JBoss Microcontainer meets OSGi

Scott Stark & Ales Justin
Red Hat inc.

Agenda I.

- Microcontainer project introduction
  - JMX vs. MC
  - State machine and IoC
  - Legacy MBeans
  - AOP integration
  - Deployers
  - Classloader
  - Other extensions

Agenda II.

- What is OSGi
  - What problems does it solve
  - How does it work
- Microcontainer and OSGi
  - What are we going to support
- JBoss5 and OSGi
  - ProfileService and OBR
- Why not use existing OSGi impls
- New OSGi personality in MC
  - Transparent changes in the MC codebase
- Simple demo

JMX vs MC

- JMX Supports
  - Management interface
  - Invocation bus for attribute accessors, operations
  - Basic ClassLoader model
- Does not support
  - Deployment model
  - Dependenies/IoC
  - Extensible component model
  - AOP
- JMX Was Not Designed as a MicroContainer
  - JBoss simply leveraged JMX for that purpose

State Machine

- At its core, the MC is a state machine
- Dependency module
  - Controller
    - Dependency State Machine
    - ControllerStates – define your own states
  - ControllerContext
    - Represents a component
    - Has dependencies
    - Receives callbacks when dependencies are satisfied for a state transition
    - ControllerMode – automatic, manual, on demand
  - DependencyInfo and DependencyItem
    - Define your own dependencies
    - Write your own dependency item
      - Standard implementations available

Dependency Injection/IOC

- Depenendcy on other components
- Can be expressed via dependency on a named component
- Can be expressed via injection of:
  - component
  - component property
- Extensible dependency mechanism
  - The DependencyItem that performs the resolution logic is an extensible aspect of the dependency metadata

Legacy MBean Support

- The MC can be extended to support any component model
- The legacy mbean services used by earlier jboss releases are supported via custom deployers.
- SARDeployer – processes the *-service.xml into ServiceDeployment/ServiceMetaData
- ServiceDeploymentDeployer – creates component deployments from the ServiceMetaData
- ServiceDeployer – creates the mbean component from the serviceMetaData
  - registers the mbean with the jmx kernel
  - handles dependencies on MBean & MC components

AOP integration

- MC has first class support for AOP via its metadata
- annotation - represents a Java5 annotation on the particular join point
  - deployment, bean, constructor, lifecycle callback, install callback, value types
- Can mix aop.xml element using xmlns="urn:jboss:aop-beans:1.0"

AOP Example

```
<deployment xmlns="urn:jboss:bean-deployer:2.0">
   <bean name="AspectManager" class="org.jboss.aop.AspectManager">
      <constructor
         factoryClass="org.jboss.aop.AspectManager"
         factoryMethod="instance"/>
   </bean>

   <aop:lifecycle-configure xmlns:aop="urn:jboss:aop-beans:1.0"
               name="JMXAdvice"
               class="org.jboss.aop.microcontainer.aspects.jmx.JMXLifecycleCallback"
               classes="@org.jboss.aop.microcontainer.aspects.jmx.JMX">
      <property name="mbeanServer"><inject bean="MBeanServer"/></property>
   </aop:lifecycle-configure>

   <aop:aspect xmlns:aop="urn:jboss:aop-beans:1.0"
               name="TxAdvice"
               class="org.acme.aspects.TxAspect"
               pointcut="execution(* *->@org.acme.Tx(..))">
      <property name="txManager"><inject bean="TxManager"/></property>
   </aop:aspect>

   <bean name="BeanWithAspects"
      class="org.jboss.test.microcontainer.support.SimpleBeanImpl">
      <annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX</annotation>
      <annotation>@org.acme.Tx</annotation>
   </bean>

   <bean name="PlainBean"
      class="org.jboss.test.microcontainer.support.SimpleBeanImpl">
   </bean>

</deployment>
```

Deployers

- Structural Deployers
  - Specification defined – jar, war, ear
  - Overridable via jboss-structure.xml
- Aspectized Deployers
- Run in phases: NOT_INSTALLED, PARSE, DESCRIBE, CLASSLOADER, POST_CLASSLOADER, REAL, INSTALLED
  - Non linear – JCA → ServiceMetaData → ServiceDeployer
- Attachments
  - Metadata for deployment
  - Can define Input/Output order for deployers
- VDF
  - JBoss VFS usage
- ManagedDeployment/ManagedComponent support
  - Management interface for a deployment
- MainDeployer
  - Deployment process, completion check

