

# THE OSGi MENACE

---

Raimundo@BJUG

# What is OSGi

---

- ❑ Open Service Gateway initiative
  - ❑ The OSGi™ specifications define a *standardized, component oriented, computing environment for networked services*.
  - ❑ Adding an OSGi Service Platform to a networked device (embedded as well as servers), adds the capability to manage the life cycle of the software components in the device from anywhere in the network.
-

# Scope of OSGi

---

## □ Service Platform

- A standard, *non-proprietary*, software component framework for manufacturers, service providers, and developers.
- A powerful model for co-existence of different components/applications in a single JVM.
- A secure environment that executes applications in a sandbox.
- A cooperative model where applications can dynamically discover and use services provided by other applications running inside the same OSGi Service Platform.
- A flexible remote management architecture.
- A number of standardized, optional services.

## □ Programming Model

- A flexible deployment Application Programming Interface (API) that controls the life cycle of applications.
-

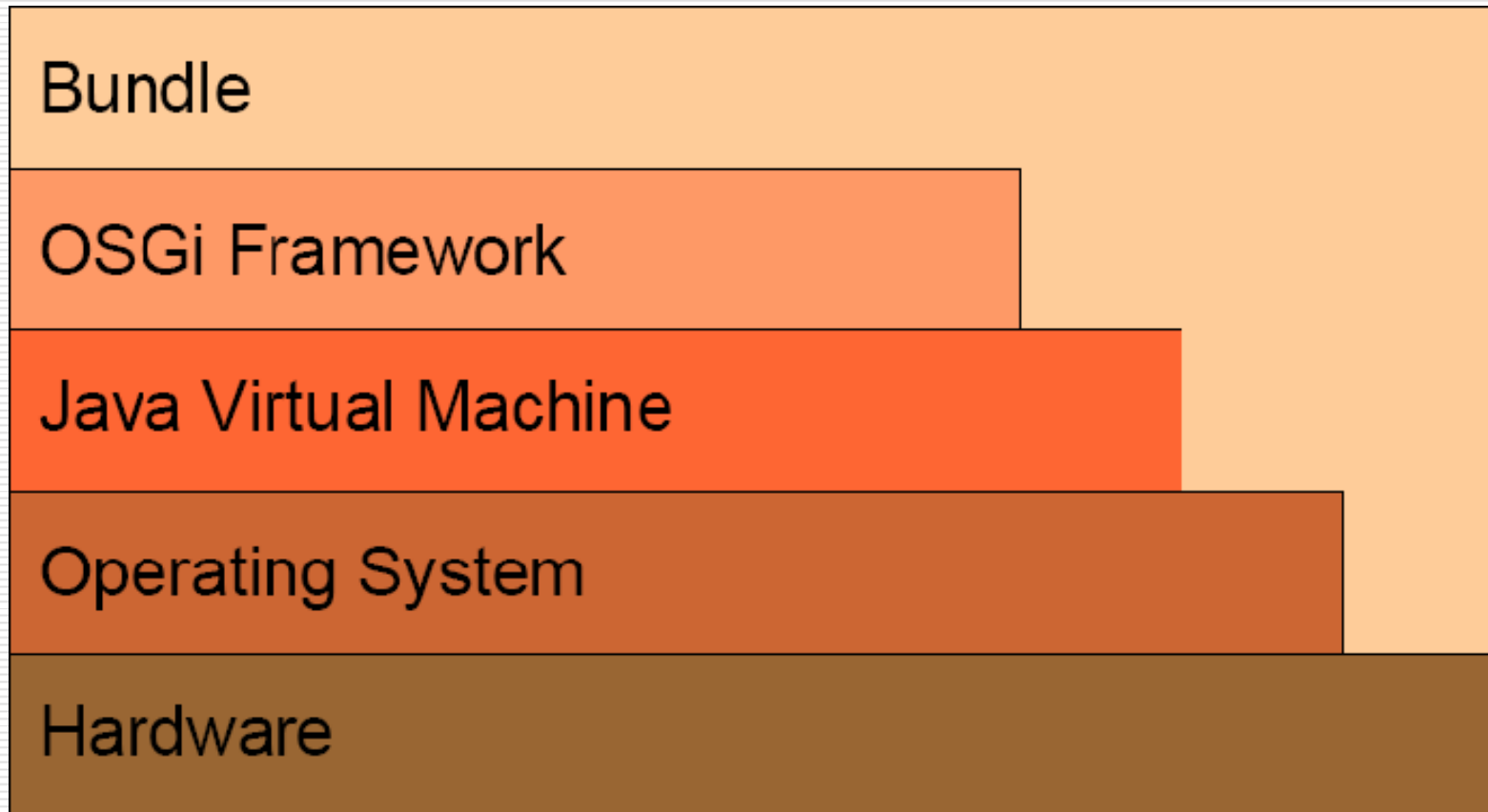
# Framework · Architecture · Platform

---

- What is a "Framework"?
    - Domain-Nature
    - Semi-Manufactured
  - What is a "Architecture"?
    - Arrangement of components
  - What is a "Platform"?
    - Solutions
    - Operations
    - Markets
-

