



The OSGi R5 Enterprise Release

David Bosschaert, Red Hat
Tim Diekmann, TIBCO
EEG co-chairs

Agenda



- Brief Summary
- What's new in R5
- What's next?

Brief Summary



- 2nd Release of *Enterprise Specification*
- Extends *Core Specification R5*
- Driving theme for this release is 'Applications' in OSGi
- Available **now** as an EA draft (official release follows soon)
 - download from www.osgi.org
 - RI and CT to be published end of June 2012

What's New in R5



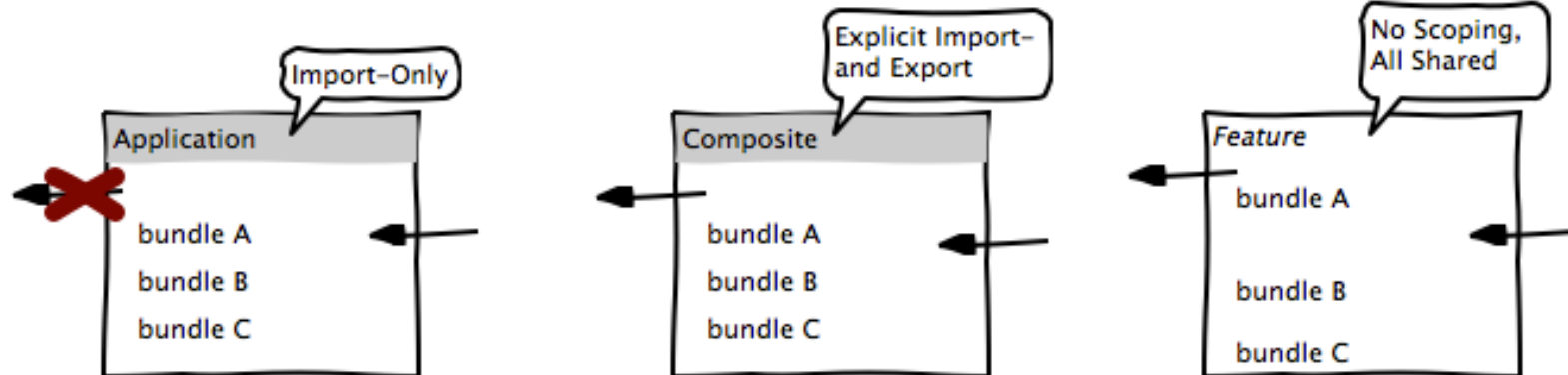
- Support for "Applications" and supporting services
 - Subsystem Service
 - Repository Service
 - Resolver Service
- Auxiliary Services and Specifications
 - Service Loader Mediator
 - Common Namespaces
- Service Updates and Enhancements
 - JMX Management Model
 - Configuration Admin
 - Declarative Services
 - Coordinator Service

Subsystem Service



- Aggregation of resources (incl. bundles, subsystems)
 - three types: *feature*, *composite*, *application*
- Declarative scoping and sharing policies allow for boundaries between different applications
- Common lifecycle of resources in the same scope
- Applications may be self-contained or require external resources
 - uses Repository and Resolver service to identify missing resources and use policies to drive provisioning
- Defines an abstract and concrete description of the subsystem, subsystem archive (*.esa)
- Based on Core framework hooks that allow for limiting visibility and duplicate installation of bundles

