Eclipse 4 Application Platform

Tom Schindl - BestSolution Systemhaus GmbH

EclipseCon Nov 2011

About Tom

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

2 typical problems in RCP

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

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

```
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

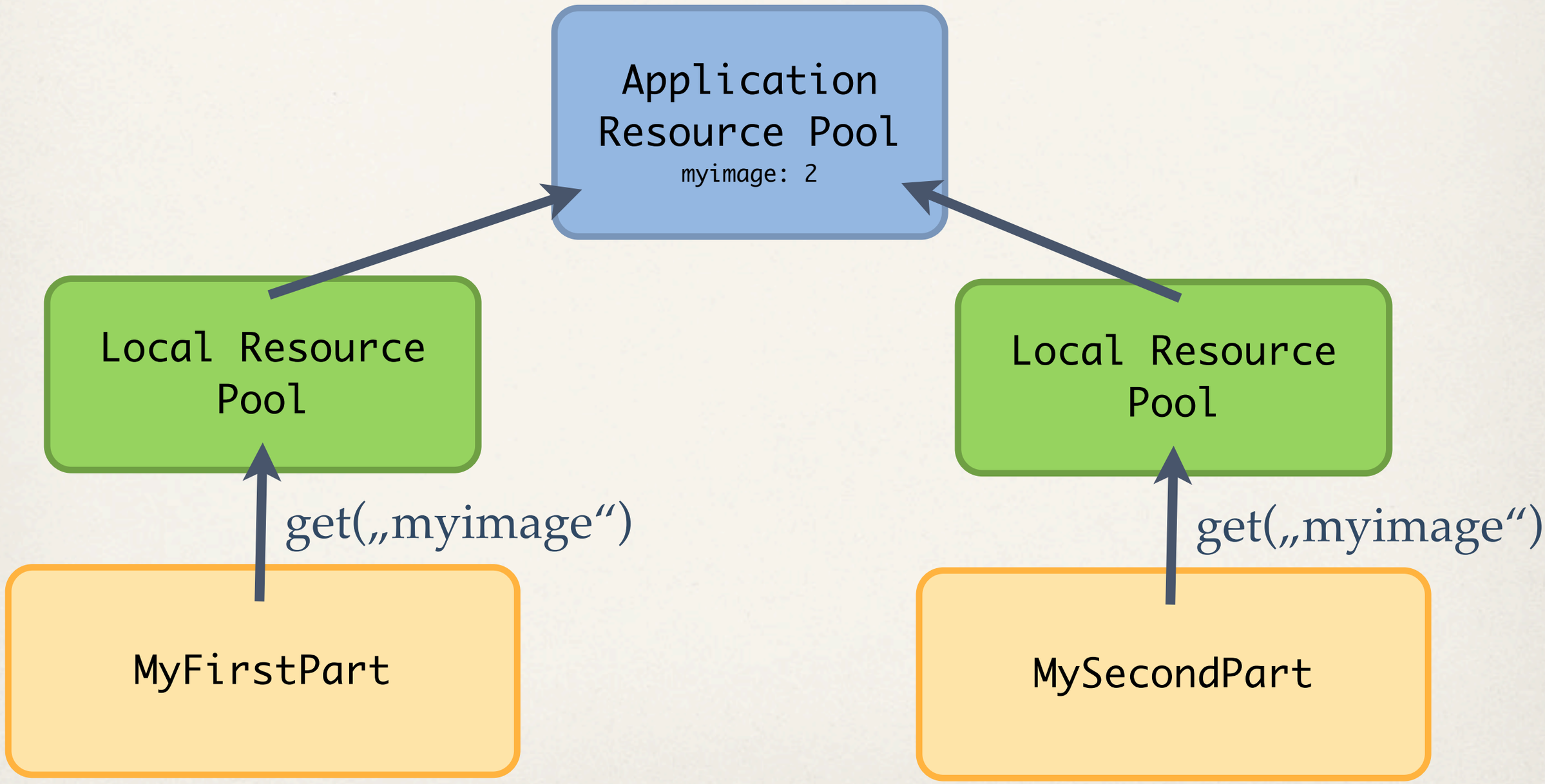
Resource management and sharing

