Rational Application Developer for WebSphere Software 8.0

Anita Rass Wan, RAD Product Manager

Rational software WebSphere software

ON DEMAND BUSINESS

RAD accelerates SOA, Java EE, Web 2.0 and Portal development for IBM middleware

Evans Data IDE User Choice Award 2010

Web 2.0

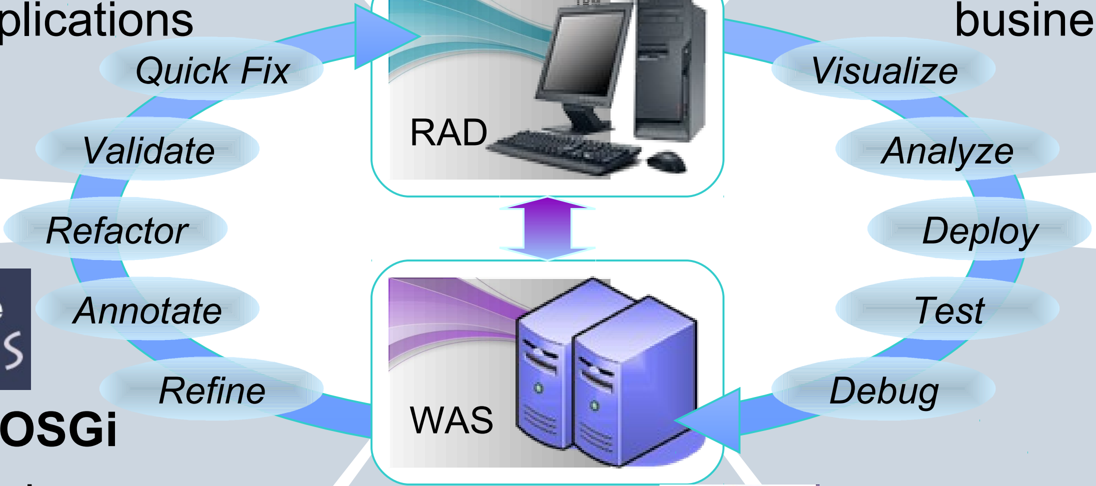
Extend SOA and Java EE assets to the glass via dynamic, rich JSF and DOJO applications

Team Collaboration

Rational Team Concert: Collaborate, Problem Determination

SOA

Assemble Web services and SCA components into heterogeneous business applications



Java EE 6 and OSGi

Develop, test and componentize Java EE 6 applications with annotation based programming

WAS Integration

Hot deploy incremental changes to WAS

Portal

Visually design portal sites and portlets for WebSphere Portal Server

RAD Themes Overview

Theme 1: Support for Standards and Standards Currency

- Tools to support JEE6 specifications (EJB 3.1, JPA 2.0, JSF 2.0, Servlet 3.0, ..), SCA
- Ongoing support for WAS feature packs; JPA 2.0 tools, OSGI (Aries) Tools

Theme 2: Improve Developer Productivity

- Support iterative creation, validation, revision and deletion of artifacts in support of JEE 6
- Advanced Web 2.0 development tools to build rich internet applications
- Comprehensive migration support from RAD 7.0.x and RAD 7.5.x

Theme 3: Improve Application Quality

- Improve the static problem determination tools (line level code coverage enhancements)
- Improve the dynamic problem determination tools: Debug tools supporting JEE6, enhanced profiling capabilities

Theme 4: Cloud support

- Use RAD to access test server environments in the IBM public cloud

Theme 5: Exploit Integrations with other IBM products

- WebSphere Application server test environment
- WebSphere Portal server test environment
- Rational Team Concert

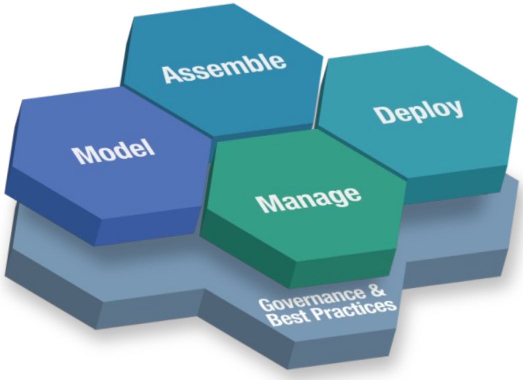
Java Enterprise Edition (JEE) 6

Optimized for WebSphere

Continuing the trend of simplification, streamlining and improved integration

RAD helps accelerate Java EE 6 annotation style development

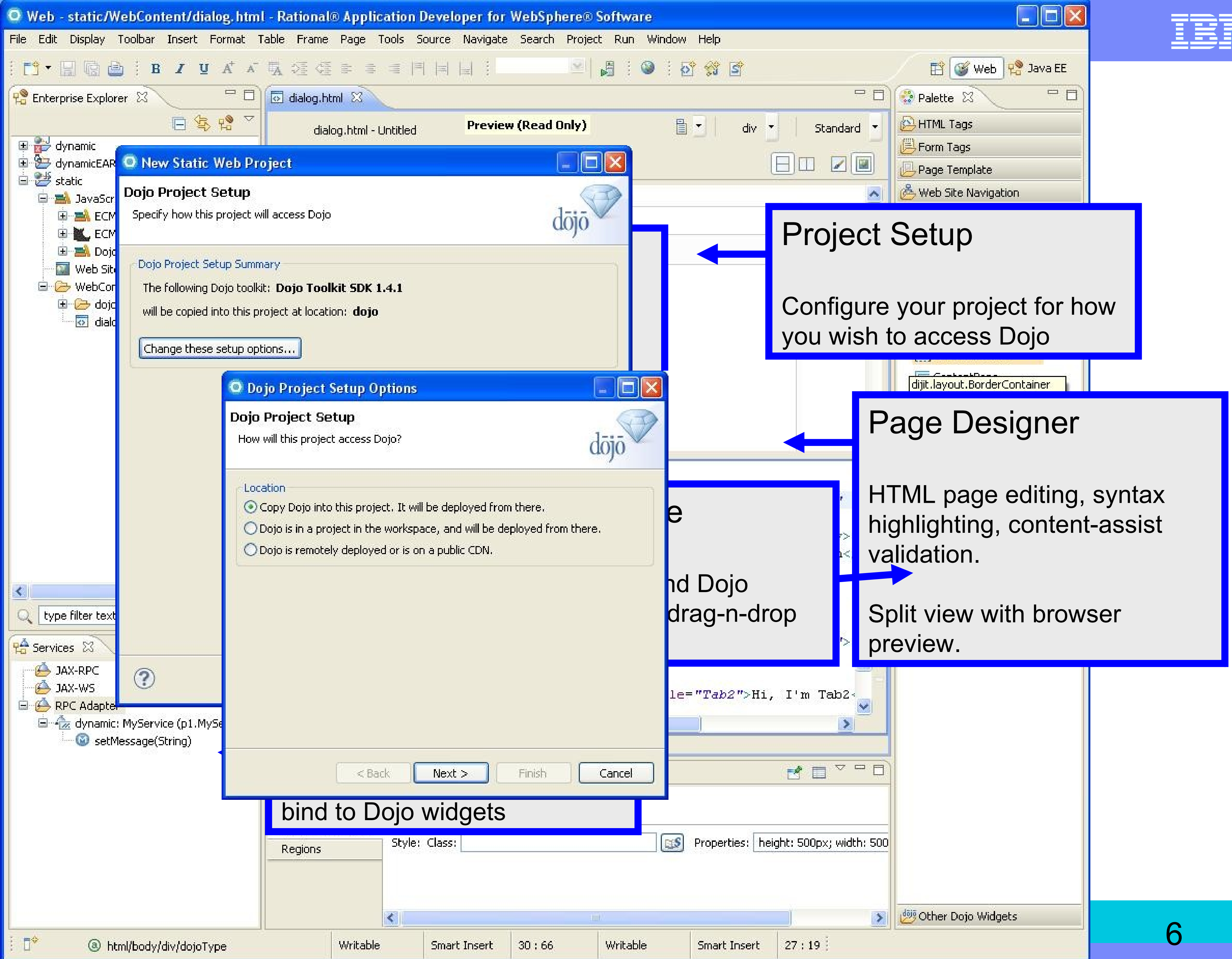
- Content assist and as you type validation
- Quickfixes for code and project configuration
- Advanced refactoring options to allow you to modify and maintain code in an iterative manner.
- Annotation view to manage and modify annotation properties



