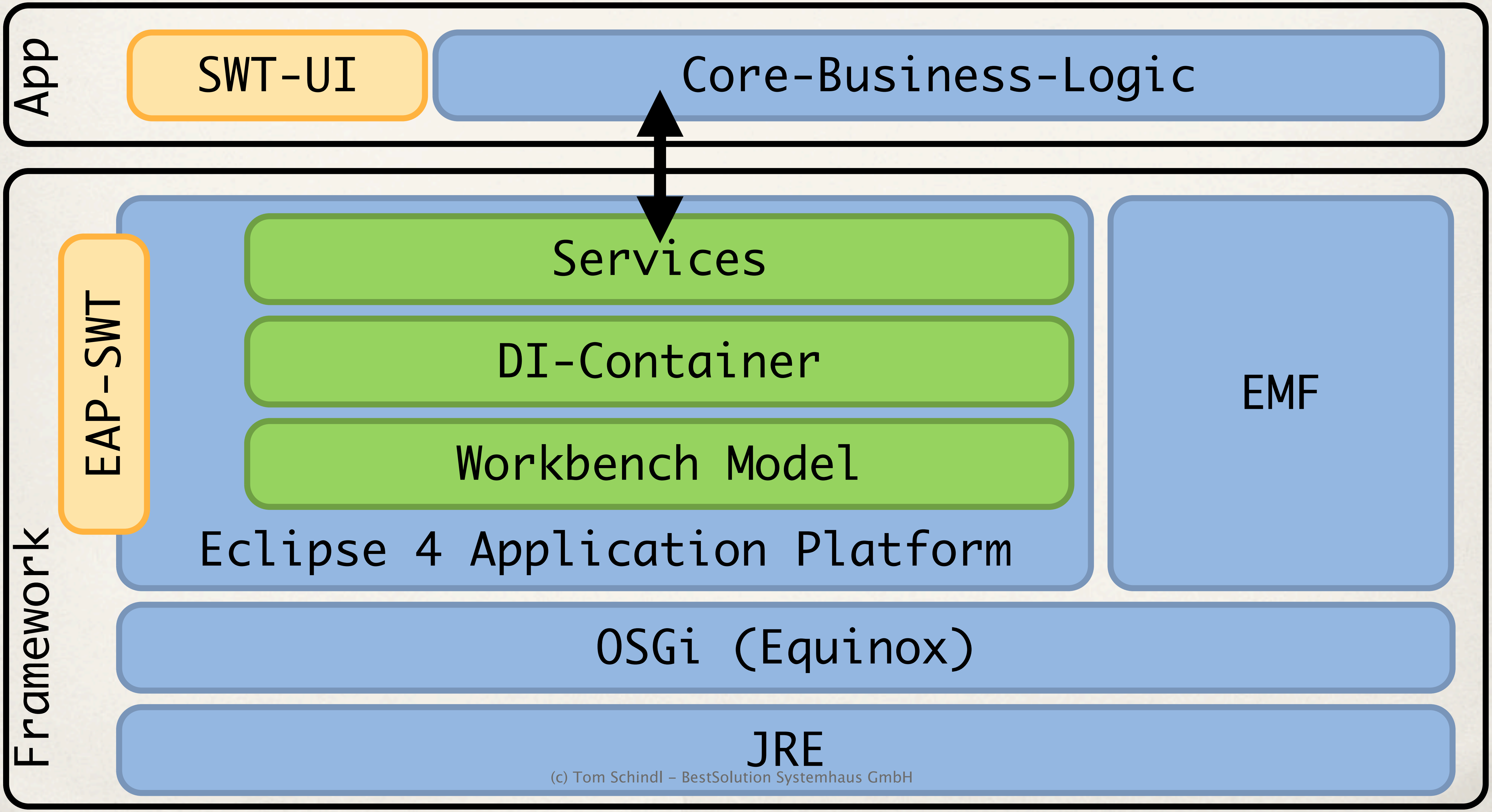
(c) Tom Schindl - BestSolution Systemhaus GmbH

Eclipse 4.1 Application Platform



(c) Tom Schindl - BestSolution Systemhaus GmbH

Eclipse 4.1 Application Platform



(c) Tom Schindl - BestSolution Systemhaus GmbH

Dependency Injection

```
public class MyPart {

  void createPartControl(Composite parent) {

  }


  void selChanged(Object value) {
  }


  void dispose() {
  }


  void setFocus() {
  }
}
```

Dependency Injection

```
public class MyPart {
  @PostConstruct
  void createPartControl(Composite parent) {

  }

  @Inject
  void selChanged(@Named("selection") @Optional Object value) {
  }

  @PreDestroy
  void dispose() {
  }

  @Focus
  void setFocus() {
  }
}
```

Dependency Injection

```java
public class PartRenderer {
   public void createContrib(Composite c, IEclipseContext ctx) throws Exception {
      ctx.set("org.eclipse.swt.widgets.Composite",s);

      MyPart part = ContextInjectionFactory.make(MyPart.class, ctx);
   }
}
```

Dependency Injection

Dependency Injection

- **What is subject of injection**

Dependency Injection

- **What is subject of injection**
  - All objects stored in the IEclipseContext-Hierarchy

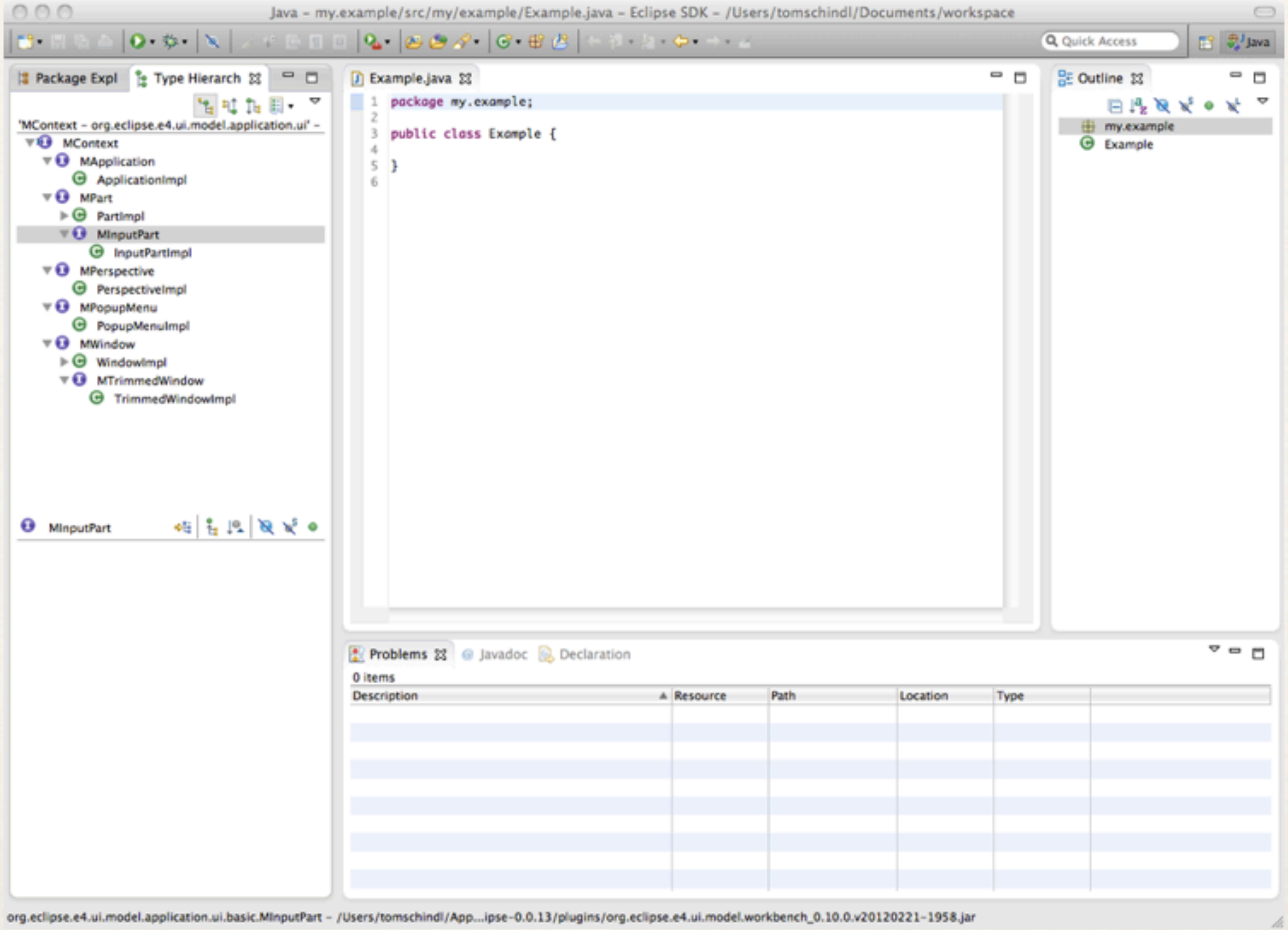
Dependency Injection

- **What is subject of injection**
  - All objects stored in the IEclipseContext-Hierarchy
  - All Preferences

