



Java Development with the KMC Platform



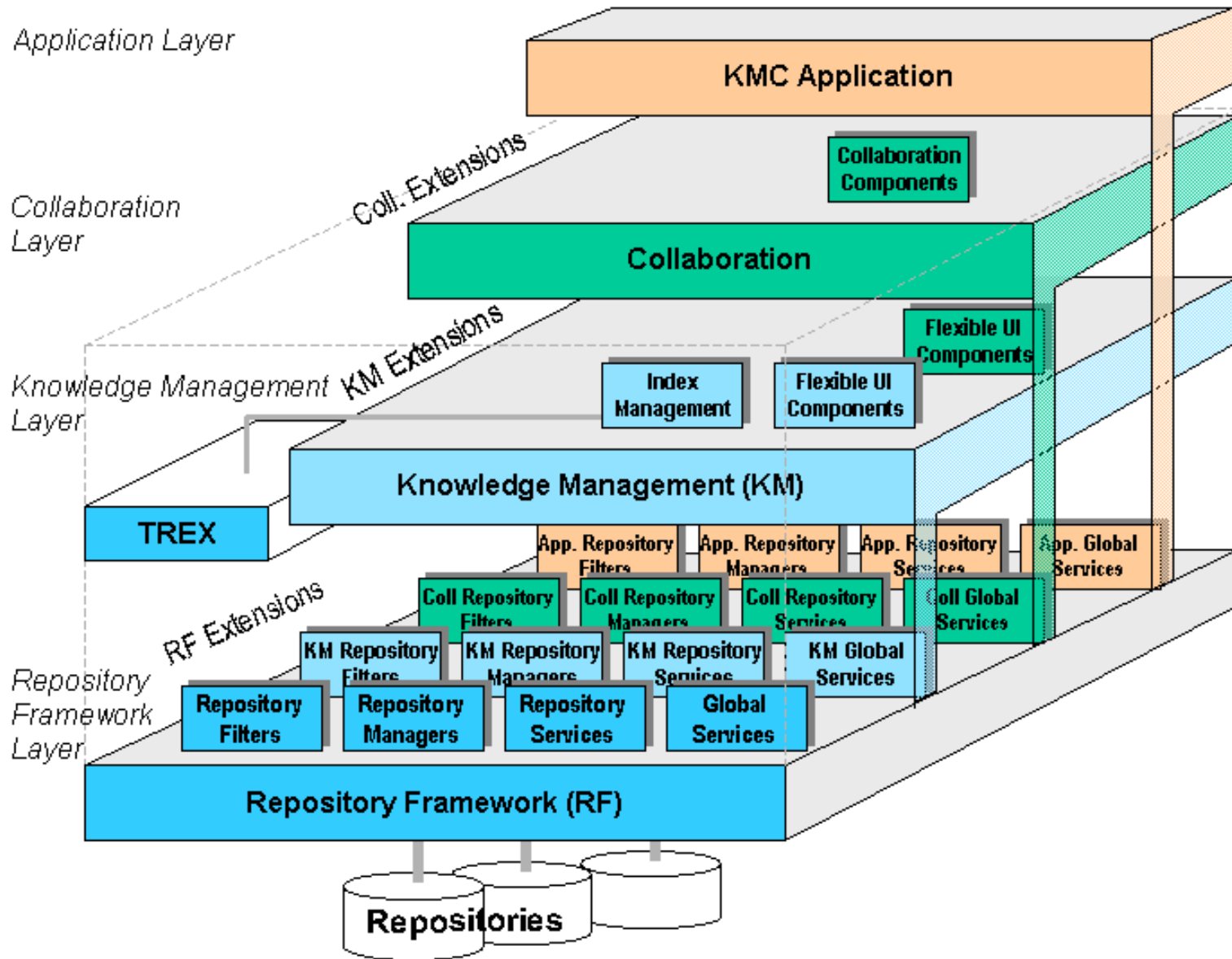
Thilo Brandt, SAP AG



KMC architecture for developers

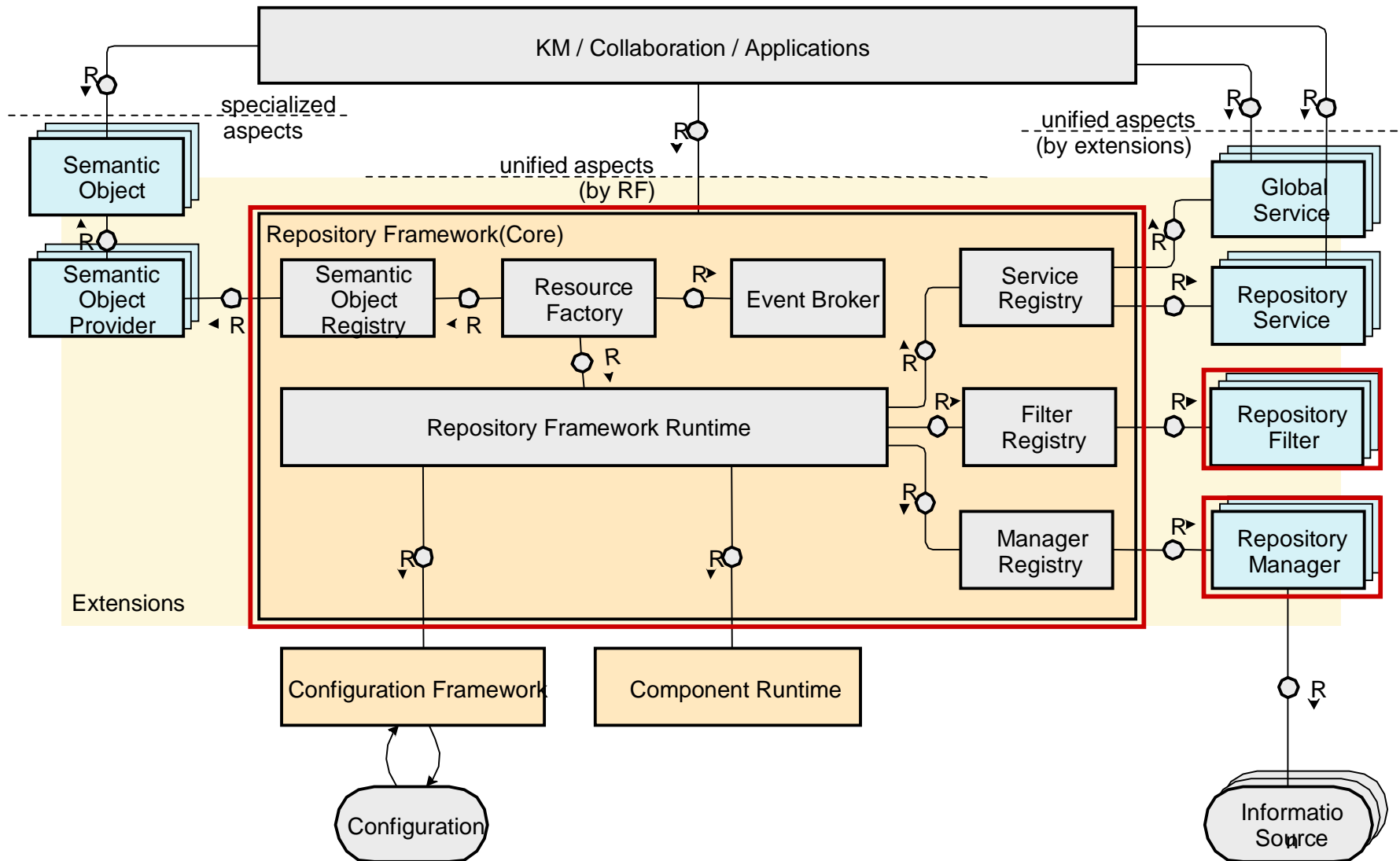
SAP NetWeaver Developer Studio for KMC

KMC Architecture For Developers



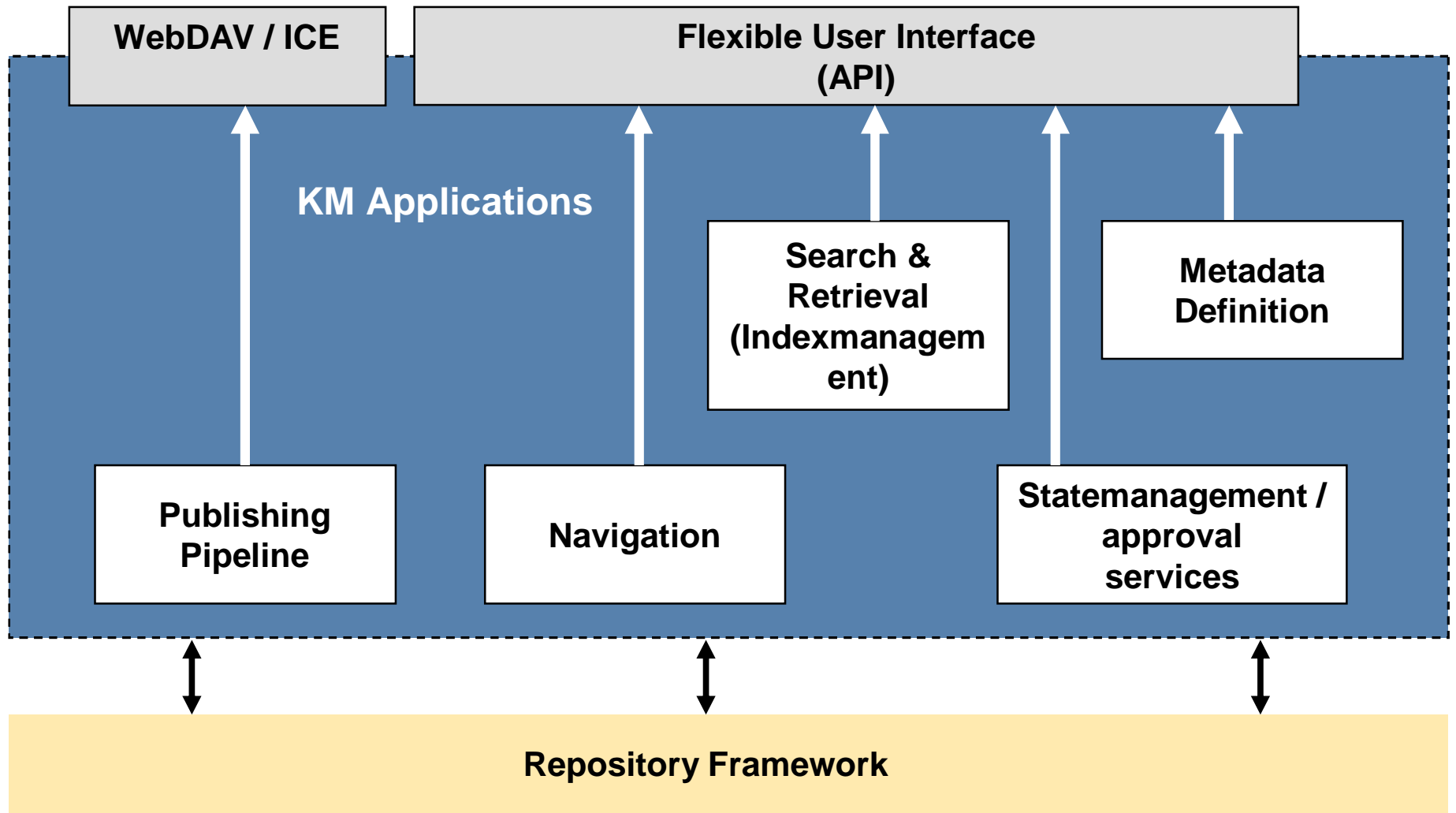
KMC Architecture For Developers

Repository Framework architecture



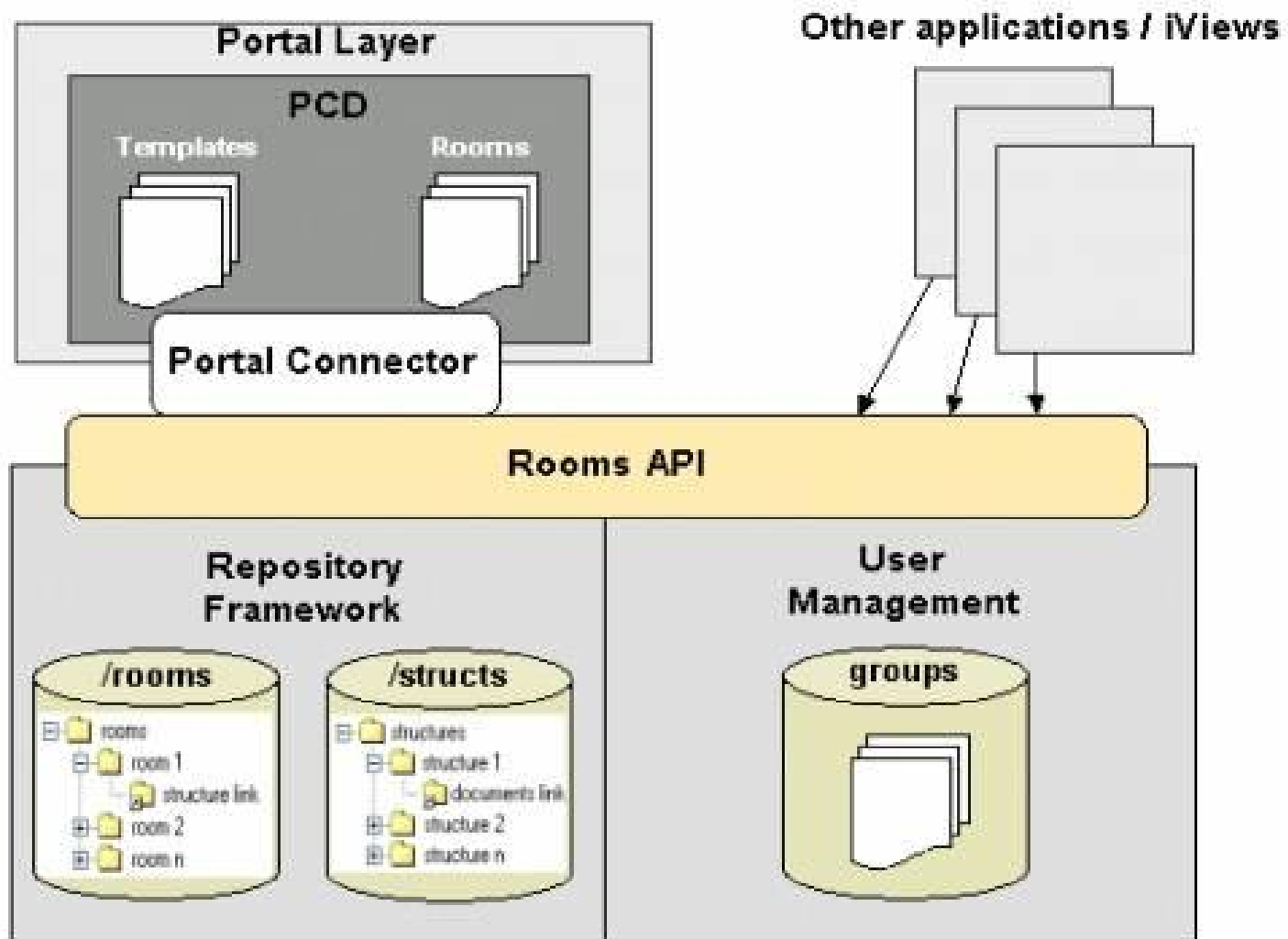
KMC Architecture For Developers

Knowledge Management architecture

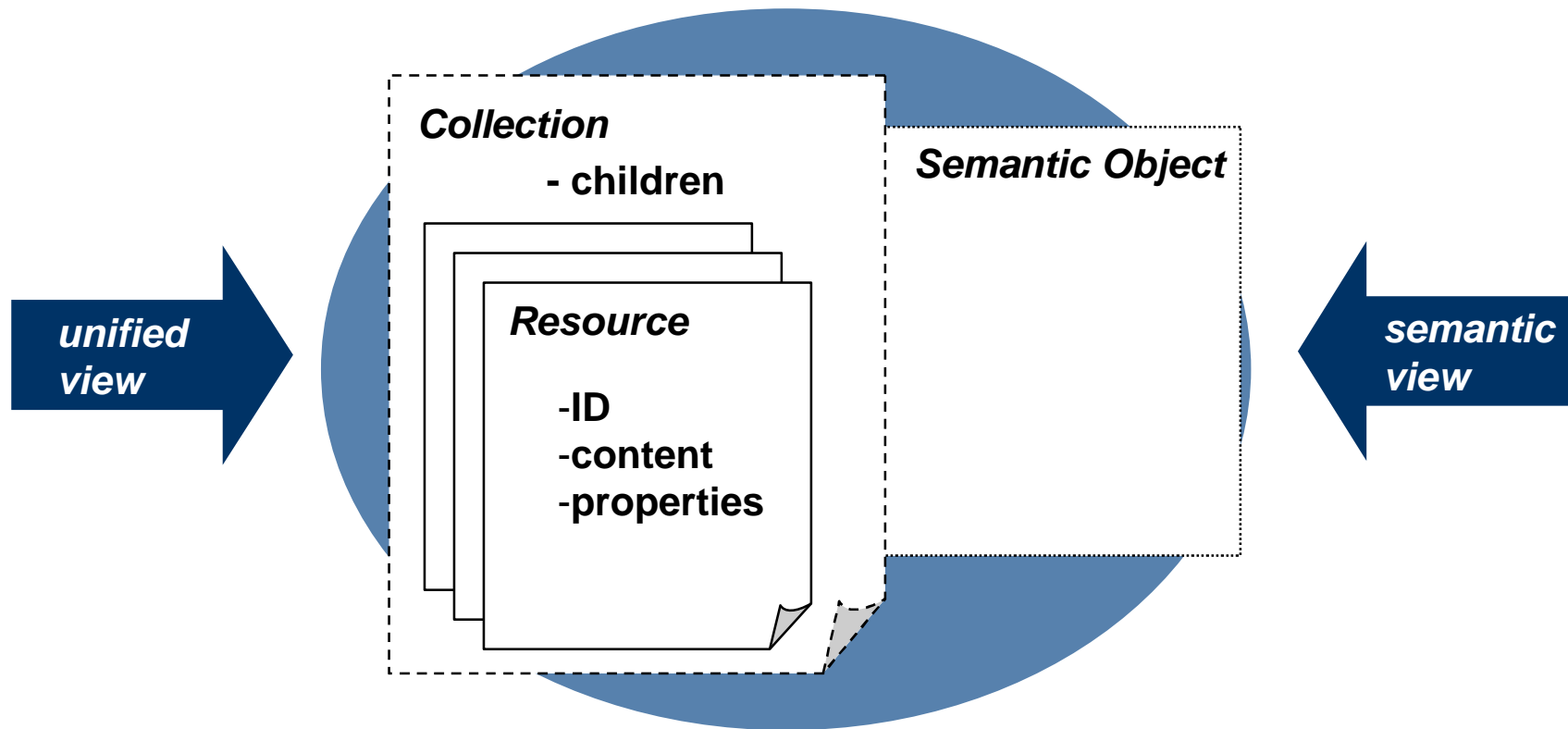


KMC Architecture For Developers

Collaboration architecture

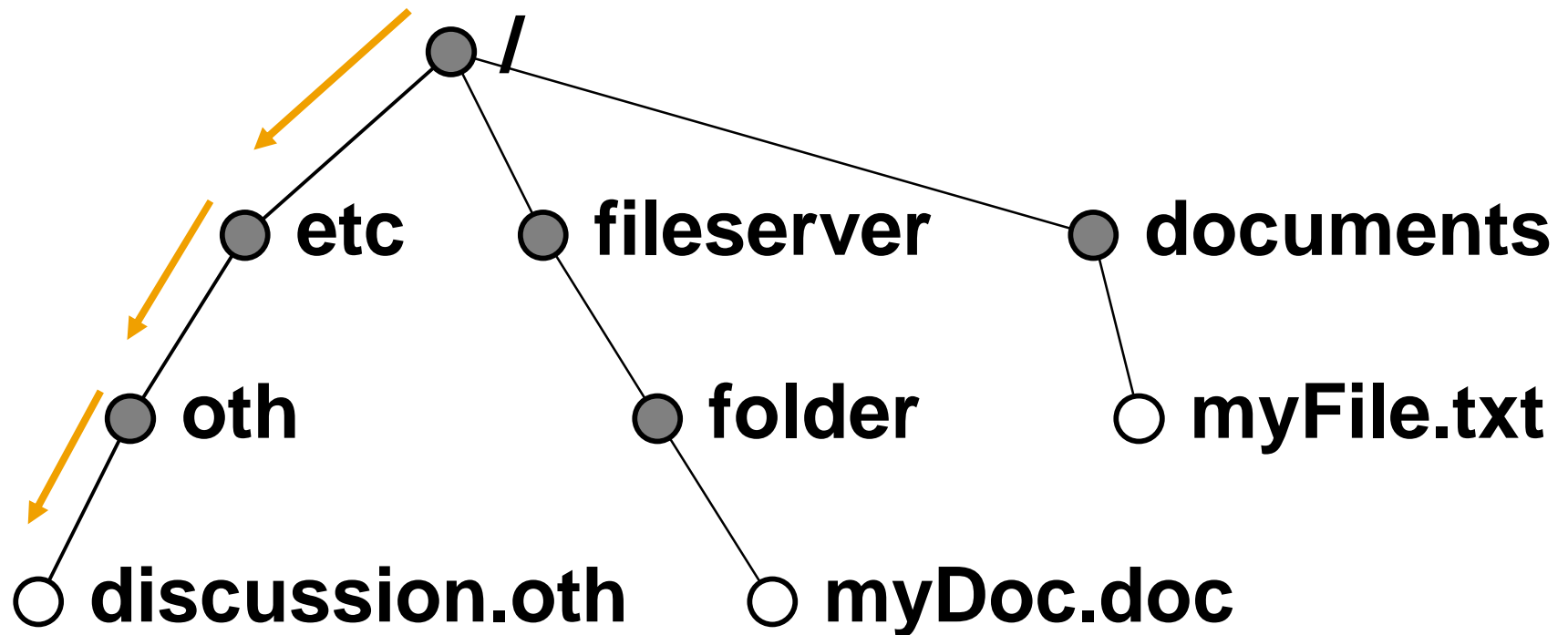


Repository Framework Concepts – Objects



Repository Framework Concepts – Objects

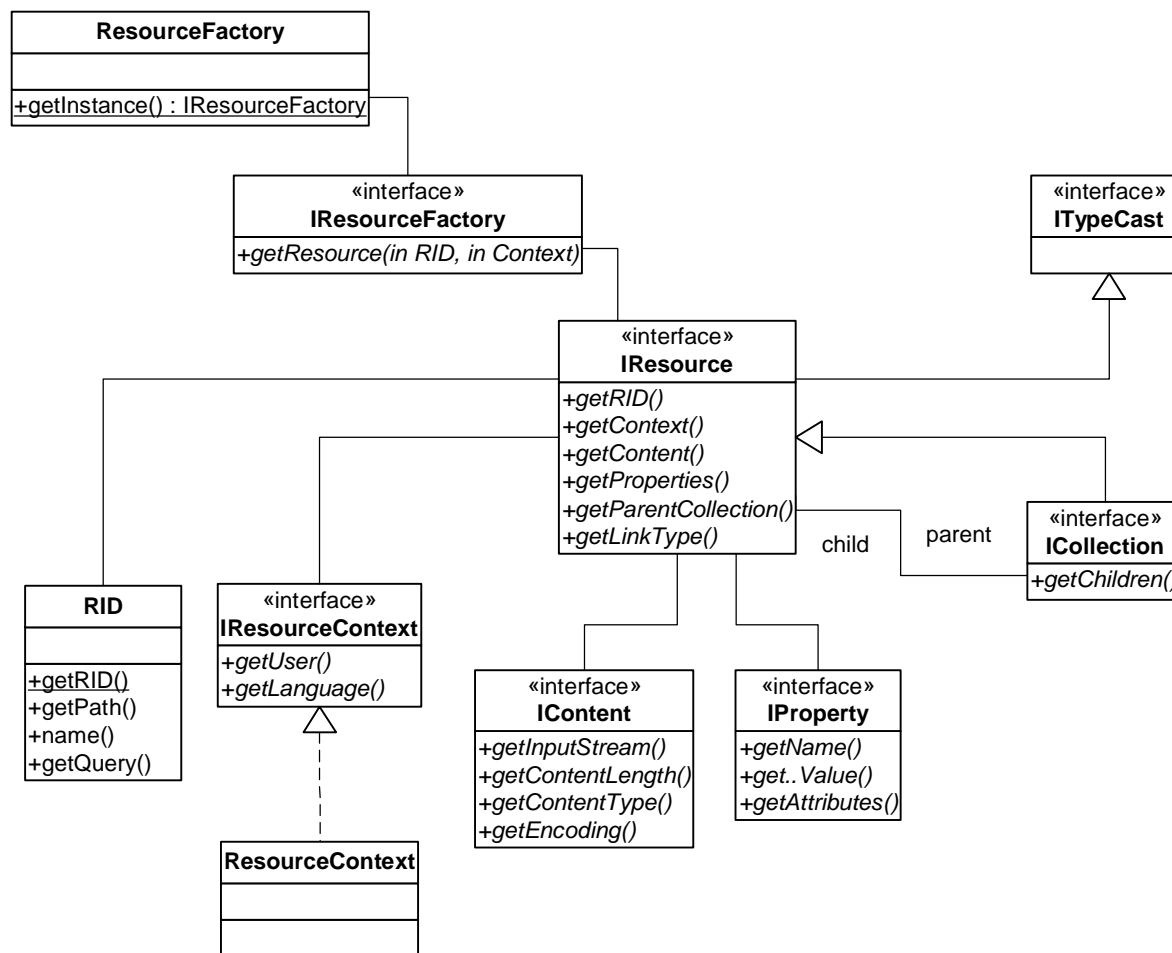
Objects hierarchy and addressability



Example RID: /etc/oth/discussion.oth

Repository Framework Concepts – Objects

Class diagram of RF objects



Repository Framework Concepts – Objects

Sample coding for retrieving RF objects

```
IUser user = ...;  
  
IResourceFactory factory =  
    ResourceFactory.getInstance();  
  
RID rid = RID.getRID("/etc");  
  
IResourceContext context = new ResourceContext(user);  
  
IResource resource =  
    factory.getResource(rid, context);
```

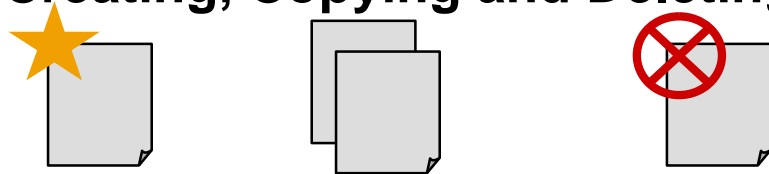
Repository Framework Concepts – Objects

Sample coding for checking for collections and retrieving ist children elements

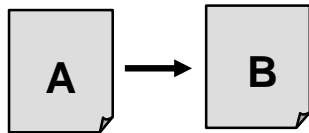
```
if(resource.isCollection()) {  