Optimized for WebSphere

RAD - Simplified Web 2.0 Development

- Use RAD to build rich internet applications to improve online user experience and increase customer satisfaction
- Lower the barriers of adoption of Web 2.0 technologies
  - Visual development of Web 2.0 pages
  - Source level tools to aid with Javascipt and dojo development
  - Debug capabilities with Firebug integration
  - Multiple browser previews
  - Expose server side assets as Web 2.0 consumable services
- Lightweight Ajax test server to rapidly preview web pages that include images, scripts, services, and data



Web - static/WebContent/dialog.html - Rational® Application Developer for WebSphere® Software

le="Tab2">Hi, I'm Tab2

New Static Web Project

Dojo Project Setup

Specify how this project will access Dojo

Dojo Project Setup Summary

The following Dojo toolkit: **Dojo Toolkit SDK 1.4.1**

will be copied into this project at location: **dojo**

Change these setup options...

Dojo Project Setup Options

Dojo Project Setup

How will this project access Dojo?

Location

● Copy Dojo into this project. It will be deployed from there.

○ Dojo is in a project in the workspace, and will be deployed from there.

○ Dojo is remotely deployed or is on a public CDN.

< Back | Next > | Finish | Cancel

Project Setup

Configure your project for how you wish to access Dojo

Page Designer

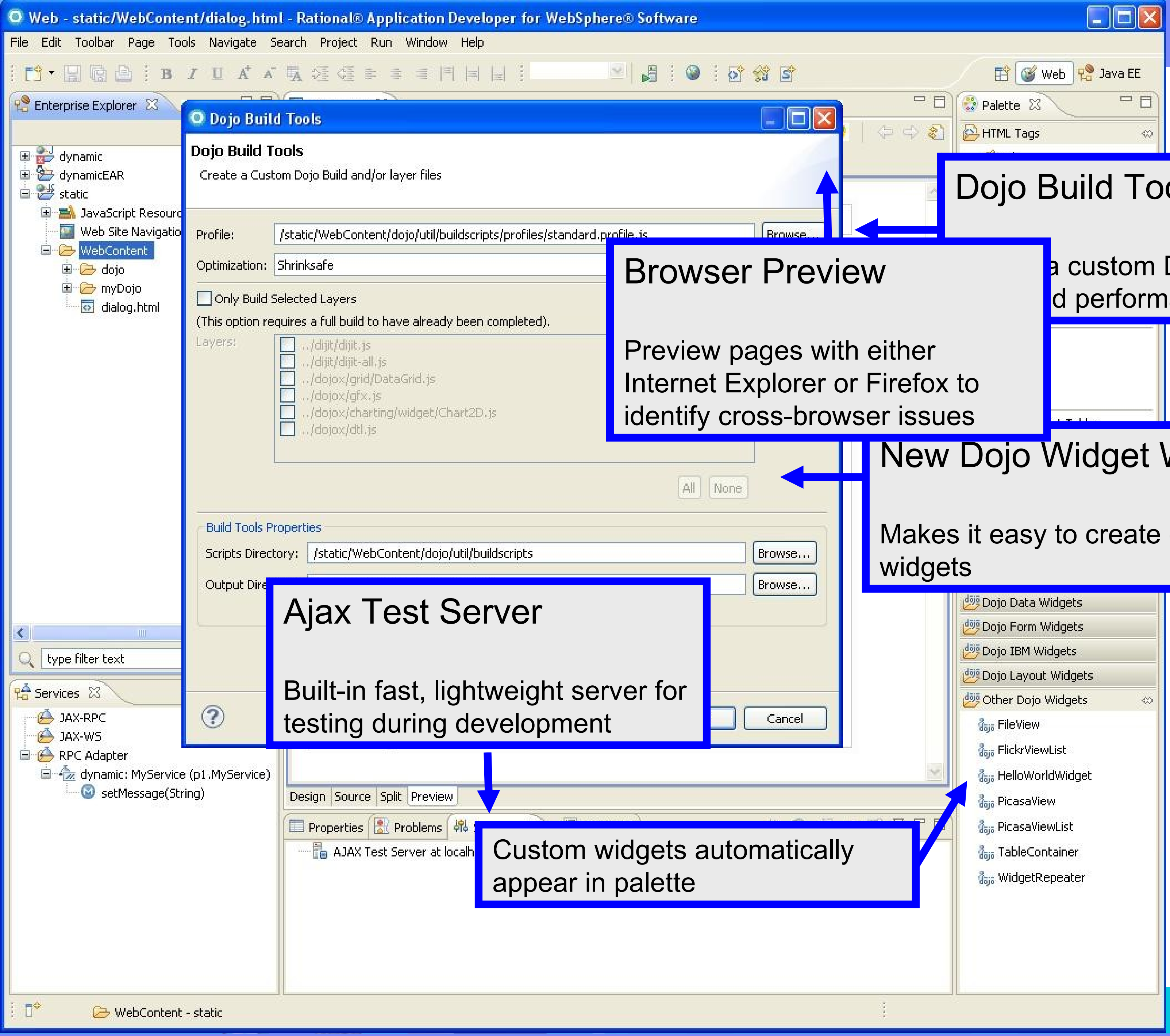
HTML page editing, syntax highlighting, content-assist validation.

Split view with browser preview.

e

nd Dojo drag-n-drop

bind to Dojo widgets



Web - static/WebContent/dialog.html - Rational® Application Developer for WebSphere® Software

Dojo Build Tools

Create a Custom Dojo Build and/or layer files

Profile: /static/WebContent/dojo/util/buildscripts/profiles/standard.profile.js [Browse...]

Optimization: Shrinksafe

□ Only Build Selected Layers

(This option requires a full build to have already been completed).

Layers:
- □ ../dijit/dijit.js
- □ ../dijit/dijit-all.js
- □ ../dojox/grid/DataGrid.js
- □ ../dojox/gfx.js
- □ ../dojox/charting/widget/Chart2D.js
- □ ../dojox/dtl.js

[All] [None]

Build Tools Properties

Scripts Directory: /static/WebContent/dojo/util/buildscripts [Browse...]

Output Dire [Browse...]

