



J-Spring

16 april 2008 Spant! - Bussum



# Integrating OSGi in an enterprise java (web)application

Hallo Khaznadar

GX Creative online development



## About me

Hallo Khaznadar

- Past
  - Senior Engineer at Compuware Europe B.V.
  - Architect at Ponte Vecchio B.V.
  - Software/Hardware R&D Engineer/Architect at Inzicht Informatietechniek B.V. / Euro Partners B.V.
- Current
  - Product Architect at GX Creative Online Development B.V.
- Education
  - B.Sc. Computer Engineering at Baghdad University.
- Interests
  - Embedded (software/Hardware), distributed computing, web based applications.
- Email
  - [Hallo.Khaznadar@gx.nl](mailto:Hallo.Khaznadar@gx.nl)



## About <GX>

<GX>

GX creative online development

Web Content Management System (WCMS)

Privately held / 140+ employees

**Nijmegen, Amsterdam, Eindhoven (NL)**

Founded in 1995

First Java based CMS in 1998

<http://www.gxdeveloperweb.com>



## Presentation Overview

- Introduction to WCMS.
- Introduction to OSGi.
- Why OSGi ?
- OSGi explained.
- OSGi in GX WM.
- Conclusion.
- Demo.
- Q&A.



## Introduction to WCMS

**Web Content Management System.**

A web content management system is a software system used to manage and control a large, dynamic collection of web materials (HTML documents and their associated resources).



## Introduction to WCMS

- What are the key features of a WCMS?

- Automated templates
- Easily editable content
- Workflow management
- Document management
- Content virtualization



Easy Functional  
extension

- **Scalable feature sets**
- **Web standards upgrades**



Hard to achieve within  
a framework without a  
backing infrastructure.



## Introduction to OSGi

Open Service Gateway initiative

<http://www.osgi.org/>



## Introduction to OSGi

- What does OSGi provide?
  - **Component-Based.**
  - **Service-Oriented.**
  - **Standard Manageable software lifecycle.**

dynamic module system for Java.



## Introduction to OSGi

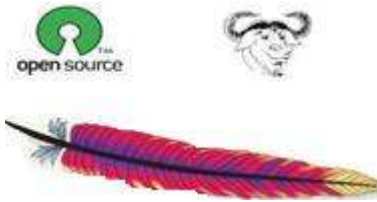
- The history of OSGi and its present.



1999



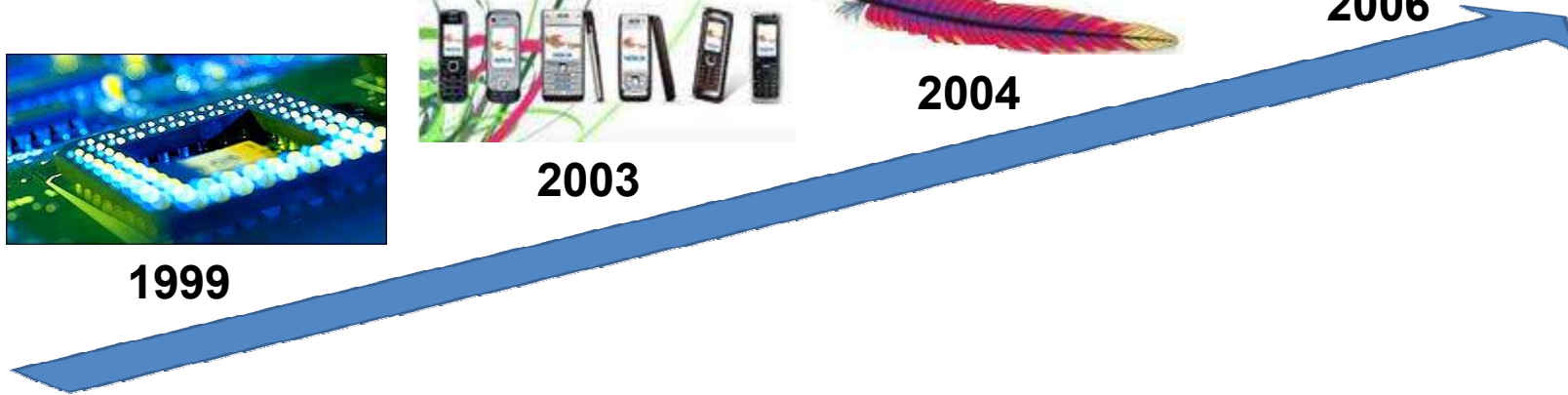
2003



2004



2006





# Introduction to OSGi

- The open source implementations:
  - Felix (Apache implementation)
  - Equinox (The eclipse implementation of OSGi and the foundation for Eclipse plug-in model).
  - Knopflerfish.
  - ...and others.
- Significant Enterprises:
  - BEA.
  - Eclipse.
  - IBM (WebSphere, Lotus).
  - Siemens Communications.
- Applications Frameworks:
  - Spring OSGi Dynamic Modules.
  - IPOJO.
  - Sling.
- Automotive Electronics:
  - BMW.
  - VOLVO.



