JBoss AS7 OSGi

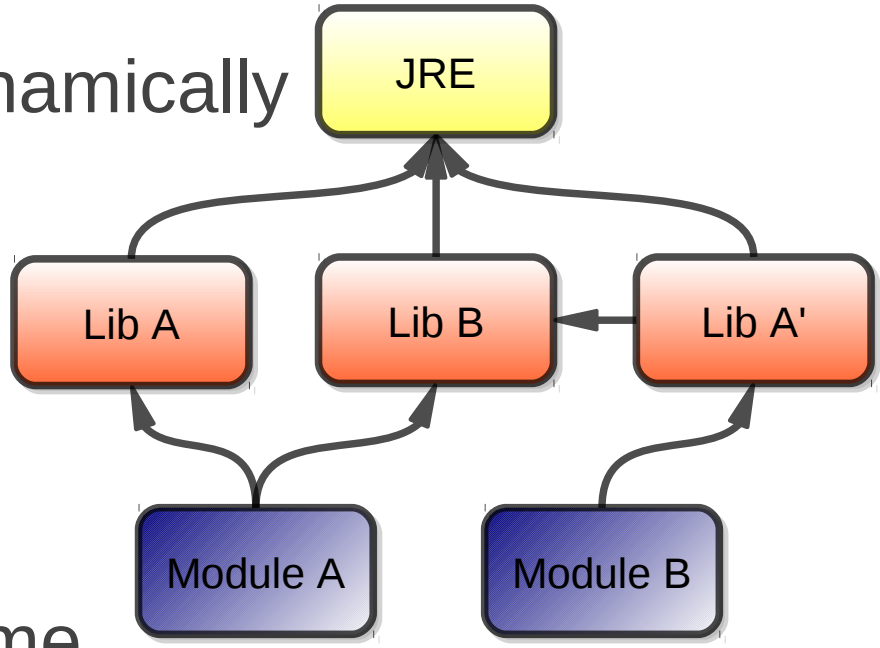
Thomas Diesler
thomas.diesler@jboss.com

Agenda

- OSGi Core
- Enterprise OSGi
- OSGi in AS7
- Demos

OSGi Core: Modularity

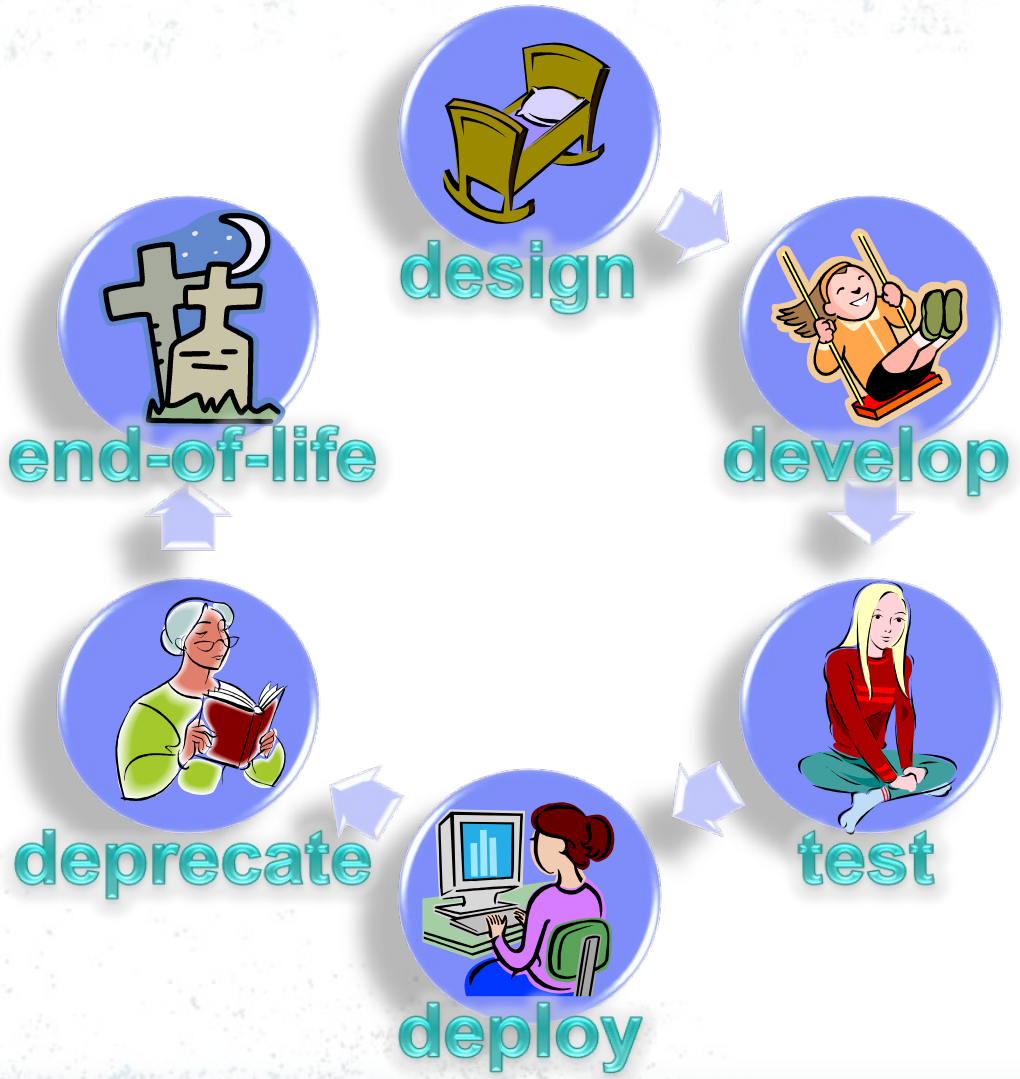
- OSGi modules are called 'Bundles'
- Computed classloader delegation graph
- Bundles can be added, removed & updated dynamically
- Side-by-side versioning
- Private truly private
- No more Jar Hell
- OSGi Bundle can be a plain Jar at the same time