[Cancel]

Dojo Build Tools

a custom Dojo build for
d performance

Browser Preview

Preview pages with either Internet Explorer or Firefox to identify cross-browser issues

New Dojo Widget Wizard

Makes it easy to create custom widgets

Ajax Test Server

Built-in fast, lightweight server for testing during development

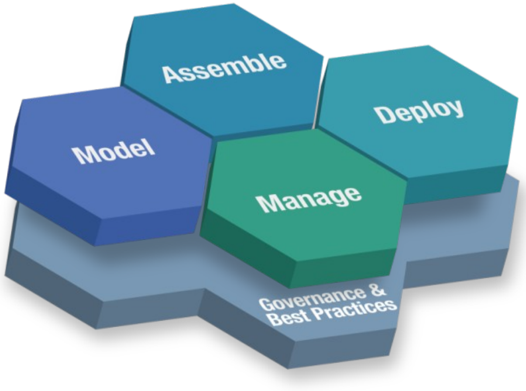
Custom widgets automatically appear in palette

Improve Application Quality

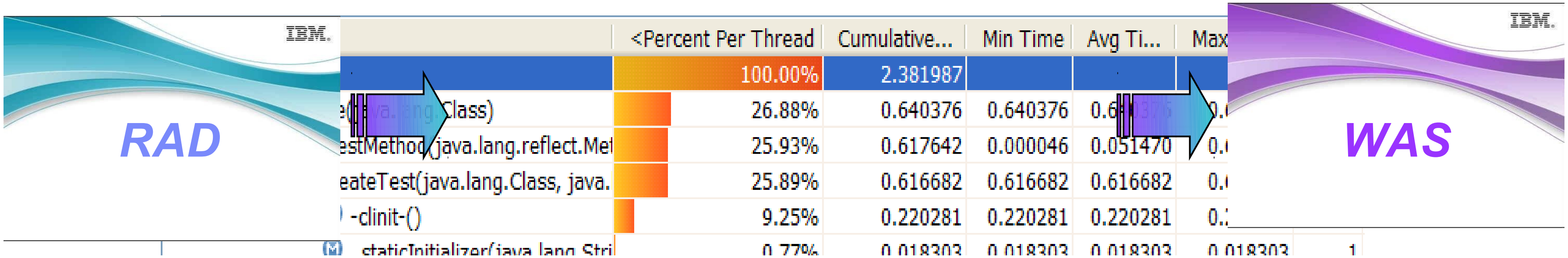
Improve Quality

Problem determination tools help find problems during the development lifecycle of your application

Static tools that help find problems in the developed code via line level code coverage or static analysis



Dynamic tools that help find problems at runtime whether it is debugging the code or profiling the code on the WebSphere Application Server



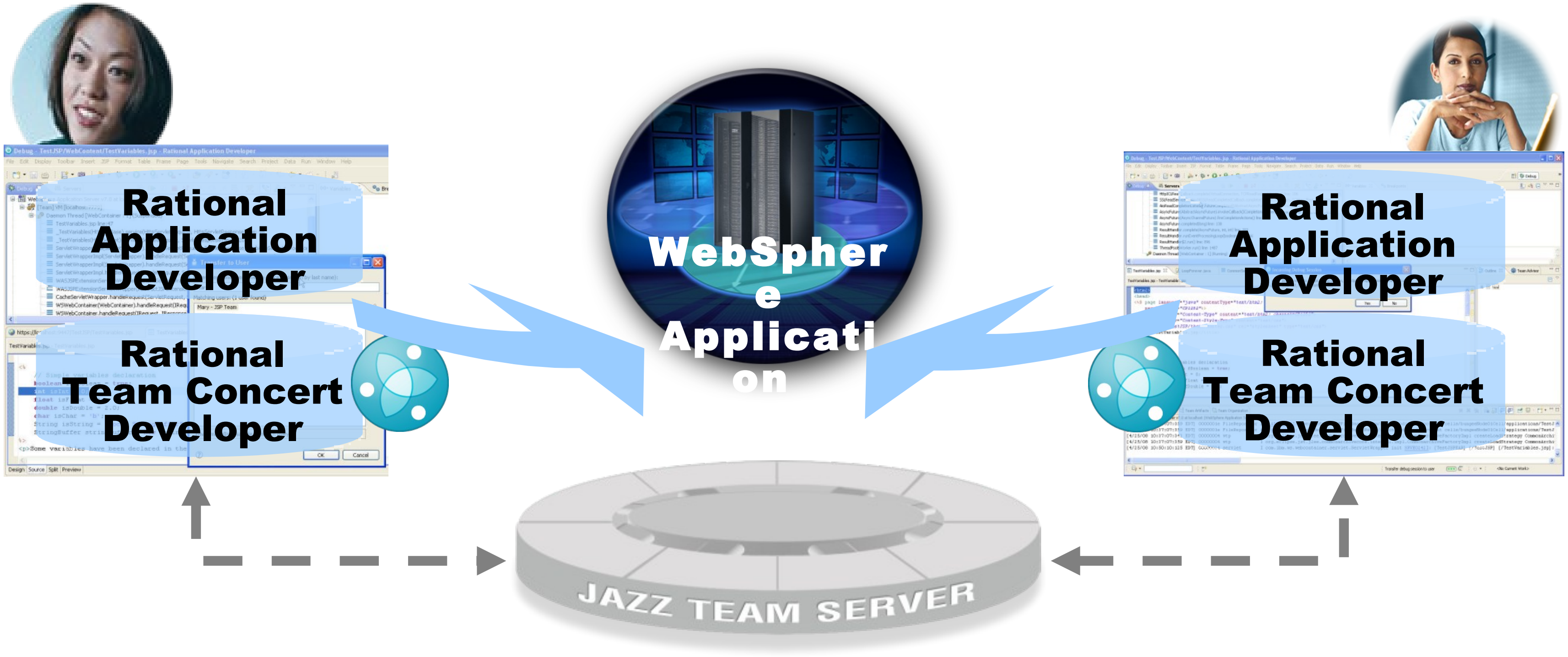
	<Percent Per Thread	Cumulative...	Min Time	Avg Ti...	Max	
	100.00%	2.381987				
e(java.lang.Class)	26.88%	0.640376	0.640376	0.640376	0.(	
estMethod(java.lang.reflect.Met	25.93%	0.617642	0.000046	0.051470	0.(	
eateTest(java.lang.Class, java.	25.89%	0.616682	0.616682	0.616682	0.(	
-clinit-()	9.25%	0.220281	0.220281	0.220281	0.:	
staticInitializer(java.lang.Stri	0.77%	0.018303	0.018303	0.018303	0.018303	1

