

OSGi Training

Christian Schneider
Open Source Architect
cschneider@talend.com

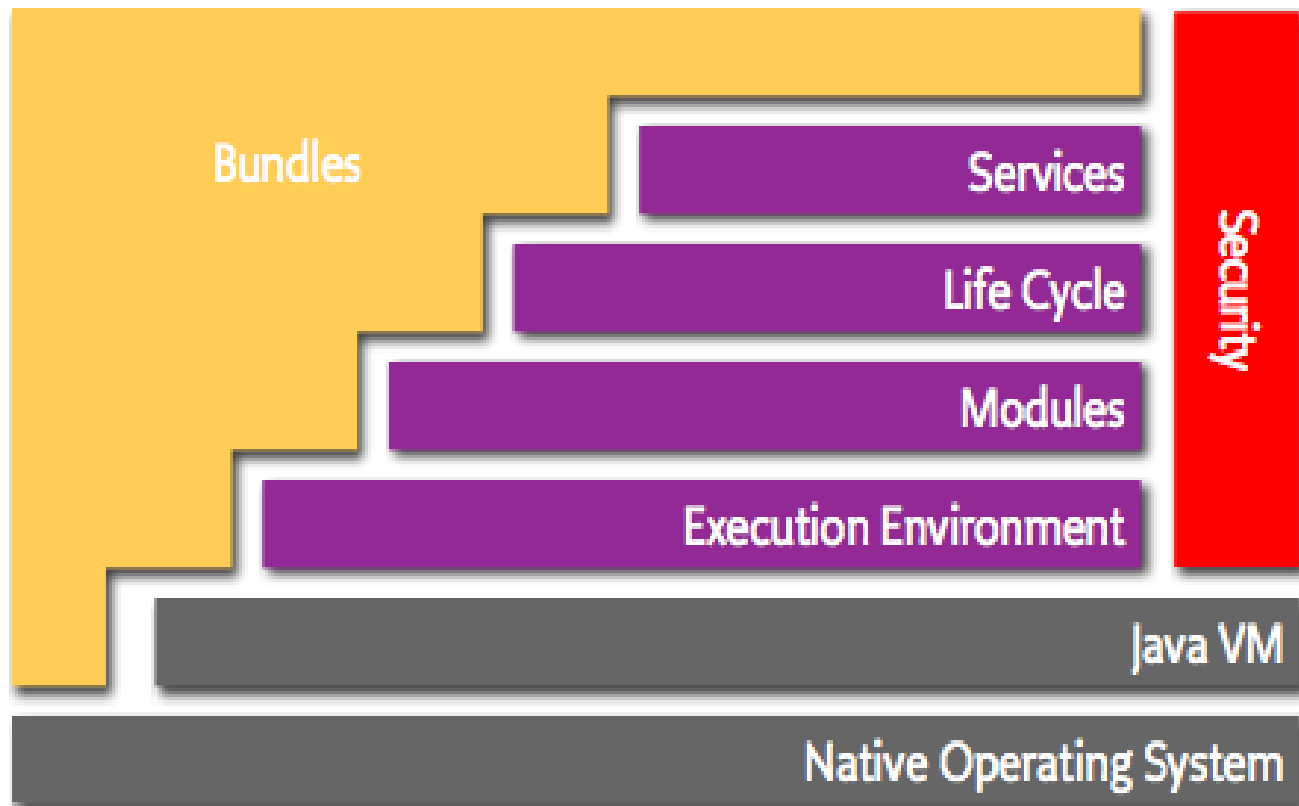


What is OSGi ?

OSGi : OpenSoftware Gateway initiative

OSGi is a dynamic and modular platform for Java, created in 1999 by IBM, Oracle, Sun, Ericson, etc. It's the OSGi Alliance and OSGi Expert Group.

OSGi Architecture



Execution Environment (Framework)

→ OSGi framework

- Runtime kernel that allows to run bundles
- Bundles can be managed at runtime

Modules (Bundle)