Dependency Injection

- **What is subject of injection**
  - All objects stored in the IEclipseContext-Hierarchy
  - All Preferences
  - All objects stored in the OSGi-Service-Registry

IEclipseContext



Java - my.example/src/my/example/Example.java - Eclipse SDK - /Users/tomschindl/Documents/workspace

'MContext - org.eclipse.e4.ui.model.application.ui' -

```
▼ MContext
  ▼ MApplication
      ApplicationImpl
  ▼ MPart
    ▶ PartImpl
    ▼ MInputPart
        InputPartImpl
  ▼ MPerspective
      PerspectiveImpl
  ▼ MPopupMenu
      PopupMenuImpl
  ▼ MWindow
    ▶ WindowImpl
    ▼ MTrimmedWindow
        TrimmedWindowImpl
```

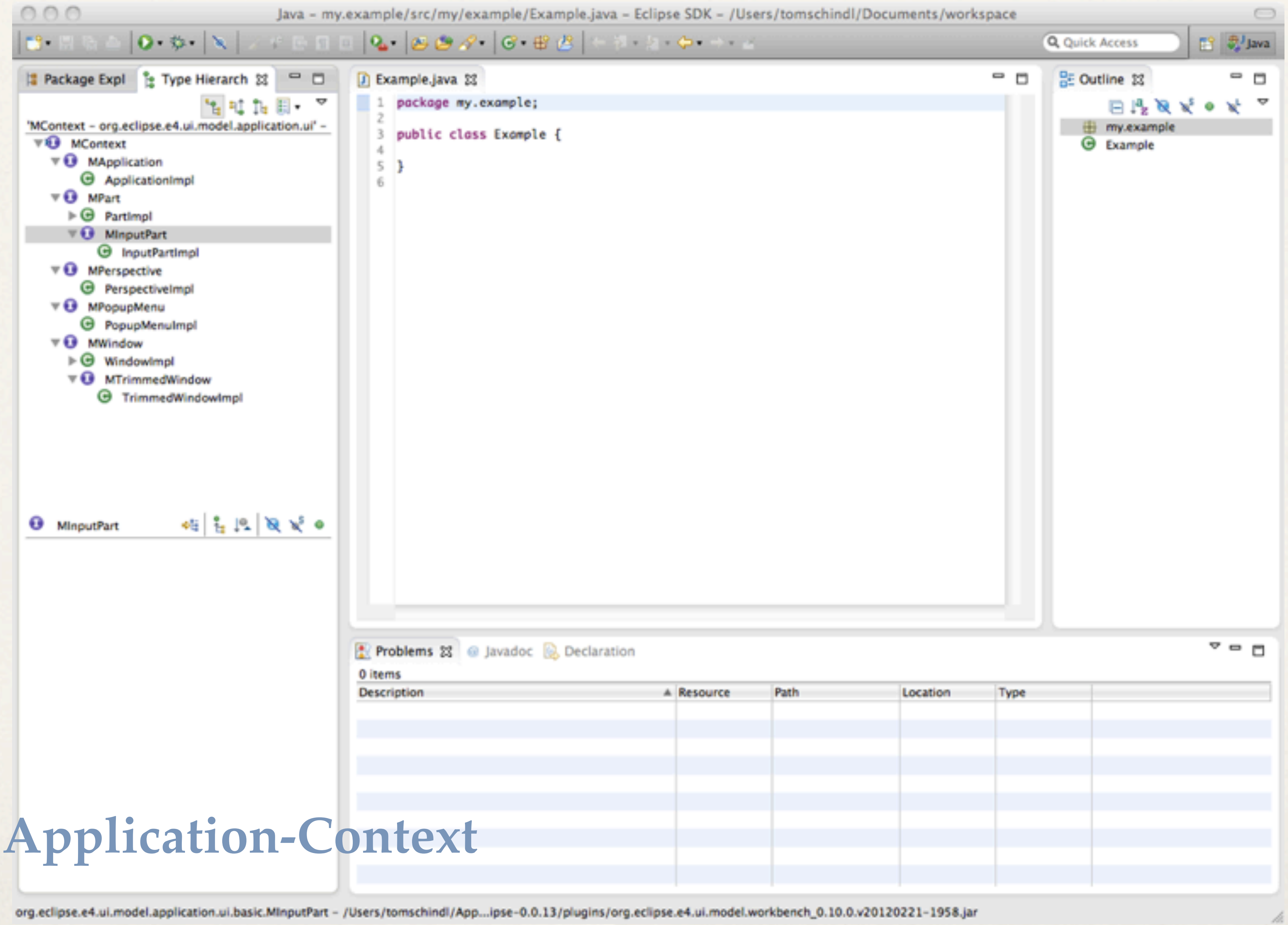
```
package my.example;

public class Example {

}
```

0 items

Description	Resource	Path	Location	Type

org.eclipse.e4.ui.model.application.ui.basic.MInputPart - /Users/tomschindl/App...ipse-0.0.13/plugins/org.eclipse.e4.ui.model.workbench_0.10.0.v20120221-1958.jar

IEclipseContext



Java - my.example/src/my/example/Example.java - Eclipse SDK - /Users/tomschindl/Documents/workspace
Quick Access
Java
Package Expl
Type Hierarch
Example.java
Outline
Problems
Javadoc
Declaration

```
'MContext - org.eclipse.e4.ui.model.application.ui' -
▼ MContext
  ▼ MApplication
      ApplicationImpl
  ▼ MPart
    ▶ PartImpl
    ▼ MInputPart
        InputPartImpl
  ▼ MPerspective
      PerspectiveImpl
  ▼ MPopupMenu
      PopupMenuImpl
  ▼ MWindow
    ▶ WindowImpl
    ▼ MTrimmedWindow
        TrimmedWindowImpl

MInputPart
```

```java
package my.example;

public class Example {

}
```

```
my.example
Example
```

