

Progress. | Fuse.
OSGi Overview

BUSINESS
MAKING
PROGRESS™

freeman.fang@gmail.com ffang@apache.org

Apache Servicemix Committer/PMC member

Apache Cxf Committer/PMC member

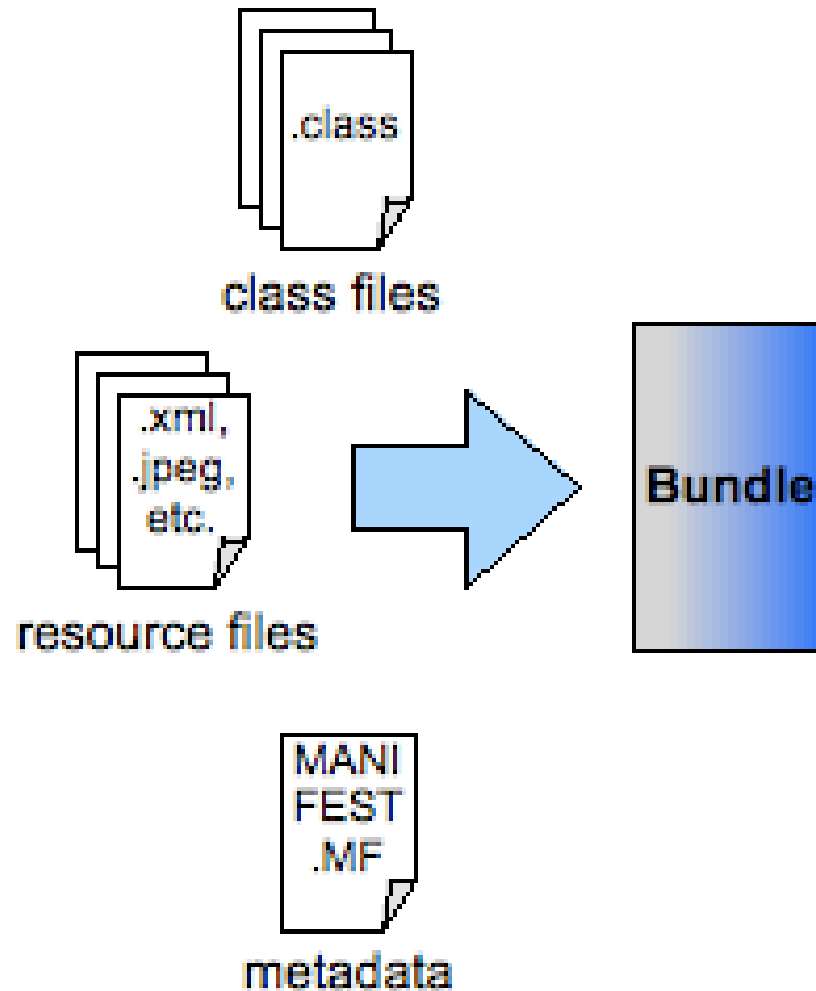
Apache Karaf Committer/PMC member

Apache Felix Committer



- Why OSGi
- What is OSGi
- How OSGi Works
- Apache projects related to OSGi

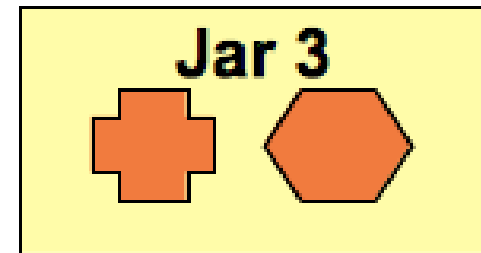
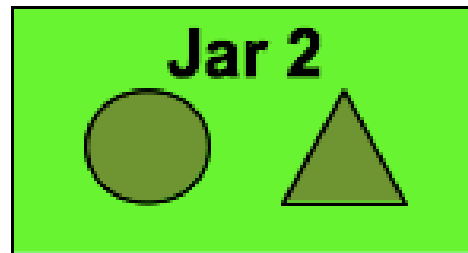
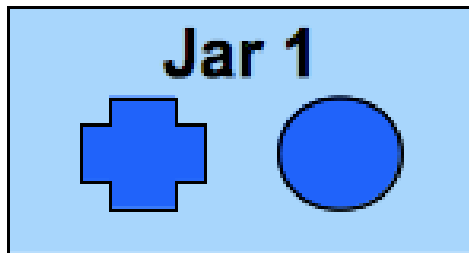
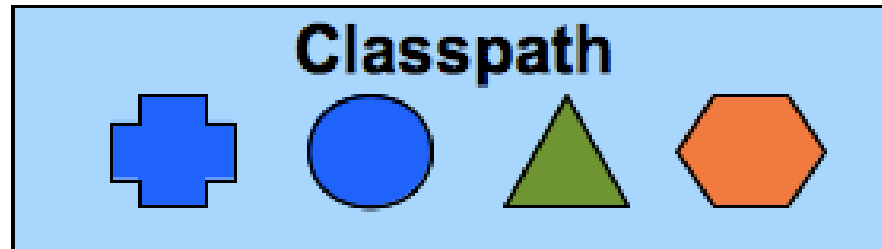
- modularity
- more strict access control
- classloader issues without OSGi
- dynamic



- poor code visibility control
- error-prone classpath concept
- rigid deployment and management support

```
package mypackage.greeter;  
public interface Greeter {  
    public void sayHi(String name);  
}
```

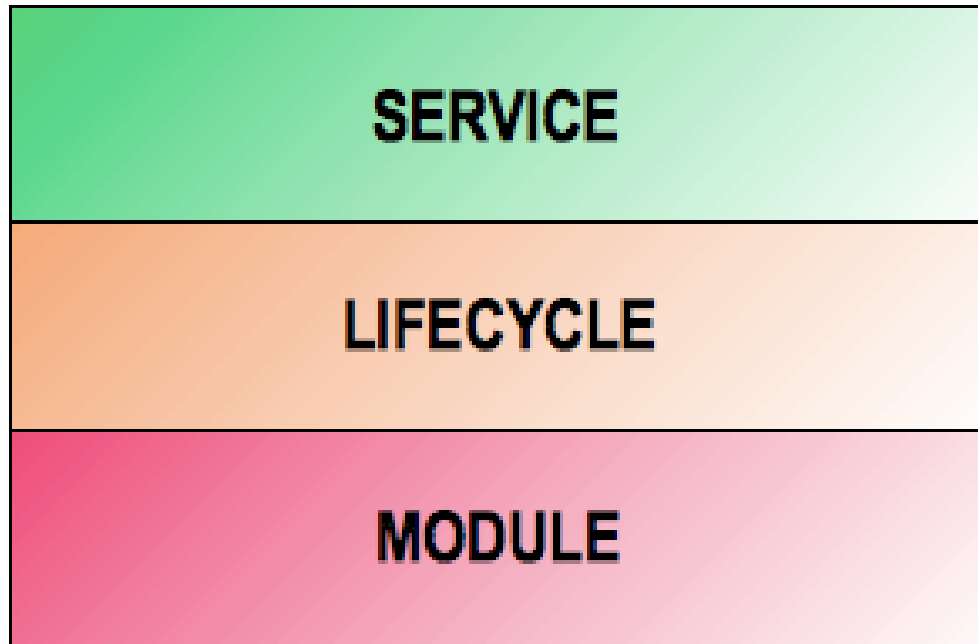
```
package mypackage.greeter.impl;  
public class GreeterImpl implements Greeter {  
    public void sayHi(String name) {  
        System.out.println("Hello " + name);  
    }  
}
```



- can we hot replace a jar from classpath
- can we hot redeploy a component from container
- what's the underlying classloader issue

- early detection vs exception during runtime
- great plugin component support

- Open Services Gateway Initiative
- OSGi Alliance
- SOA in JVM
- Modularity layer for JAVA platform



- The term bundle is what OSGi uses to refer its specific realization of the module concept
- What's the definition of bundle? A physical unit of modularity in the form of a JAR file containing code, resources, and metadata, where the boundary of the JAR file also servers as the encapsulation boundary for logically modularity at execution time.
- What's the differeneec between bundle jar and normal jar?
- How to create a bundle jar?
- How to OSGi-fied a normal jar?