OSGi Core: Services

- “SOA inside the JVM”
- Services looked up by type and/or custom filter
  - “I want a service that implements org.acme.Payment where location=US”
  - One or many
- Dynamic! Services can be updated without taking down the consumers
  - OSGi Service Consumers react to dynamism

The Dawning of Life



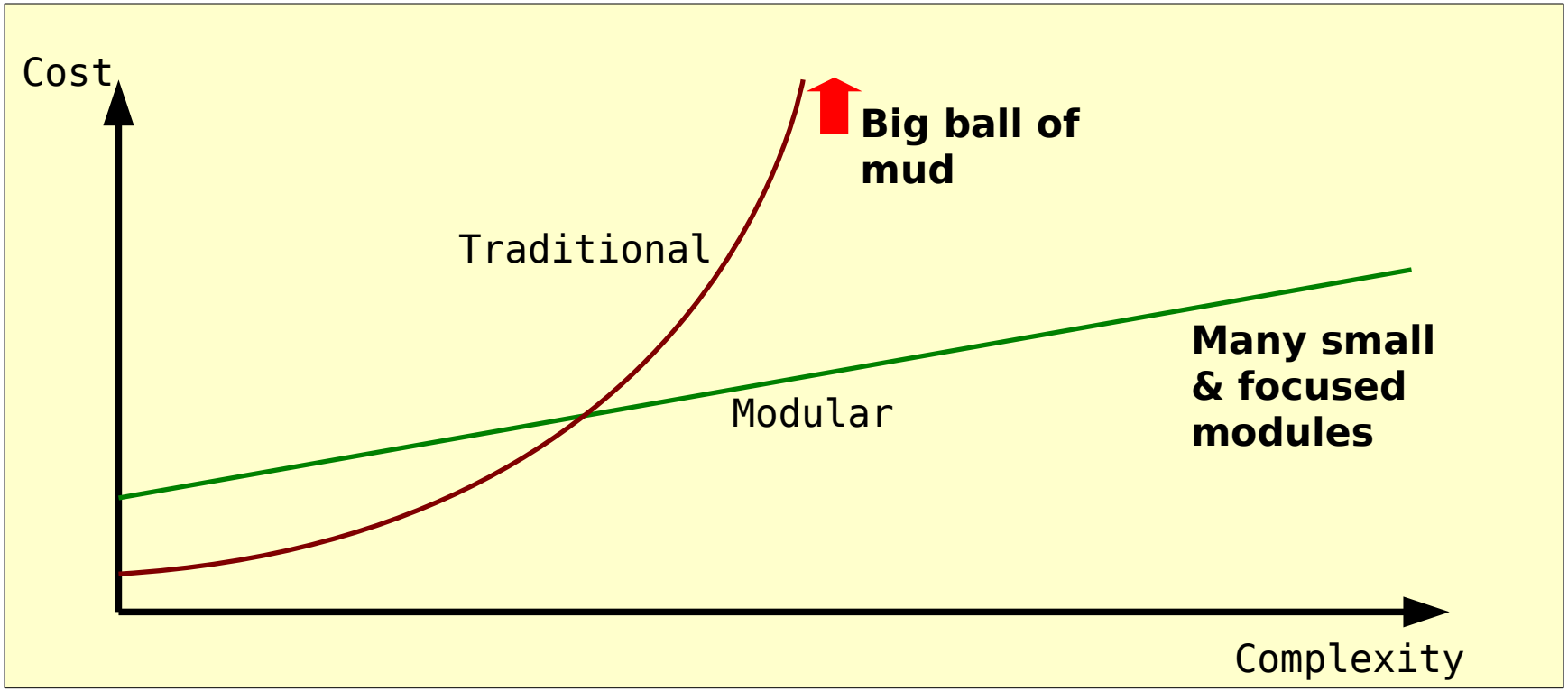
Attention on Modules

Appreciation of module independence

Concern about module life-cycle and governance

…but the problem already existed!

Cost vs Complexity



Modularity highly improves maintainability*

OSGi Bundles vs. JBoss Modules

- OSGi is a standard
- Uses a complex resolver
- Wiring based on rich metadata. Package version just one criteria
- Many compendium specifications and other 3rd party bundles
- Attachable Fragments
- Sensitive to execution env

- Modules not portable
- Uses explicit wiring
- Wiring based on module name. Multiple “slots” are supported
- Exist in the context of AS7
- Very fast resolution

OSGi Service vs. MSC Services

- Keyed by Interface they implement
- LDAP filter supported
- Lifecycle tied to owner bundle or explicit API
- Service dependency and injection via DS, Blueprint
- ServiceListener supported
- ServiceFactory supported
- ConfigurationAdmin

- Keyed by unique service name
- No filtering, name only
- Lifecycle tied to owner owner or explicit API
- Service dependency and injection by MSC API
- ServiceListener supported
- Instance per service name
- ConfigAdmin in AS7

Enterprise OSGi – Today

- Mapping of a number of EE specs
  - Webapp
  - JTA
  - JDBC
  - JPA
  - JNDI
  - JMX
- Remote Services
- Blueprint standard (based on Spring)

Enterprise OSGi - Future