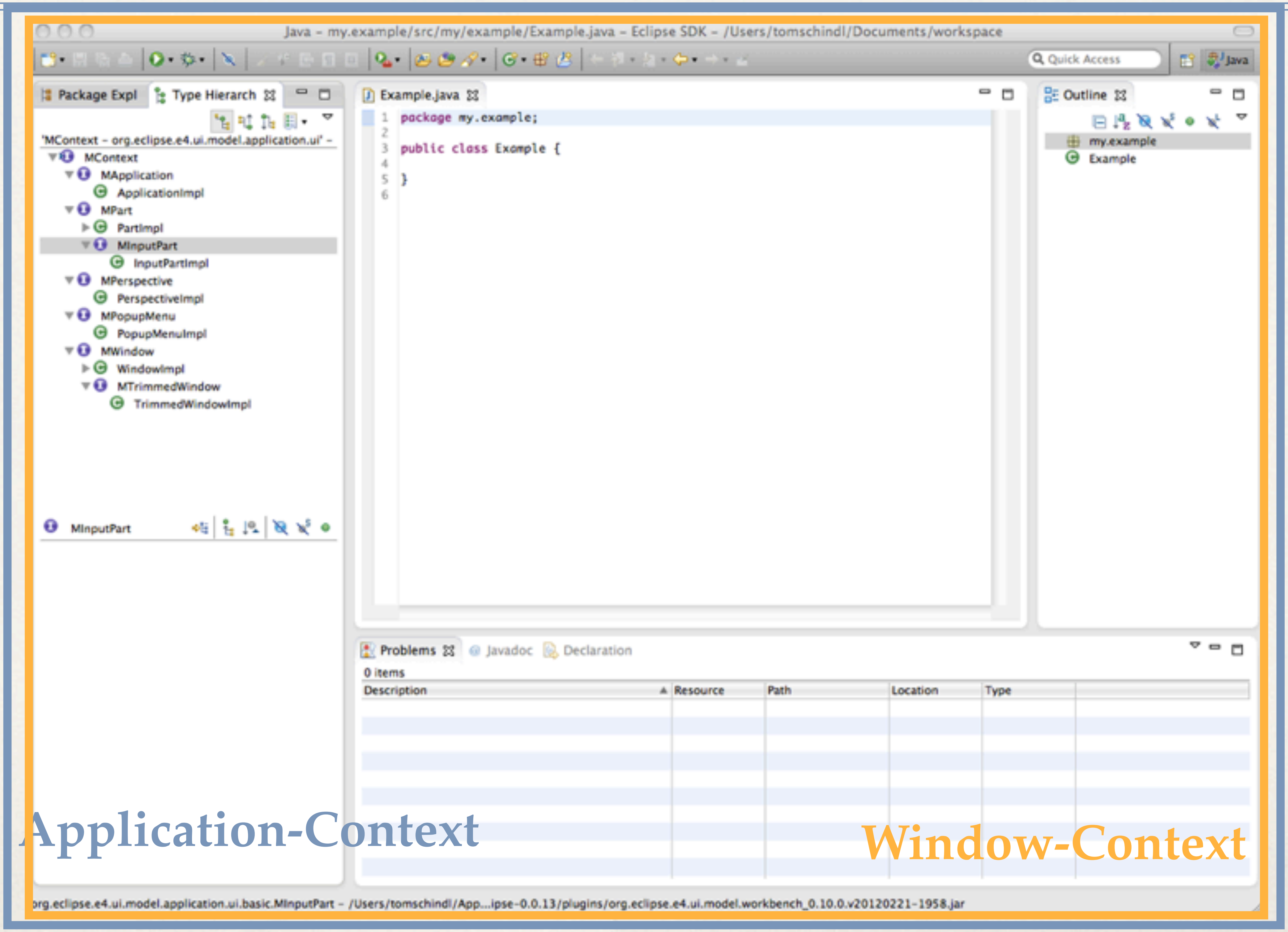
0 items

Description	Resource	Path	Location	Type

org.eclipse.e4.ui.model.application.ui.basic.MInputPart - /Users/tomschindl/App...ipse-0.0.13/plugins/org.eclipse.e4.ui.model.workbench_0.10.0.v20120221-1958.jar

Application-Context

IEclipseContext



'MContext – org.eclipse.e4.ui.model.application.ui' –

- MContext
  - MApplication
    - ApplicationImpl
  - MPart
    - PartImpl
    - MInputPart
      - InputPartImpl
  - MPerspective
    - PerspectiveImpl
  - MPopupMenu
    - PopupMenuImpl
  - MWindow
    - WindowImpl
    - MTrimmedWindow
      - TrimmedWindowImpl

MInputPart

```
package my.example;

public class Example {

}
```

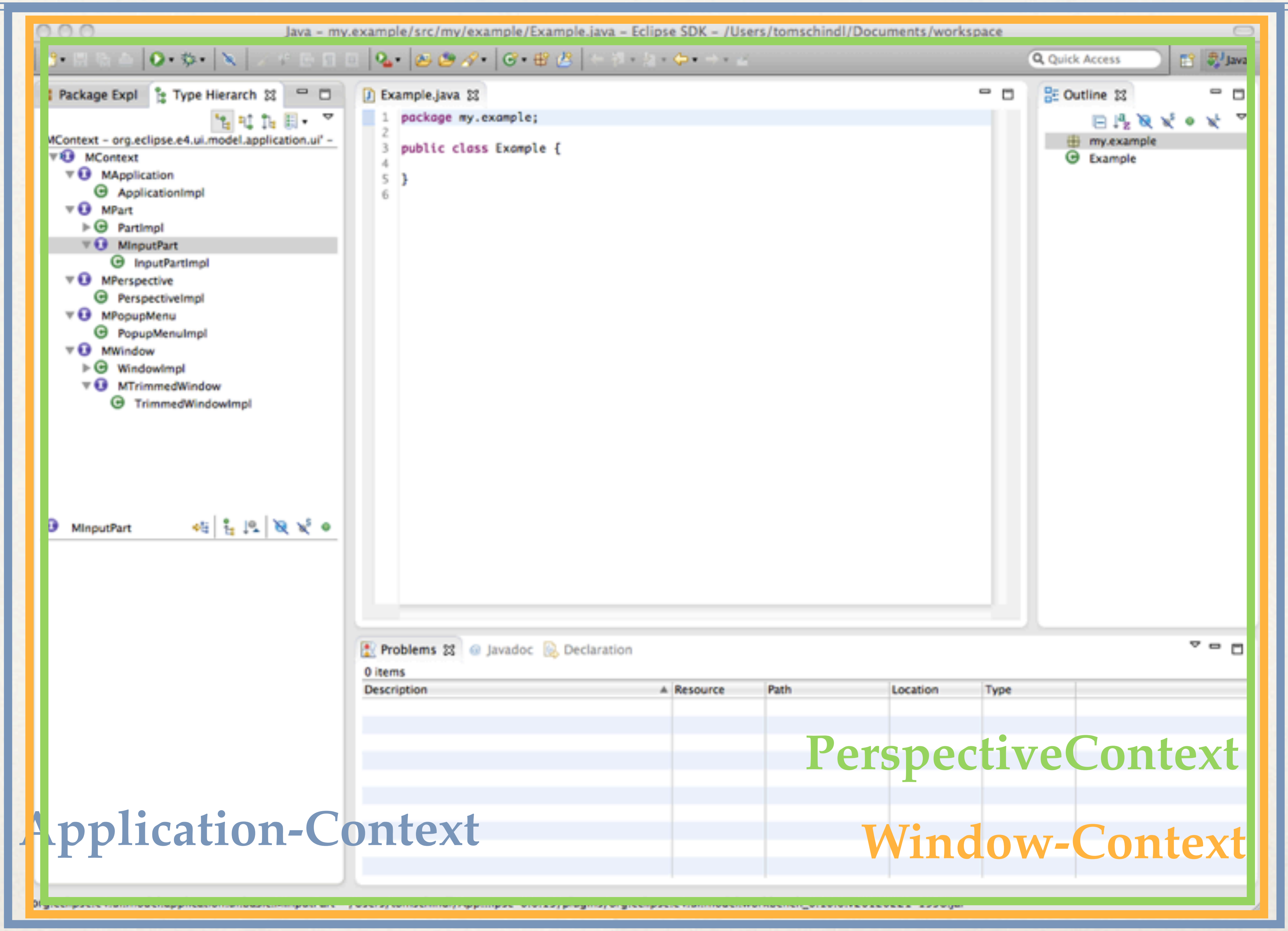
- my.example
- Example

0 items

Description	Resource	Path	Location	Type

rg.eclipse.e4.ui.model.application.ui.basic.MInputPart – /Users/tomschindl/App...ipse-0.0.13/plugins/org.eclipse.e4.ui.model.workbench_0.10.0.v20120221-1958.jar