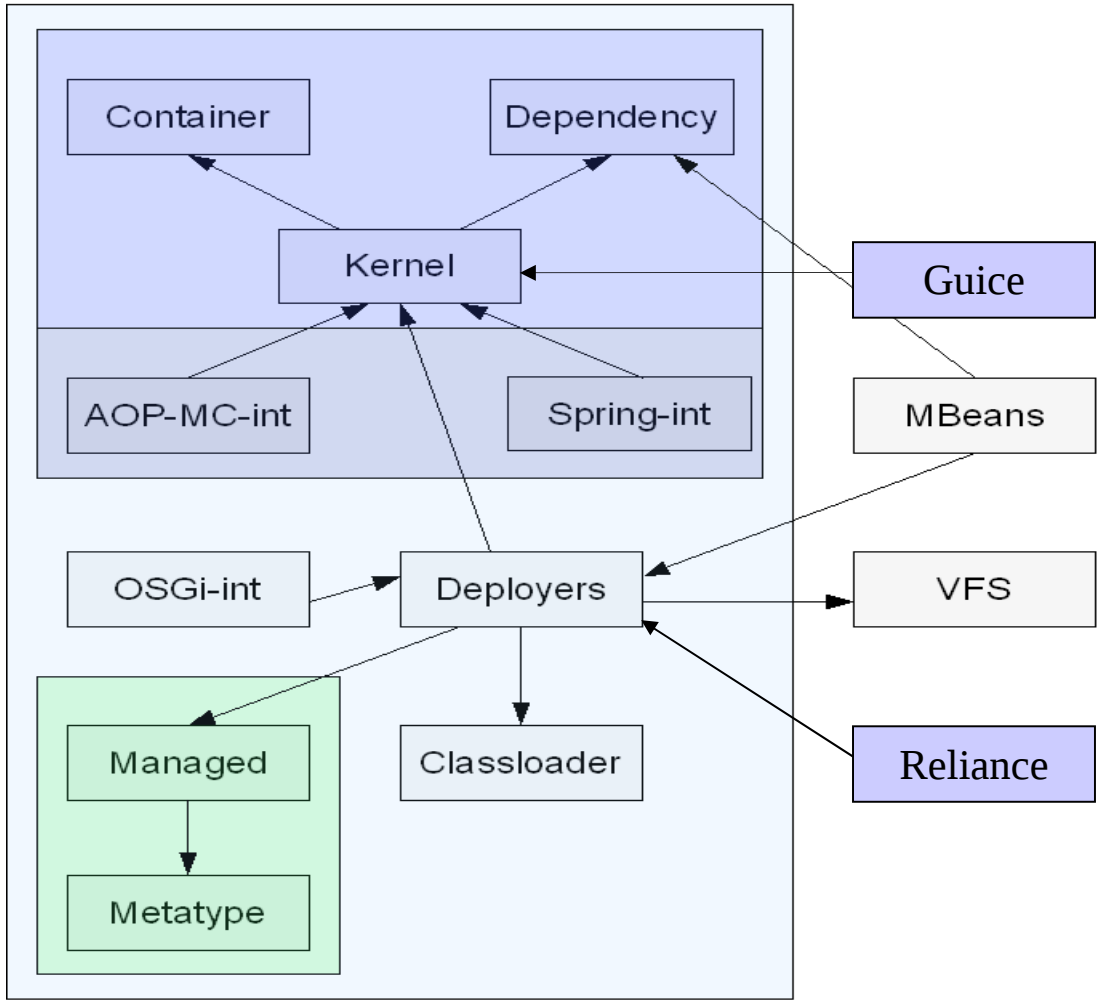
Classloader

- Update of the ULR peer class loading model to support OSGi bundle/Java modules
- Simplification / restriction
  - Limit error prone implementations
  - SPI → Base architecture
    - Protected / Package modifiers
- ClassLoaderPolicy
- ClassLoaderDomain
  - Hierarchy
  - Default CLDomain
- ClassLoaderSystem
  - Domain factory
- Loader
  - DelegateLoader
    - FilteredDelegateLoader
- ParentPolicy
  - ClassFilter

Other extensions

- Managed and Metatype
  - Open MBean style types used by ManagedDeployments
- Spring
  - Schema support
  - Lifecycle, dependency by MC
- Guice
  - Beans ‘exchange’
- Reliance
  - Drools integration
  - jBPM integration