- RFC 152 Subsystems / Applications
- RFC 112 Bundle Repository
- RFC 146 JCA
- RFC 167 ServiceLoader
- RFP 133 Cloud Computing

OSGi Core in AS7

- AS7 is a fully compliant OSGi 4.2 Core Framework
  - TCK tests pass 100%
- Just deploy your OSGi bundles
  - Through the standard AS7 deployment channels

OSGi Core Implementation

- Implemented from scratch on top of jboss-modules and jboss-msc
- For the OSGi resolver we use the one from Apache Felix
  - Resolver computes the bundle classloader wirings
  - Based on OSGi Metadata
    - e.g. Import-Package / Export-Package

OSGi in AS7 - XServices

- Use OSGi services in EE components
- Use EE components in OSGi services
- JBoss modules/msc as integration layer
- Every bundle is a module. Not every module is bundle
- Modules can be given OSGi/Resolver metadata

Enterprise OSGi in AS7

- AS7 out of the Box:
  - OSGi Core, Compendium APIs
  - Enterprise OSGi JMX, JNDI, JTA, WebApp, Blueprint
  - OSGi ConfigAdmin, EventAdmin, HttpServer
  - Customized WebConsole

OSGi in AS7 – Next Steps

- OSGi 4.3 specification
- OSGi Enterprise Applications
- Provide more JBoss functionality as OSGi services:
  - Web Apps using JBoss web
  - CDI integration
  - Possibly JPA/Hibernate support

Demo

- Show how an EJB3 can access an OSGi payment service
- Show how a Servlet can access an OSGi payment service

Q & A

Links

- AS7 home page
  http://www.jboss.org/jbossas
- AS7 OSGi home page
  http://www.jboss.org/jbossas/osgi
- OSGi Specifications
  http://www.osgi.org/Download/Release4V42
- JBoss OSGi Diary
  http://jbossosgi.blogspot.com

Acknowledgements

- Cost vs Complexity, Hal Hildebrand http://bit.ly/i42uBo
- Dawning of Life, Graham Charters, IBM