IEclipseContext



```
package my.example;

public class Example {

}
```

0 items

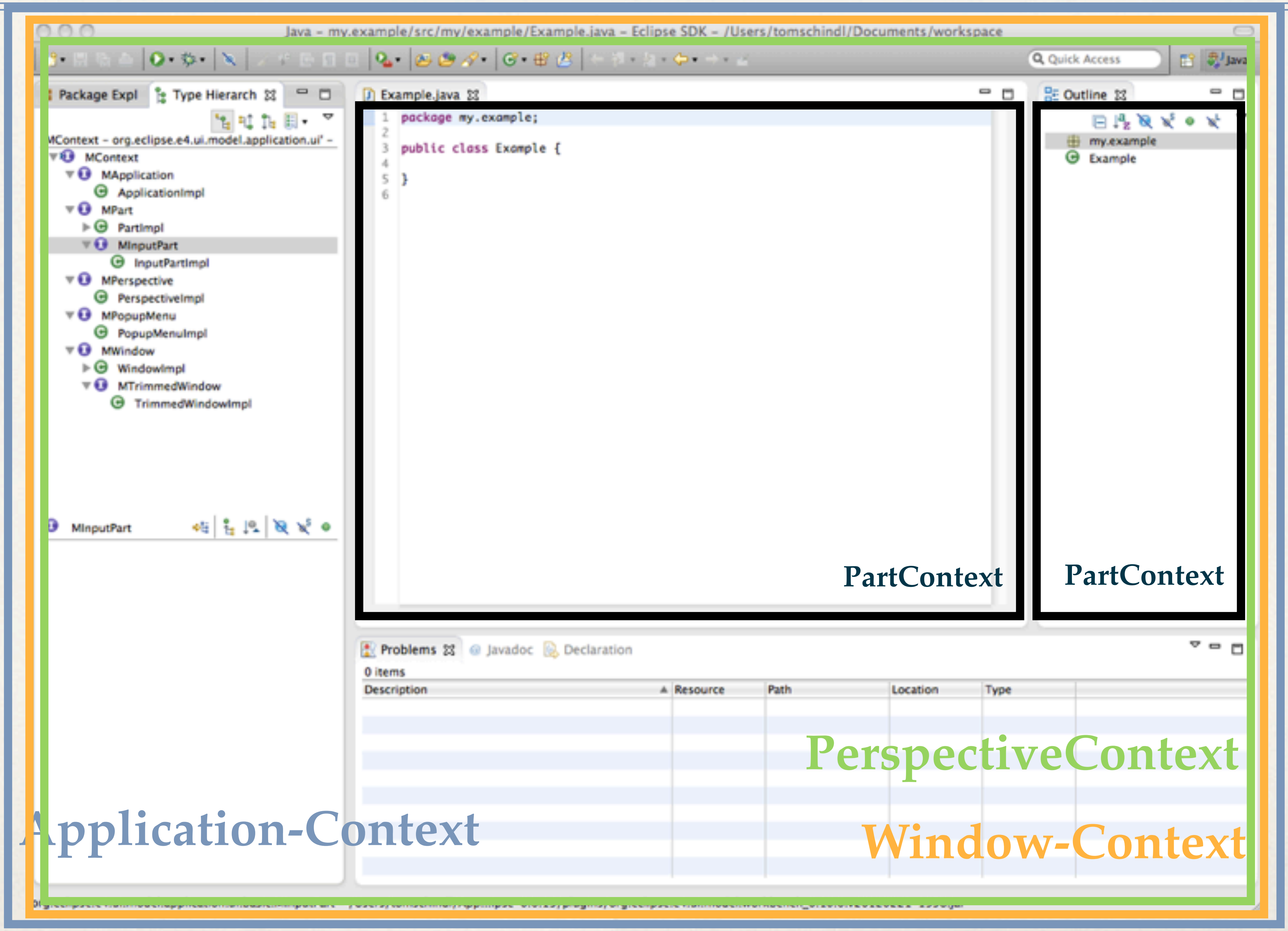
Description	Resource	Path	Location	Type

PerspectiveContext

Application-Context

Window-Context

IEclipseContext



```java
package my.example;

public class Example {

}
```

PartContext

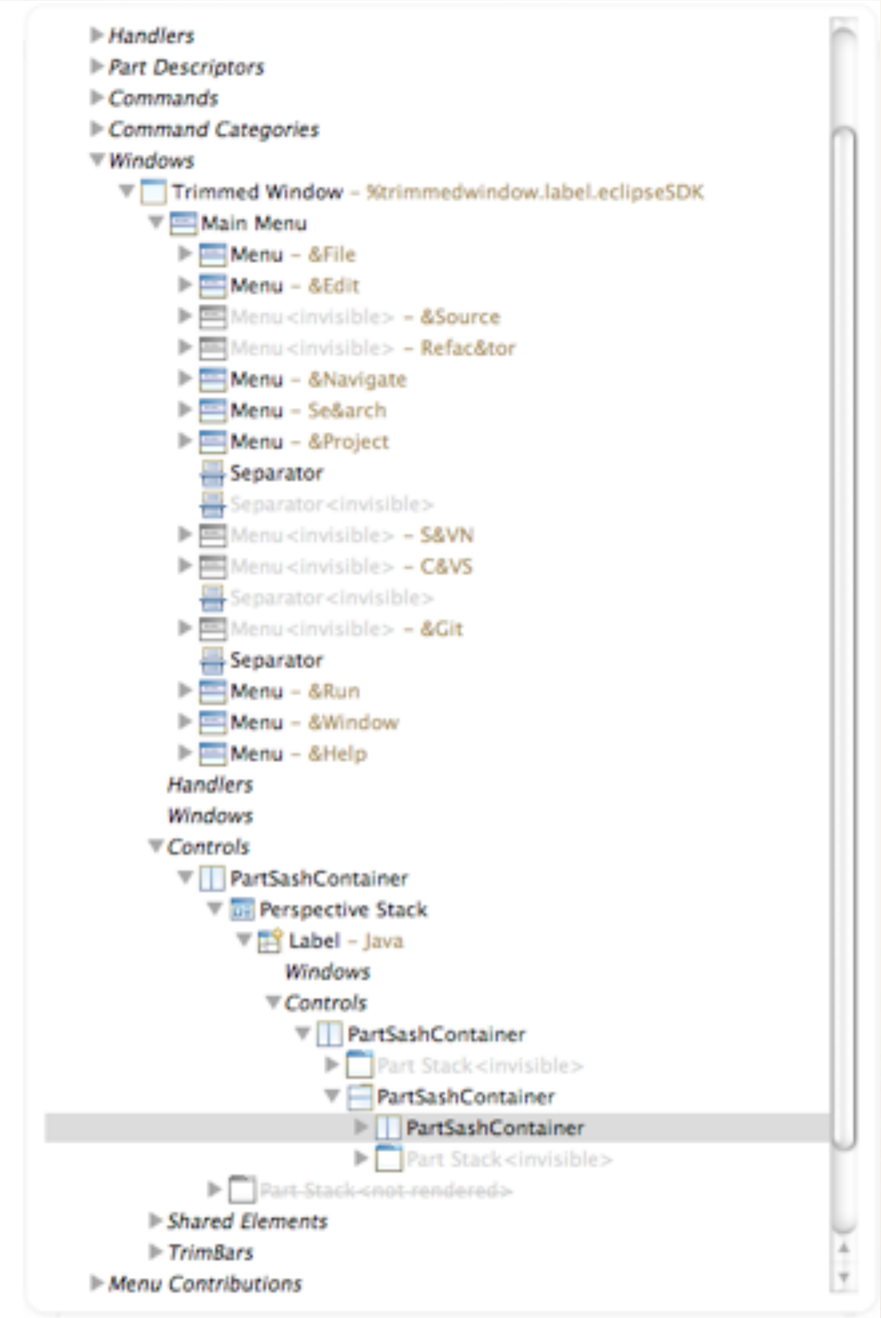
PartContext

PerspectiveContext

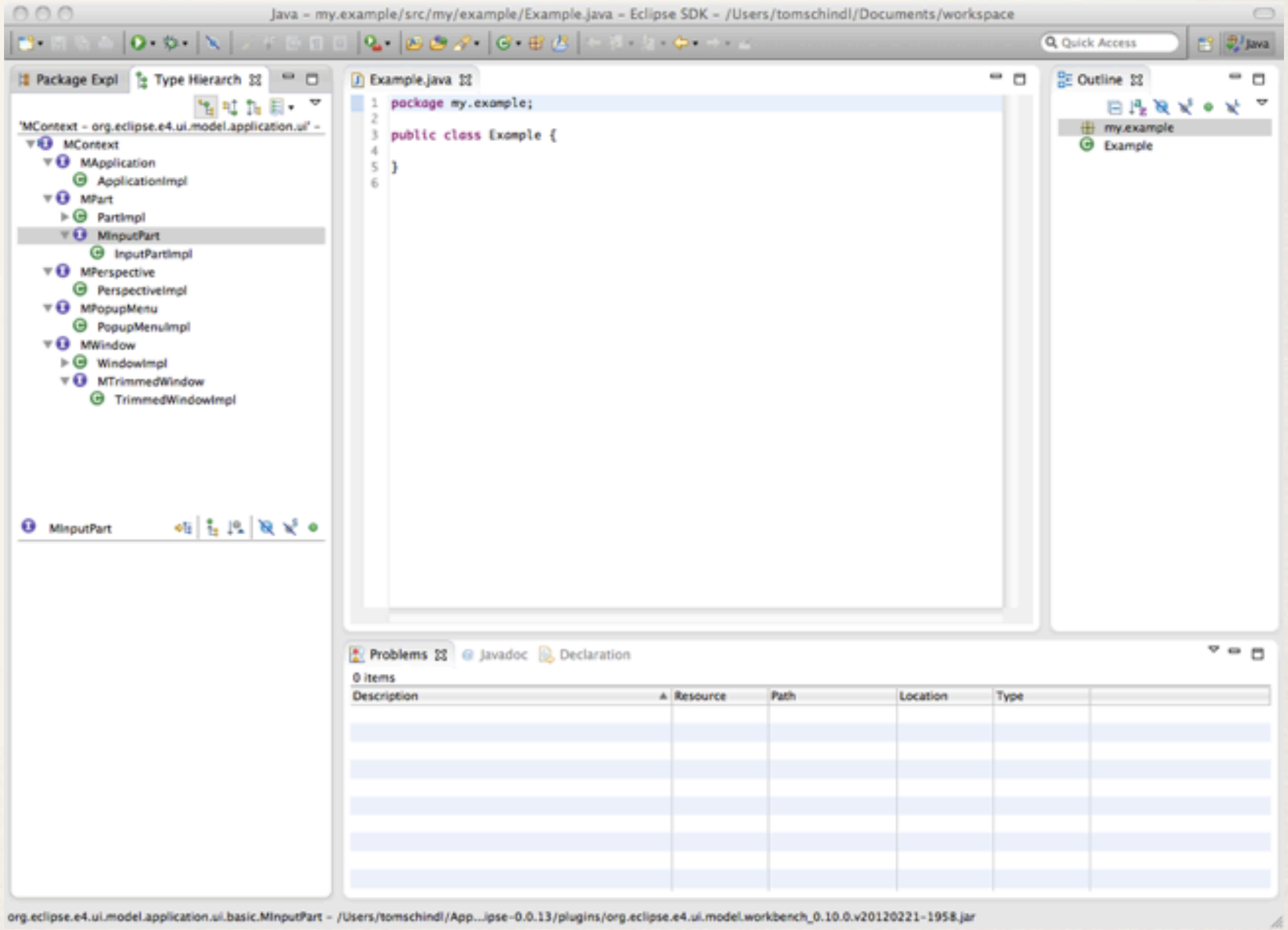
Application-Context

Window-Context

Application-Model



```
▶ Handlers
▶ Part Descriptors
▶ Commands
▶ Command Categories
▼ Windows
  ▼ Trimmed Window - %trimmedwindow.label.eclipseSDK
    ▼ Main Menu
      ▶ Menu - &File
      ▶ Menu - &Edit
      ▶ Menu <invisible> - &Source
      ▶ Menu <invisible> - Refac&tor
      ▶ Menu - &Navigate
      ▶ Menu - Se&arch
      ▶ Menu - &Project
        Separator
        Separator <invisible>
      ▶ Menu <invisible> - S&VN
      ▶ Menu <invisible> - C&VS
        Separator <invisible>
      ▶ Menu <invisible> - &Git
        Separator
      ▶ Menu - &Run
      ▶ Menu - &Window
      ▶ Menu - &Help
    Handlers
    Windows
    ▼ Controls
      ▼ PartSashContainer
        ▼ Perspective Stack
          ▼ Label - Java
              Windows
            ▼ Controls
              ▼ PartSashContainer
                ▶ Part Stack <invisible>