# OSGi Service Platform

---



# OSGi Terminology

---

## □ Framework

- Container that provides deployed services their environment and manages them

## □ Service

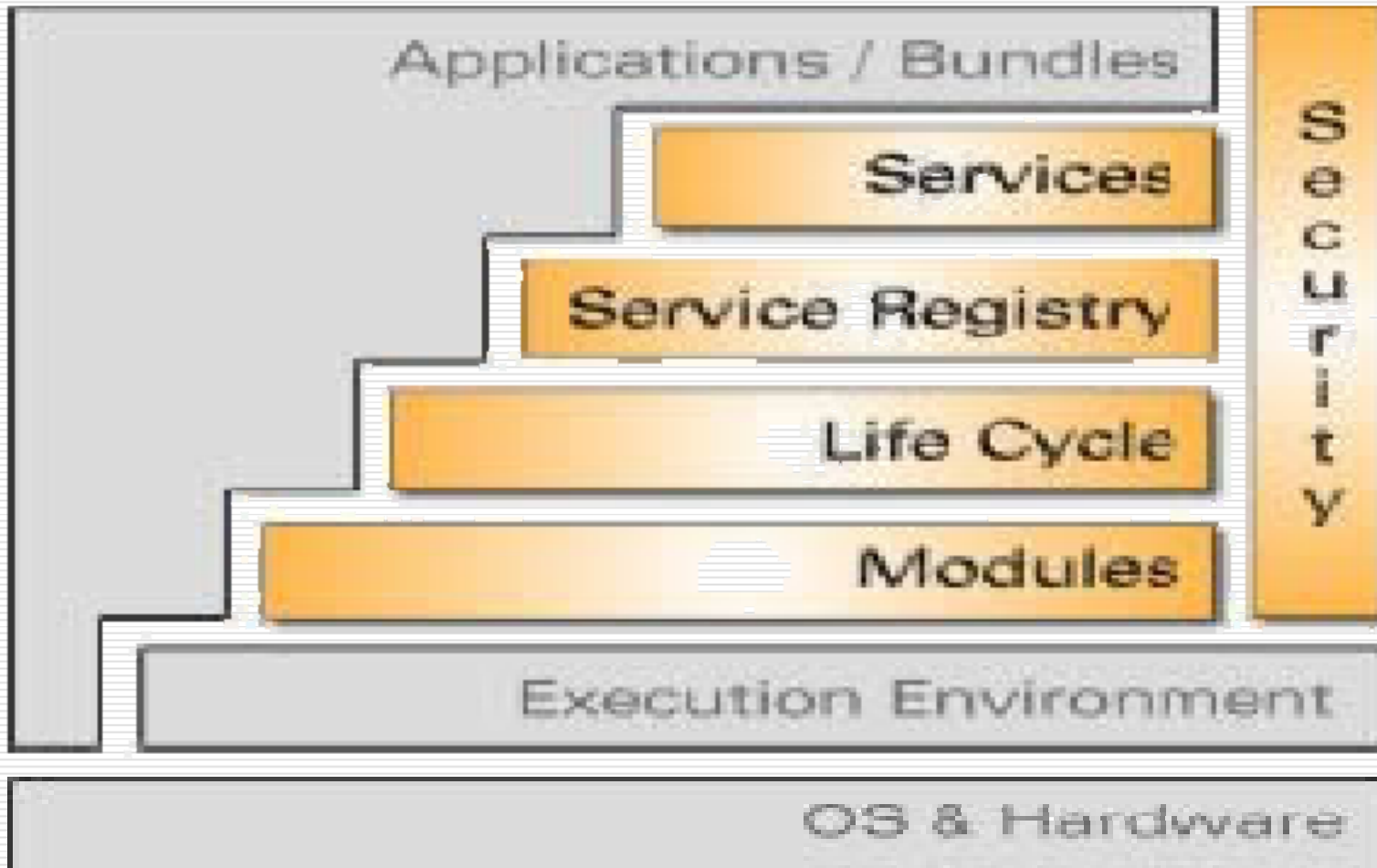
- Java interface with defined semantics

## □ Bundle

- Component, service implementation and deployment unit
  - Bundles register services with the framework
  - Bundles interact only via services
-