Improve Quality Through Team Collaboration

Improve Quality

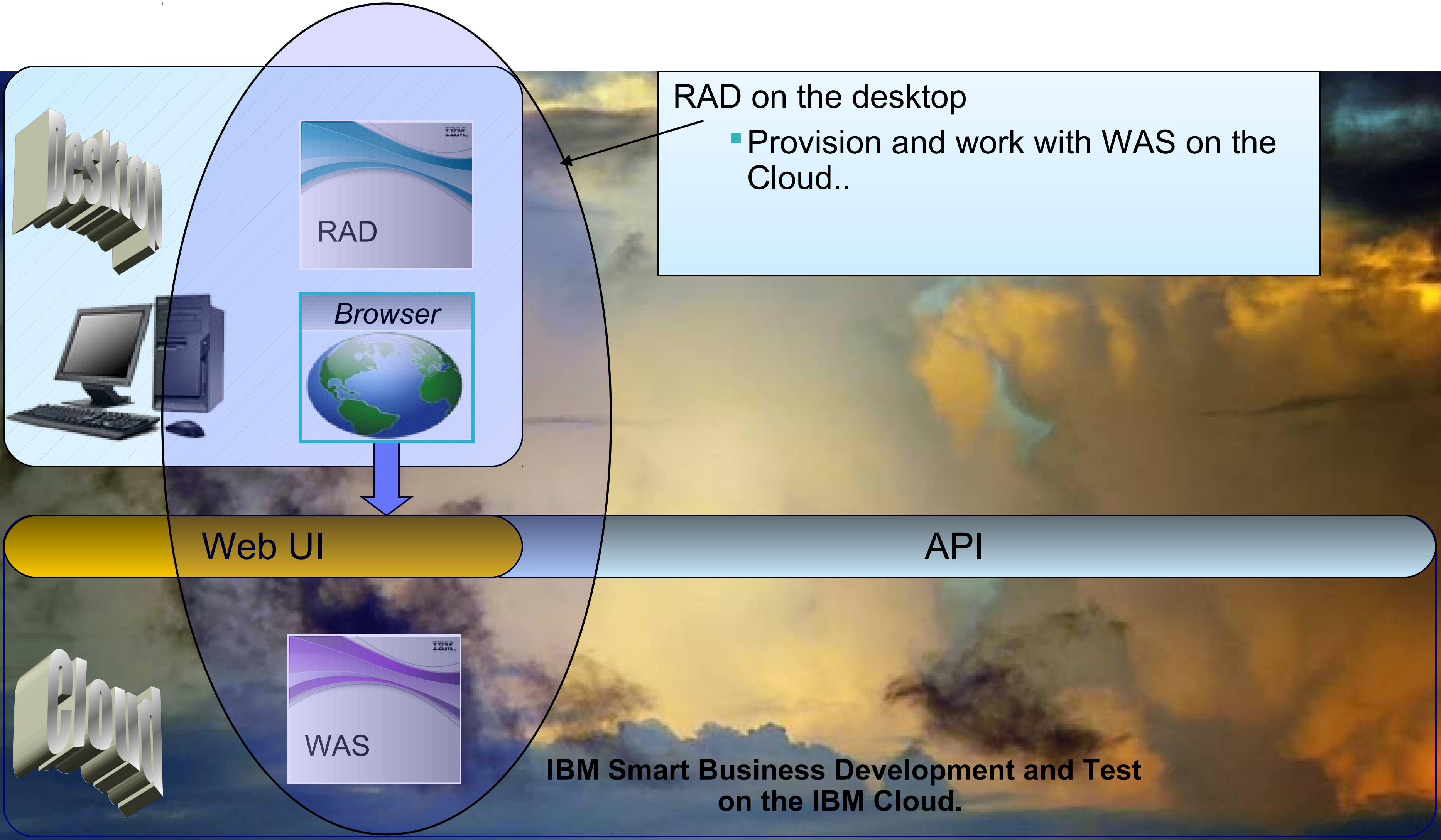
RAD and RTC integrate to:

- Share live debug sessions between team members
- Share code coverage information from automated testcase execution
- Improve test coverage and quality based on code coverage results


Rational Application Developer
Rational Team Concert Developer
WebSphere Application
Rational Application Developer
Rational Team Concert Developer
JAZZ TEAM SERVER

Introducing - Support for the Cloud

Optimized for WebSphere



RAD on the desktop

- Provision and work with WAS on the Cloud..

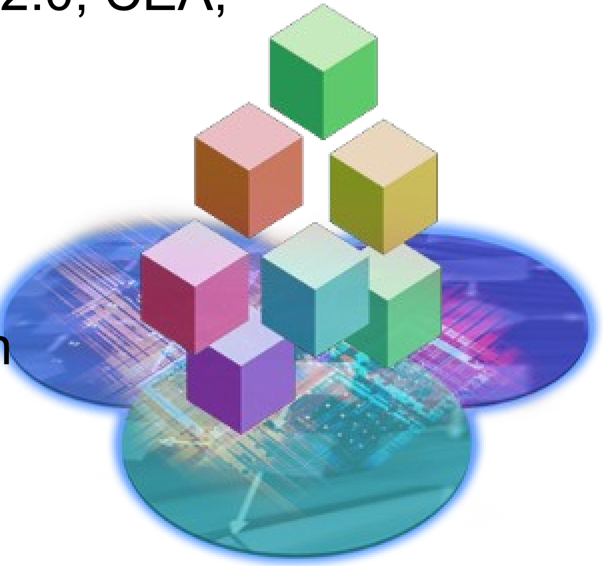
IBM Smart Business Development and Test on the IBM Cloud.

Optimized for WebSphere

WebSphere Application Server Integration

RAD the fastest development, deployment, integrated test and profiling for WebSphere Application Server

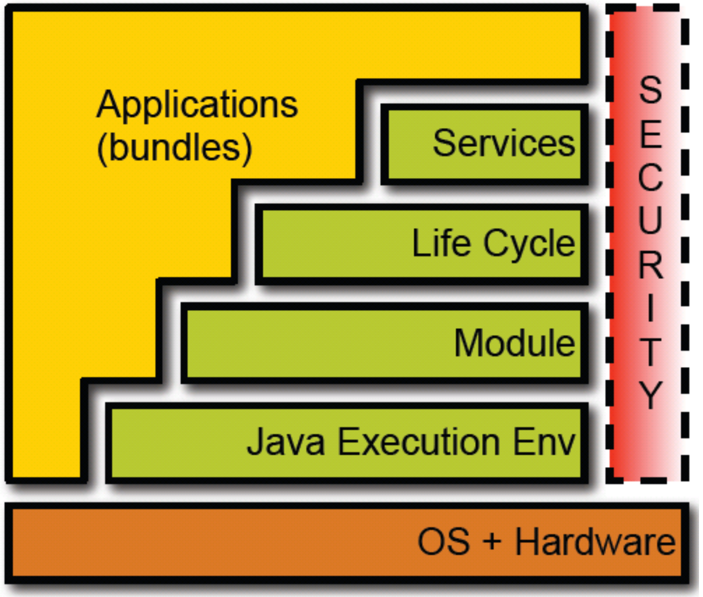
- RAD includes WebSphere Application Server integrated test environments
  - V8 beta, V7.0, V6.1, V6.0 (remote only)
  - Feature Packs – Web 2.0, SCA, OSGi, JPA 2.0, CEA, XML, Web Services, EJB 3
- Migration Support
  - Support to help you migrate your application to the latest server, or the latest specification



Increase Componentization with OSGI and RAD

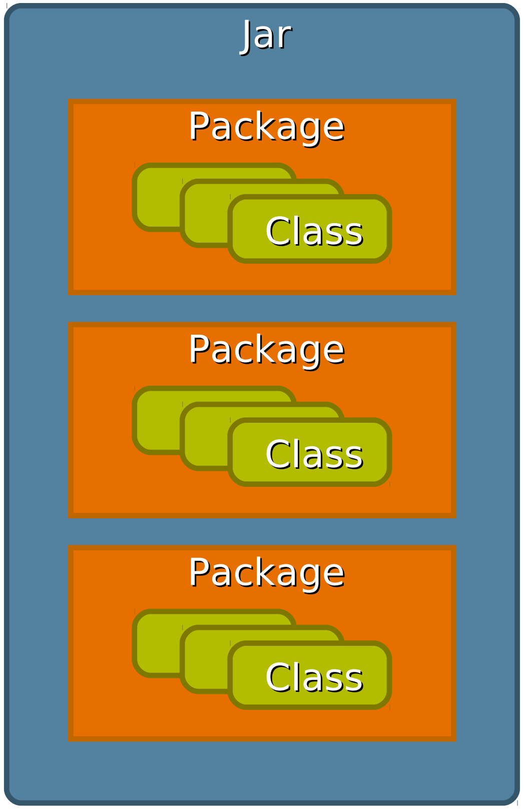
Optimized for WebSphere

Create modular applications to reduce application complexity, ensure easier integration, and administration with built-in versioning, and dependency management.