## Introduction to OSGi



**PHILIPS**

sense and simplicity

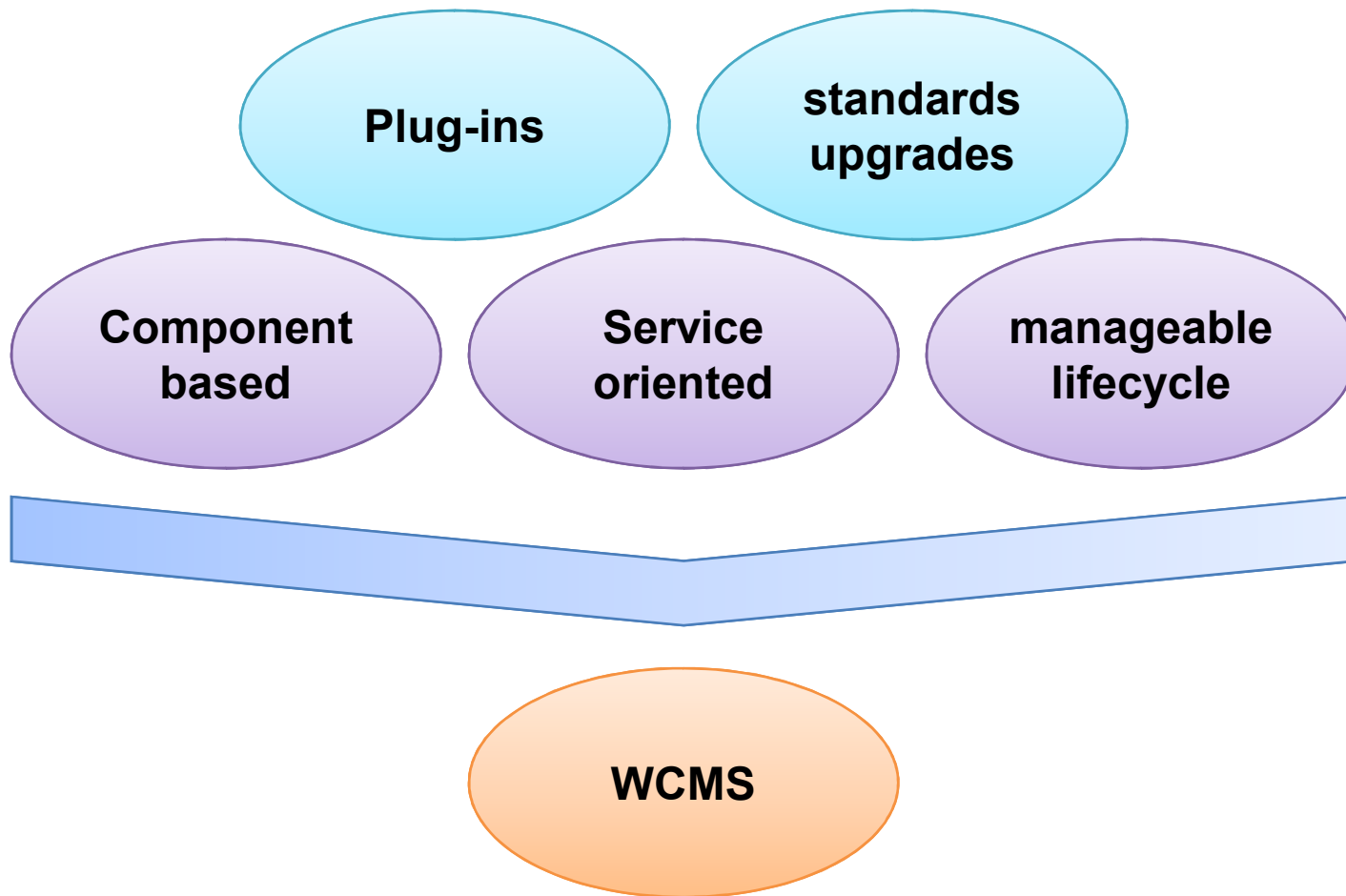
**NOKIA**

**SIEMENS**





## Why OSGi ?





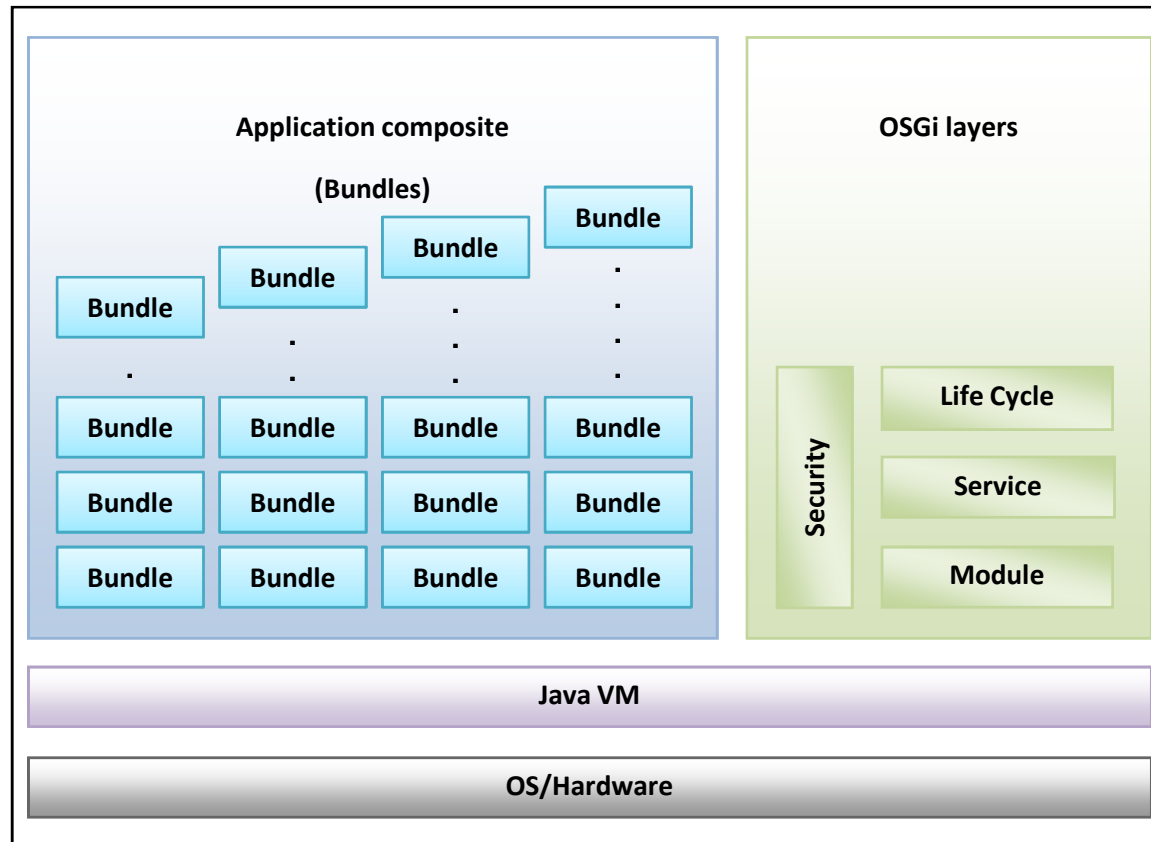
## OSGi explained

- How does OSGi do what it promises?
  - Partition the system into modules (**bundles**).
  - Strict visibility rules.
  - Bundle versioning.
  - Lifecycle management of a bundles.
  - Service oriented capabilities to the VM.



# OSGi explained

- Application Architecture with OSGi

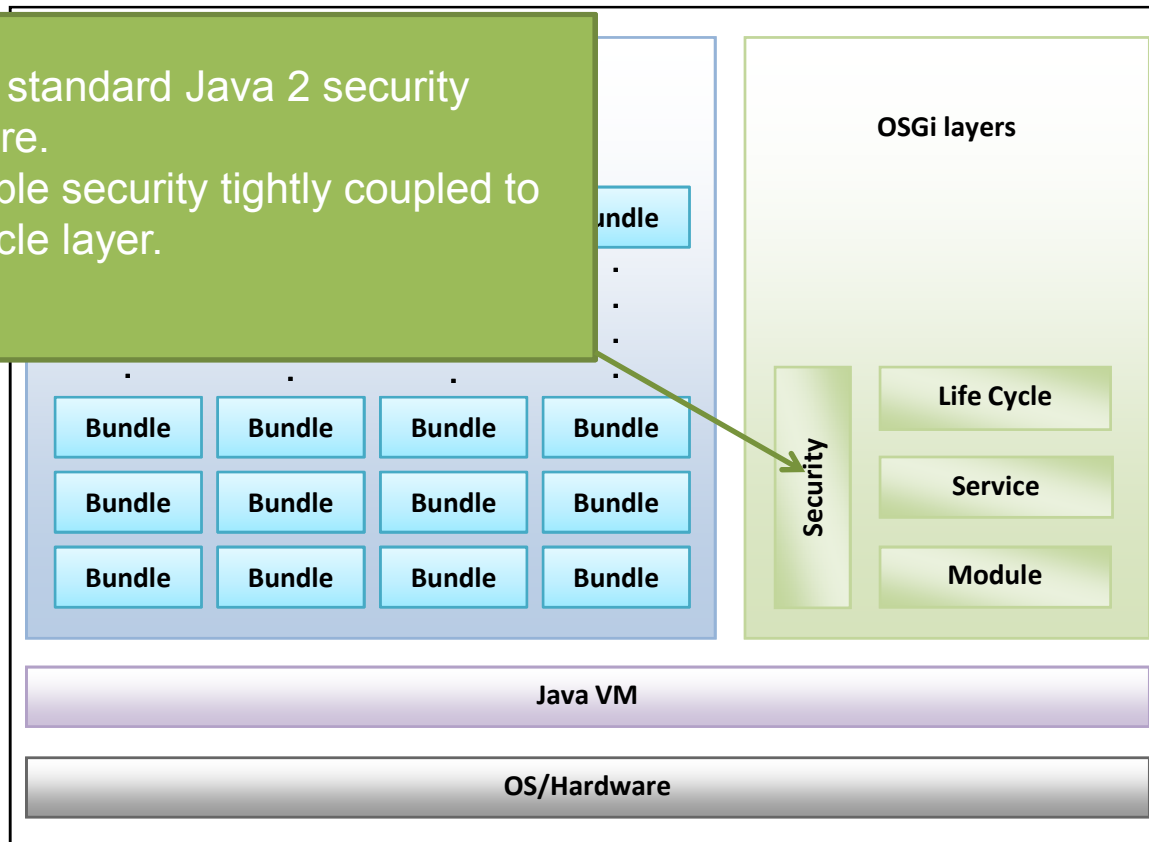




# OSGi explained

- Application Architecture with OSGi

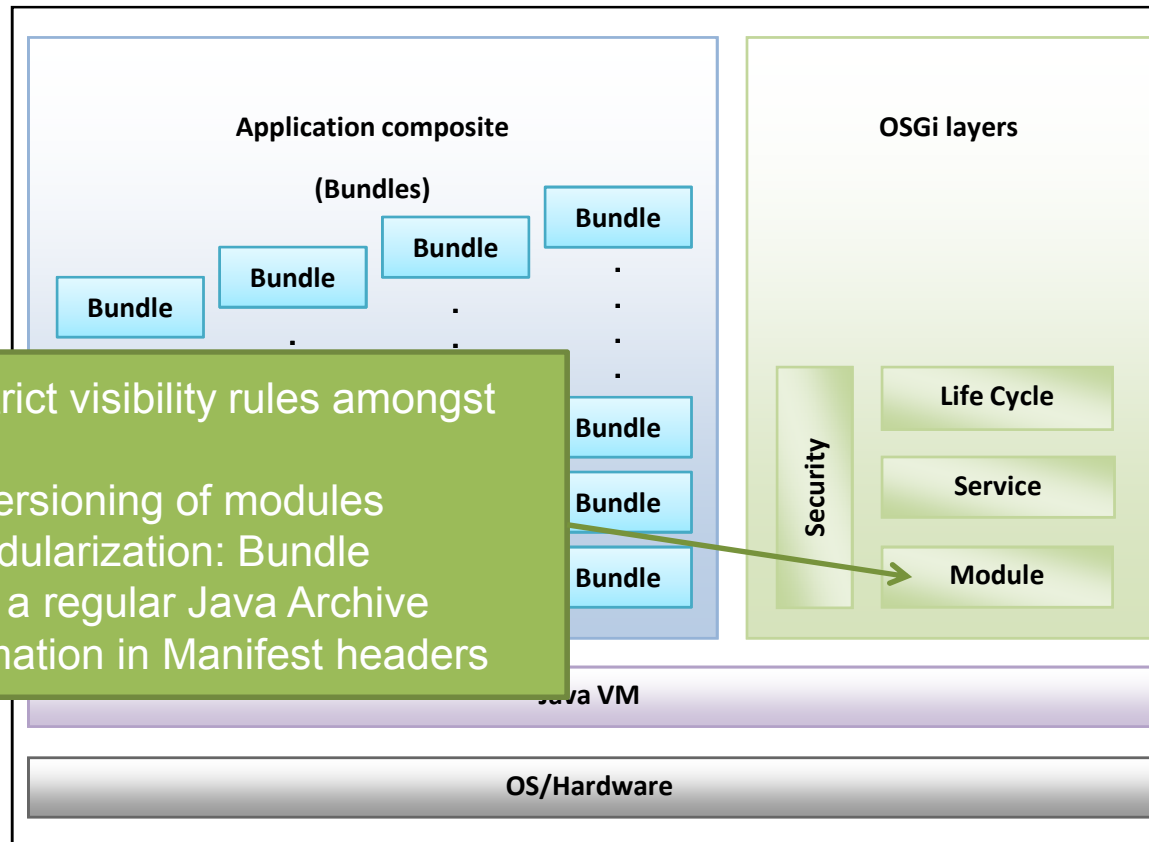
- Based on standard Java 2 security architecture.
- Manageable security tightly coupled to the life cycle layer.
- Optional.