Subsystem Types



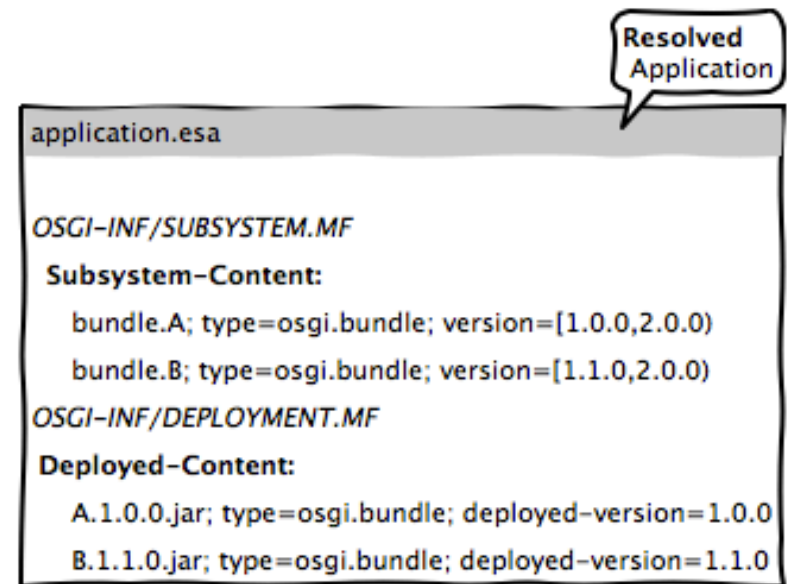
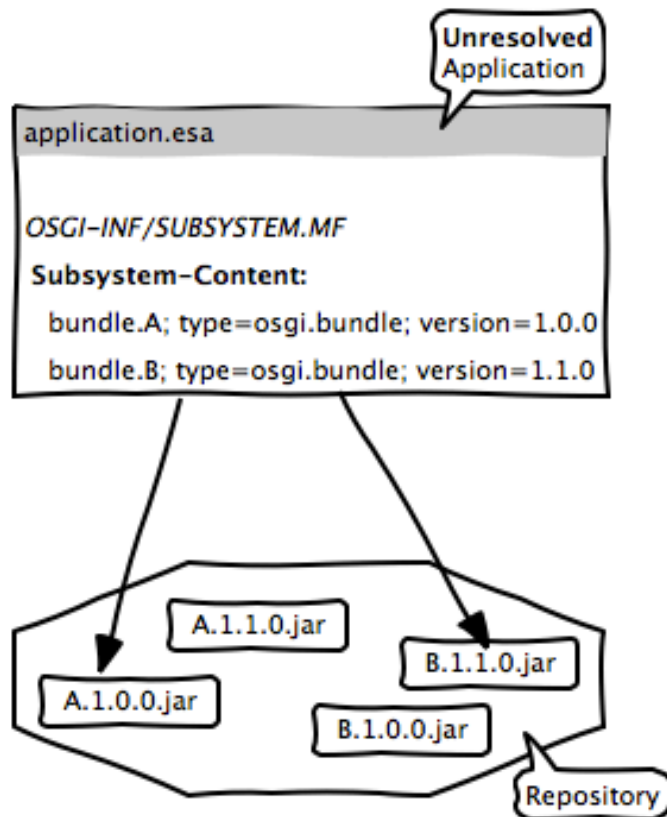
Subsystem Archives



Self-contained Application

```
application.esa
OSGI-INF/SUBSYSTEM.MF
Subsystem-Content:
  bundle.A; type=osgi.bundle; version=[1.0.0,2.0.0)
  bundle.B; type=osgi.bundle; version=[1.1.0,2.0.0)
lib/
  A.1.0.0.jar
  B.1.1.0.jar
```