Microcontainer overview



What is OSGi

- OSGi… ?
  - Open Service Gateway Initiative
  - Embedded world
  - OSGi alliance and new EEG

- ‘The Dynamic Module System for Java’
  - JSR 291

Service Oriented Dynamic Module System

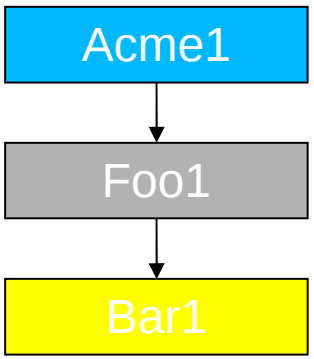
- Modules aka Bundles
- Strict visibility rules
- Dependency resolution process
- Versioning

- Runtime Module Install / Start / Stop / Uninstall

- Service Registry
- Runtime Changes Notifications

What does it solve

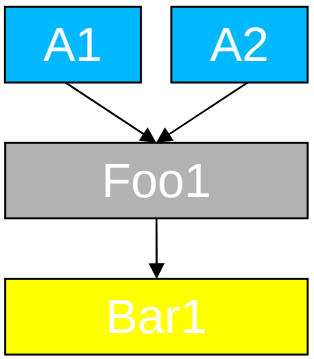
- Visibility
  - Blackbox approach
  - Explicit export
- Versioning



- Module lifecycle control

What does it solve

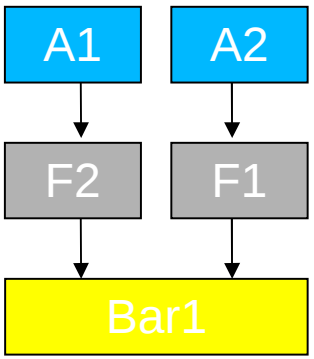
- Visibility
  - Blackbox approach
  - Explicit export
- Versioning



- Module lifecycle control

What does it solve

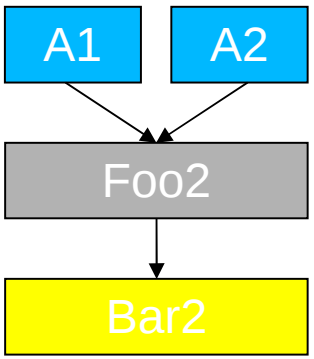
- Visibility
  - Blackbox approach
  - Explicit export
- Versioning



- Module lifecycle control

What does it solve

- Visibility
  - Blackbox approach
  - Explicit export
- Versioning



- Module lifecycle control

How does it work

- Plain jar file
  - MANIFEST.MF & BundleSymbolicName header
- Export package directive & version
  - Export package: org.jboss.vfs;version=2.0.0.GA
- Import package & version range
  - Import package: org.jboss.kernel;version="[2.0,3.0)"
  - Optional
  - Dynamic
- Other constraints
  - e.g. specific bundle dependency

Service oriented

- Service Registry
  - Exporting service to registry
    - Interface name
    - Optional properties
  - Services lookup
    - Interface name
    - LDAP filter
  - Declarative Services Support
  - Service Tracker

- Only see compatible services

Microcontainer and OSGi

- Classloading
  - New Classloader module
- Core OSGi Framework API
  - Façade on top of existing MC API
    - BundleContext
    - ServiceRegistry
    - Notifications
- Declarative Services Support
  - New EEG Component model
- Dynamicity

JBoss5 and OSGi

- ProfileService
  - Defines the bootstrap, deployers, and application contained in a named profile
  - Explicit service for the legacy implicit server/<name> contents
- OSGi Bundle Repository (OBR)
  - A Bundle Repository access api
    - Resolution based on capabilities, requirements
- Integration part
  - Profile service defines an OBR compatible repository api
  - OBR implementation can be plugged in