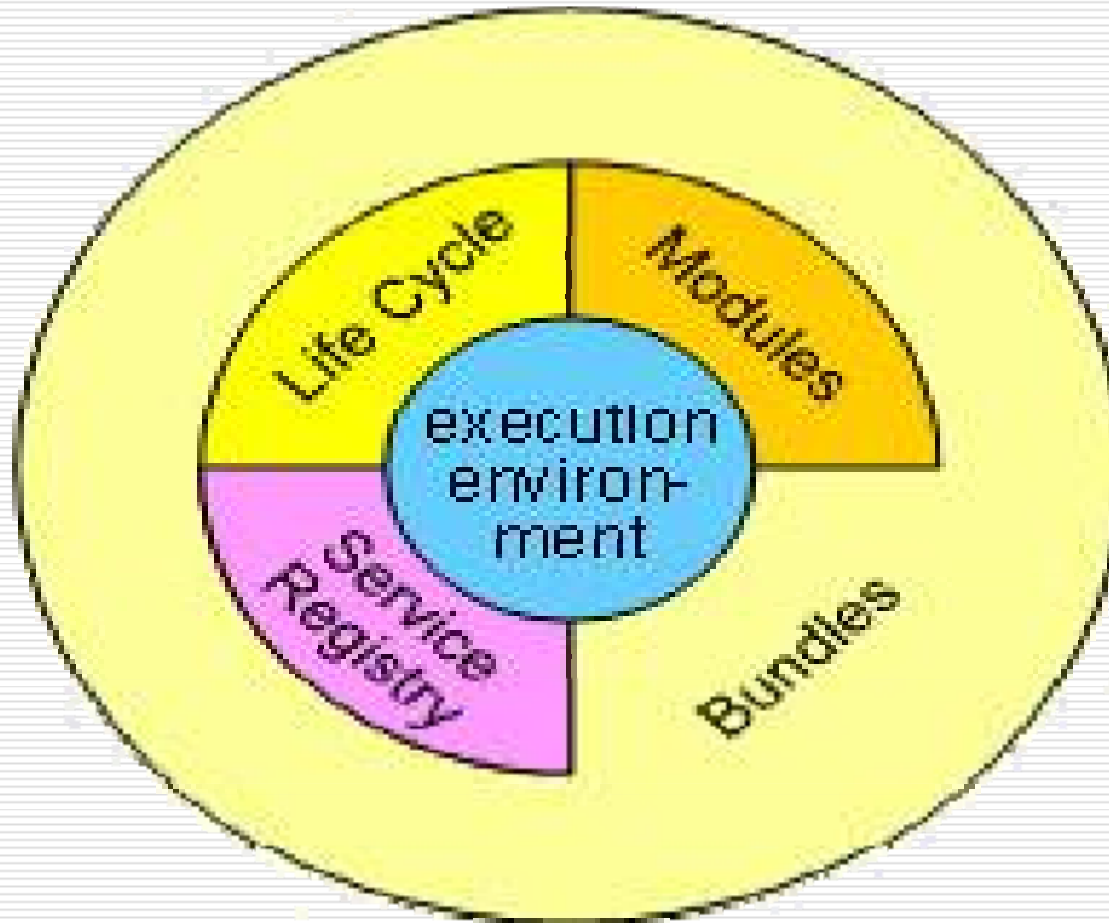
# OSGi Framework

---



# OSGi Framework

---



# Execution Environment

---

- The specification of the Java environment.
  - Configurations and Profiles
    - CDC-1.0/Foundation-1.0
    - OSGi/Minimum-1.0
    - JavaEmbedded-1.2
    - JavaCard
    - CLDC-1.0/MIDP-1.0
    - PersonalJava-1.2
    - J2EE-1.2
    - J2SE-1.3
-

# Modules

---

- Modules layer defines the class loading policies.
    - Sharing Package
    - Not only Classpath but also Modularization
-

# Life Cycle

---

- Life Cycle layer adds bundles that can be installed, started, stopped, updated and uninstalled.
    - Dynamically
    - Notification
-

# Service Registry

---

- The service registry provides a cooperation model for bundles that takes the dynamics into account.
    - interact only via services
    - Deployment API
-

# Standard Services

---

- Framework Services
    - Permission Admin
    - Package Admin
    - Start Level Admin
  - System Services
    - Log
    - Configuration Admin
    - Device Access
    - User Admin
    - IO Connector
    - Preferences
  - Protocol Services
    - Http
    - UPnP
    - Jini
  - Miscellaneous Services
    - Wire Admin
    - XML Parser
-