    ICollection collection = (ICollection)resource;  
  
    IResourceList children =  
        collection.getChildren();  
}
```

Repository Framework Concepts – Basic Actions

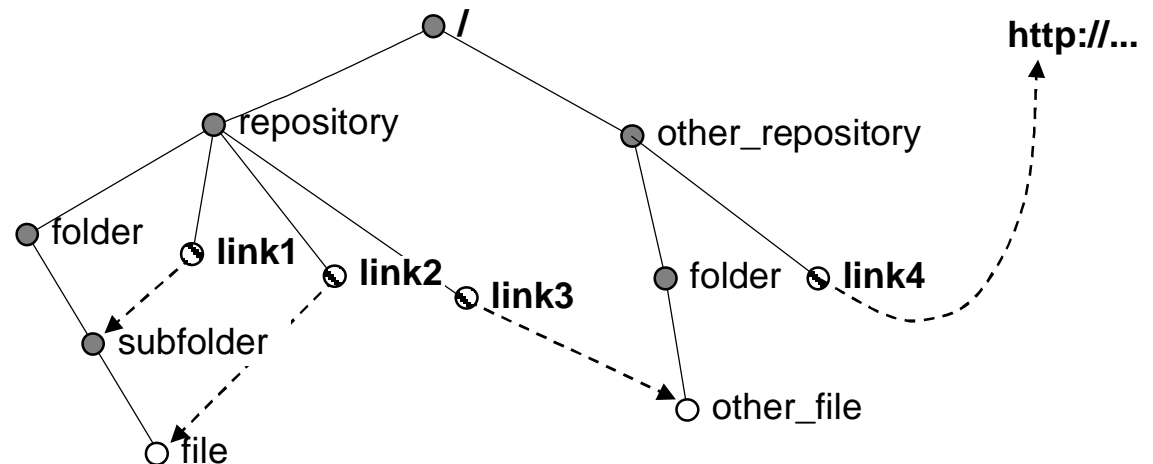
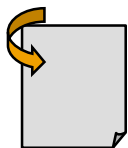
n Creating, Copying and Deleting



n Renaming and Moving



n Links



Repository Framework Concepts – Basic Actions

Sample coding for creating new resource objects

```
IResourceFactory factory =  
    ResourceFactory.getInstance();  
  
IResourceContext context =  
    new ResourceContext(user);  
  
ICollection parent =  
    (ICollection)factory.getResource("/documents",  
                                    context);  
  
IResource resource =  
    parent.createResource("file", null, null);
```

Repository Framework Concepts – Basic Actions

Sample coding for creating new collection objects

```
IResourceFactory factory =  
    ResourceFactory.getInstance();  
  
IResourceContext context =  
    new ResourceContext(user);  
  
ICollection parent =  
    (ICollection)factory.getResource("/documents",  
                                    context);  
  
ICollection collection =  
    parent.createCollection("folder", null);
```

Repository Framework Concepts – Basic Actions

Sample coding for copy, move, rename and delete a resource

```
IResource copied =  
    resource.copy( "/documents/copy", null );  
  
IResource moved =  
    copied.move( "/documents/folder", null );  
  