```
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()
    );
  }
}
```

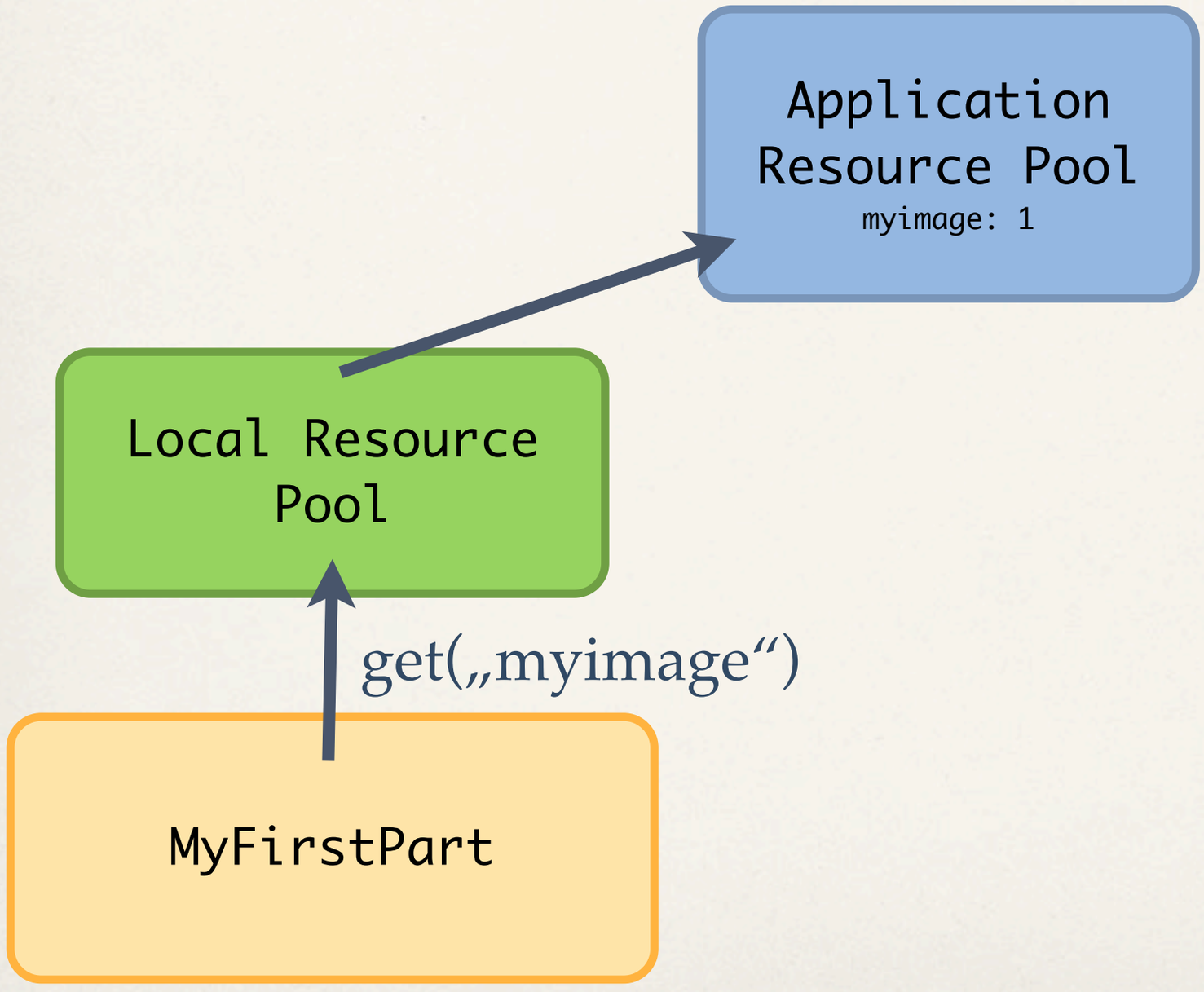
Resource management and sharing

```
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

Dependency Injection

```
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);
    }
}
```

Context Functions

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

```
<?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

