OSGi, an introduction

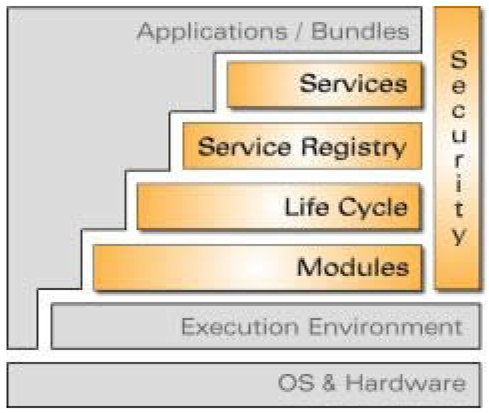
Jfokus 2007, R. Varttinen

OSGi an introduction

- Agenda
  - OSGi, the Open Services Gateway Initiative
    - Entities
    - Layers
    - Why Java?
  - Server Side and other …
    - Newton
    - Spring
    - Eclipse/Equinox
    - Knopflerfish
    - Apache Felix
  - Q&A

Intro, OSGi entities

- OSGi environment
- Framework
- Bundles
- Services
- Filters
- Events
- Security

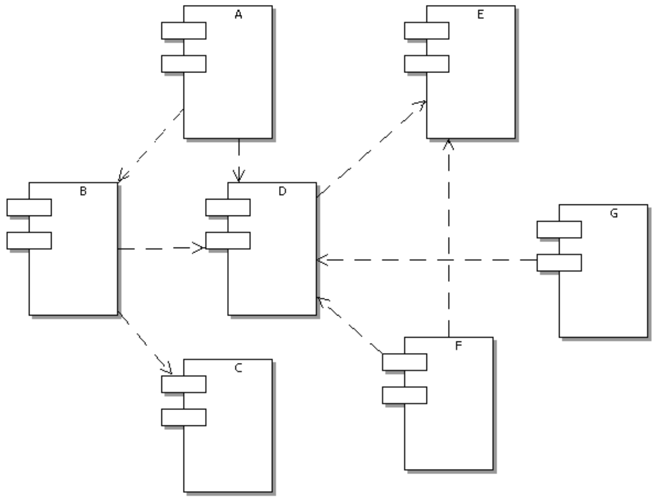


OSGI, Framework

- At the core of OSGi is the Framework
- The framework provides a standardized environment for applications (called *bundles*)
- The Framework is divided into a number of layers;
  - Module – classloading modularization
  - Life Cycle – API for life cycle management
  - Service Registry – way of communication between bundles
  - Security layer - spans the entire suite of the other layers

OSGi, Bundles

- Bundle is the deliverable
- Registers none, one or several services
- Find services registered by other bundles
- Framework itself as a system bundle



OSGi, Services

- An object registered with the Framework by a bundle to be used by other bundles
- A service is defined in a Java interface
- Different bundles, from different vendors, can implement the same service
  - Filtering
- OSGi defines a standard set of services
  - Logging, HTTP, etc.
- Notifications for life cycles of services
  - Events

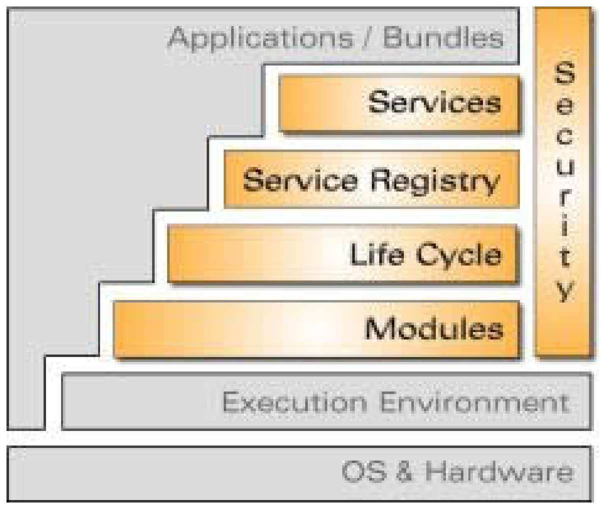
OSGi, Security

- Uses Java security based on permission classes
- A permission class defines what can be done, what it means
- JAR signing
- Optional