Subsystem Archives

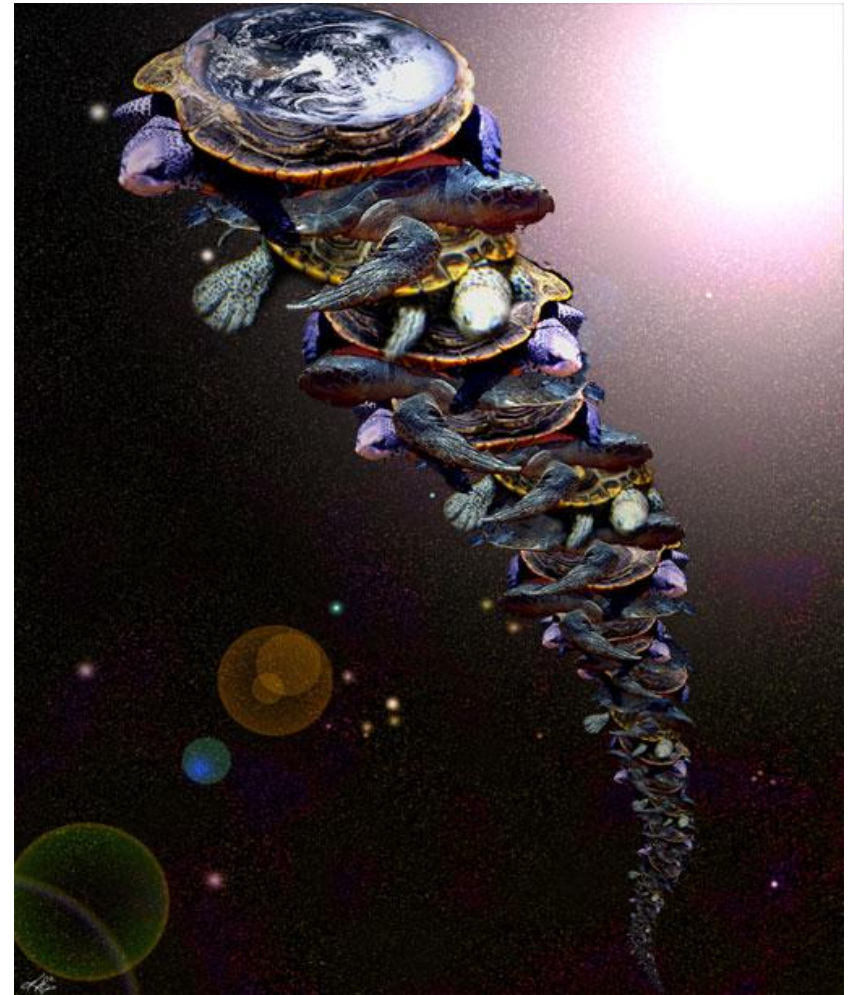


Nested Subsystems



"Turtles All The Way Down"

- A subsystem can contain a subsystem, thereby, allowing subsystems inside subsystems
- The framework itself is the root subsystem that all other subsystems are installed into

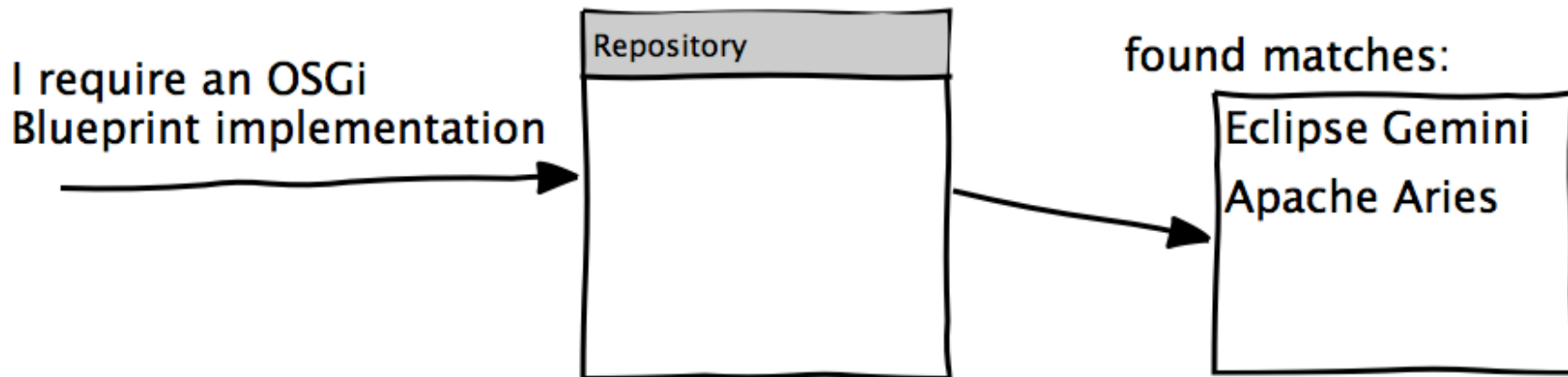
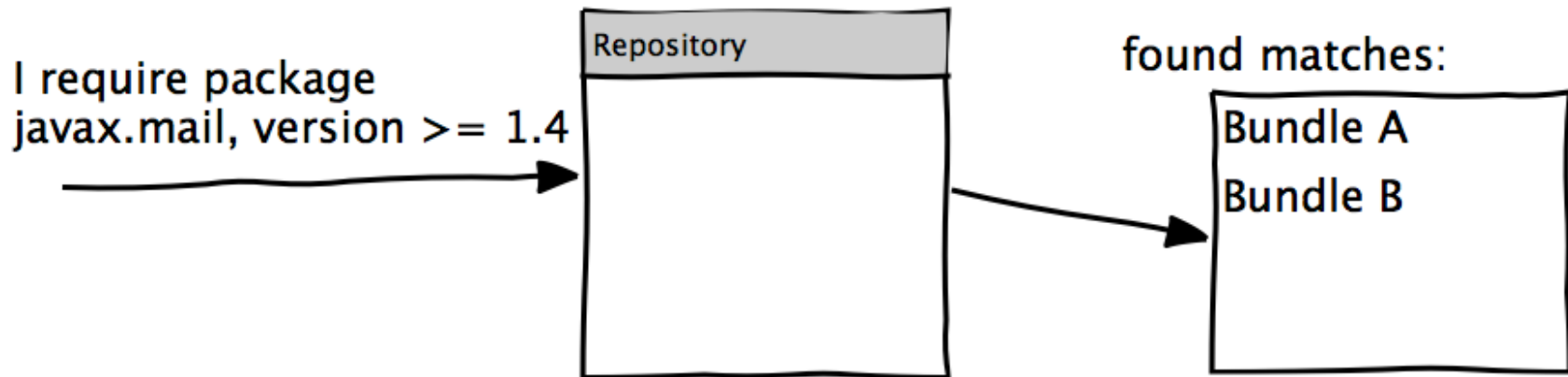