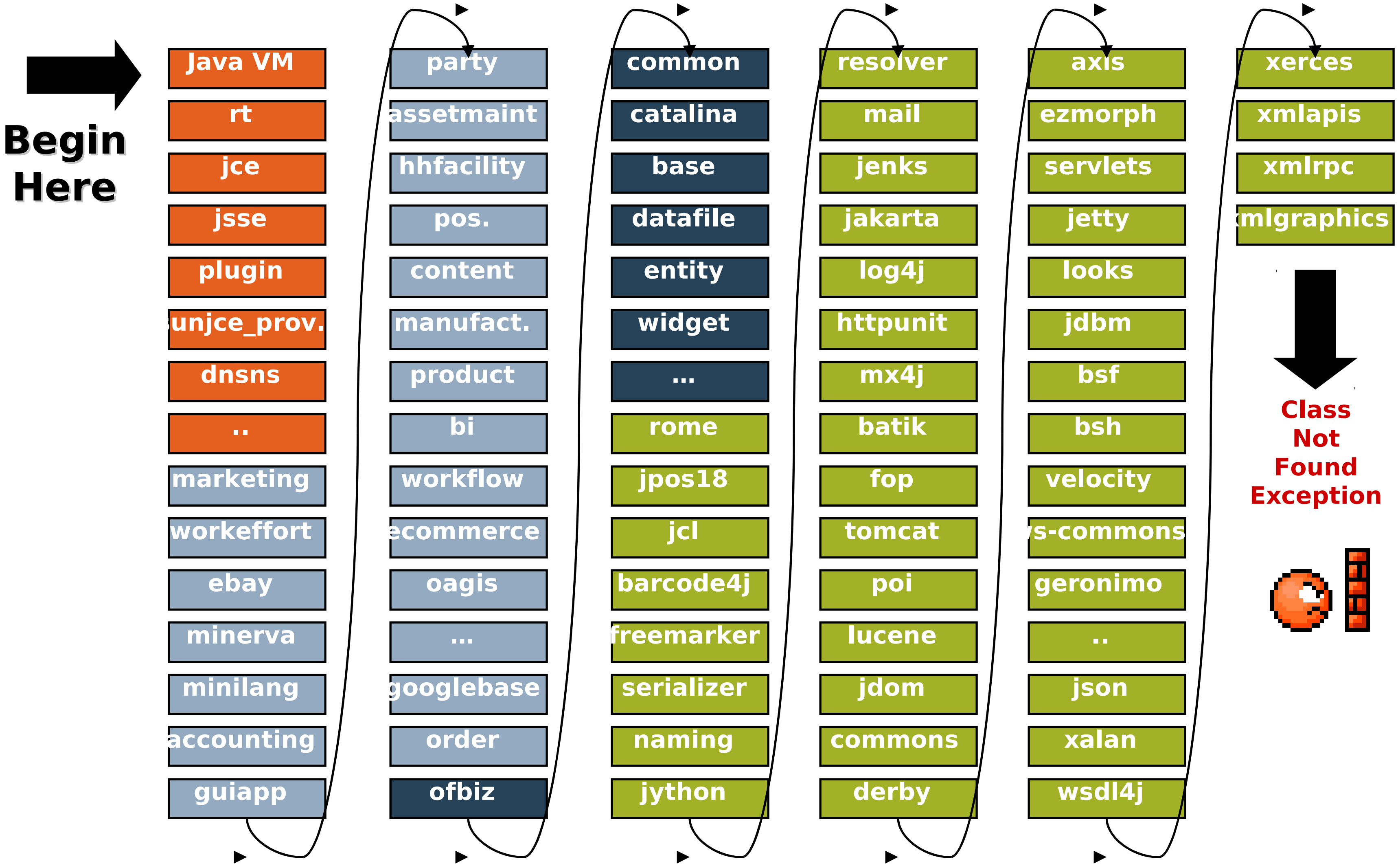
- WAS OSGi FeP provides the application-level OSGi infrastructure required by such web applications, integrated into the application server runtime
- Reduce disk and memory footprint with a shared repository of OSGi modules across different applications
- OSGi bundle versioning enables multiple versions of the same library to be loaded concurrently within a server

Modularization in Java



- Java Platform Modularity
  - Classes encapsulate data and logic.
  - Packages contain classes.
  - Jars contain packages.
- Class visibility
  - private, package private, protected, public.
- What’s missing?
  - No “jar scoped” access modifiers.
  - No way for a jar to declare its dependencies.
  - No versioning.
  - Jars have no modularization characteristics.
- At runtime there is just a collection of classes on a classpath.

Consequences



Begin Here

Java VM
rt
jce
jsse
plugin
unjce_prov.
dnsns
..
marketing
workeffort
ebay
minerva
minilang
accounting
guiapp

party
assetmaint
hhfacility
pos.
content
manufact.
product
bi
workflow
ecommerce
oagis
…
googlebase
order
ofbiz

common
catalina
base
datafile
entity
widget
…
rome
jpos18
jcl
barcode4j
freemarker
serializer
naming
jython

resolver
mail
jenks
jakarta
log4j
httpunit
mx4j
batik
fop
tomcat
poi
lucene
jdom
commons
derby

axis
ezmorph
servlets
jetty
looks
jdbm
bsf
bsh
velocity
s-commons
geronimo
..
json
xalan
wsdl4j

xerces
xmlapis
xmlrpc
mlgraphics

Class Not Found Exception

OSGi Bundles and Class Loading

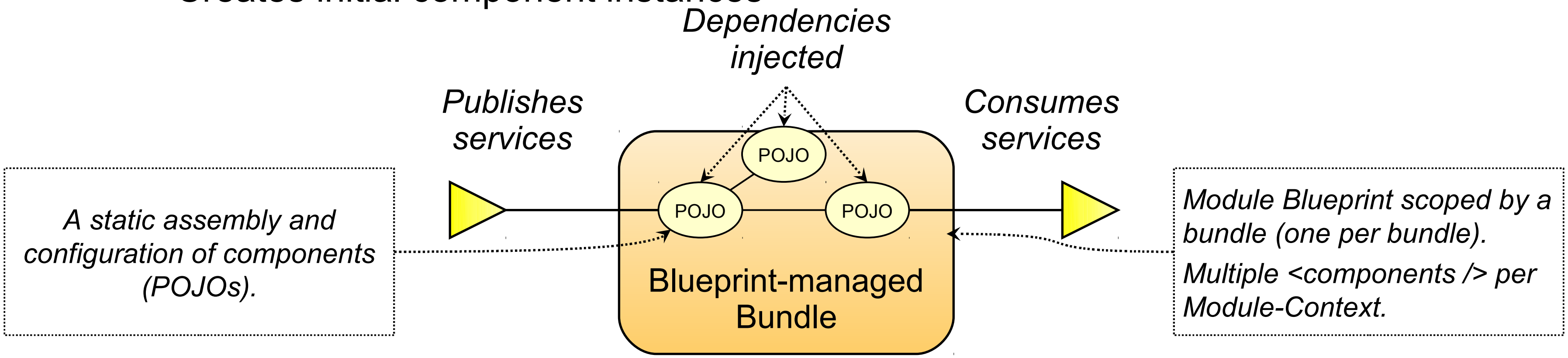
- OSGi Bundle – A jar containing:
  - Classes and resources.
  - OSGi Bundle manifest.
- What’s in the manifest:
  - Bundle-Version: Multiple versions of bundles can live concurrently.
  - Import-Package: What packages from other bundles does this bundle depend upon?
  - Export-Package: What packages from this bundle are visible and reusable outside of the bundle?

- Class Loading
  - Each bundle has its own loader.
  - No flat or monolithic classpath.
  - Class sharing and visibility decided by declarative dependencies, not by class loader hierarchies.
  - OSGi framework works out the dependencies including versions.