                ▼ PartSashContainer
                  ▶ PartSashContainer
                  ▶ Part Stack <invisible>
        ▶ ~~Part Stack <not rendered>~~
    ▶ Shared Elements
    ▶ TrimBars
▶ Menu Contributions
```



Java - my.example/src/my/example/Example.java - Eclipse SDK - /Users/tomschindl/Documents/workspace

```
'MContext - org.eclipse.e4.ui.model.application.ui' -
▼ MContext
  ▼ MApplication
      ApplicationImpl
  ▼ MPart
    ▶ PartImpl
    ▼ MInputPart
        InputPartImpl
  ▼ MPerspective
      PerspectiveImpl
  ▼ MPopupMenu
      PopupMenuImpl
  ▼ MWindow
    ▶ WindowImpl
    ▼ MTrimmedWindow
        TrimmedWindowImpl
```

MInputPart

```java
package my.example;

public class Example {

}
```

```
my.example
Example
```

0 items

Description	Resource	Path	Location	Type

org.eclipse.e4.ui.model.application.ui.basic.MInputPart - /Users/tomschindl/App...ipse-0.0.13/plugins/org.eclipse.e4.ui.model.workbench_0.10.0.v20120221-1958.jar

Workbench-Model

- Holds the whole application information
- It‘s a live model (similar to the Browser-DOM)
    - One can interact with it
- Acts as the glue between the structure and your custom Java-Code

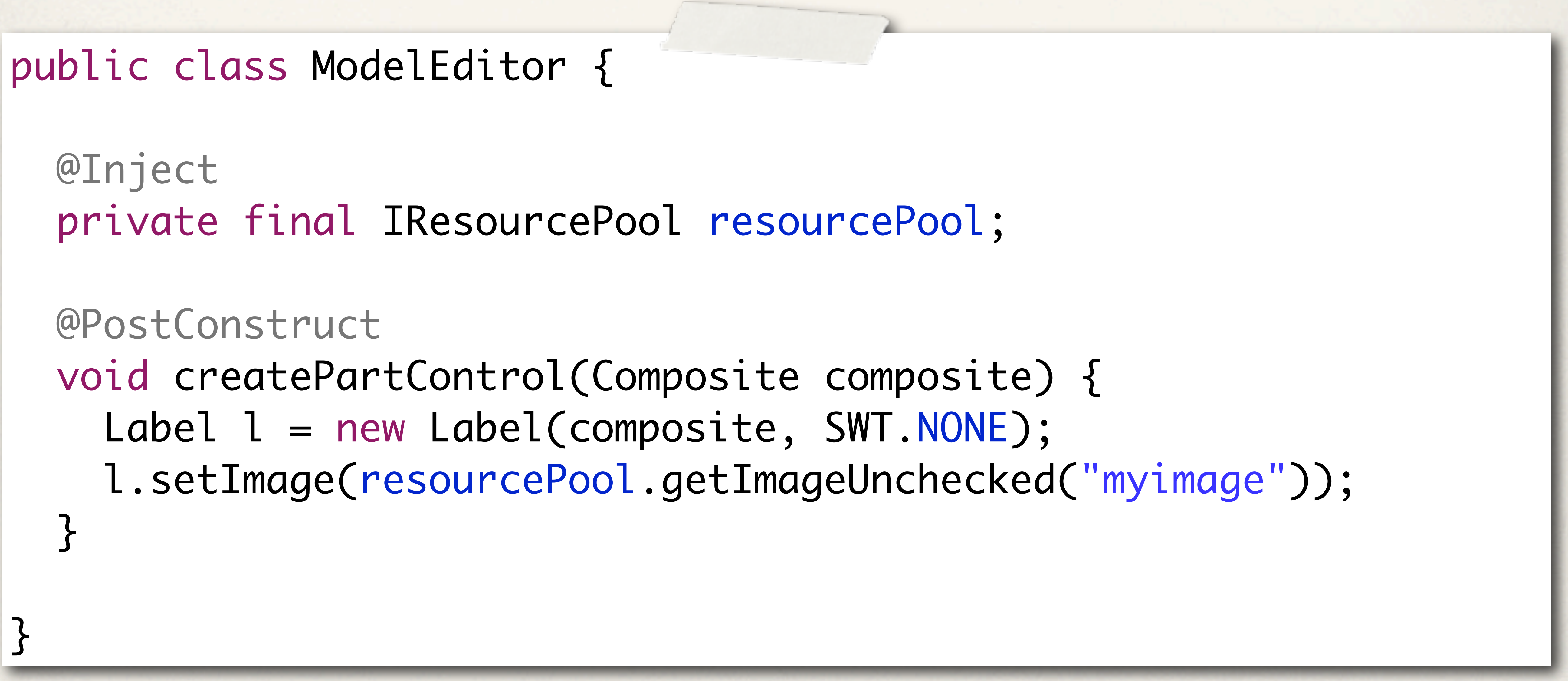
2 typical problems in RCP

- Native Resource sharing / managing
- Support locales in your application
  - NLS
  - ResourceBundle