# Programming Model

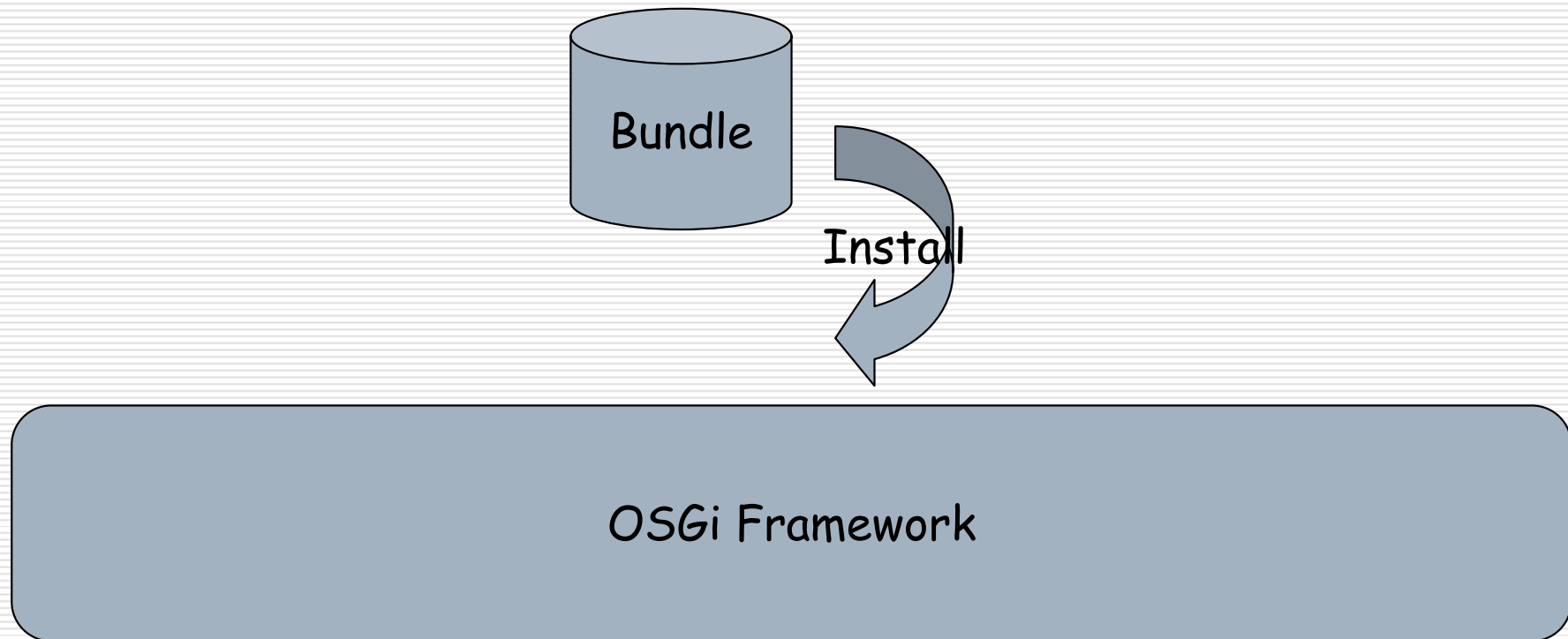
---

- Microkernel architecture
  - Deployment API
    - Glue Code
    - Integration