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: MyService bundle
Bundle-SymbolicName: com.sample.myservice
Bundle-Version: 1.0.0
Bundle-Activator: com.sample.myservice.Activator
Import-Package:
com.something.i.need;version="1.1.2"
Export-Package: com.myservice.api;version="1.0.0"
```

Blueprint Components

- When bundles are activated – they are checked to see if they are blueprint bundles.
- A blueprint bundle contains one or more Blueprint XML files
- A Blueprint bundle is created and responsible for:
  - Parsing the Blueprint XML files
  - Instantiating
  - Wiring the components together
- The Blueprint container makes sure:
  - the mandatory service references are satisfied
  - Registers all the services into the service registry
  - Creates initial component instances

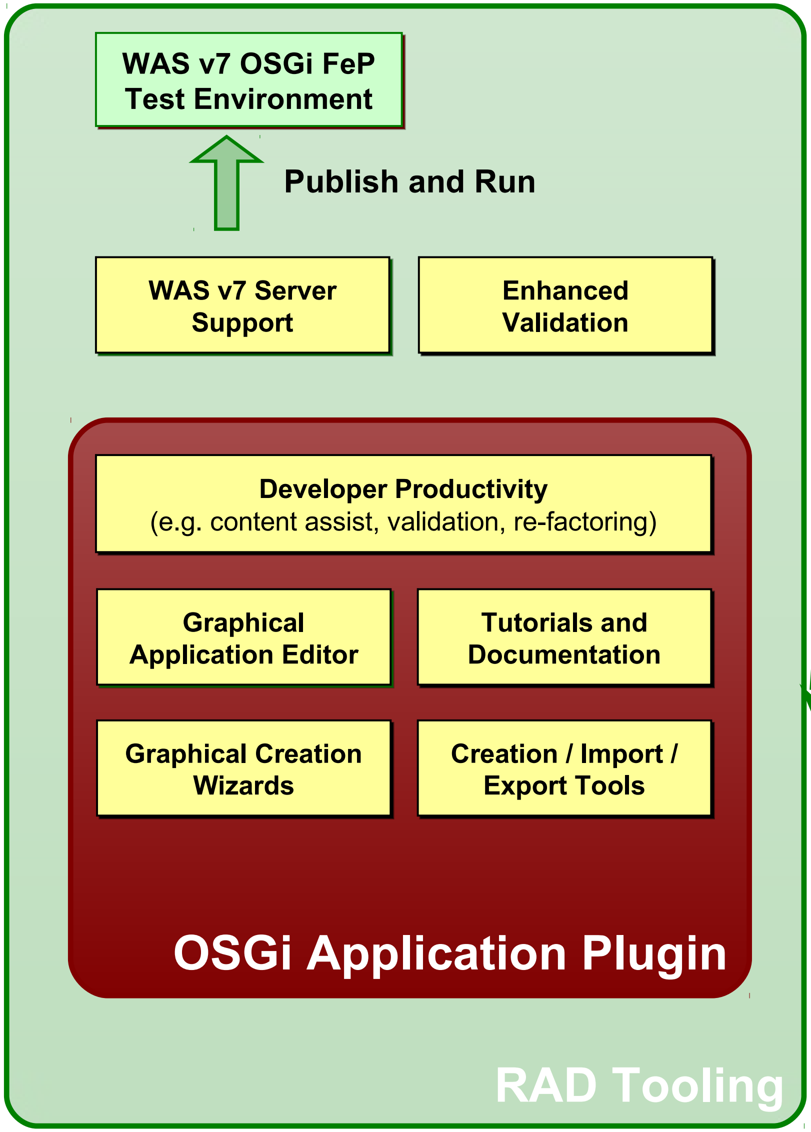


A static assembly and configuration of components (POJOs).

Module Blueprint scoped by a bundle (one per bundle).

Multiple <components /> per Module-Context.

OSGi Application Tools



WAS v7 OSGi FeP Test Environment
Publish and Run
WAS v7 Server Support
Enhanced Validation
Developer Productivity
(e.g. content assist, validation, re-factoring)
Graphical Application Editor
Tutorials and Documentation
Graphical Creation Wizards
Creation / Import / Export Tools
OSGi Application Plugin
RAD Tooling

OSGi Application Support in RAD

Provide integrated development and test of OSGi Applications on the WebSphere platform

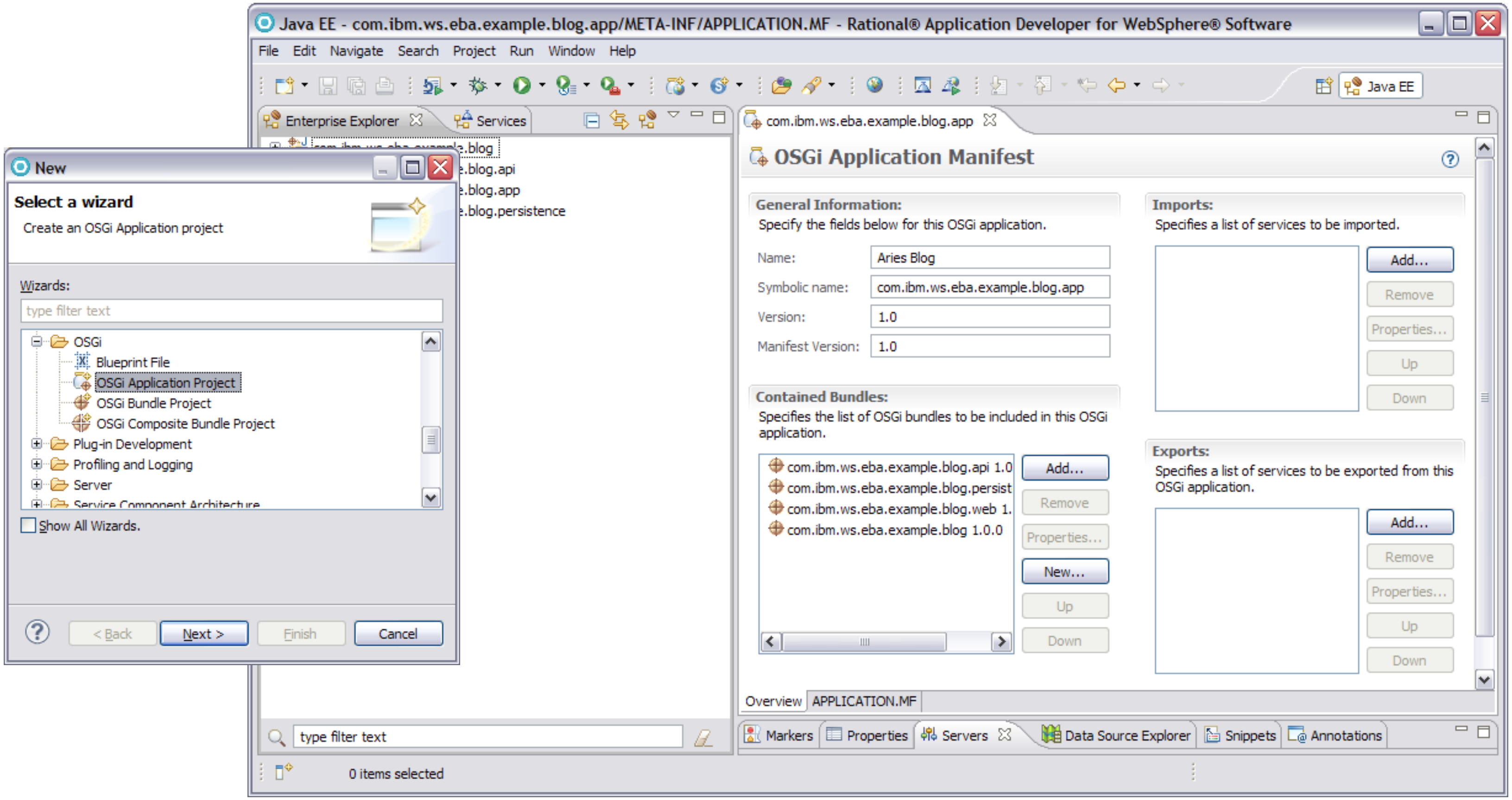
- Integrated with Web Tools, JEE productivity tools, and other capabilities in RAD
- Supports deployment to WAS v7 OSGi FeP and includes the FeP in the WAS Test Environment
- Enhanced validation for application.mf and blueprint.xml

New in V8

RAD OSGi Application Support

RAD provides project support for OSGi application project , bundle projects, composite bundle projects and blueprint files