IResource renamed =  
    moved.rename( "renamed_copy" );  
  
renamed.delete();
```


Repository Framework Concepts – Basic Actions

Sample coding for checking for links

```
if(LinkType.EXTERNAL.equals(resource.getLinkType())) {  
    // external link to an URL  
    URL target = resource.getTargetURL();  
} else if  
(LinkType.INTERNAL.equals(resource.getLinkType())) {  
    // internal link to another resource  
    IResource target = resource.getTargetResource();  
} else {  
    // not a link  
}
```

Repository Framework Concepts – Basic Actions

Sample coding for content creating and retrieval

```
String out = new String("my content");

ByteArrayInputStream data = new
    ByteArrayInputStream(out.getBytes());

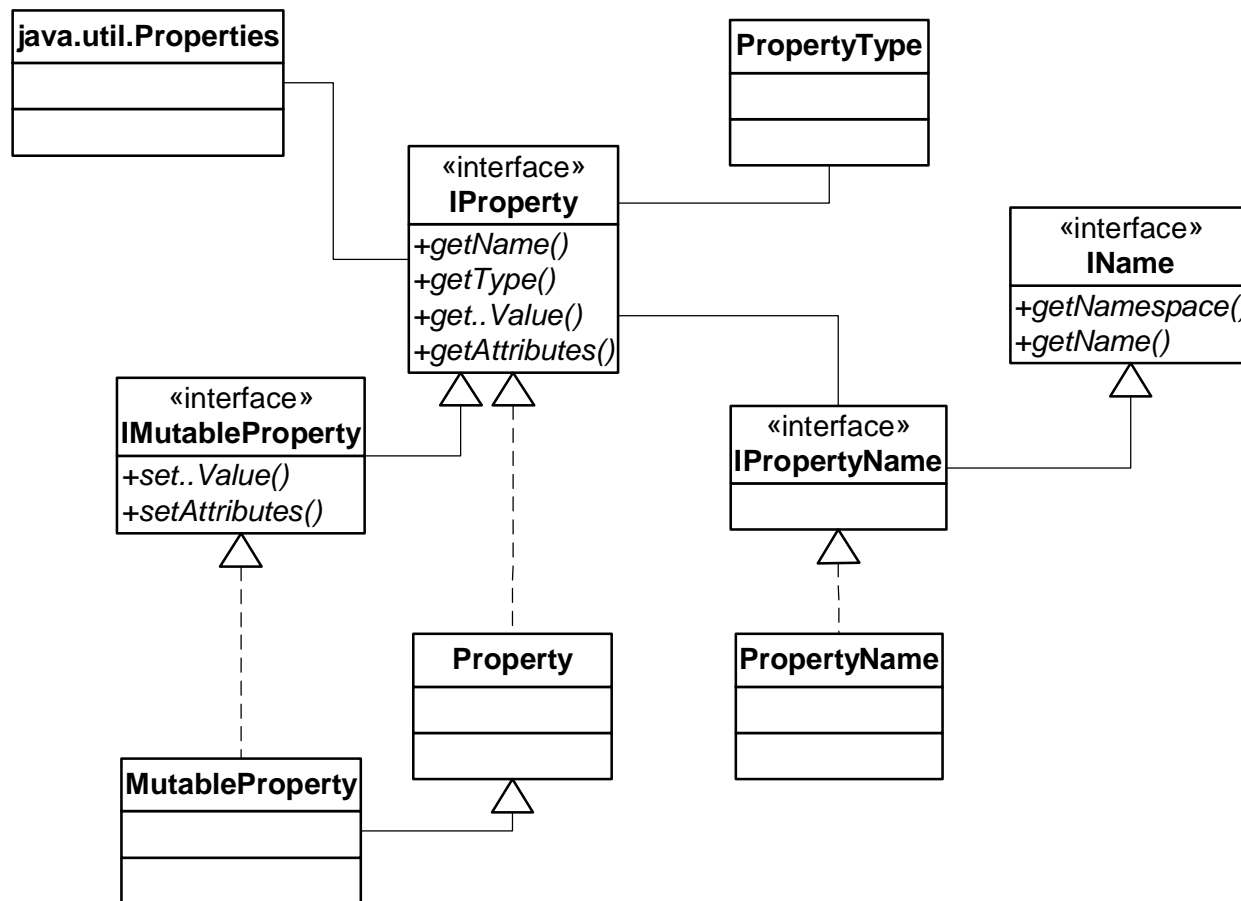
IContent newContent =
    new Content(data, "text/plain",
                data.available());

resource.updateContent(newContent);

IContent oldContent = resource.getContent();
```

Repository Framework Concepts – Basic Actions

Class diagram for metadata handling (properties)



Repository Framework Concepts – Basic Actions

Sample coding for metadata retrieval

```
String namespace = "http://sample.com/xmlns/sample";  
String name = "property";  
  
IPropertyName propertyName =  
    new PropertyName(namespace, name);  
  
IProperty property =  
    resource.getProperty(propertyName);  
  
if( property != null ) {  
    // property exists  
    String value = property.getValueAsString();  
}
```