# OSGi explained

- Application Architecture with OSGi

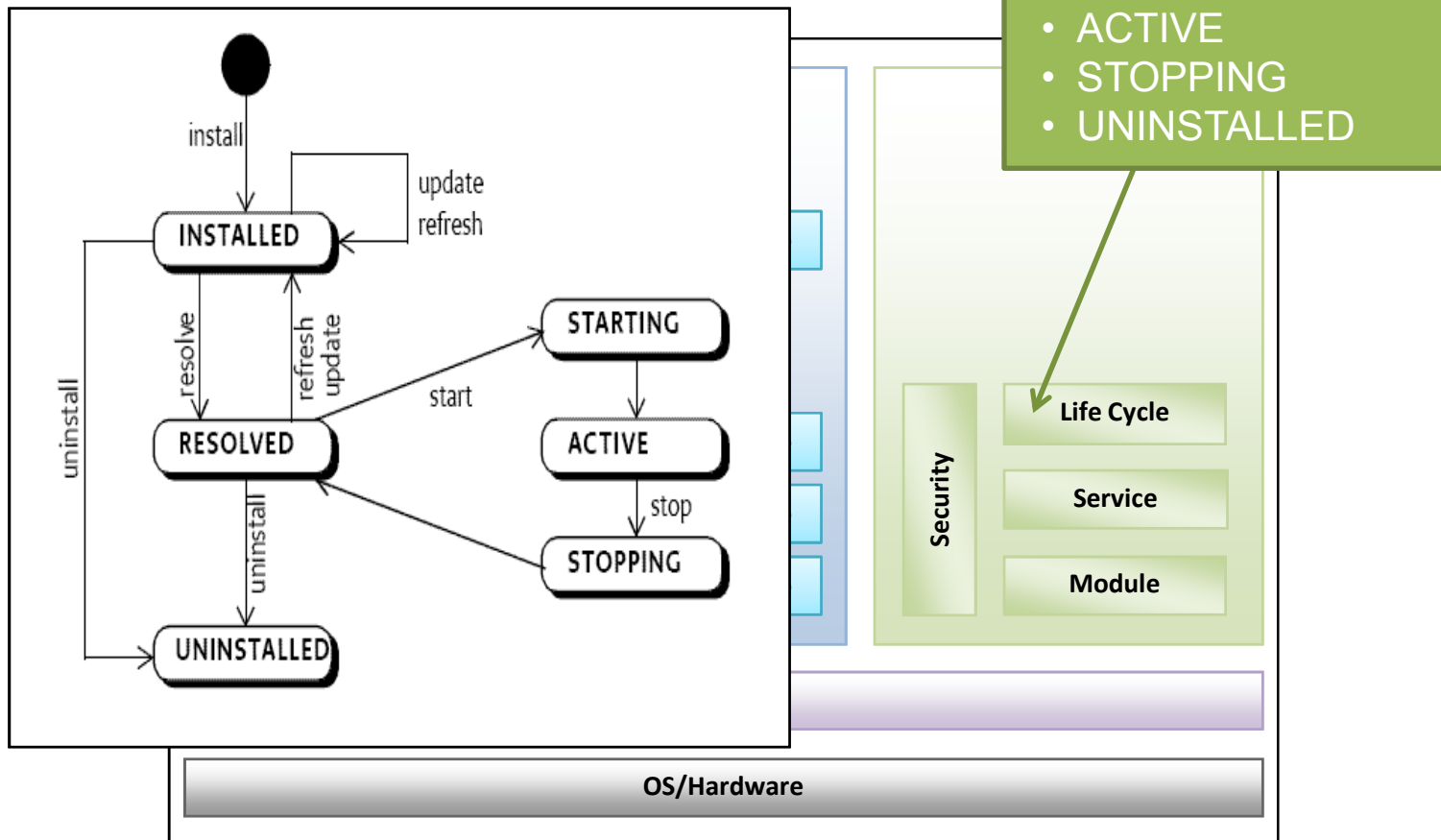


- Enforces strict visibility rules amongst modules
- Supports versioning of modules
- Unit for modularization: Bundle
- A bundle is a regular Java Archive
- Meta information in Manifest headers



# OSGi explained

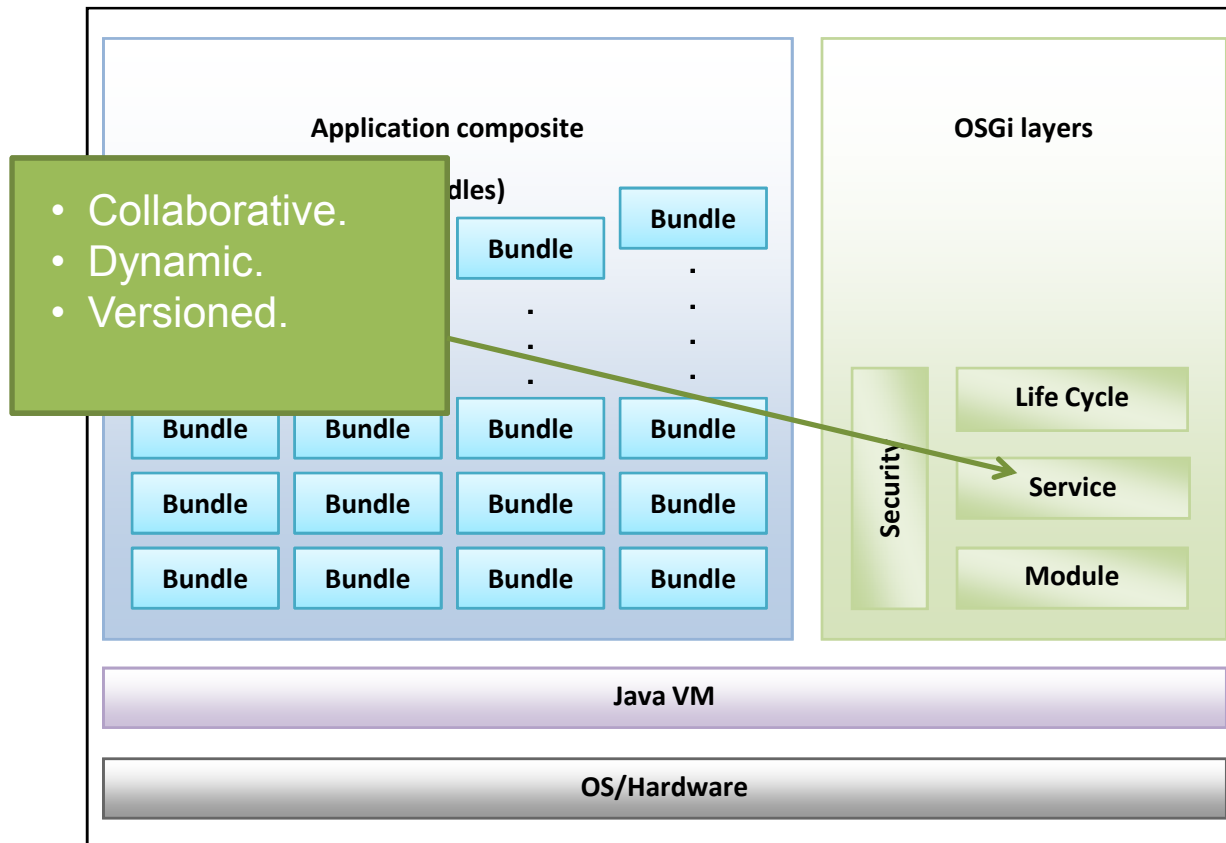
- Application Architecture with OSGi





# OSGi explained

- Application Architecture with OSGi



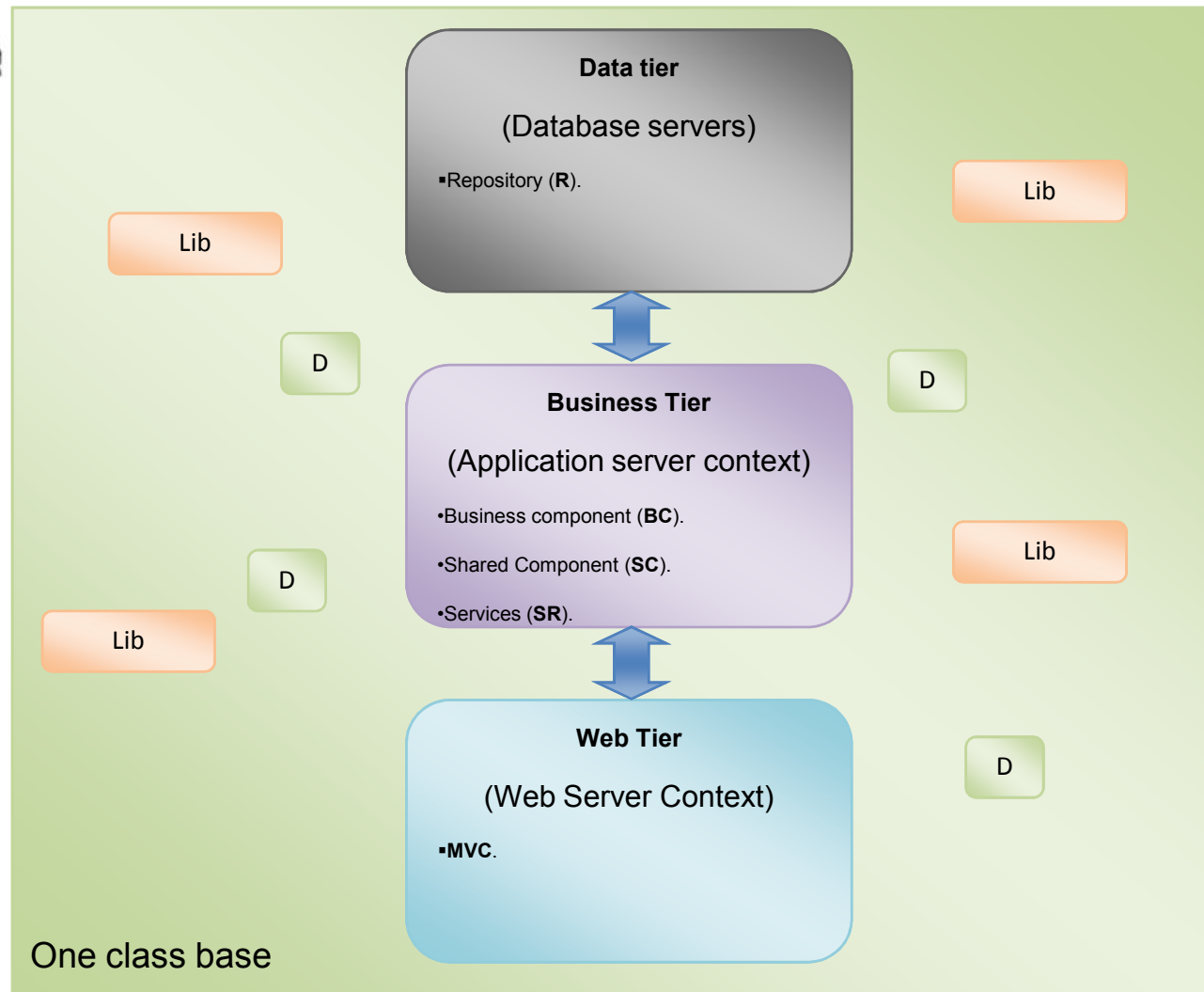


## OSGi explained

- Key concepts.
  - Controlled versioned visibility.
  - Bundle Class loading delegation model.
  - Controlled dynamic operation.