RAD provides an editor to define the OSGi application manifest to supply or require dependencies (via import & export)



Java EE - com.ibm.ws.eba.example.blog.app/META-INF/APPLICATION.MF - Rational® Application Developer for WebSphere® Software

New

Select a wizard

Create an OSGi Application project

Wizards:

- OSGi
  - Blueprint File
  - OSGi Application Project
  - OSGi Bundle Project
  - OSGi Composite Bundle Project
- Plug-in Development
- Profiling and Logging
- Server
- Service Component Architecture

□ Show All Wizards.

OSGi Application Manifest

General Information:
Specify the fields below for this OSGi application.

Name: Aries Blog
Symbolic name: com.ibm.ws.eba.example.blog.app
Version: 1.0
Manifest Version: 1.0

Contained Bundles:
Specifies the list of OSGi bundles to be included in this OSGi application.

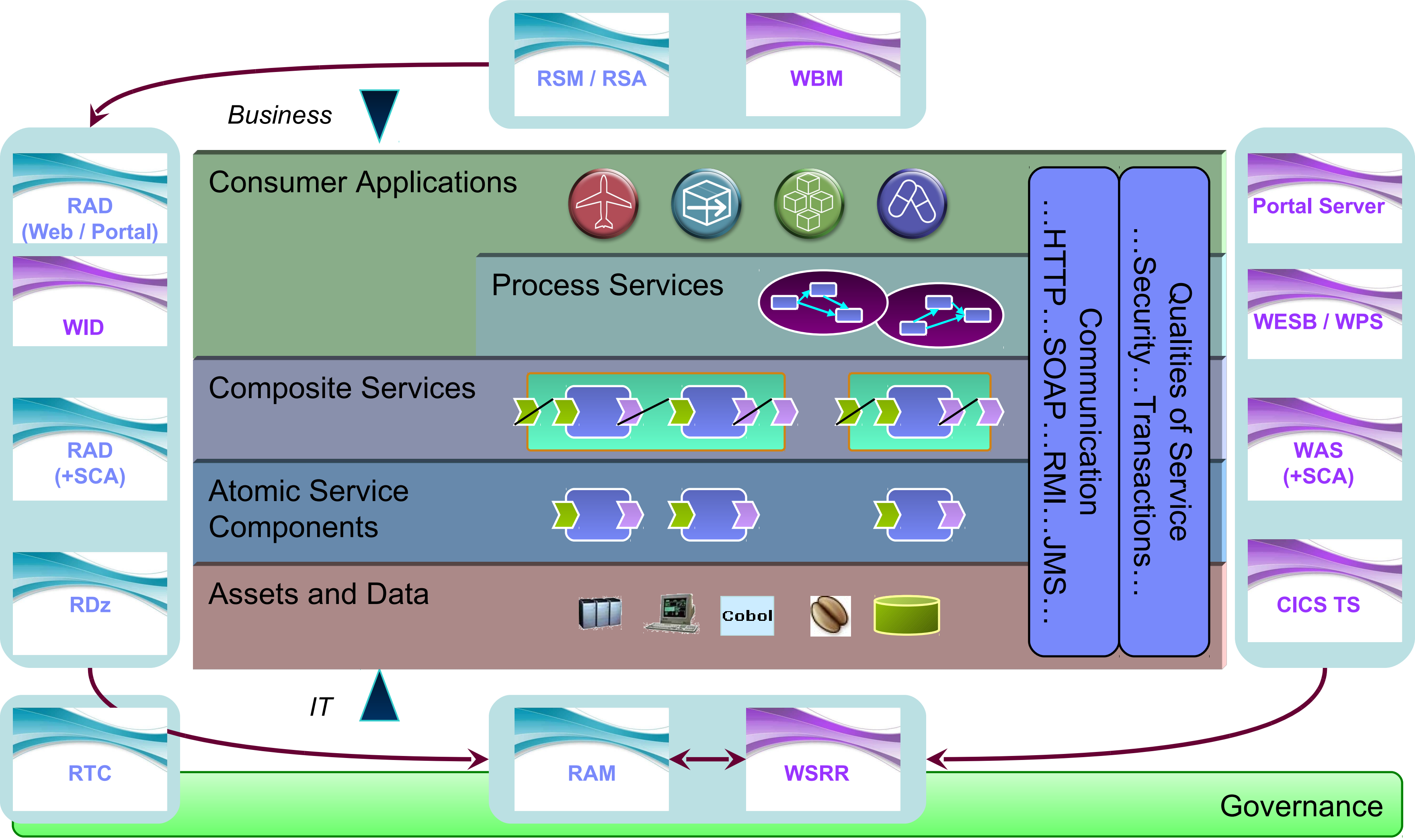
- com.ibm.ws.eba.example.blog.api 1.0
- com.ibm.ws.eba.example.blog.persist
- com.ibm.ws.eba.example.blog.web 1.
- com.ibm.ws.eba.example.blog 1.0.0

Imports:
Specifies a list of services to be imported.

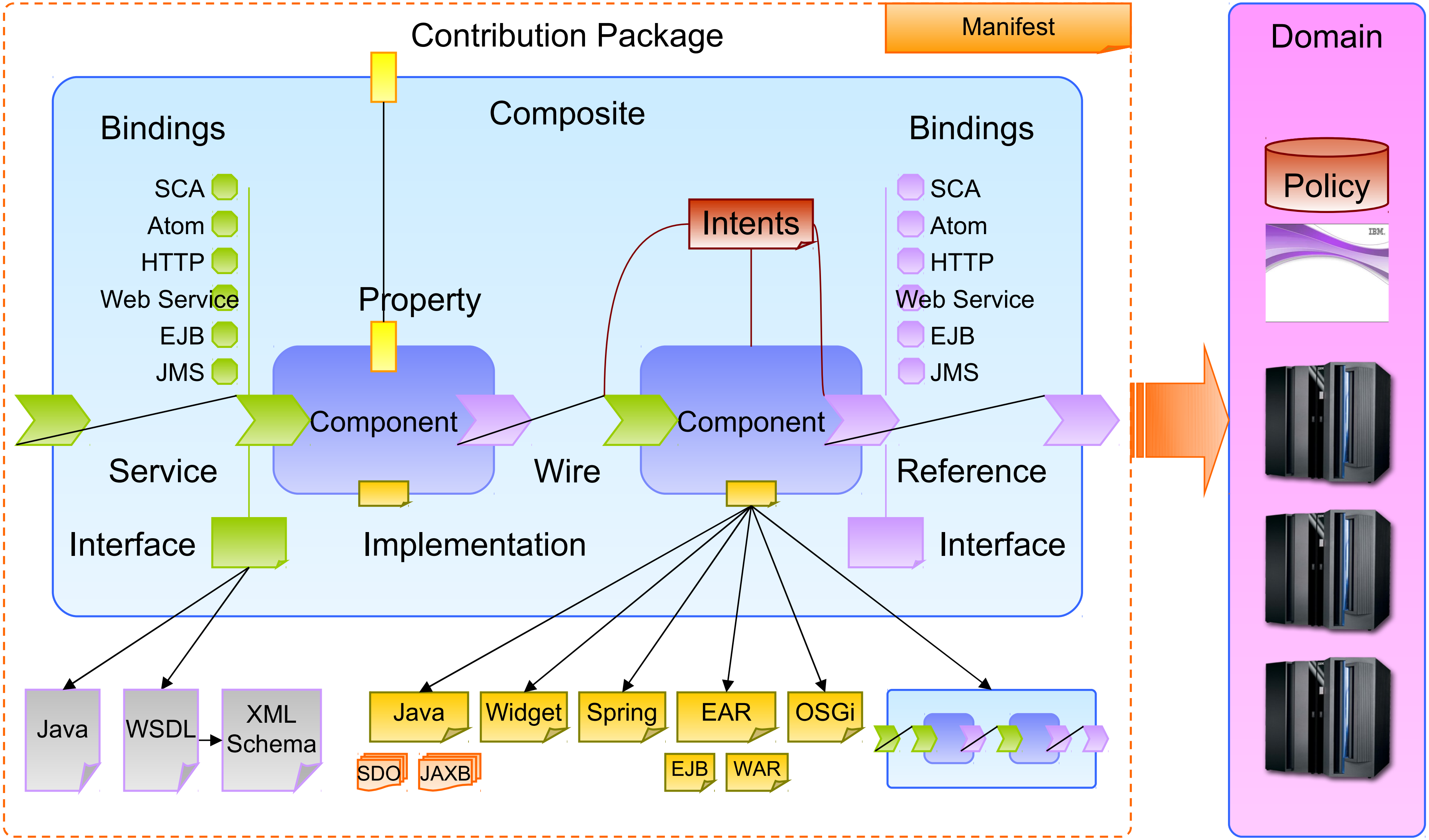
Exports:
Specifies a list of services to be exported from this OSGi application.

0 items selected

SOA Solution Layering



SCA Assembly



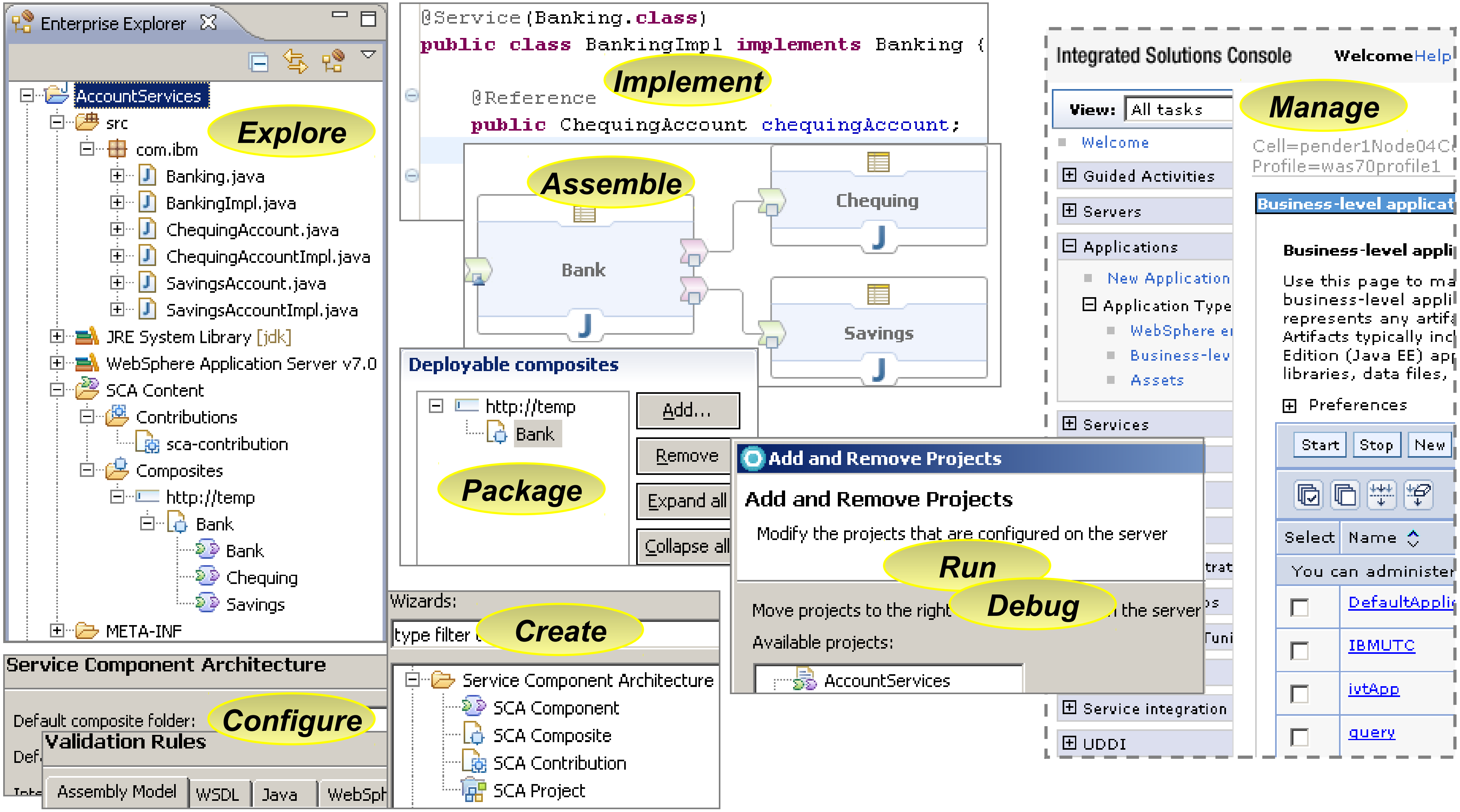
SCA Composition

- An SCA composite consists of 1-n components, each of which:
  - specifies a consistent, neutral, SOA-centric view of the services it provides and the services it requires.
  - specifies its implementation technology.
  - specifies access paths to and from services.
- SCA composites can contain components which utilize a variety of implementation technologies.
- SCA composites promote services it provides and depends upon.