OSGi, Layers …

- OSGi, layers
  - Security
  - Module
  - Life cycle
  - Service

- Framework

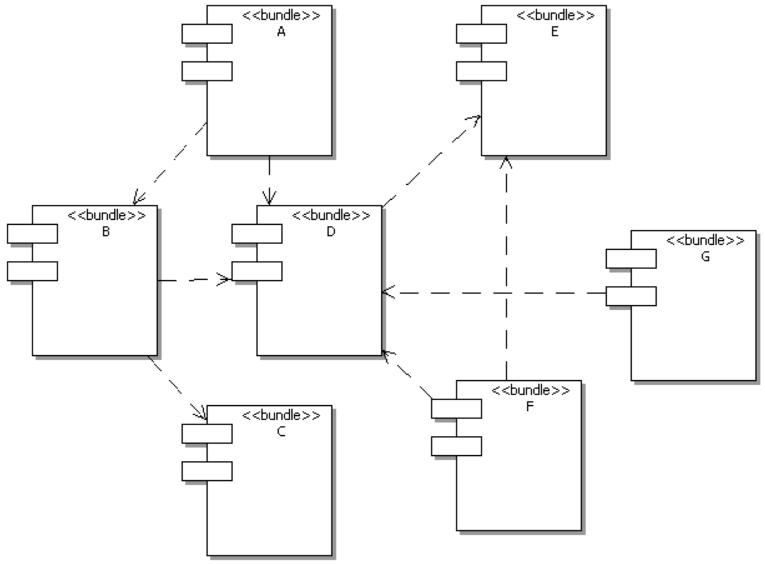


OSGi, Security layer

- Uses Java security based on Permission classes
- A Permission class defines what can be done
- Permissions administrated using the admin service

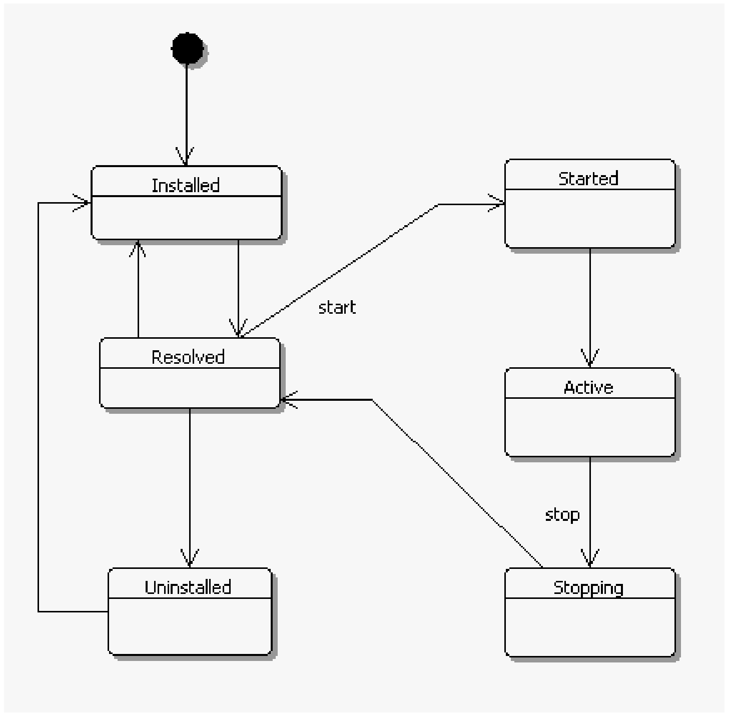
OSGi, Module Layer

- Handles bundles
- One class loader per bundle
- Loads native code
- Fragment bundles
- Extension bundles



OSGi, Life Cycle layer

- Provides API for managing bundles
- Based on the Module Layer



OSGi, Service Layer

- Provides a service model
  - Discover, and get notified about, services based on their interfaces or properties
  - Bind to one or more services by
    - program control
    - deployment configuration
- "SOA" within a VM
- The OSGi Alliance provides many standardized services
  - Logging, HTTP, Permission admin, etc.

OSGi, Why Java?

- Portable byte code
  - Independent on OS or CPU architecture
- Mature platform
- Active developer community and broad industry acceptance
  - Large code base
  - Large number of developers
- Security integrated in the language

OSGi, Server side and other

- Newton
  - http://newton.codecauldron.org/
- Spring and OSGi
  - http://www.springframework.org/osgi/specification
- Eclipse and Equinox
  - http://www.eclipse.org/equinox/
- Knopflerfish
  - http://www.knopflerfish.org/
- Apache Felix
  - http://cwiki.apache.org/FELIX/

Finally, some tips …

- Read the specification!
  - Easy and straightforward
- OSGi homepage: http://www.osgi.org
- Write your own bundles; http://www.eclipse.org

Q & A

- Any questions?

- robert.varttinen@ciber.com
- http://robertvarttinen.blogspot.com/