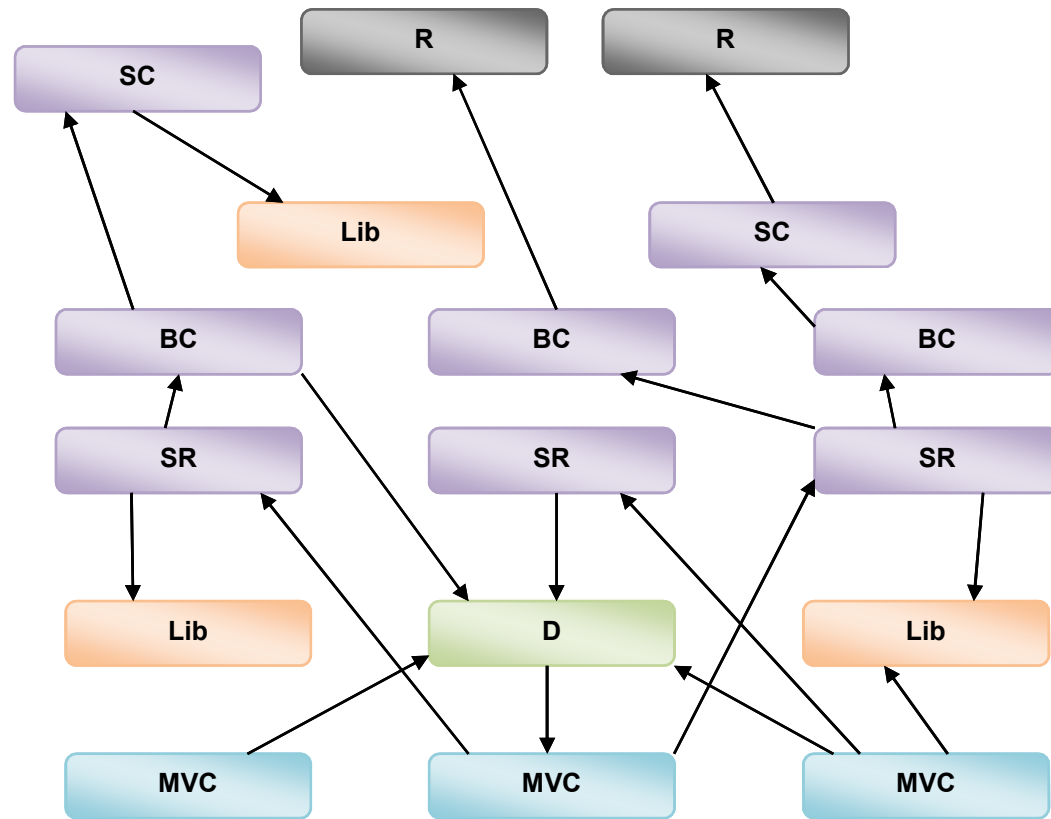


# OSGi explained





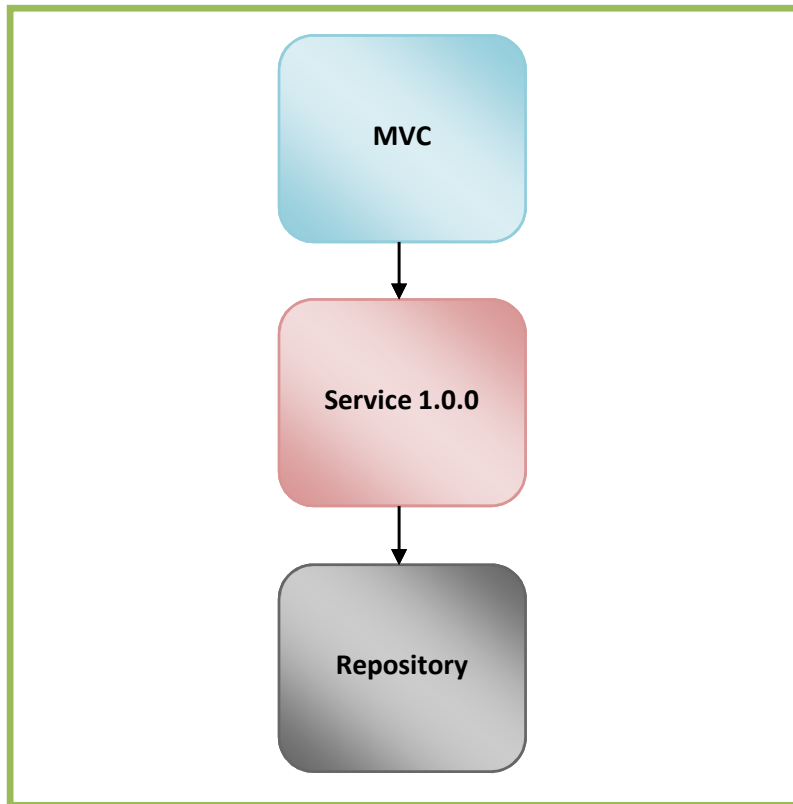
# OSGi explained



Bundle class space

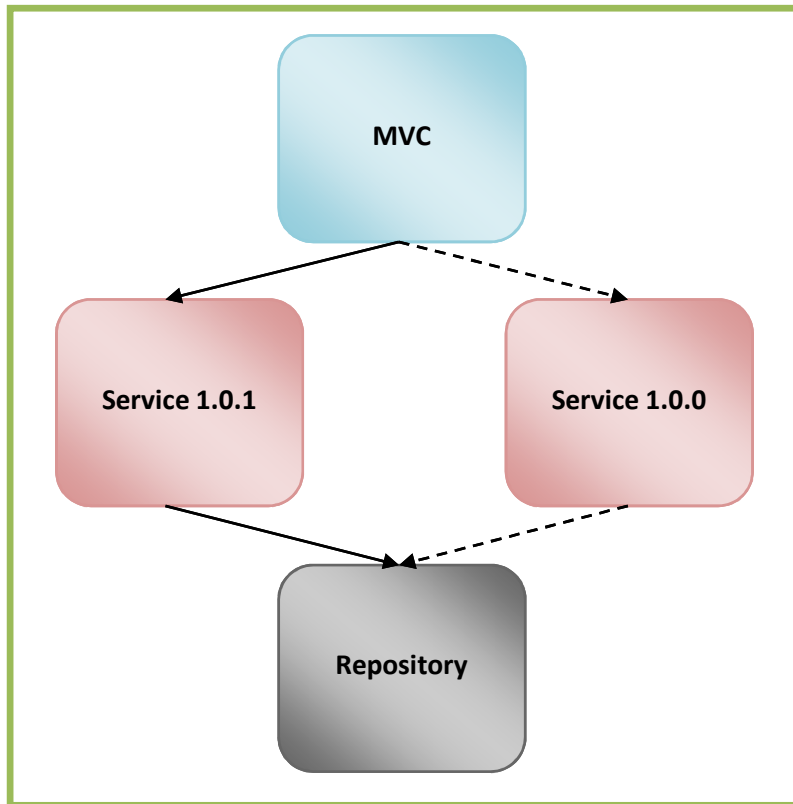


# OSGi explained





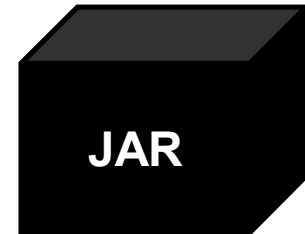
# OSGi explained





## OSGi explained

- OSGi Bundle  
META-INF/MANIFEST.MF

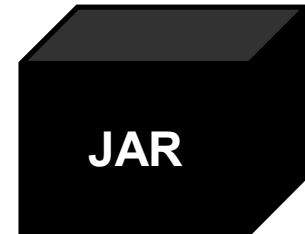


```
Bundle-Name: Logging service
Bundle-SymbolicName: nl.gx.webmanager.services.LogService
Bundle-Description: An implementation of the GX WebManager LogService using
commons logging
Bundle-UpdateLocation: http://update.gx.nl/repo/services/LogService.jar
Bundle-Activator: nl.gx.webmanager.services.LogService.LogServiceActivator
Bundle-Vendor: GX Creative Online Development
Bundle-Version: 1.1.2
Export-Service: nl.gx.webmanager.services.LogService
Import-Package: org.apache.commons.logging
```