```
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

```
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

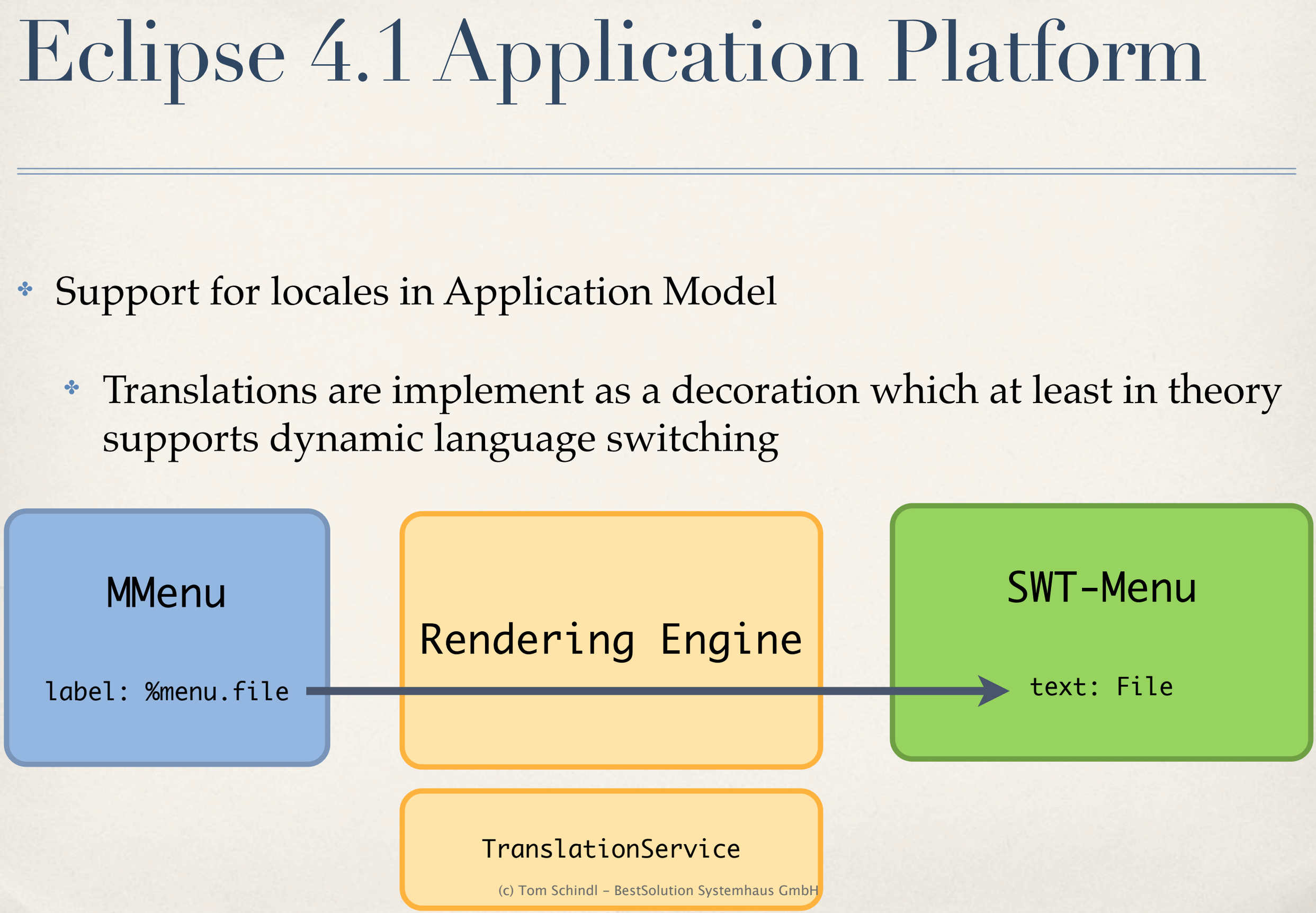
```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/v1.1.0"
   name="org.eclipse.e4.tools.services.translationsupplier">

   <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

```
public class TranslationObjectSupplier extends ExtendedObjectSupplier {

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


  }
}
```