Resource management and sharing

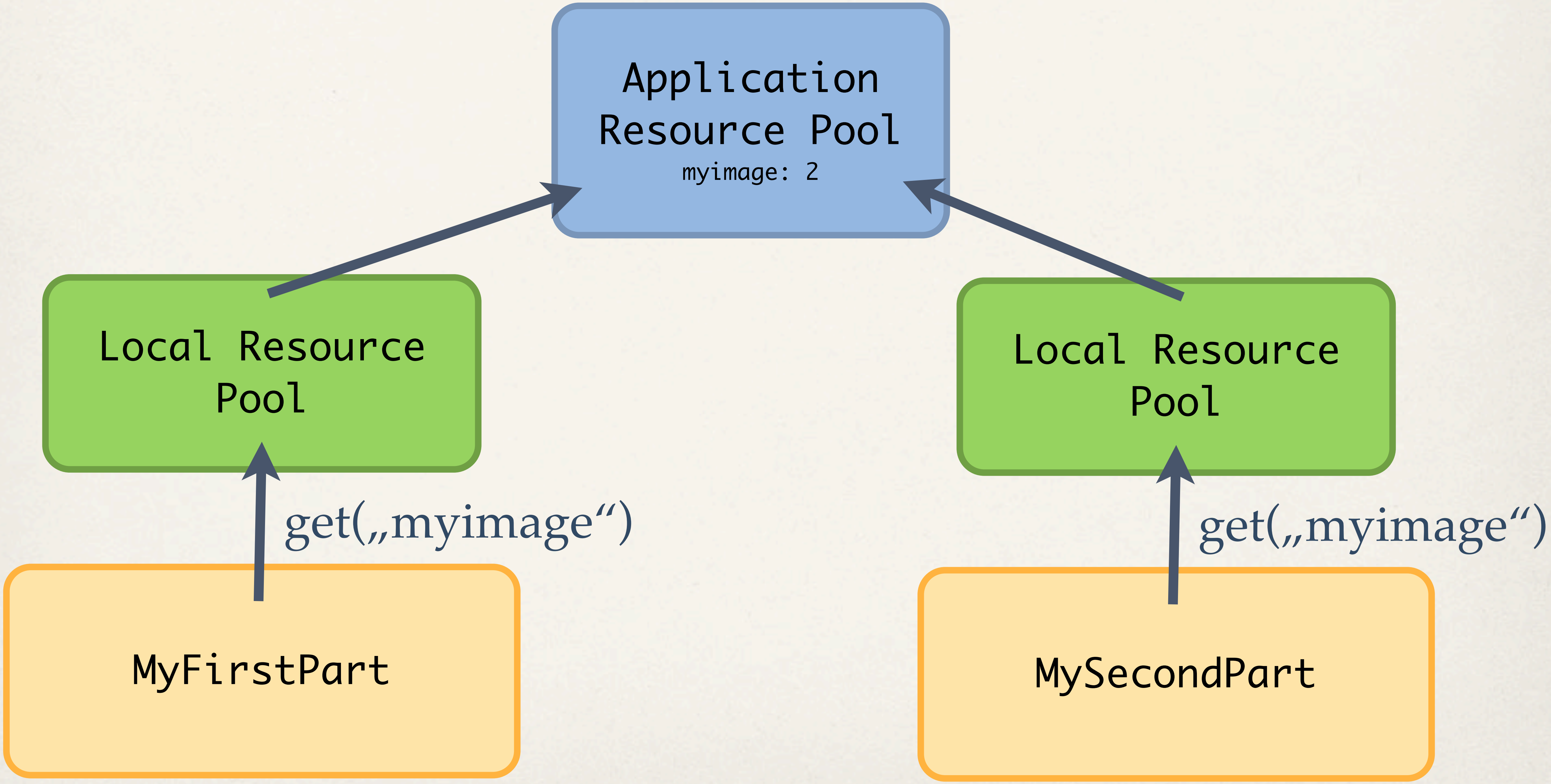
```java
public class ModelEditor {

  @PostConstruct
  public void createPartControl(Composite composite) {
    Label l = new Label(composite, SWT.NONE);
    l.setImage(
      Activator.imageDescriptorFromPlugin(
      Activator.PLUGIN_ID, "/resource/myimage.png").createImage()
    );
  }
}
```

Resource management and sharing

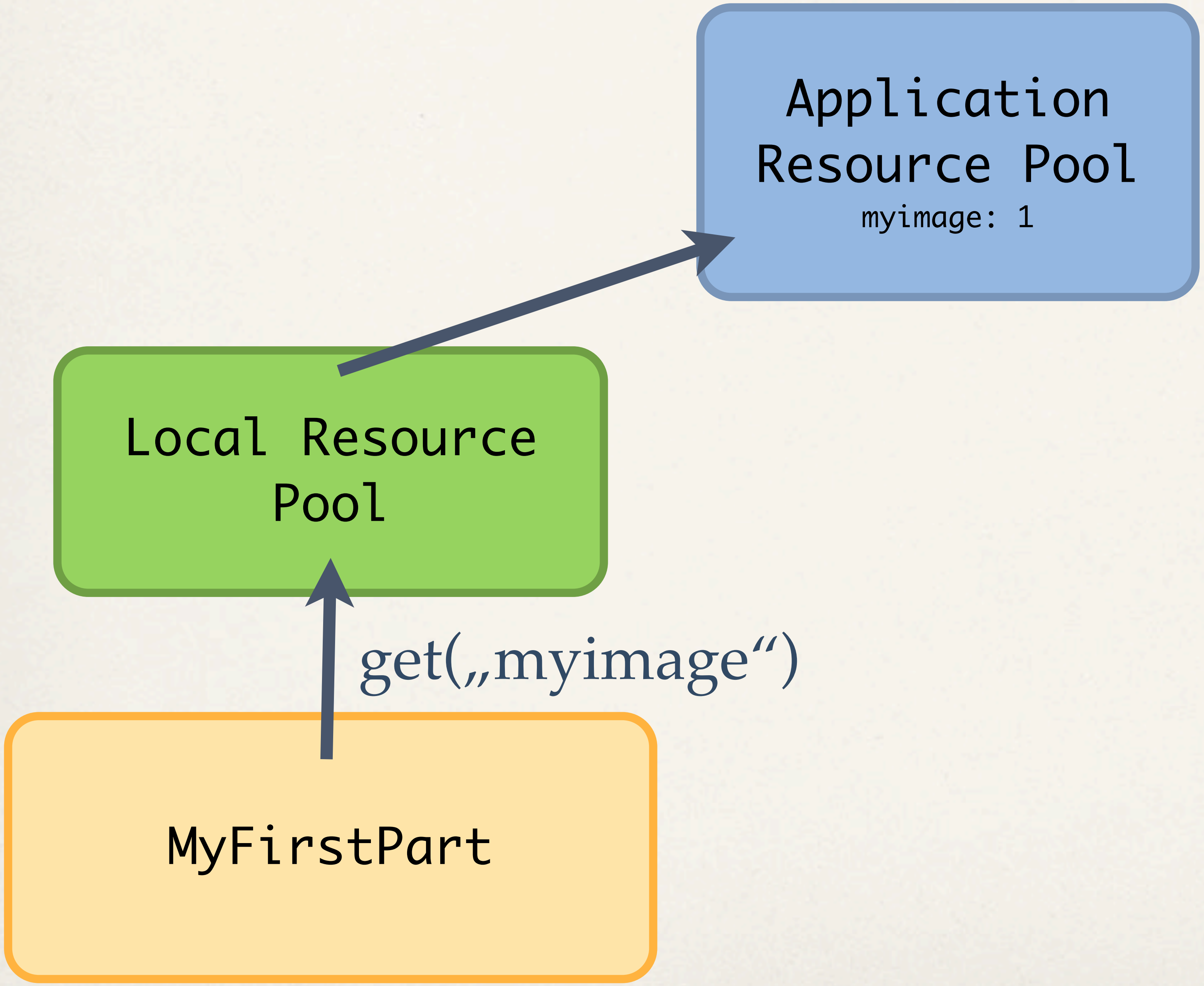


```
public class ModelEditor {

  @Inject
  private final IResourcePool resourcePool;

  @PostConstruct
  void createPartControl(Composite composite) {
    Label l = new Label(composite, SWT.NONE);
    l.setImage(resourcePool.getImageUnchecked("myimage"));
  }

}
```