## OSGi explained

- OSGi Bundle  
META\_INF/MANIFEST.MF

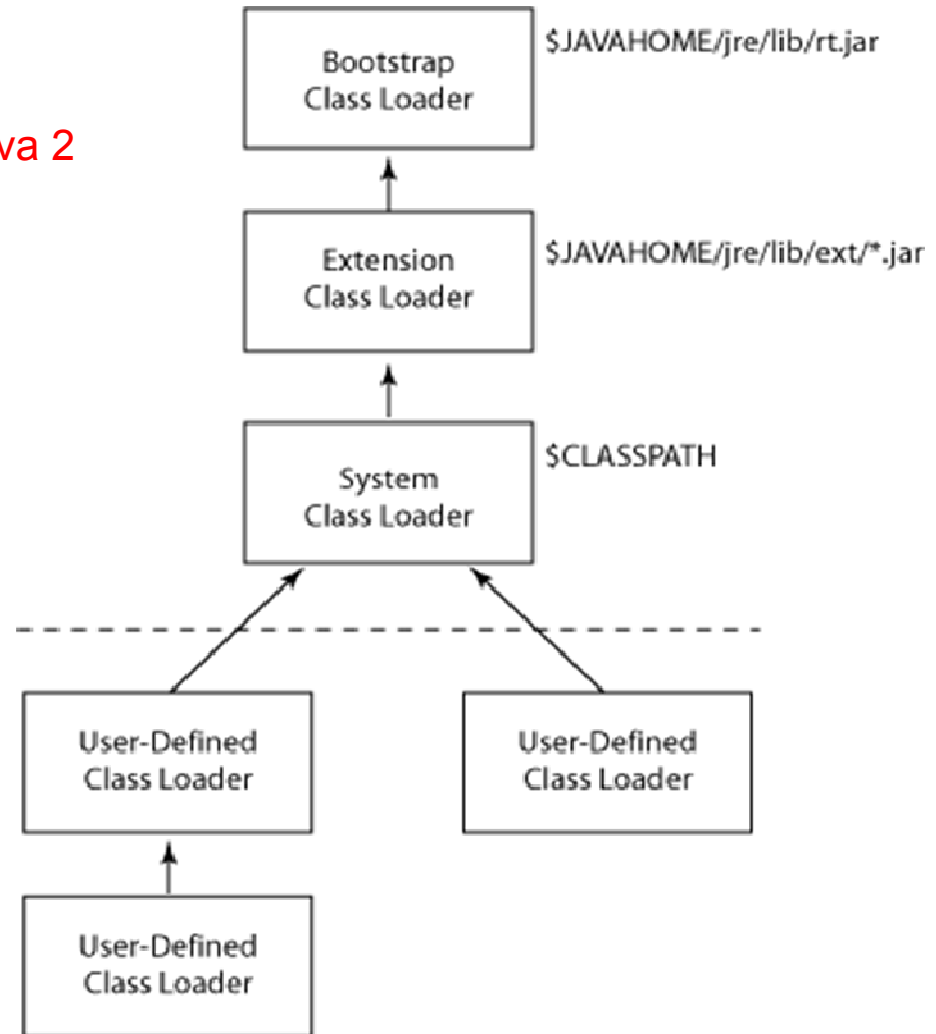


```
Built-By: hallok  
Bundle-Vendor: GX Creative Online Development  
Bundle-Name: GX WebManager JUnit Test Runner  
Bundle-Description: Executes JUnit test cases in the WebManager environment  
Bundle-Activator: nl.gx.webmanager.services.testrunner.impl.TestRunnerActivator  
Import-Package:  
javax.xml.parsers,javax.xml.transform,javax.xml.transform.dom,javax.xml.transform.  
sax,javax.xml.transform.stream,junit.framework,nl.gx.webmanager.services.testrunne  
r,org.apache.felix.dependencymanager  
Export-Package: nl.gx.webmanager.services.testrunner  
Bundle-Classpath: .,ant-1.6.2.jar,ant-junit-1.6.2.jar,ant-nodeps-1.6.2.jar
```



# OSGi explained

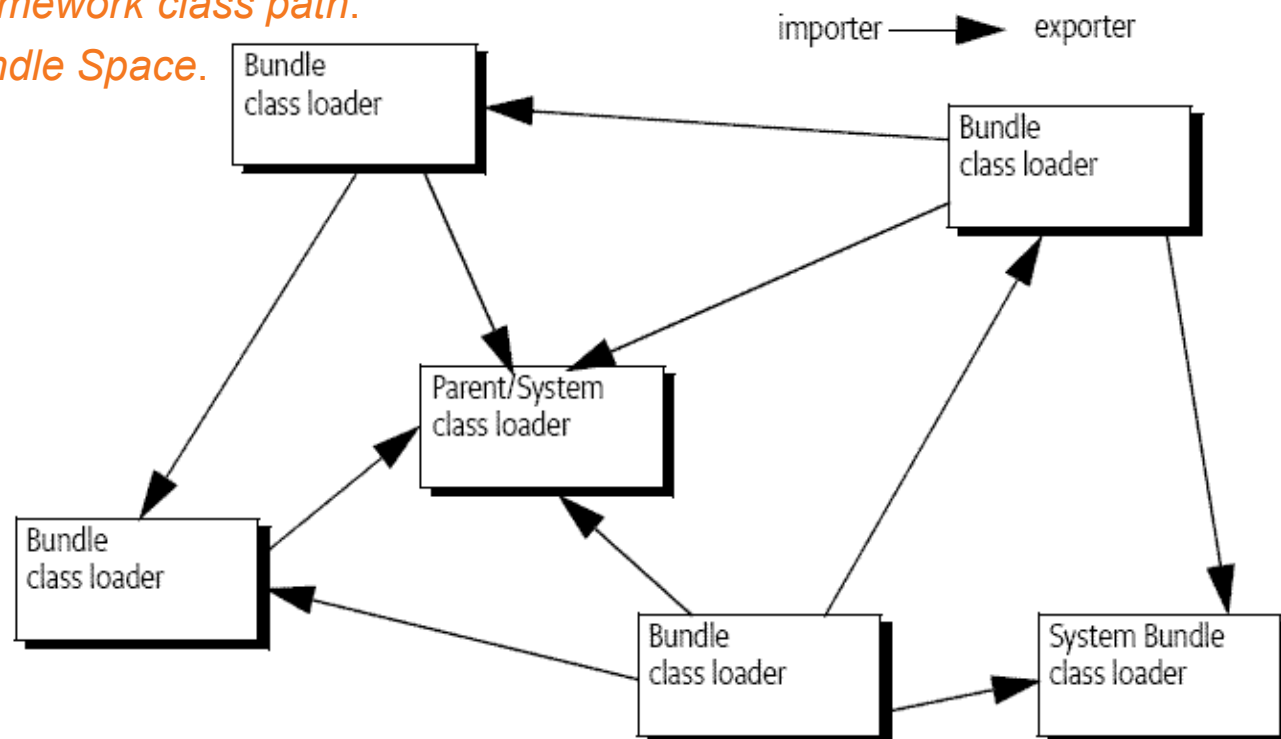
- Traditional class loading
  - Delegation model since Java 2
  - Parent first delegation





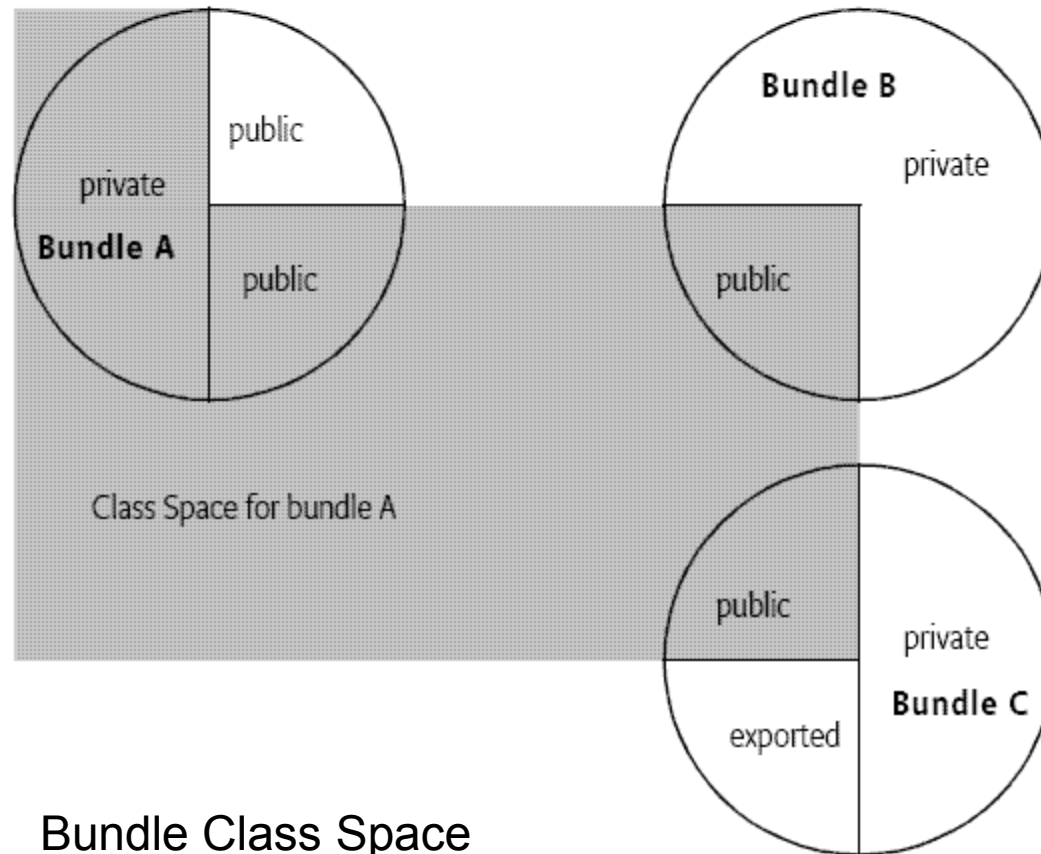
# OSGi explained

- Bundle class loading:
  - Bundle class loading delegation model
    - *Boot class path.*
    - *Framework class path.*
    - *Bundle Space.*