- Human-readable information
- Bundle identification
- Code visibility

Bundle-Name: ServiceMix :: CXF Binding Component

Bundle-Description: A CXF Binding Component

Bundle-DocURL: <http://www.apache.org/>

Bundle-Category: JBI Component, Binding Component

Bundle-Vendor: The Apache Software Foundation

...

Bundle-SymbolicName: servicemix-cxf-bc

Bundle-Version: 2010.02.0.SNAPSHOT

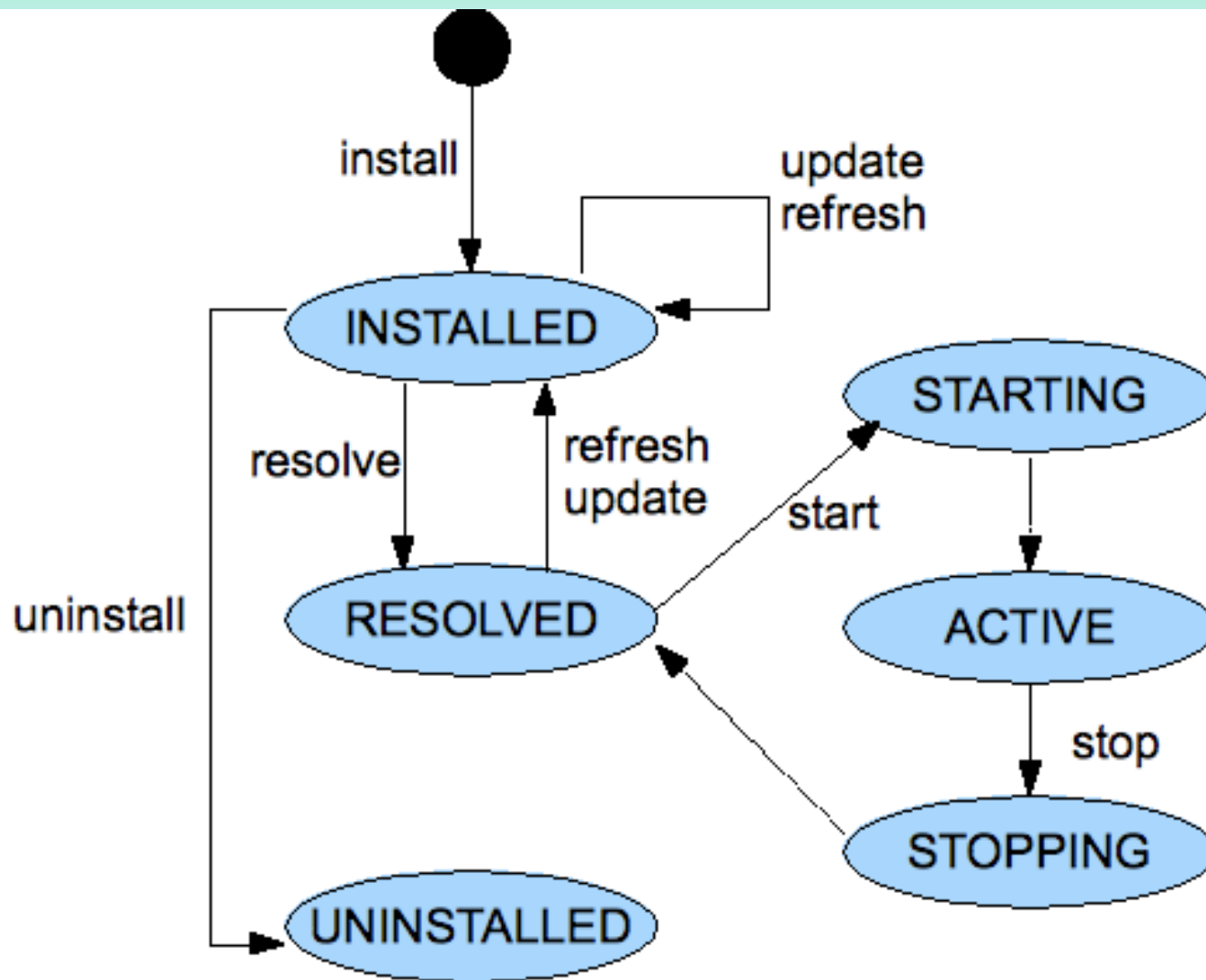
Bundle-ManifestVersion: 2

Bundle-Classpath: ./images/,embeded.jar

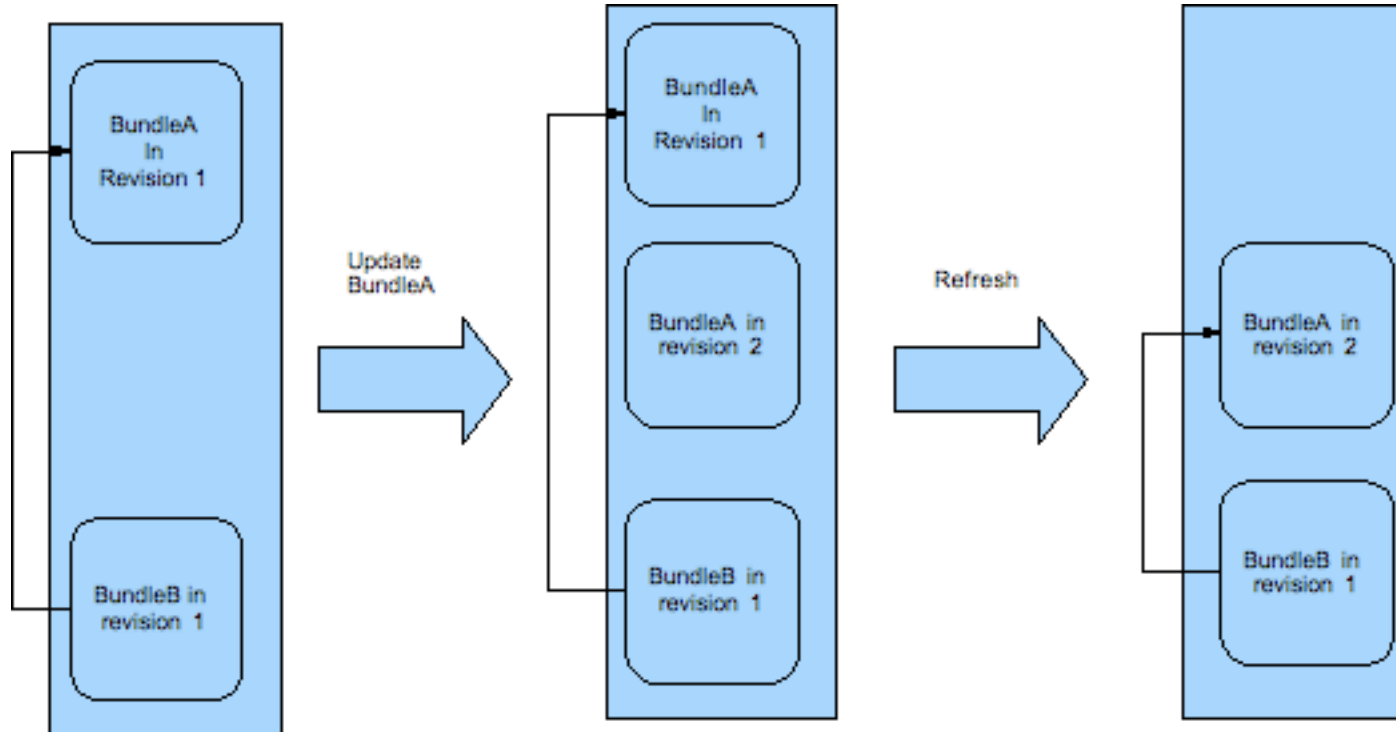
Export-Package: ...