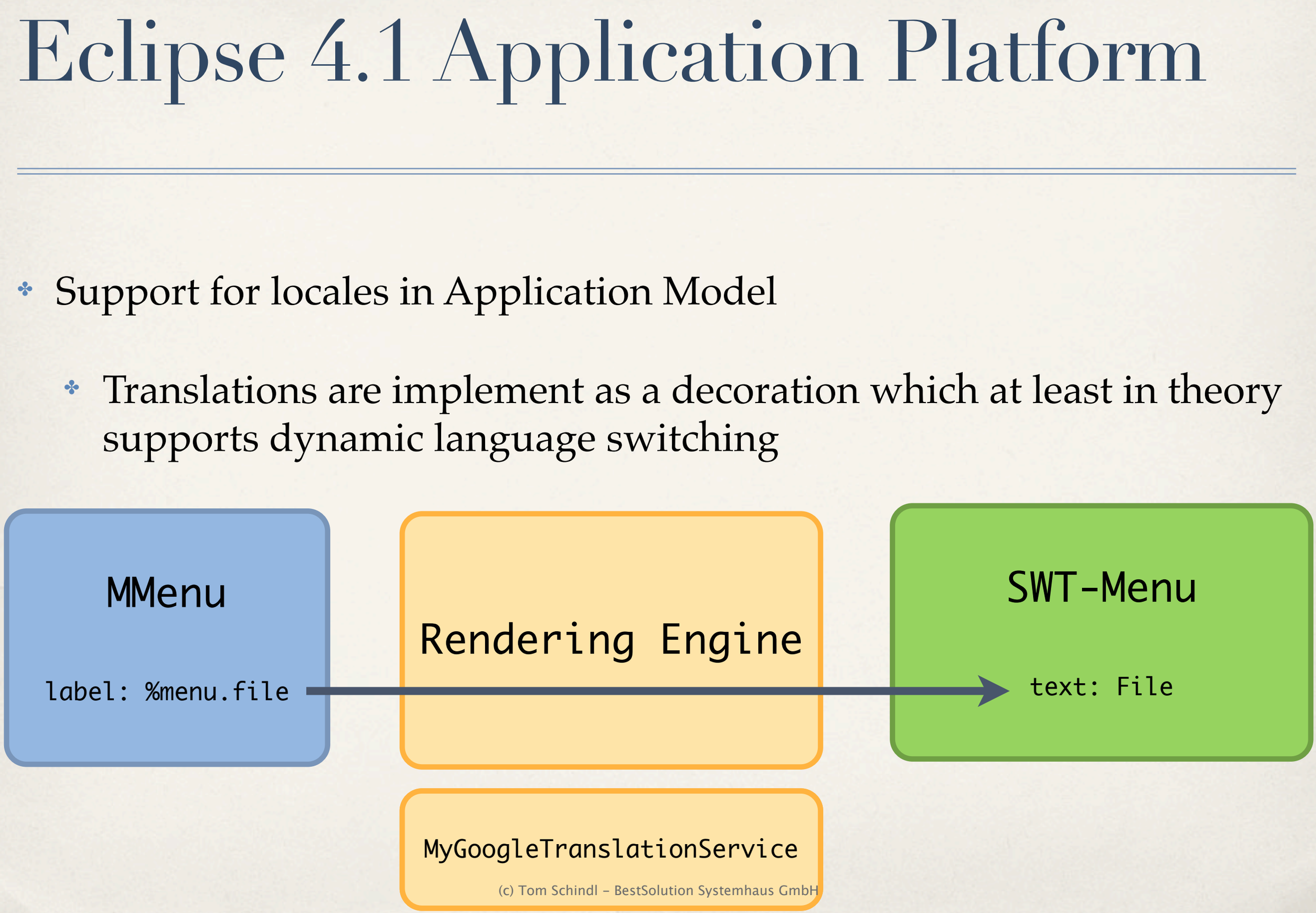
Eclipse 4.1 Application Platform

- 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



Introducing the e4-bridge

- Provides a DI-Container inside the 3.x workbench

```
public class IconViewPart extends DIViewPart<MyPart> {

   public IconViewPart() {
      super(MyPart.class);
   }
}
```

Resources

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