# OSGi explained

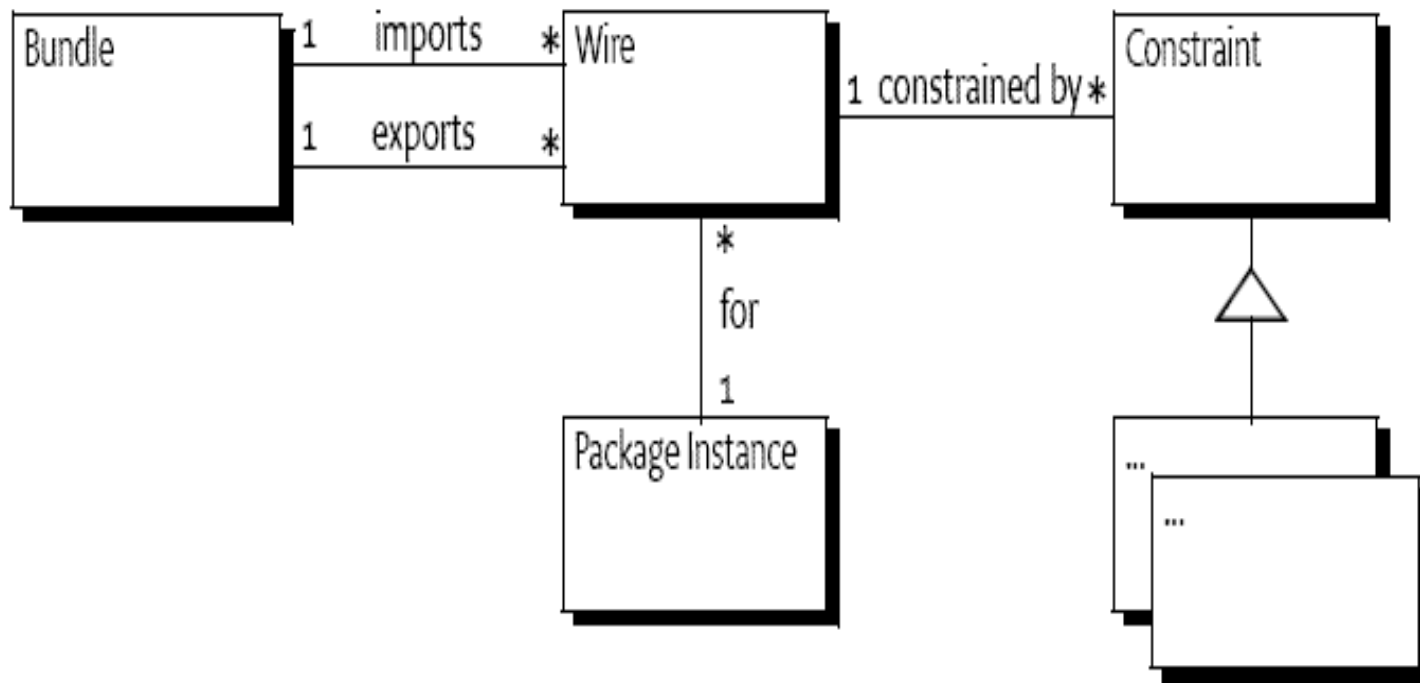


Bundle Class Space



# OSGi explained

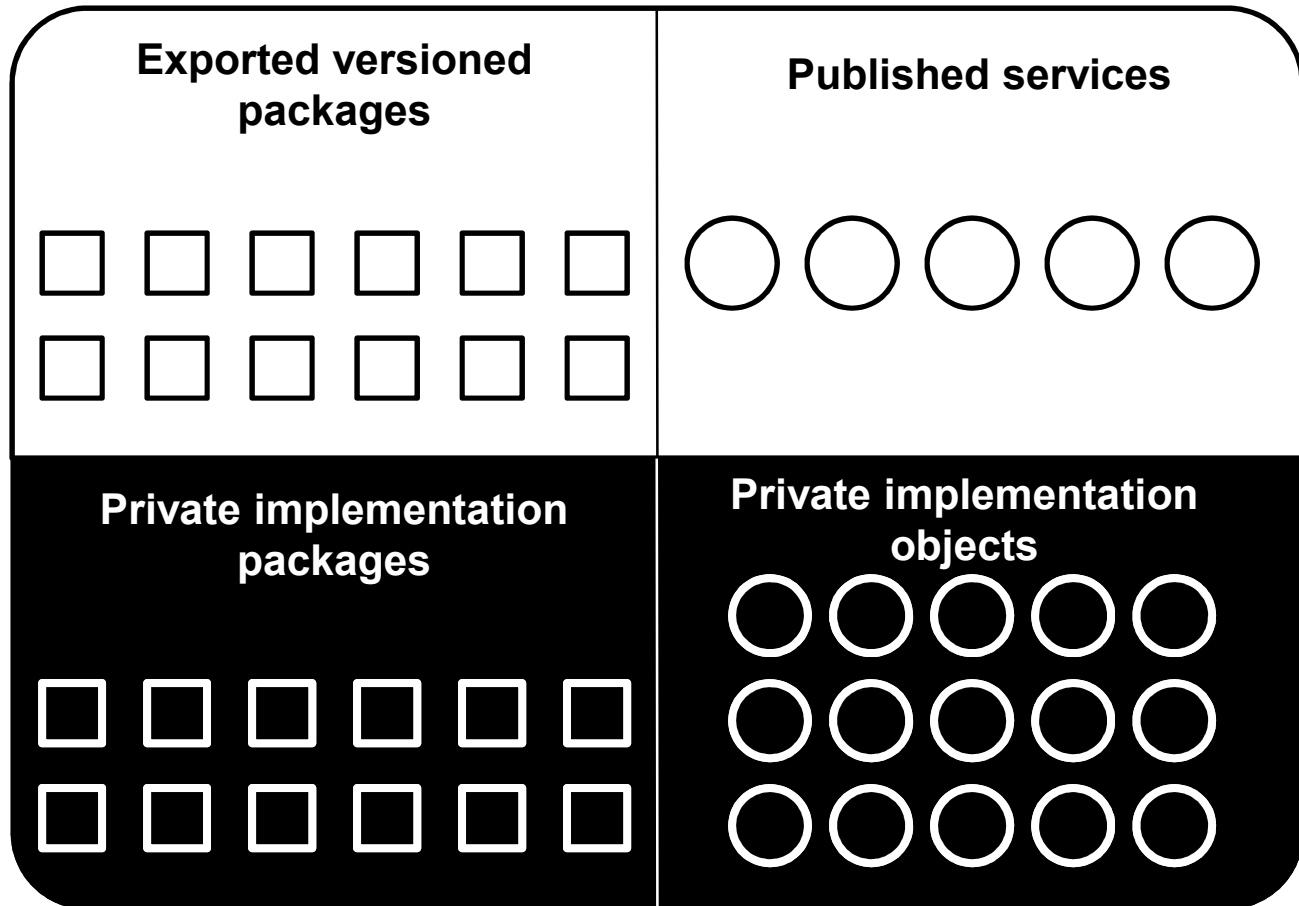
- Bundle Resolving.





# OSGi explained

**Bundle anatomy**





## OSGi explained

Summarizing the roles of the different parts of an OSGi based framework

The service object is owned by, and runs within, a bundle.

This bundle must register the service object with the Framework service registry so that the service's functionality is available to other bundles under control of the Framework.

Dependencies between the bundle owning the service and the bundles using it are managed by the Framework.

The Framework maps services to their underlying service objects, and provides a simple but powerful query mechanism that enables a bundle to request the services it needs.

The Framework also provides an event mechanism so that bundles can receive events of services that are registered, modified, or unregistered.

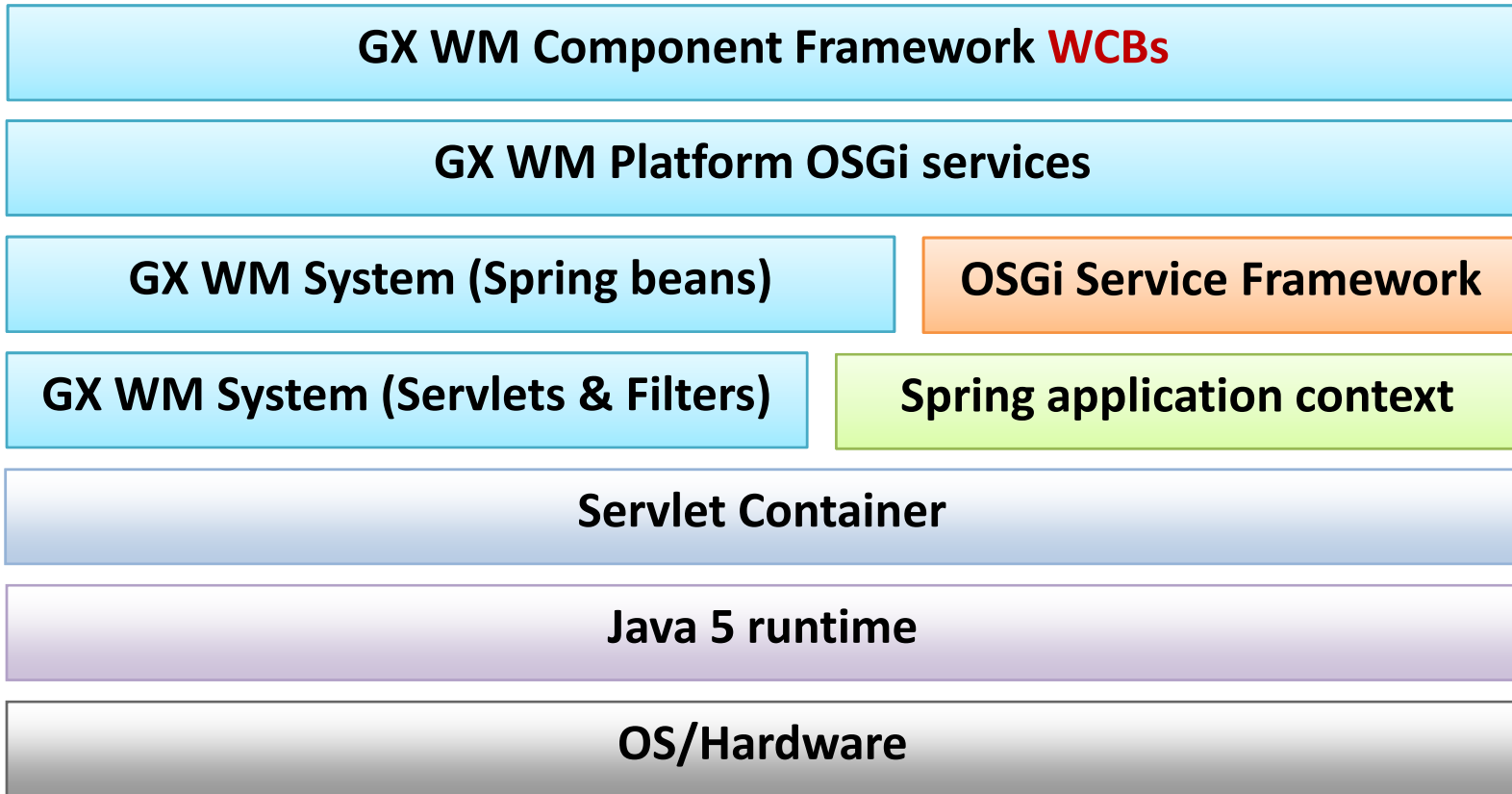


## OSGi in GX WM

### **WebManager Architecture**



## OSGi in GX WM



**WebManager Architecture**



## OSGi in GX WM

- WCBs

- What is a WCB?

- WebManager **C**omponent **B**undle

- What does it do?

- A WCB is an OSGi bundle that publishes one or more components.