Repository Framework Concepts – Advanced concepts

n **Eventing**

n **Security**

n **Locking**

n **ID Mapping**

n **Ordering**

n **Property Search**



KMC architecture for developers

SAP NetWeaver Developer Studio for KMC

THE BEST-RUN BUSINESSES RUN SAP



Requirements

n Install a local JDK

n Install a local SAP NetWeaver Developer Studio

u <http://service.sap.com/instguidesNW04> -> Installation -> Installation Guide -> SAP NetWeaver Developer Studio)

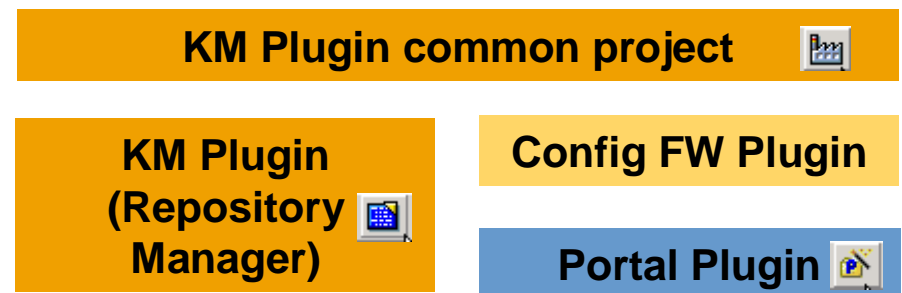
n Remote or local SAP Enterprise Portal with KMC

Portal-Plugins

- n **Creates an empty portal application project**
 - u **Creates source development folders for API & Core coding**
 - u **Creates a filesystem structure for PAR file creation**
 - l **Public area for client content**
 - l **Private area for server side content**
 - u **Creates a deployment descriptor file (portalapp.xml)**
- n **Config Framework plugin**
- n **Support for Portal specific components (portal services, portal components)**
- n **Support for WebServices**

KM-Plugins

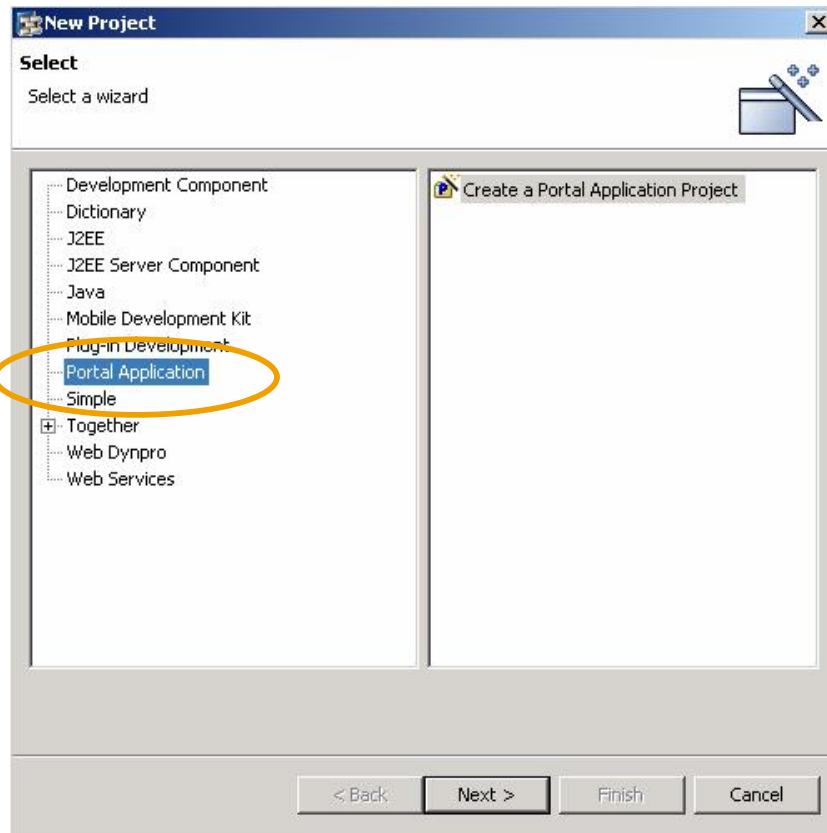
- n **Creates source code skeletons for CM components**
- n **Creates configuration entries (XML files) for config framework**
- n **Creates Wrapper classes for registration at CrtClassLoader**



SAP NetWeaver Developer Studio For KMC

Create a new Portal Application project (1)

n From the menu, choose: **File** -> **New ...** -> **Project**



Or choose the appropriate icon in the toolbar:



SAP NetWeaver Developer Studio For KMC

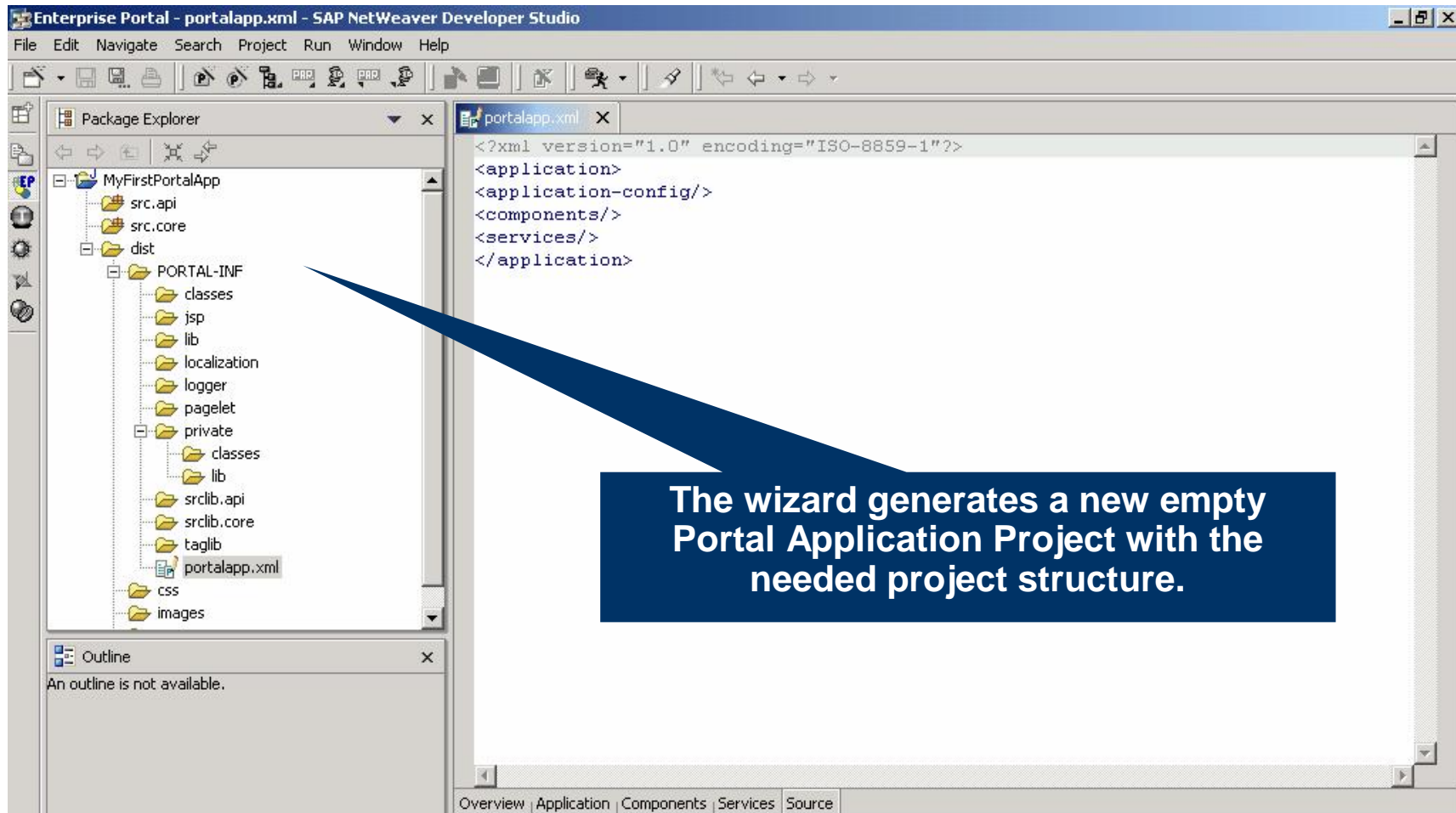
Create a new Portal Application project (2)

Define a **project name** and the **root folder** for your project:

The screenshot shows a 'New' dialog box for creating a 'Portal Application Project'. The 'Project name' field contains 'MyFirstPortalApp' and the 'Project root folder' field contains 'd:\SAP'. The dialog includes a 'Finish' button and a 'Cancel' button at the bottom.

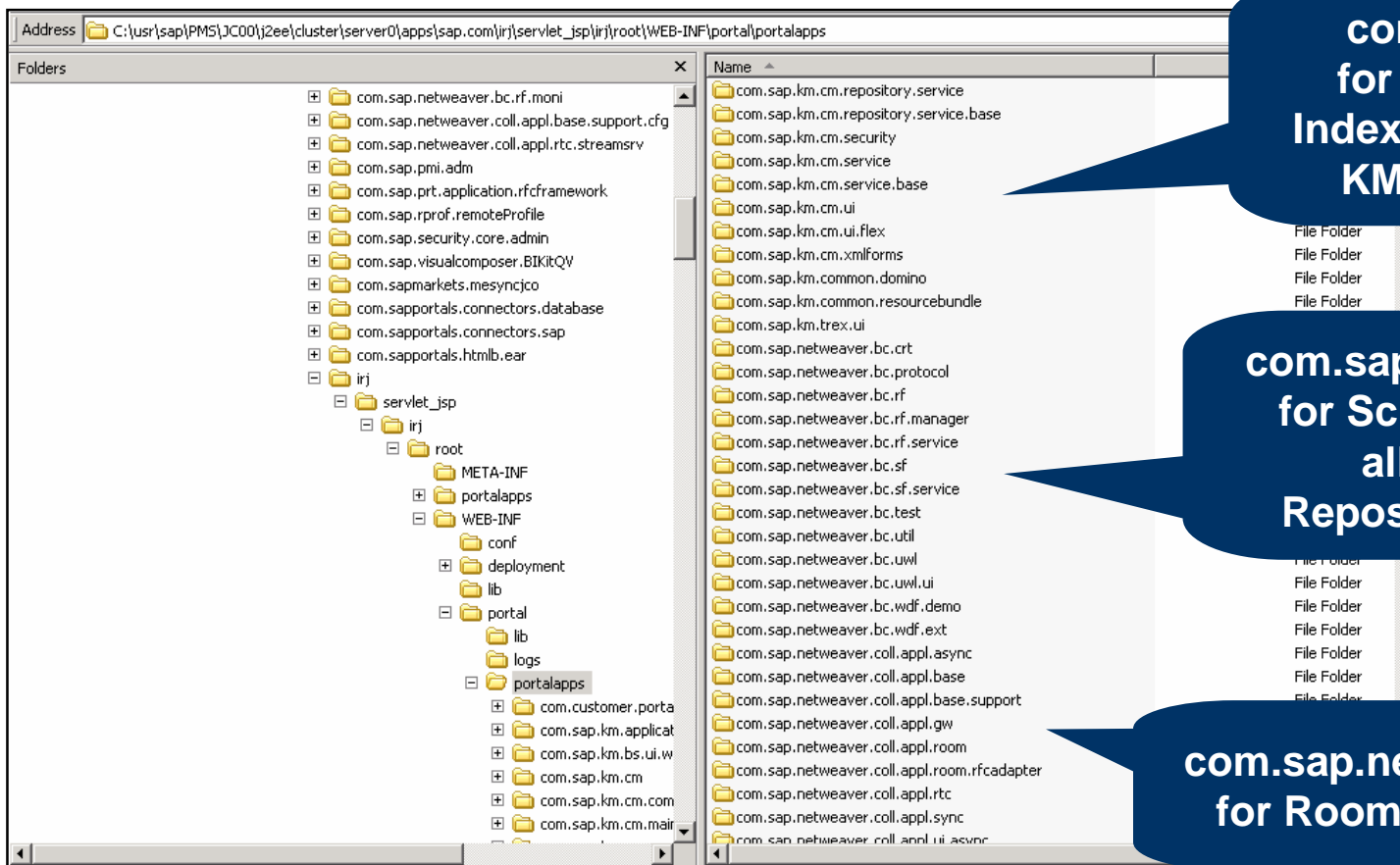
SAP NetWeaver Developer Studio For KMC

Create a new Portal Application project (3)