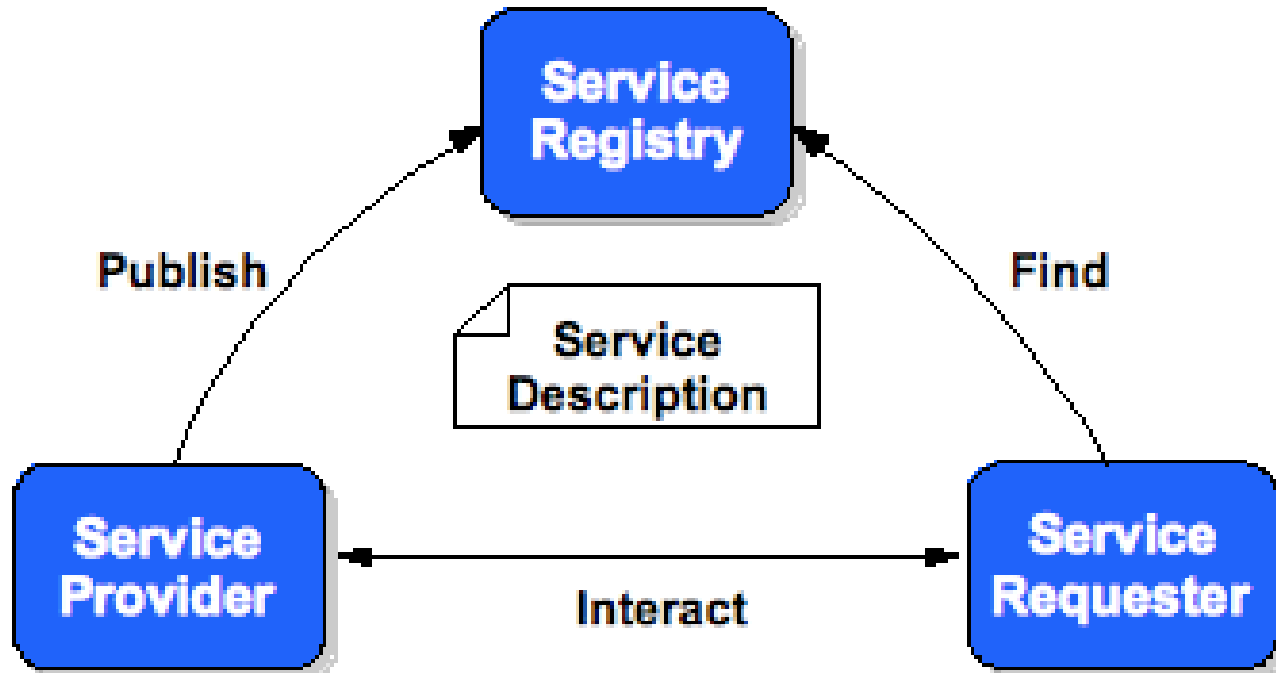
Import-Package: ...