Repository Service



- A Service to access Repositories (local or remote)
 - Bundle or other Resource
 - Query on Generic Capabilities (introduced in Core R4.3 spec)
- Provides resources that can be installed and used in Resolve operations
- Can be used standalone or with Subsystems / Resolver
- Defines XML Schema for repository description
 - Interchange format

Repository Service Examples



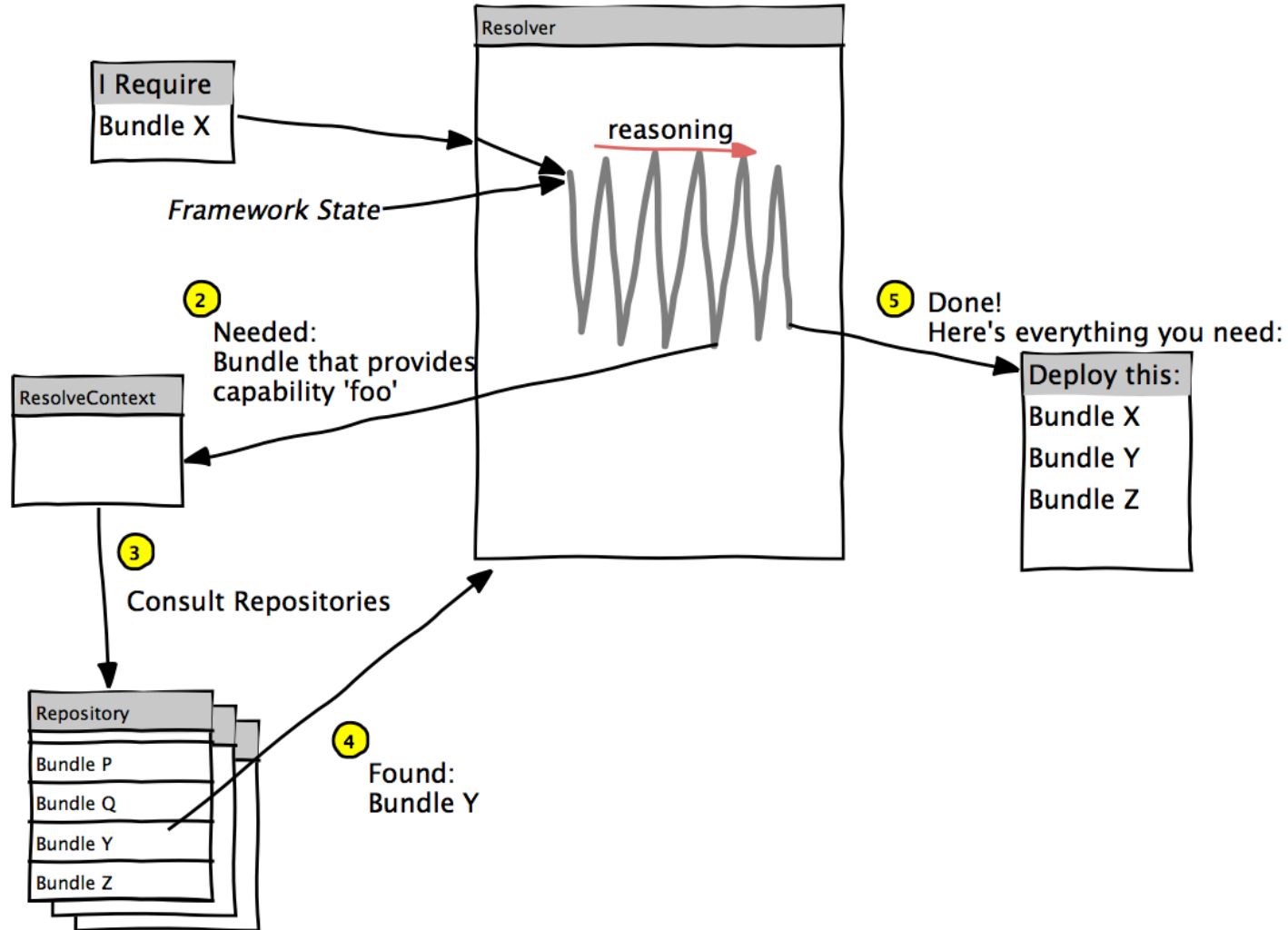


Resolver Service

- Allow management agents to do provisioning
 - *"Find me everything I need to deploy so that this bundle resolves"*
- Can resolve any constraint expressed as a generic capability/requirement
 - Not limited to predefined OSGi namespaces
- Designed to work with Repository Services for dependencies
- Can be used for what-if scenarios
 - *"Will this combination of bundles resolve?"*

Resolver Example

① I need to install Bundle X



OSGi ServiceLoader Mediator



java.util.ServiceLoader is non-modular by design

- expects global visibility: classloader issues
- encapsulation issues: exposure of internals
- hard to get working in general

OSGi ServiceLoader Mediator spec:

1. Services found in **META-INF/services** registered in OSGi ServiceRegistry with **serviceloader.mediator** property
2. Clients using **ServiceLoader.load()** can be made to work **as-is** generally involves weaving, other implementations possible



Common Namespaces

- Generic Requirement-Capability model
 - Introduced in Core 4.3
 - Can be used by Bundles, Resolver, Repository, etc
 - Capability types defined using Namespaces
- Enterprise R5 adds Capability Namespaces for:
 - Services:
 - *"I provide a Configuration Admin service"*

```
Provide-Capability: osgi.service;  
objectClass=org.osgi.service.cm.ConfigurationAdmin
```