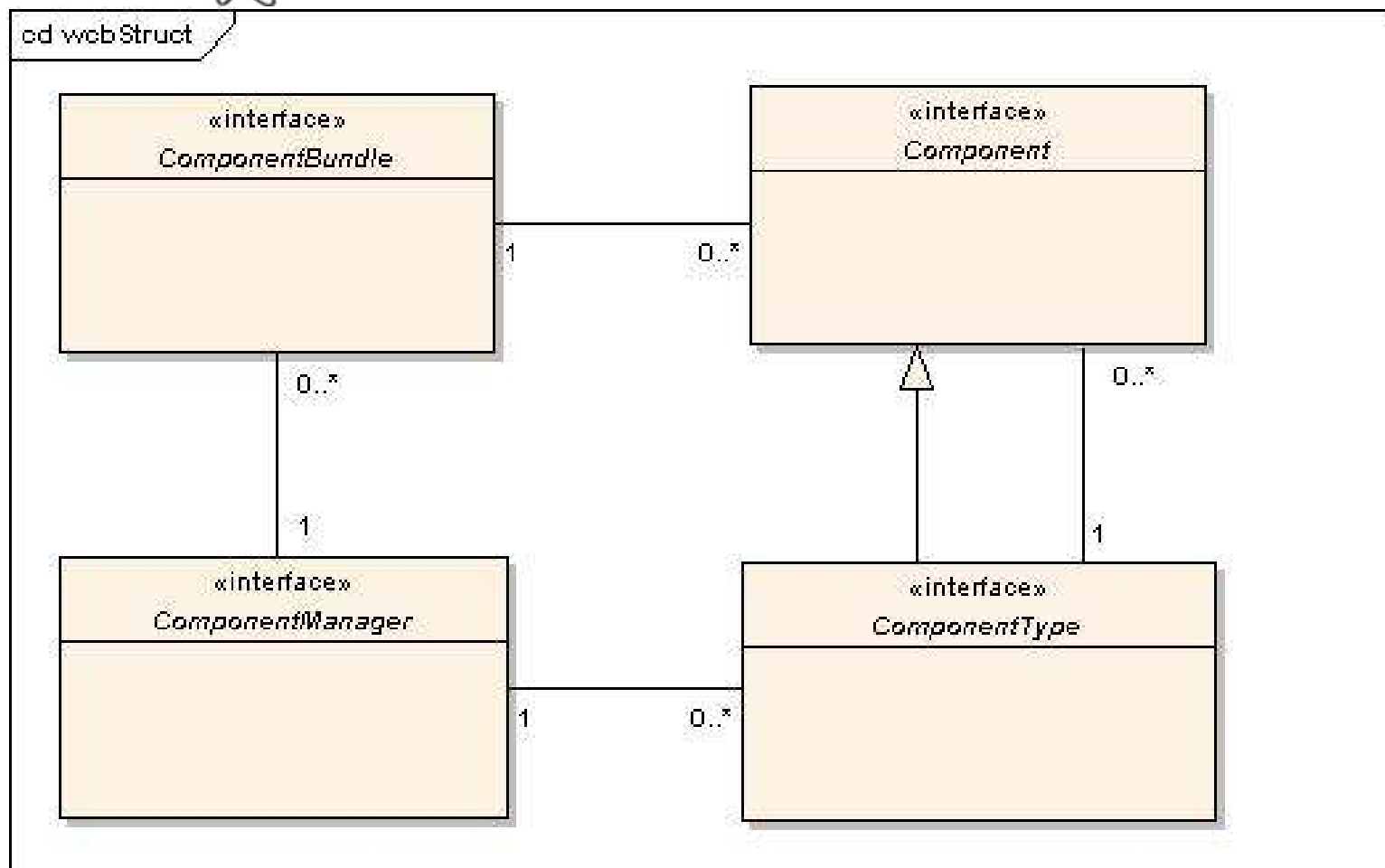
- What is a component?

- WebManager specific Such.

- element, panel, media item, service, task.**



## OSGi in GX WM





## OSGi in GX WM

What are the other WM specific capabilities of a WCB ?

Dependency injected automatically

Licensable.

Authorization capability.

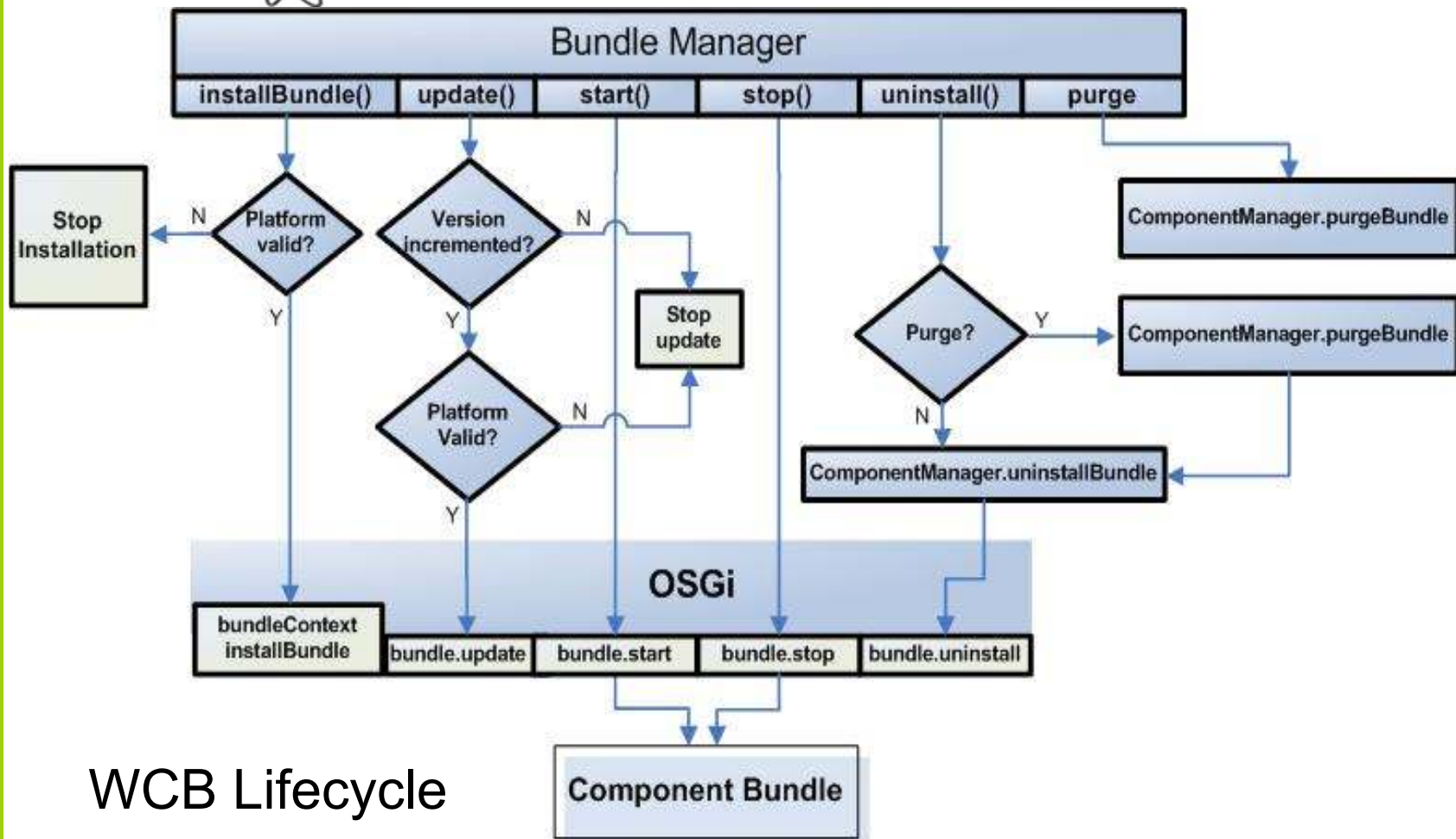
Resources.

Persistent capable.

Extended life cycle.