Resource management and sharing



Resource management and sharing



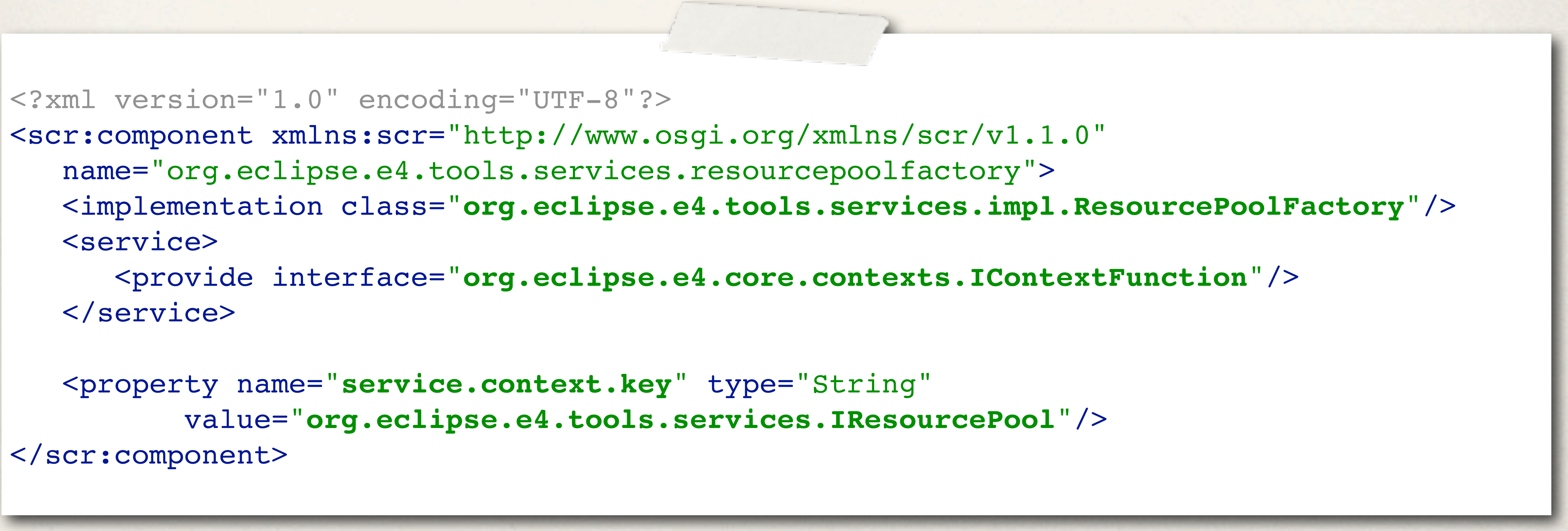
Resource management and sharing

Application Resource Pool

myimage: 0

Context Functions

- Allows to lazily create the injection instance
- Contributed as an OSGi-Service



```xml
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/v1.1.0"
   name="org.eclipse.e4.tools.services.resourcepoolfactory">
   <implementation class="org.eclipse.e4.tools.services.impl.ResourcePoolFactory"/>
   <service>
      <provide interface="org.eclipse.e4.core.contexts.IContextFunction"/>
   </service>

   <property name="service.context.key" type="String"
          value="org.eclipse.e4.tools.services.IResourcePool"/>
</scr:component>
```

Context Functions

```java
public class ResourcePoolFactory extends ContextFunction {

  @Override
  public Object compute(IEclipseContext context) {
    return
     ContextInjectionFactory.make(ResourcePool.class, context);
  }

}
```

Context Functions

```
class ResourcePool implements IResourcePool {

  public Image getImage(String key) {
     // load image or increment refcount

  }

  @PreDestroy
  public void dispose() {
    // decrease refcount of images loaded
  }

}
```

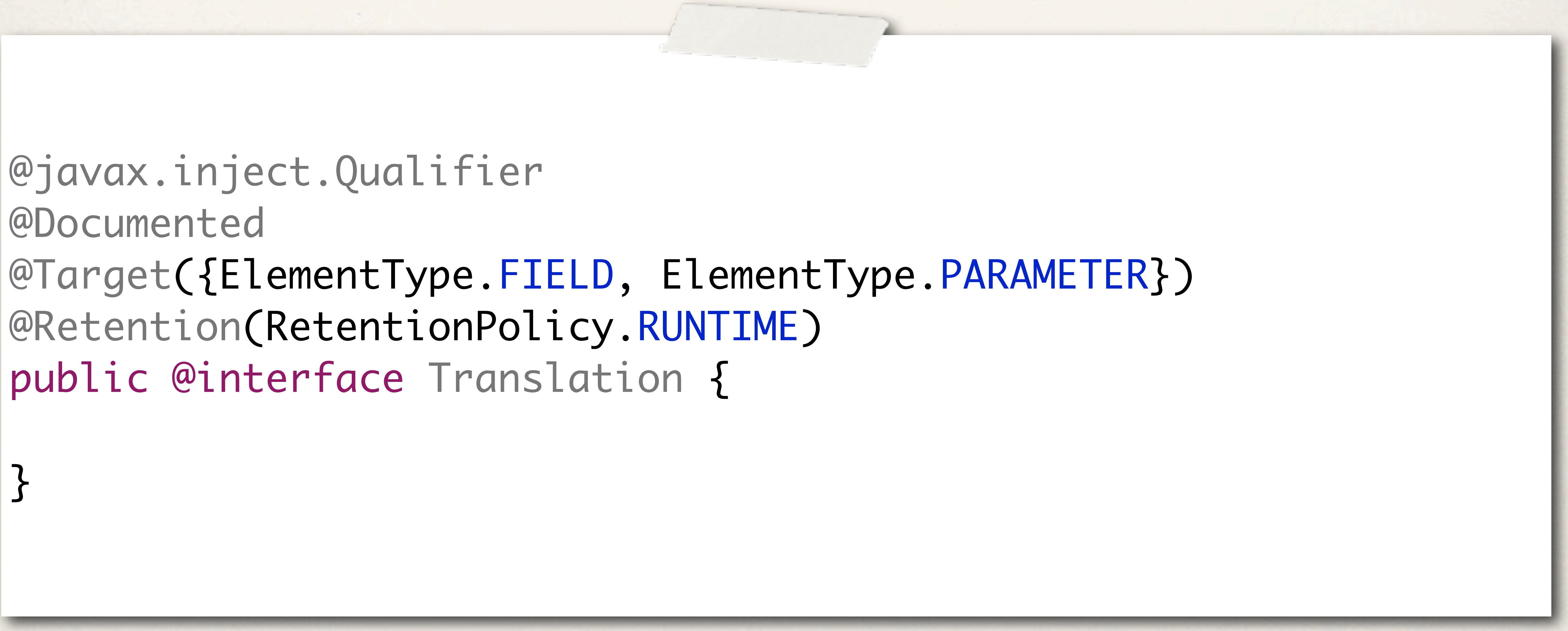
Locale Support

```java
public class Messages {
  public static String MyLabel;

  NLS.initializeMessages(Messages.class.getName(), Messages.class);
}


public class ModelEditor {




  @PostConstruct
  public void createPartControl(Composite composite) {
    Label l = new Label(composite, SWT.NONE);
    l.setText(Messages.MyLabel);
  }

}
```

Locale Support

```
public class Messages {
  public String MyLabel;



}



public class ModelEditor {

  @Inject
  @Translation
  Messages Messages;

  @PostConstruct
  public void createPartControl(Composite composite) {
    Label l = new Label(composite, SWT.NONE);
    l.setText(Messages.MyLabel);
  }

}
```