→ Bundle

- Jar with additional meta data

Bundle-Version: 1.1.0

Bundle-SymbolicName: org.myorg.mybundle



Version and Name are unique
in the framework

Export-Package: org.myorg.mypackage;version=1.0

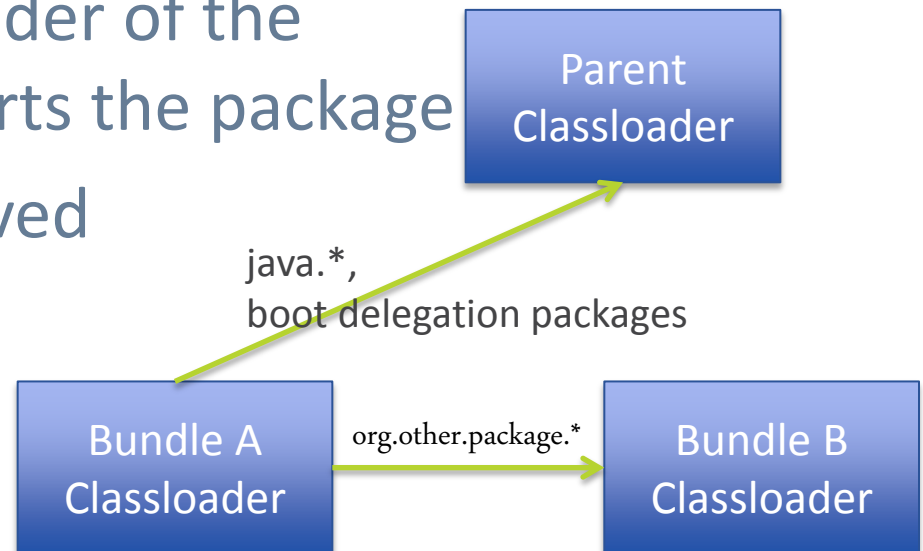
Import-Package: org.other.package;version=[1.0;1.5)

Bundle-Activator: org.myorg.Activator

- Classes are resolved on package level not on bundle level
- Bundles have to explicitly specify the packages they need and offer. Everything else is private

Modules (Class loading)

- Each bundle has its own class loader
- `java.*` and packages specified in boot delegation resolved from Parent classloader (Framework classloader)
- For each Import-Package the bundle classloader delegates to the classloader of the bound bundle that exports the package
- All other packages resolved internally from bundle



Modules (Classloading 2)

→ The classloader of a class is the (delegated) classloader that ultimately loaded the class

Example:

Class A from Bundle BundleA

Class B from Bundle BundleB

Code inside class A:

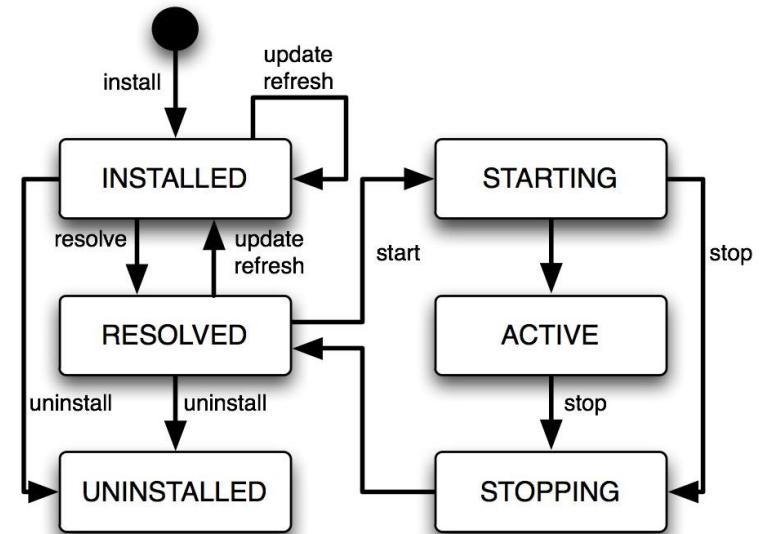
```
this.getClass().getClassloader() -> Classloader of Bundle A
```

```
B b = new B();
```

```
b.getClass().getClassloader() -> Classloader of Bundle B
```

Lifecycle

→ Reflects dynamic nature of OSGi



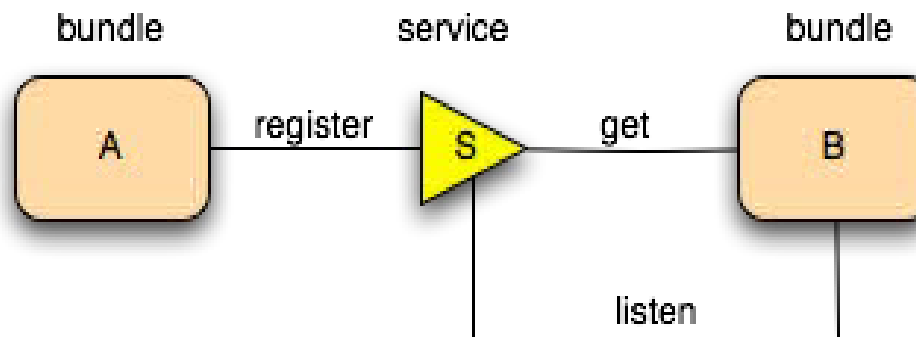
→ Resolve: Imported packages are bound to bundles that export them

→ Start: Activator start method is called

→ Stop: Activator stop method is called

Services

- OSGi Service: Java object instance + Properties
- Bundles can register / unregister services at runtime
- Bundles can listen to service lifecycle -> ServiceReference
- Service object retrieved using `context.getService(reference)`
- Calls on service objects are pure java method calls. No magic
- Never hold service objects over longer periods. Unget after use



Apache Karaf, Blueprint and OSGi services

→ See apachecon talk

https://docs.google.com/presentation/d/1xa5XWWJ40X_WYb8MiDbRGtHx20f4CfXHfi7cg4RHahk/edit