SAP NetWeaver Developer Studio For KMC

**Integrate the KM libraries from your Portal installation:
take the public JAR files from the par files**



**com.sap.km.*
for Flexible UI,
Indexmanagement,
KM API Usage**

**com.sap.netweaver.bc.*
for Scheduler, Filter,
all Services,
Repository Manager**

**com.sap.netweaver.coll.*
for Room Extensions**

Hint: copy all needed jar files to a local folder

SAP NetWeaver Developer Studio For KMC

Go to the SDN – Knowledge Management – Quick Links: Knowledge Management and Collaboration Developer's Guide

Knowledge Management and Collaboration (KMC)

Knowledge Management and Collaboration (KMC)

Knowledge Management provides functionalities to manage unstructured information (like documents) and link them with structured information (e.g. Business Objects). It also provides search and navigation functionalities.

Collaboration empowers features to allow teams to collaborate using the portal (like chatting, online meetings or sharing documents in a team room).

| | |
|---|---|
| Introduction to the KMC Platform | Introduction to the KMC Platform |
| The Foundations Repository Framework (RF) | The Foundations Repository Framework (RF) |
| -> Concepts | -> Concepts |
| -> Components and Their Functions | -> Components and Their Functions |
| -> Extensions | -> Extensions |
| -> Services | -> Services |
| The Search Engine (REX) | The Search Engine (REX) |
| The Knowledge Management Components (KMC) | The Knowledge Management Components (KMC) |
| -> KMC Global Services | -> KMC Global Services |
| -> KMC Repository Services | -> KMC Repository Services |
| -> Flexible ... | -> Flexible ... |
| Collaboration | Collaboration |
| The Development Environment | The Development Environment |
| -> Setting Up the IDE | -> Setting Up the IDE |
| -> Setting Up a Project with Wizards | -> Setting Up a Project with Wizards |
| -> Setting Up a Project manually | -> Setting Up a Project manually |
| Using and Extending the KMC Platform | Using and Extending the KMC Platform |
| -> Introduction to the KMC API | -> Introduction to the KMC API |
| -> Repository Framework (RF) | -> Repository Framework (RF) |
| -> How to Use the RF's Client API | -> How to Use the RF's Client API |
| -> How to Use the RF's API | -> How to Use the RF's API |

Overview Package Class Use Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.sapportals.wcm.rendering.collection

Class AbstractComponent

[contained in: com.sap.km.cm.ui.flex.par - km.shared.ui.flex.collection_api.jar]

java.lang.Object

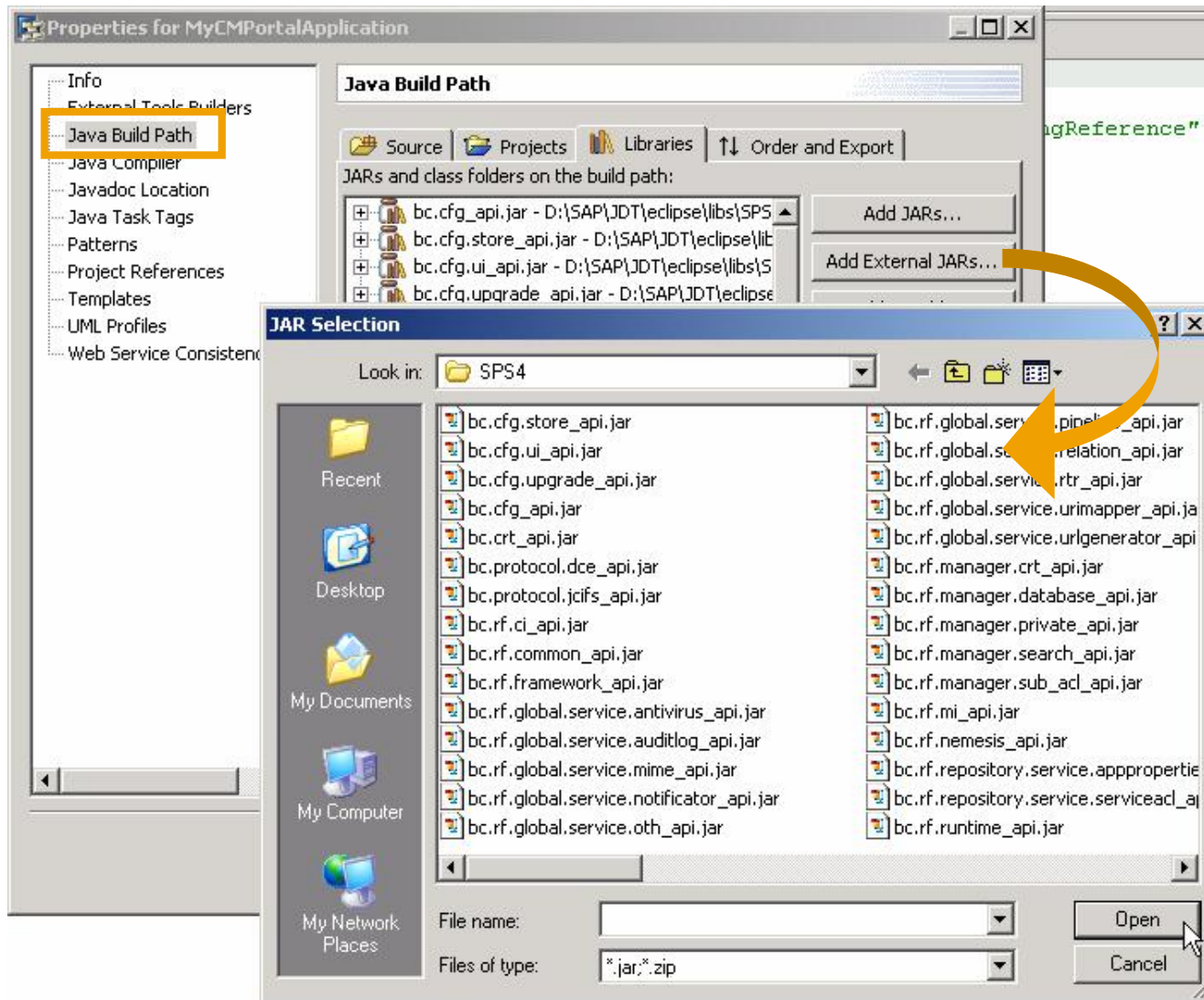
|--com.sapportals.wcm.rendering.collection.AbstractComponent

All Implemented Interfaces:

[IComponent](#)

SAP NetWeaver Developer Studio For KMC

Add KM libraries to your Portal Application KM project

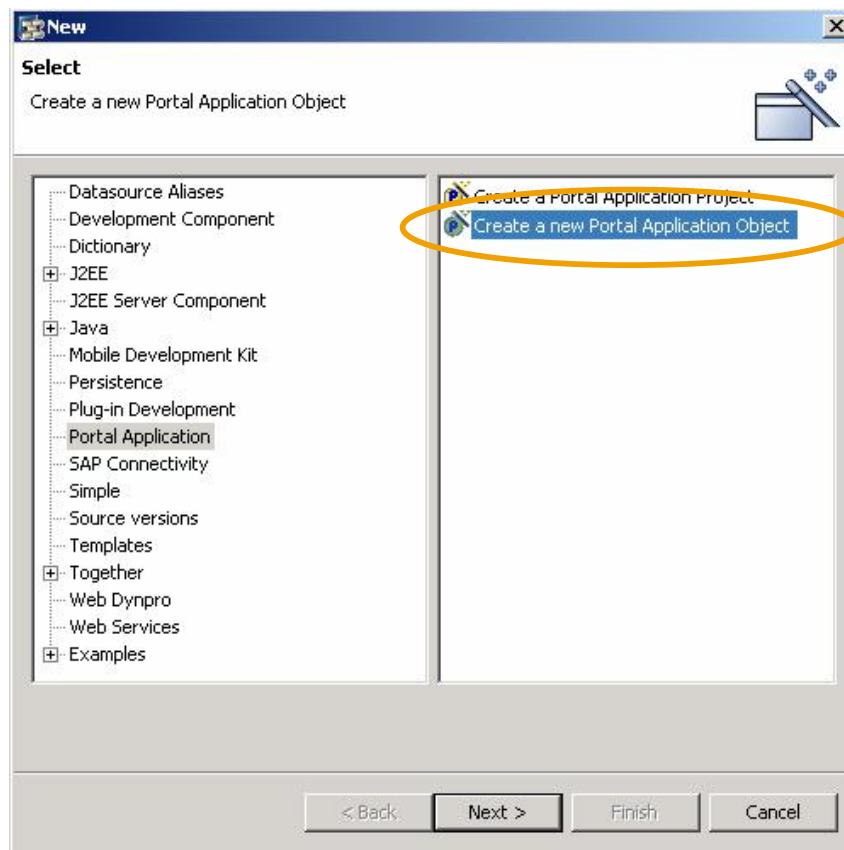


SAP NetWeaver Developer Studio For KMC

Create a new Portal Application Object (1)

n From the menu, choose: **File è New ... è Create a new Portal Application Object**

n OR **File è New ... è Other è Create a new Portal Application Object**



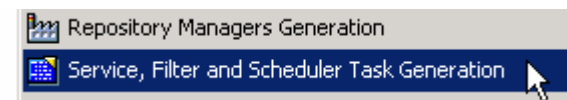
OR choose the appropriate icon in the toolbar:



OR KM wizards directly via icons in the toolbar

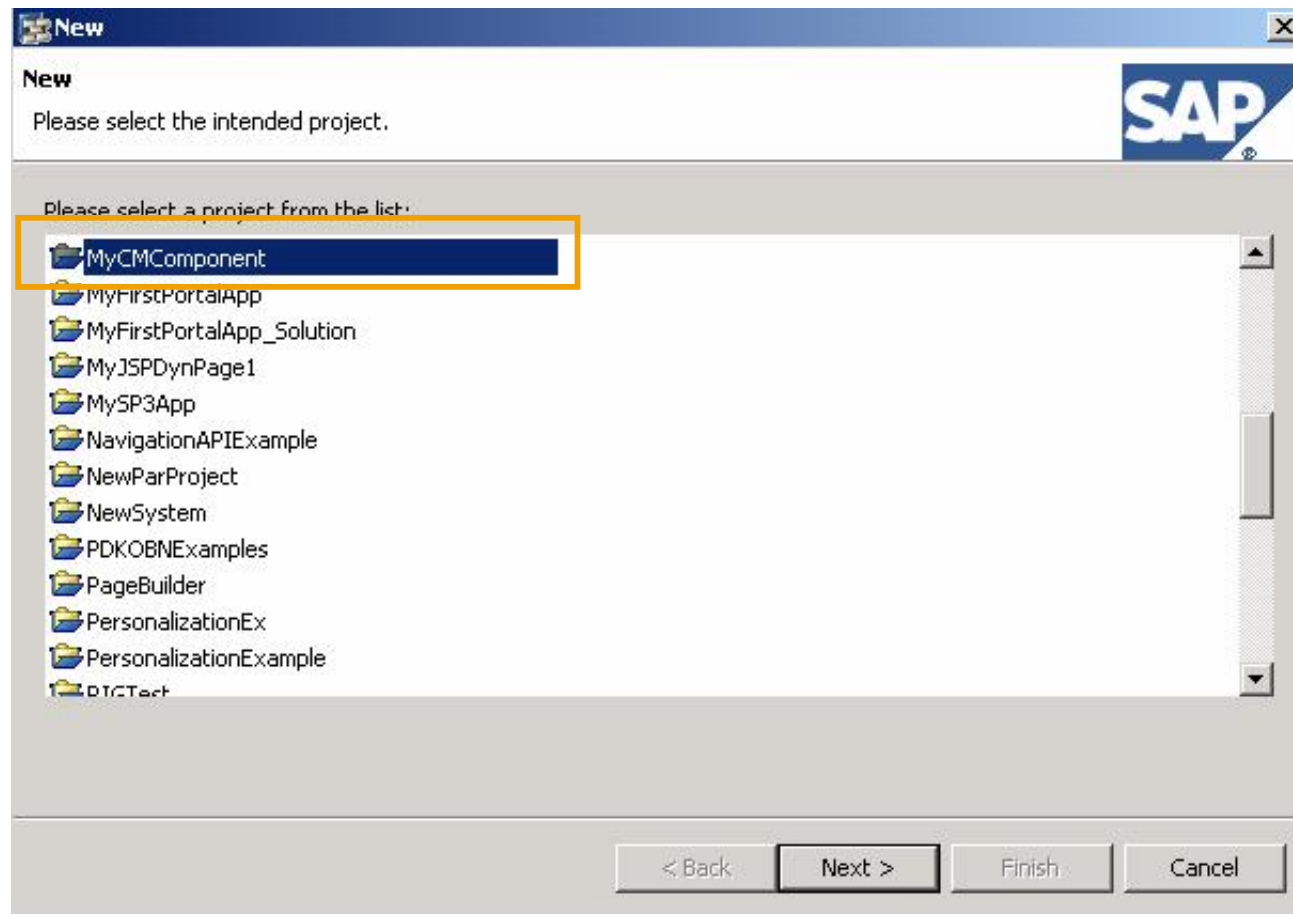


OR right-clicking your project



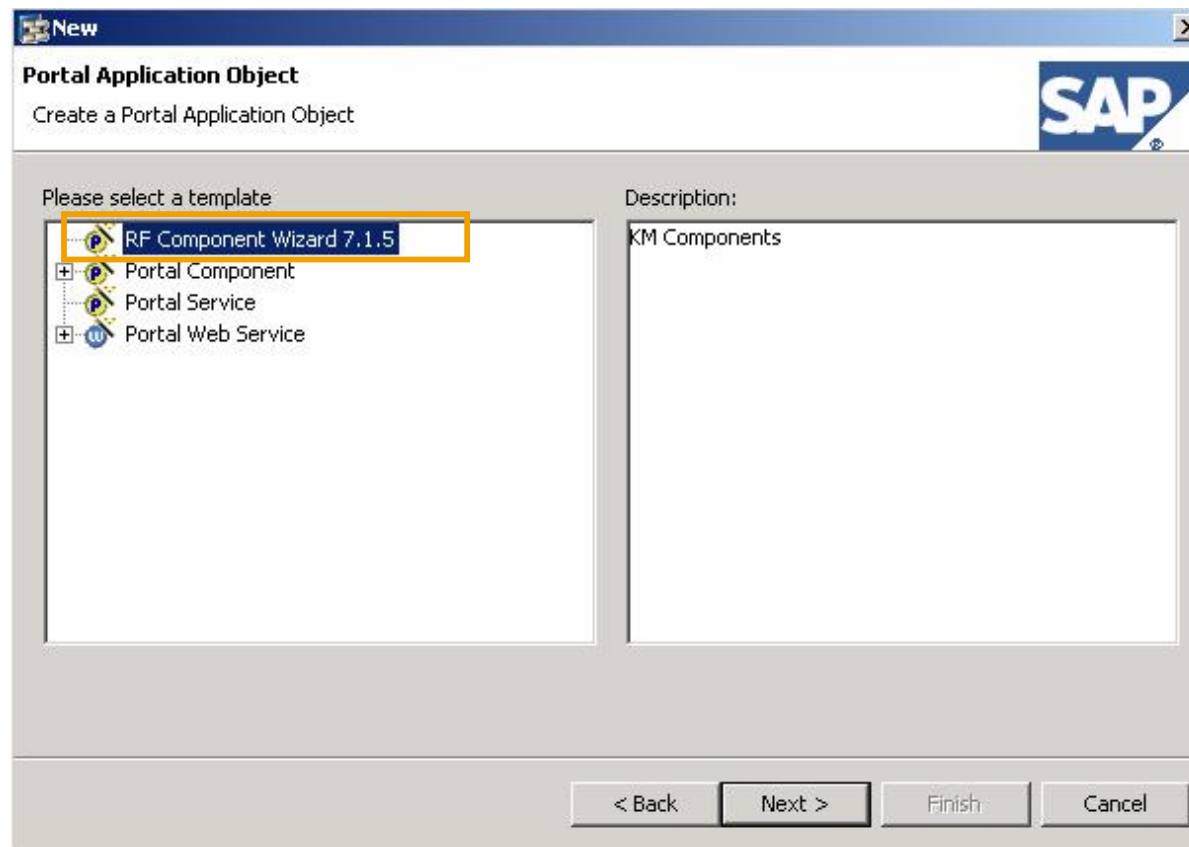
Create a new Portal Application Object (2)

Select the Portal Application Project and click „Next“:



Create a new Portal Application Object (3)

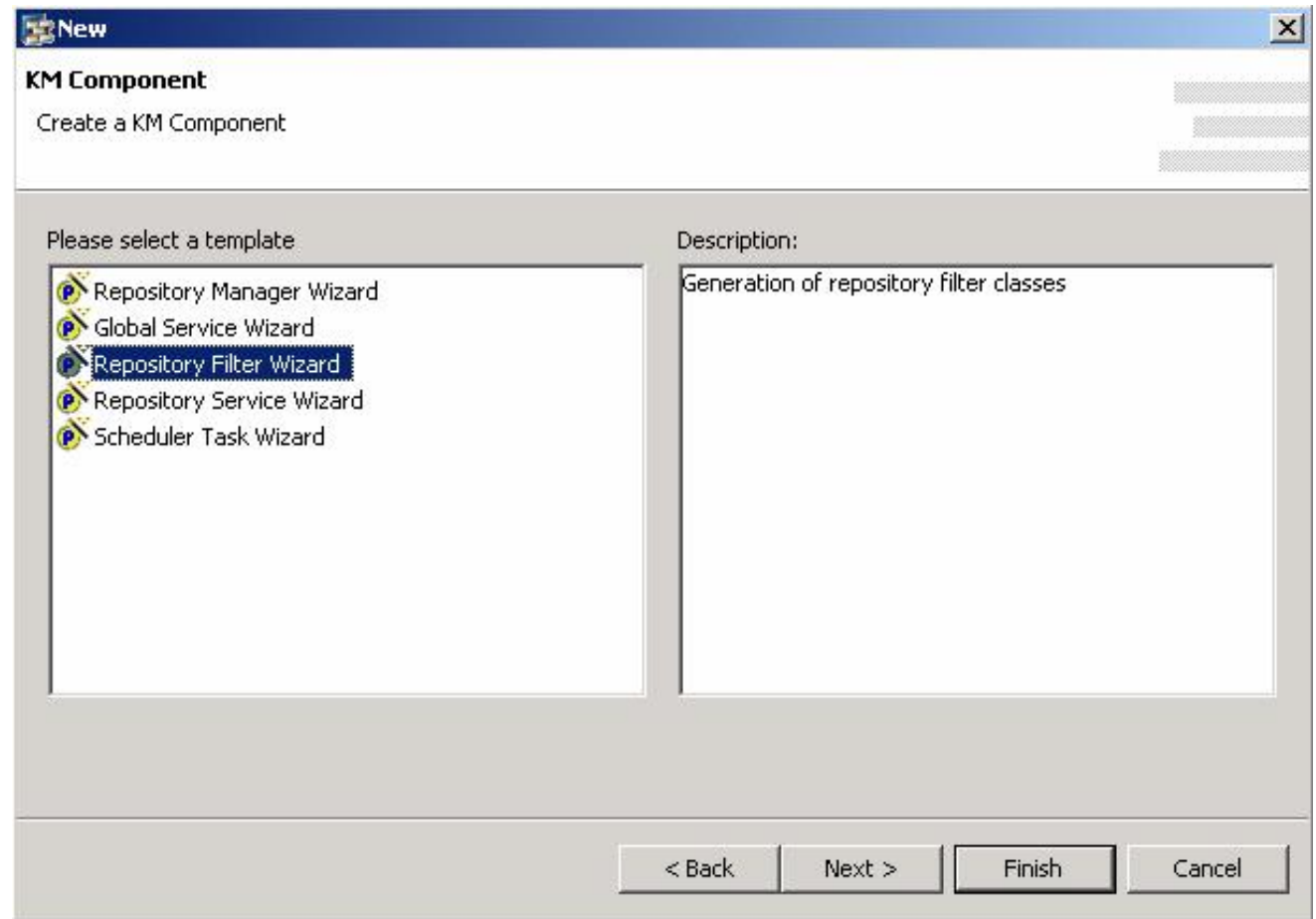
Create a new Repository Framework Component for your Portal Application Project:



Create a new Portal Application Object (4)

Select which KM Component you want to implement:

- n Repository Manager
- n Global Service
- n Repository Filter
- n Repository Service
- n Scheduler Task



Example: Create a Repository Filter è Namespace Filter

The screenshot shows the 'KMC Wizard' dialog box, specifically the 'Repository Filter Wizard' step. The title bar reads 'KMC Wizard' and the subtitle is 'Repository Filter Wizard'. Below the subtitle, it says 'Generation of repository filter classes'. The dialog is divided into several sections:

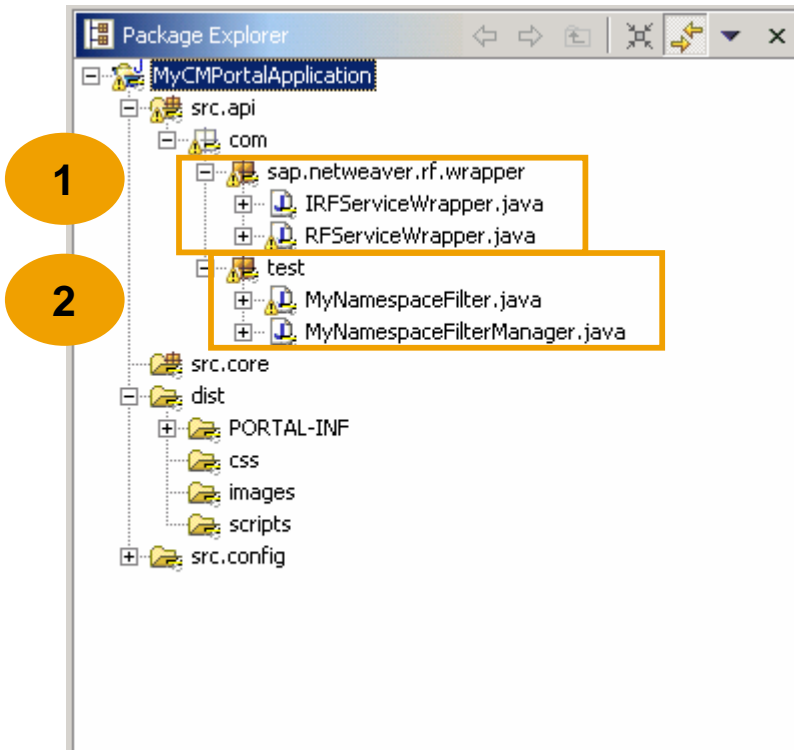
- Config Framework Version:** A dropdown menu is set to 'NW04'.
- Generate project classpath with BC libraries from plugin:** A checked checkbox.
- Class name:** A text field containing 'MyNamespaceFilter'.
- Package:** A text field containing 'com.test'.
- Name (if different from class name):** An empty text field.
- Types:** A dropdown menu is open, showing a list of options: 'amespace', 'Namespace', 'Property', 'Content', and 'mychmforapplication'. The 'Namespace' option is highlighted by the mouse cursor.