 - Extenders:
 - *"I require a Blueprint implementation"*

```
Require-Capability: osgi.extender;  
filter:="(osgi.extender=osgi.blueprint)"
```
 - Contracts:
 - Instead of individual imports: *"I need the Servlet 2.5 API"*

```
Require-Capability: osgi.contract;  
filter:="(&(osgi.contract=Servlet)(version>=2.5))"
```



Service Updates and Enhancements

- JMX Management Model
 - API updates and support Bundle Wiring API
- Configuration Admin Service
 - `SynchronousConfigurationListener`
 - Allow 'transactional' configuration updates across PIDs when used with the Coordinator Service
 - Targeted PIDs
 - Address specific bundle when multiple bundles listen to the same PID, e.g. MS and MSF from different versions
 - Allow multiple bundles to share the same configuration
- Declarative Services
 - Build-time only annotations that are processed by tools like BndTools to create runtime XML component declarations
- Coordinator Service
 - Allows coordination across APIs
 - (e.g. `ConfigurationAdmin` updating multi-PID updates)

What's Next



- "OSGi In The Cloud"
 - see workshop during EclipseCon 2012
- Blueprint enhancements
 - Transactions
 - Configuration Admin integration
 - Namespaces
- WebApp enhancements and update
- Asynchronous services / Scala actors support
- Javascript support
- CDI integration
- EJB support
- JavaEE package version handling
- ... more ...



More

- OSGi Website: www.osgi.org
- OSGi Core Specification R5:
<http://www.osgi.org/download/osgi.core-5.0.0-pfd.pdf>
- OSGi Enterprise Specification R5:
<http://www.osgi.org/download/osgi.enterprise-5.0.0-pfd.pdf>
- OSGi BoF Tuesday 19:00
- "OSGi In The Cloud" workshop at EclipseCon 2012
 - Thursday March 29, 09:00 am
 - register: <http://osgi-cloudworkshop-2012.eventbrite.com>