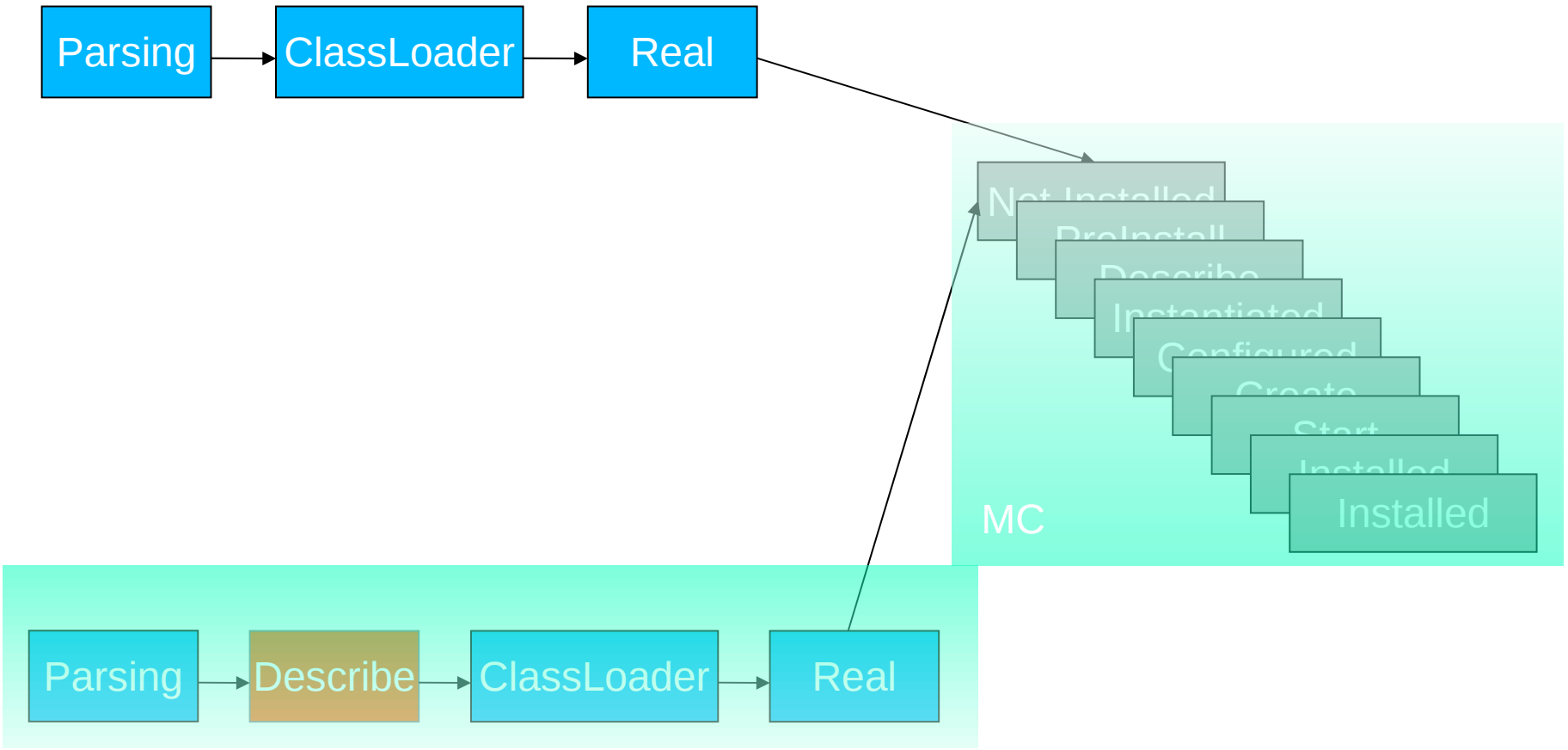
Why not use existing OSGi impl

- Features we need
  - Fine grained dependency
  - AOP support
  - Legacy JMX
  - Structure and real deployers
  - Scoped metadata
  - Open mbeans
  - VFS
- Delegate Classloading
  - Based on VFS

New OSGi personality

- New, new, new … ☺
  - ClassLoader module
    - OSGi aware policy
  - ControllerContexts / ‘personalities’
    - Introduction of Dependency to Deployers
      - See next slide
    - OSGi Services Support
  - Deployers
    - OSGi Manifest.MF Parser
    - DeploymentResolution
    - OSGi Classloader

Deployers and dependencies



Demo

- Deployers
  - PackageVersionParser
  - PVResolutionDeployer
  - ‘OSGi’ ClassLoaderDeployer
- ‘OSGi‘ ClassLoaderPolicy
  - From PVMetaData
- FooService → Acme1
- BarService → Acme2

Wrap up

- Home:
  - Labs page:
    - http://labs.jboss.com/portal/jbossmc
- Design
  - Available on the wiki:
    - http://www.jboss.org/wiki/Wiki.jsp?page=JBossMicrocontainer
- Development:
  - Roadmap on JIRA – JBMICROCONT
  - Discussions in the forums

- Questions?