At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

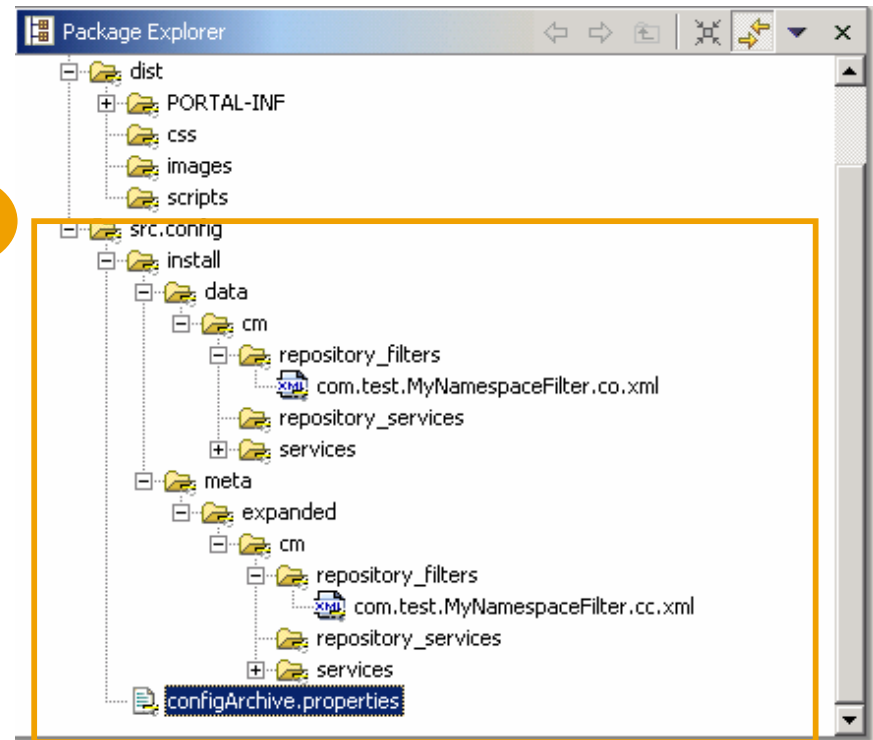
SAP NetWeaver Developer Studio For KMC

Project structure

1. Portal service wrapper in src.API
2. CM Component classes
3. Config Archive for config framework



3
+



SAP NetWeaver Developer Studio For KMC

A standard Portal Service with interface and implementation class uses the CrtClassLoader Registration

The Service KEY has to be defined:

```
package com.sap.netweaver.rf.wrapper;  
  
import com.sapportals.portal.prt.service.IService;  
  
public interface IRFServiceWrapper extends IService  
{  
    public static final String KEY = "/* To be inserted */";  
}
```

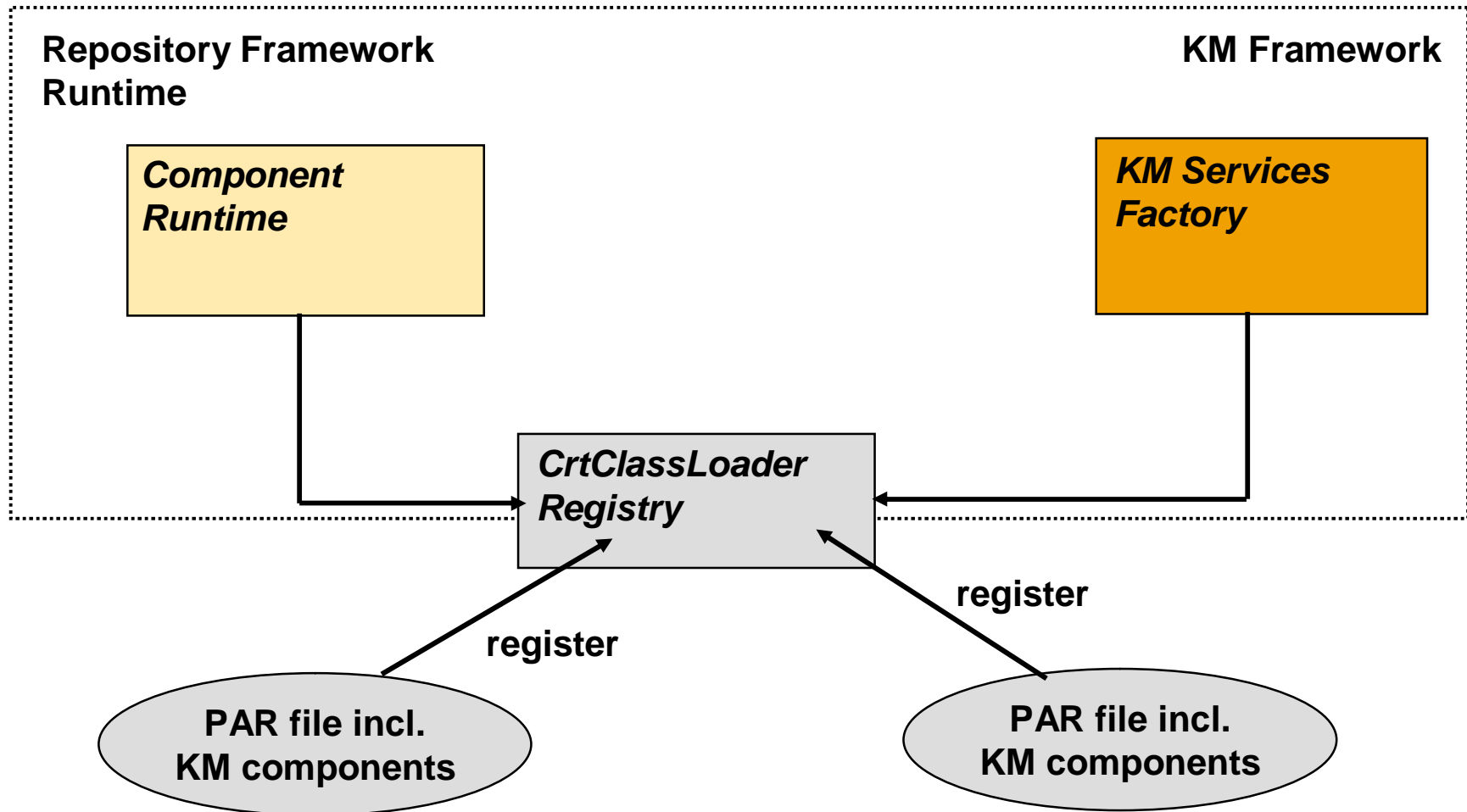
Portal Service KEY:=
<Servicename>

Naming convention:

<ServiceName> := <ApplicationName>.<Service>

<ApplicationName> := parfilename

CrtClassLoader Registration



A standard Portal Service implements the CrtClassLoader Registration

```
public class RFServiceWrapper implements
    IRFServiceWrapper {

    public void init(IServiceContext serviceContext){
        mm_serviceContext = serviceContext;
        CrtClassLoaderRegistry.addClassLoader(
            this.getClass().getClassLoader()
        );
    }
}
```

Registers this service wrappers classloader at the KM CrtClassLoader registry.

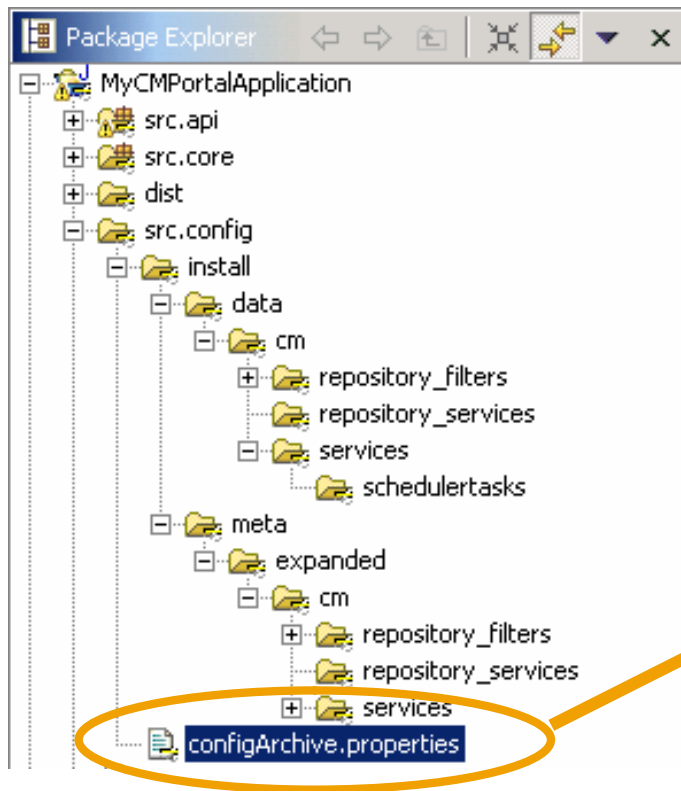
è **Mandatory implementation for all KM components**

Using the configuration framework (KM base services)

- n **KM components are exposed as configurables**
- n **Access the configuration data with base service wrapper classes (IConfigurable)**
- n **Deploy pre-defined Classes definitions and configurable instances with a KM component PAR files**

SAP NetWeaver Developer Studio For KMC

Create / Check KM component configuration



configArchive.properties:

ca.name=MyCMPortalApplication.prjconfig

ca.version=6.0.1.1

ca.creation.time=1117

ca.creation.date=20040826

ca.creation.user=unknown

ca.creation.machine=unknown

**ca.dependencies: bc.util.prjconfig,
bc.sf.prjconfig, bc.sf.service.prjconfig,
bc.rf.prjconfig**

cma.name=MyCMPortalApplication.prjconfig

cma.version=6.0.1.1

cma.storage=sfs

cma.creation.time=1117

cma.creation.date=20040826

cma.creation.user=unknown

cma.creation.machine=unknown

**cma.dependencies: bc.util.prjconfig,
bc.sf.prjconfig, bc.sf.service.prjconfig,
bc.rf.prjconfig**

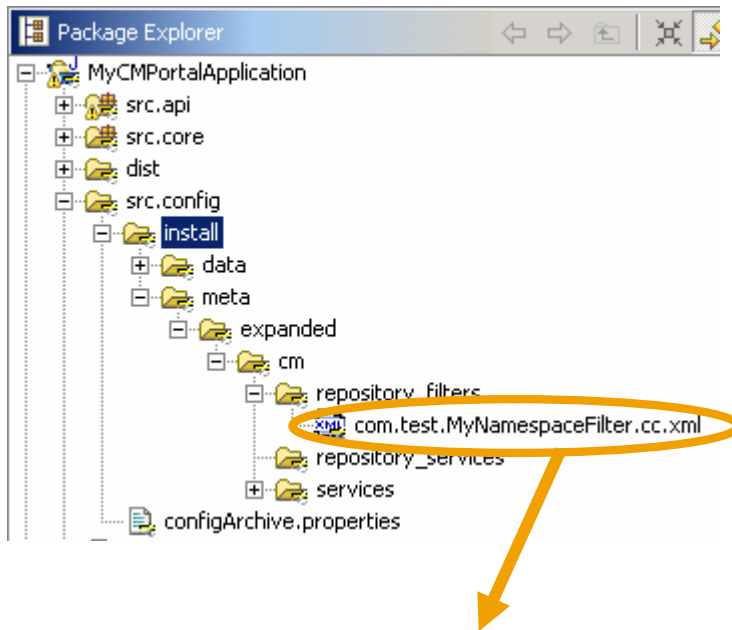
Structure:

src.config/install/data/cm/...

src.config/install/meta/expanded/cm/

...