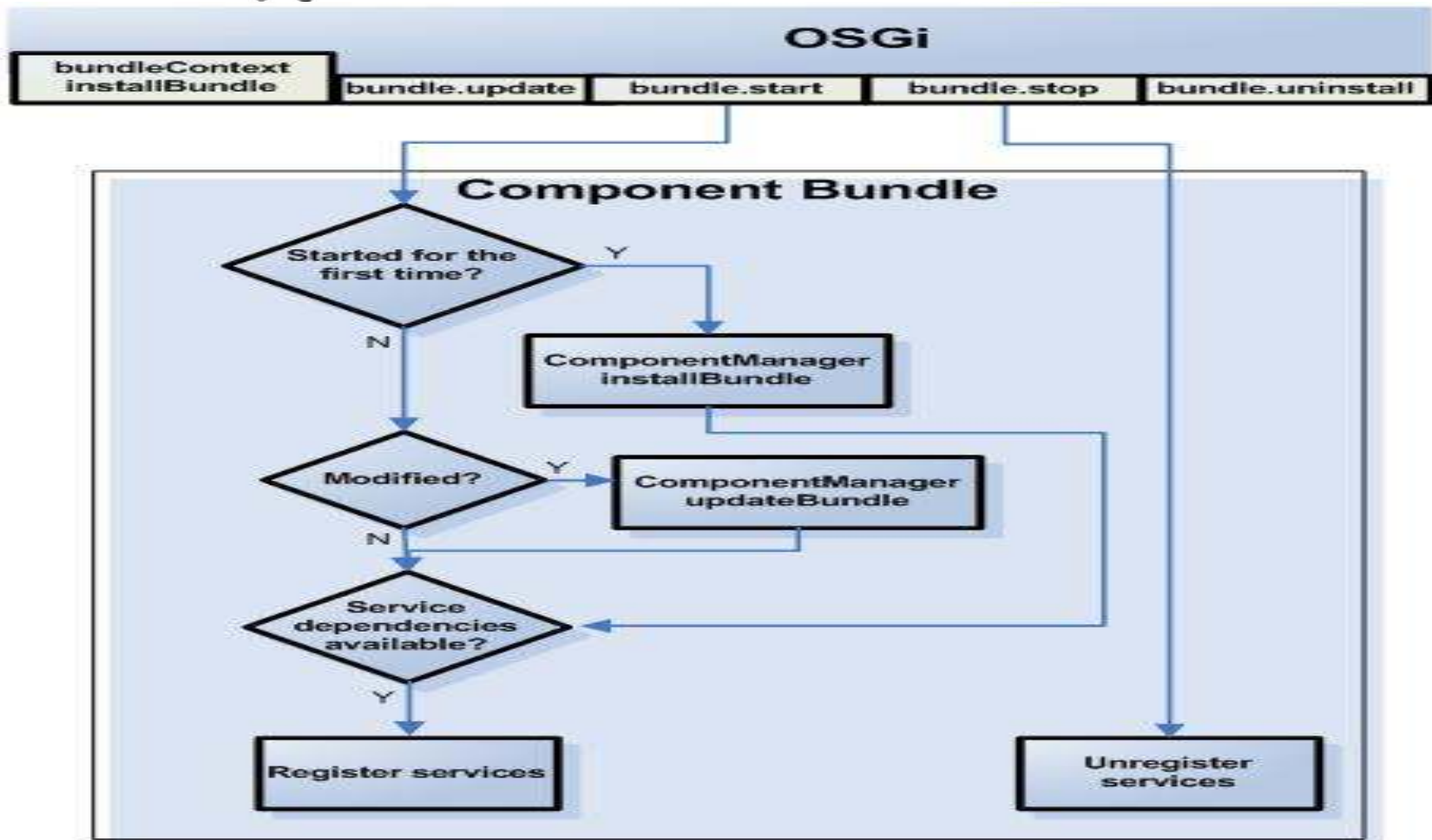
# OSGi in GX WM



WCB Lifecycle



# OSGi in GX WM





## OSGi in GX WM

- Lets take a look at some code snippets.



## OSGi in GX WM

### The Bundle Activator

- Bundle-Activator manifest entry specifies the activator.
- An OSGi bundle may specify 0 or 1 activator class. Library bundles have no need for it
- The Activator class must implement `org.osgi.framework.BundleActivator`
- The Framework instantiates and invokes the start method when the bundle enters the ACTIVE state.
- The Framework invokes the stop method when the bundle leaves the ACTIVE state.
- Receives a unique `BundleContext` that provides access to the Framework.

```
public class SimpleActivator implements BundleActivator {
    private BundleContext myBc;

    public void start(BundleContext bc) throws Exception {
        myBc = bc;
    }

    public void stop(BundleContext bc) throws Exception { }
}
```



## OSGi in GX WM

### Providing a service

- Services are registered with the Framework through the BundleContext.
- Services are registered by 1 or more interface FQNs and optional registration properties.
- The Framework returns a ServiceRegistration object for future reference.
- Registration and unregistration may be done at any point in time.

```
public void registerLongService() {
    Long i = new Long(20);
    Dictionary props = new Hashtable();
    props.put("myvalue", "20");
    mySr = bc.registerService(Long.class.getName(), i, props);
}

public void unRegisterLongService() {
    mySr.unregister();
}
```



## OSGi in GX WM

### Consuming a service

- Services are retrieved from the Framework through the BundleContext.
- The Framework returns a ServiceReference object that can be used to get the service instance and may be kept for future reference.
- Consumers must unget the ServiceReference when done to let the Framework know the service is released.
- Consumers must guard against keeping stale references to service instances.

```
public void consumerLongService() {
    ServiceReference sr =
myBc.getServiceReference(Long.class.getName());
    if (sr != null) {
        Long si = (Long) myBc.getService(sr);
        if (si != null) {
            // ...
            myBc.ungetService(sr);
        }
    }
}
```



## OSGi in GX WM

### Using ServiceListener

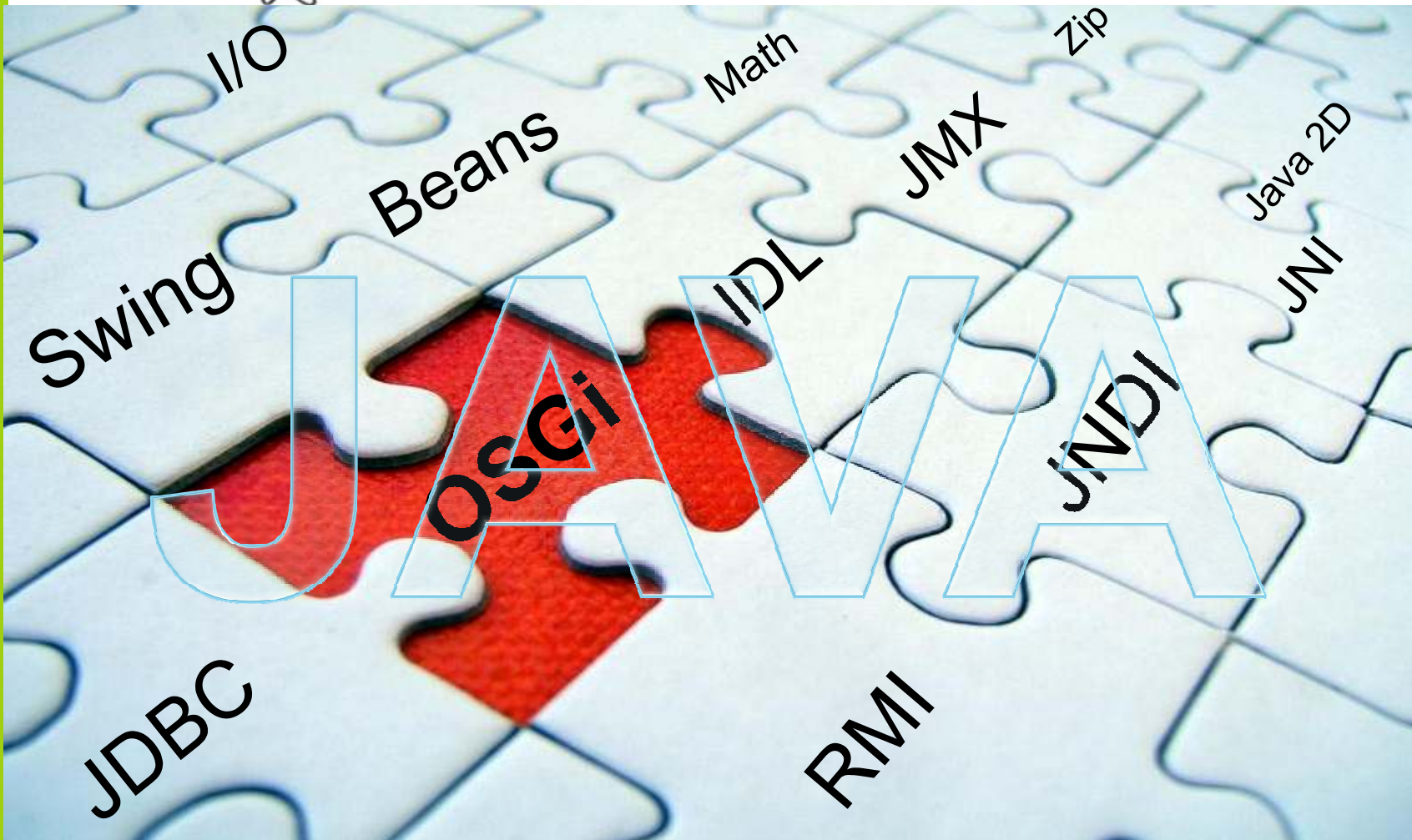
- Service listeners can be (un)registered at the Framework through the BundleContext.
- A filter can be specified to filter out irrelevant events.

```
public void registerLongServiceListener() {
    myBl = new ServiceListener() {
        ServiceReference sr = se.getServiceReference();
        switch (se.getType()) {
            case ServiceEvent.REGISTERED:
                // ...
                break;
            default: break;
        }
    };
    try {
        myBc.addServiceListener(sl, null);
    } catch (InvalidSyntaxException e) {}
}

public void unregisterLongServiceListener() {
    myBc.removeServiceListener(myBl);
}
```



## Conclusion





# GXDeveloperWeb.com

Welcome GX WebManager Developer Community! - GX DeveloperWeb - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.gxdeveloperweb.com/web/show

gxdeveloperweb.com wcmexchange.com gxwebmanager.com

<GX> GX DeveloperWeb

Software Sources Wiki Forums Blogs Search Contact

Home

### Sign in

E-mail \*, (xx@domain.com)

Password \*

**Sign in**

[Register](#) | [Forgot password](#)

### Recent blogs

April 14, 2008  
[Lacking chemistry](#)

April 14, 2008  
[Maven secrets: assembling a WCB zip](#)

April 11, 2008  
[ApacheCon EU 2008 impressions](#)

April 7, 2008  
[Baking apple pie at 347](#)

March 29, 2008  
[Maven secrets: filtering sources](#)

[All Blogs...](#)

## Welcome GX WebManager Developer Community!

GX WebManager is an Enterprise Web Content Management system. Based on a J2EE architecture, using open technologies like Spring, JCR and OSGi. It's designed to respond faster to the ever changing requirements of business and IT. Deploy new web components runtime and exchange them in the community.

**Download**

*Download GX WebManager CE!*

Download the free GX WebManager Community Edition and build your first component today.

Please give us **your feedback on our documentation!**

### On GX DeveloperWeb you will find:

- > [Documentation](#)
- > [Wiki](#)
- > [Forums](#)
- > [Blogs](#)
- > [\(Re\)sources](#)
- > [Issue tracker](#)

### What's new?

**April 15, 2008**  
[wcmexchange.com new and improved](#)

**April 11, 2008**  
[Meet us at JavaOne 2008 San Francisco](#)

**April 8, 2008**  
[GX at J-Spring 2008](#)

[More What's new...](#)

### Register now!

By becoming a member, you will be able to participate actively in the community. So you can download software, contribute to the Wiki and get access to the Issue Tracker. Further more manage your own profile and receive newsletters.

**Register now! It's free, quick and easy**

© 2008 GX creative online development B.V.

[Disclaimer](#) | [Sitemap](#)

Done



## Demo

- Switching JDBC driver.
- Two versions of the same WCB.
- WCB management console.



Q&A