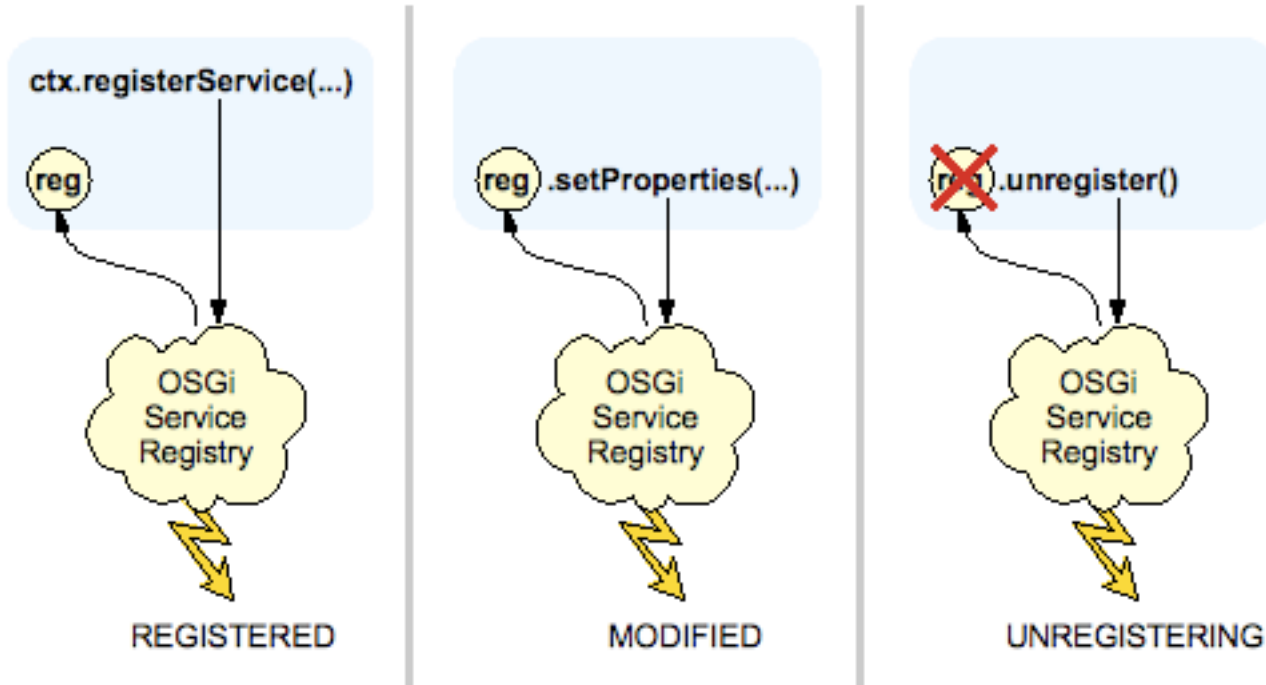
- If the class is from a package starting with “java.”
- If the class is from a package imported by the bundle.
- The bundle class path is searched for the class.



- self-contained bundle
- two steps operation







- dynamic way to use a service
- multiple implementation services provider come/go on the fly
- find some service which already be there before your bundle start

- directly from `bundleContext`
- reactively with service listener
- indirectly using a `ServiceTracker`

- core services
 - package admin service
 - start level service
 - url handlers service
 - permission admin service

- compendium services
 - http service
 - event admin service
 - configuration admin service
 - user admin service

- should my application depend on certain bundle start sequence
- should I heavily use bundle-classpath in manifest
- should I put multiple class copys in different bundles(even use private package)

- SPI discovery in JVM
- Why it doesn't work in OSGi Container
- How to resolve this problem

- Service Provider Interface
- Implementation provider jar on classpath
- FactoryProvider From API jar
- Provider-Configuration File in implementation provider jar (META-INF/services/javax.xml.soap.SOAPFactory)

- Dynamic for multiple provider
- API bundle shouldn't bind to a specific impl
- Doesn't make sense to import impl package for API bundle
- META-INF/services isn't visible for API bundle

- OSGILocator
- BundleActivator

- Apache Felix
- Apache Karaf
- Apache Servicemix
- Apache aries
- ...

- Felix is the OSGi core runtime
- Aries is OSGi 4.2 EE implementation
- Karaf provides a distribution based on Felix with several key features out-of-box
- Karaf start from Apache ServiceMix Kernel
- ServiceMix4 is a set of sub-projects such as components/maven-plugins/specs/bundles/nmr/features, but all based on Karaf

- Thanks