    - Lifecycle
-

# Component Model

---

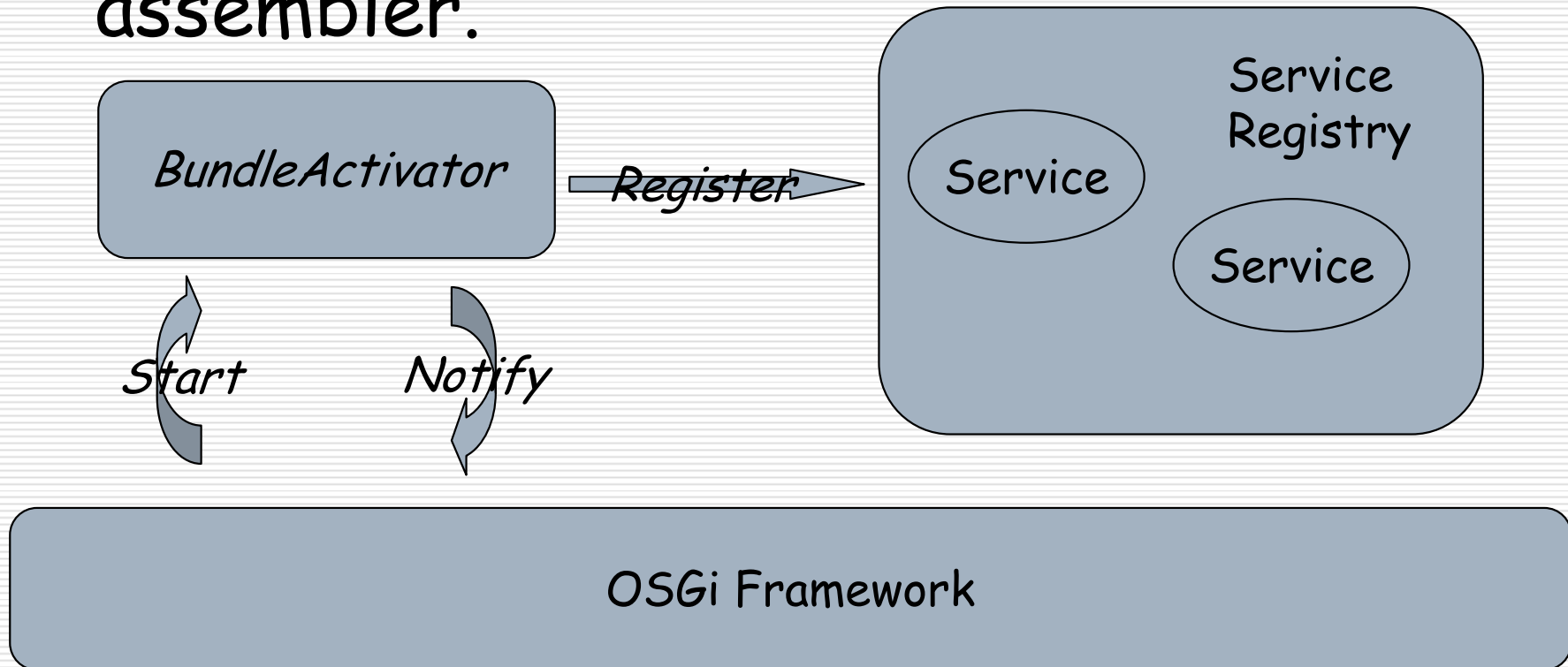
- the *BundleActivator* is a service assembler.