  - Corollary: SCA composites can hide detail that is not relevant for external consumers or providers much like chips hide circuit detail.
- Composite applications exist in other technologies:
  - EAR files are a composition of JEE implementation kinds.
  - OSGi applications are a composition of OSGi bundles.

SCA Composition Considerations

- Tightly-coupled vs. Loosely-coupled composition
  - Wire or Bind vs. Implementation.
    - Service independence, location and management.
    - Can use default binding to make wiring simpler for either style.
- Coarse-grained vs. Fine-grained Policy
  - “Intents” (abstract policies) put QoS within reach of the average developer.
  - Policy is best thought of as a constraint.
  - Coarse-grained (service level) policy can be configured in the assembly.
  - Fine-grained (operational level) policy is frequently deferred to the underlying framework implementation (e.g. JEE, OSGi) policy mechanisms.
    - Policy specified at the service level constrains all operations of the service.

Composition in Rational Application Developer



```
@Service(Banking.class)
public class BankingImpl implements Banking {

    @Reference
    public ChequingAccount chequingAccount;
```

Integrated Solutions Console   Welcome Help

View: All tasks

Cell=pender1Node04C
Profile=was70profile1

Business-level applicat

Business-level appli

Use this page to ma
business-level appli
represents any artif
Artifacts typically inc
Edition (Java EE) app
libraries, data files,

Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right ... the server

Available projects:

AccountServices

Deployable composites

Default composite folder:

More Information

- RAD Home Page
  - Product Editions and Extensions
  - Trials
  - Features and Benefits, System Requirements
- RAD WIKI on developerWorks
  - Technical Resources:
    - Demos, Documentation, Whitepapers, Sample models

Thank
You