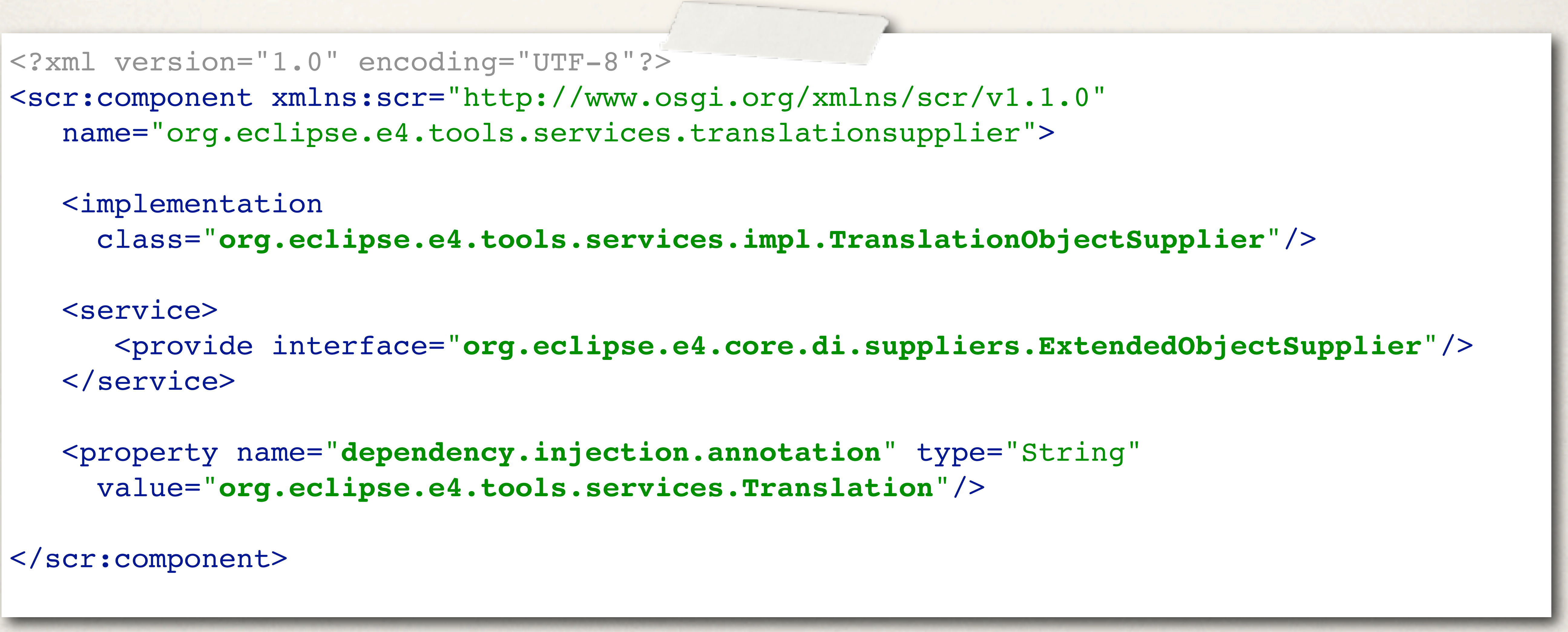
Create your own annotations



```
@javax.inject.Qualifier
@Documented
@Target({ElementType.FIELD, ElementType.PARAMETER})
@Retention(RetentionPolicy.RUNTIME)
public @interface Translation {

}
```

Teach Eclipse DI the new annotation

- Contributed through DS



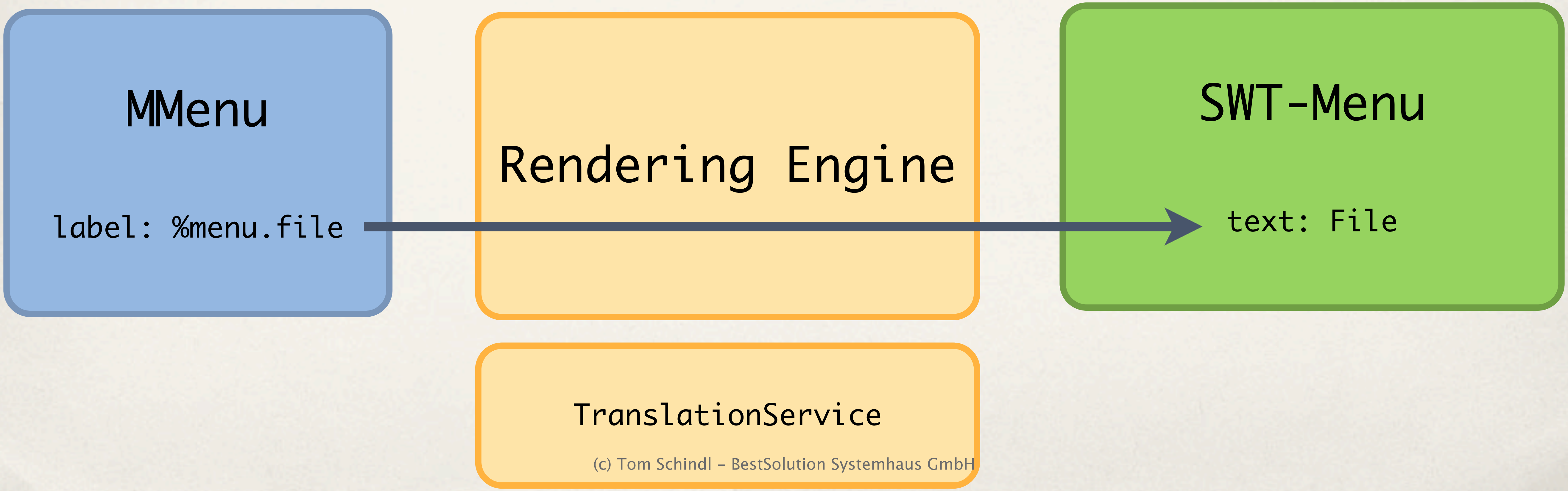
```xml
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/v1.1.0"
   name="org.eclipse.e4.tools.services.translationsupplier">

   <implementation
     class="org.eclipse.e4.tools.services.impl.TranslationObjectSupplier"/>

   <service>
      <provide interface="org.eclipse.e4.core.di.suppliers.ExtendedObjectSupplier"/>
   </service>

   <property name="dependency.injection.annotation" type="String"
     value="org.eclipse.e4.tools.services.Translation"/>

</scr:component>
```

The object supplier

```java
public class TranslationObjectSupplier extends ExtendedObjectSupplier {

  @Override
  public Object get(IObjectDescriptor descriptor, IRequestor requestor,
          boolean track, boolean group) {


  }
}
```

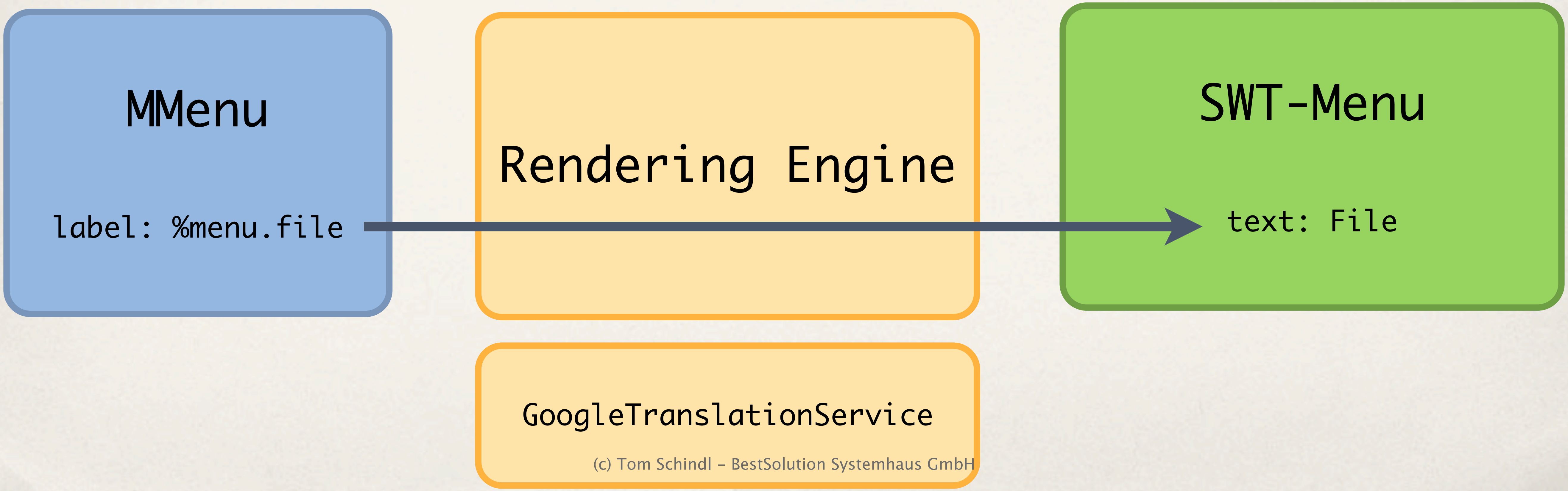
Eclipse 4.1 Application Platform

- Support for locales in Application Model
    - Translations are implement as a decoration which at least in theory supports dynamic language switching



Eclipse 4.1 Application Platform

- Support for locales in Application Model
  - Translations are implement as a decoration which at least in theory supports dynamic language switching



MMenu

label: %menu.file

Rendering Engine

SWT-Menu

text: File

GoogleTranslationService

Resources

- My blog: http://tomsondev.bestsolution.at
- e4-Wiki: http://wiki.eclipse.org/e4
- e4-newsgroup: eclipse.e4
- e4-mailinglist: e4-dev@eclipse.org
- Twitter: @tomsontom