# Component Model

---

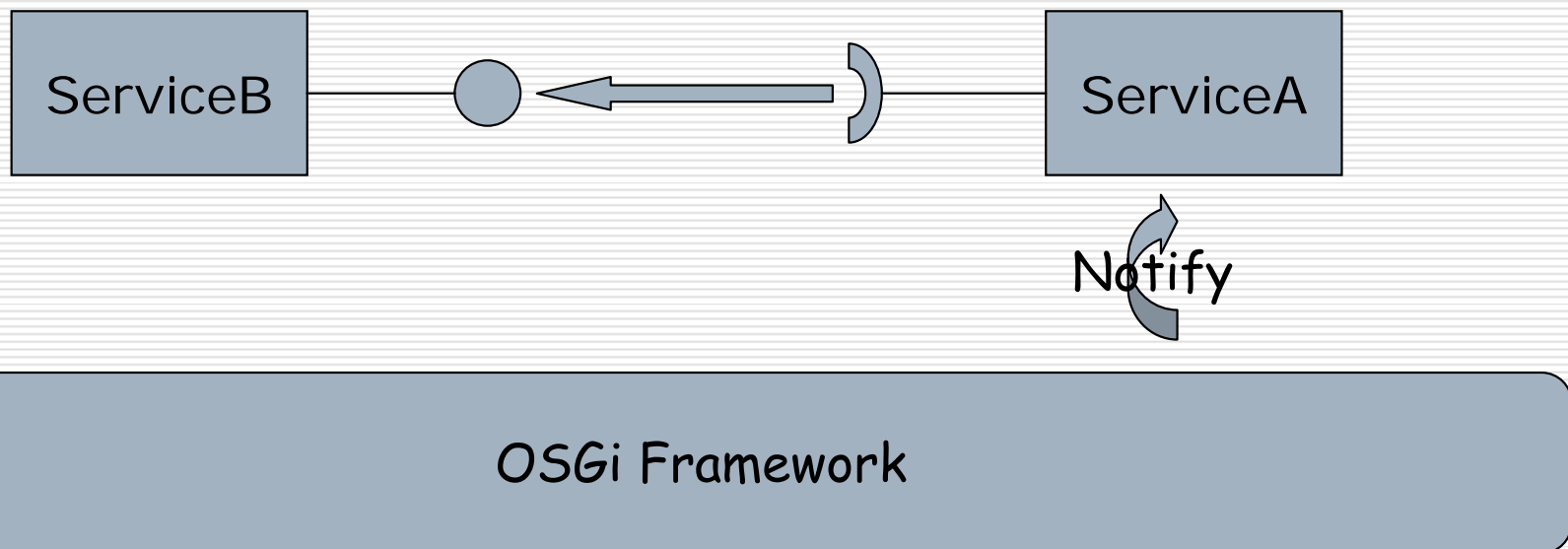
- the *BundleActivator* is a service assembler.



# Component Model

---

- the *BundleActivator* is a service assembler.



# Bundle Activator Interface

---

```
public abstract interface BundleActivator{  
  
    public abstract void start(BundleContext  
        context) throws Exception;  
  
    public abstract void stop(BundleContext  
        context) throws Exception;  
}
```

---

# Random Thought

---

- Enterprise Service Bus
    - Service-Oriented Architecture
    - Message-Oriented Architecture
    - Event-Oriented Architecture
  - Service Operation System
    - Thread vs Service
-

# OSGi Use Cases

---

- ❑ Next Generation Smart Phones
  - ❑ Shell HomeGenie
  - ❑ Eclipse
  - ❑ IBM Service Management Framework
  - ❑ High-end infotainment platform in BMW 5 Series
-

# OSGi Implementation

---

- Eclipse Equinox
  - Oscar
  - Knopflerfish Osgi
  - Xcesium Avina (early developing)
-

# Thanks

---

## □ Reference

- OSGi Specification R3
  - OSGi Whitepaper
  - The OSGi Alliance Website([www.osgi.org](http://www.osgi.org))
  - Component Deployment on OSGi Richard S.Hall, Humberto Cervantes
-