Create / Check KM component configuration



1. Contains the class definition
2. Contains instance files

2. Instance definition (com.test.MyNamespaceFilter.co.xml):

```
<Configurable configclass="com.test.MyNamespaceFilter">  
  <property name="name"  
    value="com.test.MyNamespaceFilter" />  
  <property name="active" value="true" />  
  <property name="description" />  
</Configurable>
```

1. Class definition for this namespace filter (com.test.MyNamespaceFilter.cc.xml):

```
<ConfigClass name="com.test.MyNamespaceFilter"  
  extends="RepositoryFilter">  
  <attribute name="class" type="class"  
    constant="com.test.MyNamespaceFilterManager" />  
</ConfigClass>
```

Deployment using PAR Export Wizards

n Standard portal deployment

- u maintained portalapps.xml is mandatory
- u additional information can be provided in the public part of this PAR file

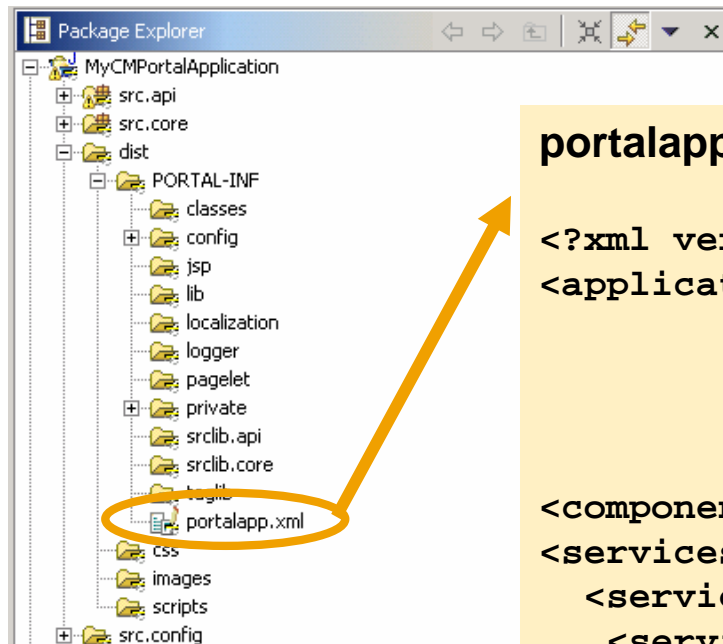
n Creates a new wrapper service for each PAR file including KM components

- u Service has no functionality, except of classloader registration

n Configuration for configuration framework

- u automatically bundle the archive containing
 - l meta data (Classes definitions)
 - l instances

Create / Check portal service configuration



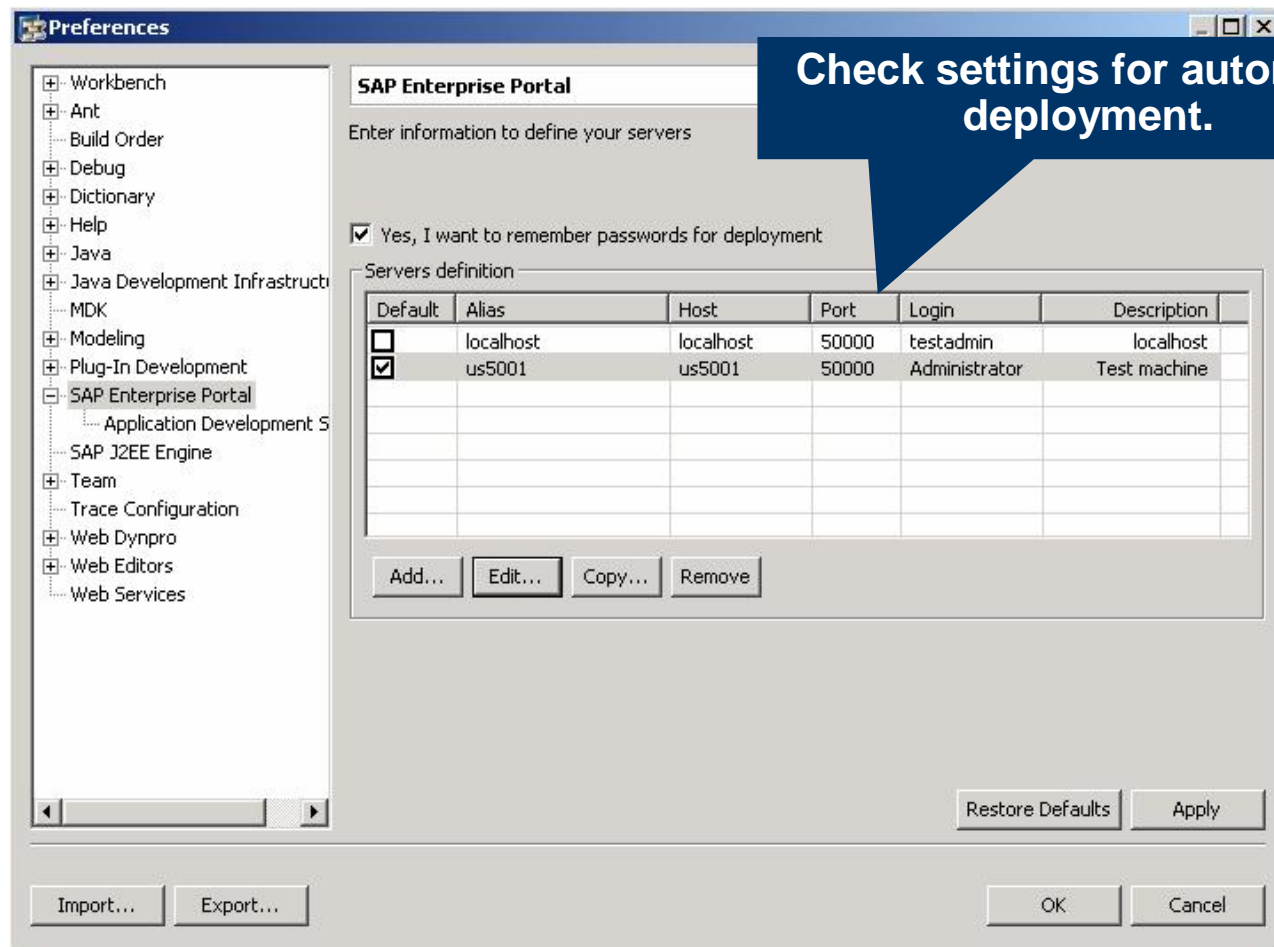
portalapp.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<application>
  <application-config>
    <property name="SharingReference"
      value="knowledgemanagement, ..." />
  </application-config>
</application>
<components/>
<services>
  <service name="RFServiceWrapper">
    <service-config>
      <property name="className"
        value="com.sap.netweaver.rf.wrapper.RFServiceWrapper" />
      <property name="startup" value="true" />
    </service-config>
  </service></services>
</application>
```

Server configuration – Connection to local/remote Portal

From the menu, choose:

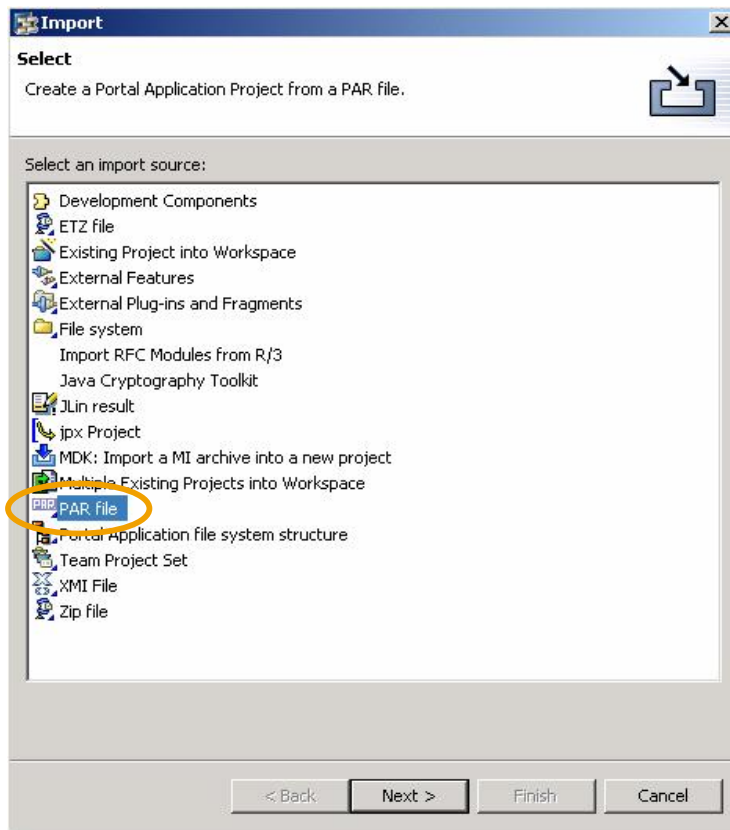
Window -> Preferences -> SAP Enterprise Portal



SAP NetWeaver Developer Studio For KMC

Create a PAR file

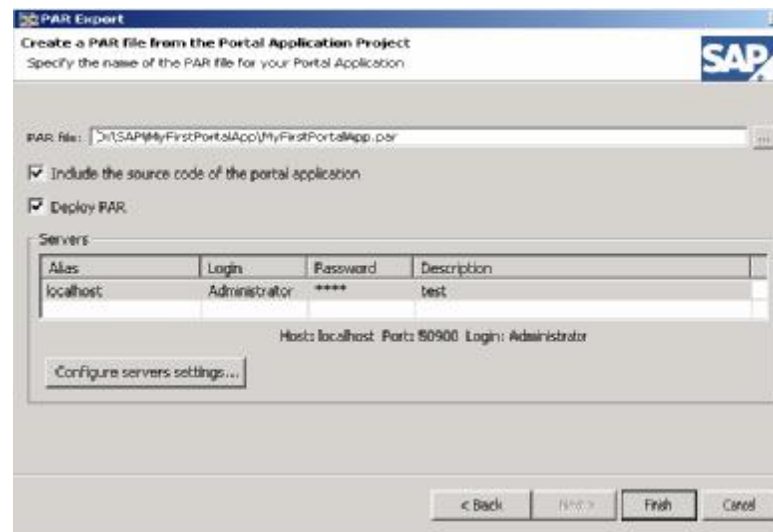
From the menu, choose: **File** è **Export** è **PAR File** – click **Next**



Or choose the appropriate icon in the toolbar:



Or right-click your project and choose: **PAR** Quick PAR Upload



After deployment -> restarting the SAP J2EE Engine is needed

Further Information

è Public Web:

www.sap.com

SAP Developer Network: www.sdn.sap.com è Knowledge Management è Code Samples

SAP Developer Network: www.sdn.sap.com è Knowledge Management è NetWeaver Developer Guide

Questions?

Q&A

Copyright 2004 SAP AG. All Rights Reserved

- n No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.
- n Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.
- n Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.
- n IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.
- n Oracle is a registered trademark of Oracle Corporation.
- n UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.
- n Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.
- n HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.
- n Java is a registered trademark of Sun Microsystems, Inc.
- n JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.
- n MaxDB is a trademark of MySQL AB, Sweden.
- n SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.
- n These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

