



Maven & the future of OSGi

Jason van Zyl

SONATYPE

Thursday, March 26th, 2009

What to make of OSGi?

- * Does OSGi suck?

- * No. But it has some severe limitations today which make adoption very difficult.

- * There needs to be an end-to-end solution for OSGi bundle/Eclipse plug-in development:

- * The OSGi runtime model is good. The modularity and classloader system is not going to be done better anytime soon.

- * The programming model has its warts, but this can be overcome with dependency injection (DI) layered upon the runtime. We have examples of this with Peaberry & Spring dm.

- * The infrastructure model is a complete disaster. Working with bundles is difficult as there are no bundles for many standard libraries, no standard bundle repositories, weak automated build support, and developer tooling leaves much to be desired.

What's in an end-to-end solution?

- * Build automation
- * Developer tooling
- * Bundle repositories (Bundle Central, the OSGi equivalent of Maven Central)
- * OSGi server & configuration model
- * Provisioning

Build automation: Tycho

- * Tycho attempts to be a complete replacement for PDE headless build.
- * PDE inside Eclipse is awesome. PDE headless build is not so awesome.
- * Systems, like PDE headless build, which generate build files are impossible to debug. If you think Maven has thrown you for some loops you haven't seen anything until you try to debug PDE headless build.

Which way do you swing?

- * MANIFEST.MF-first mode (look mom! no POM!)
 - * Hard core OSGi developers & Eclipse plugin developers. They use PDE and create and work from manifests. These developers want manifests to be the single source.
- * POM-first mode
 - * Folks coming from Maven who start using OSGi want to use the POM as the single source.

Tycho does OSGi

- * Tycho already uses Eclipse/OSGi metadata to resolve project dependencies by OSGi rules and injects these dependencies into maven project model dynamically, at build time.
- * Tycho uses JDT to make sure modularity rules are applied.
- * Tycho uses the OSGi state resolver is used to make sure the resolution that occurs during build-time matches what you will need at runtime.
- * Tycho supports bundle, fragment, feature and update site projects, as well as RCP applications.
- * Tycho knows how to run JUnit test plugins using the OSGi runtime.
- * Tycho has support for P2 repositories, Update Sites and Maven repositories.

Developer tooling: m2eclipse + PDE

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a Maven project structure with folders like 'Other Projects', 'Team', 'Devoxx', 'ErrorReporting', 'm2eclipse', and 'Maven'. The 'Maven' folder is expanded to show 'maven-core' and its sub-packages, including 'org.apache.maven' and 'lifecycle'. The main editor area shows the 'Welcome' page with sections for 'Links', 'News and Updates', 'Cheat Sheets', and 'Tasks'. The 'News and Updates' section includes 'Sonatype Blogs' and 'Plexus Users Mailing List'. The 'Tasks' section lists several issues related to Plexus component lifecycle and documentation. The bottom status bar shows '343M of 832M' and various system icons.

Package Explorer: Other Projects, dependency, plexus-component-annotations, plexus-component-metadata, plexus-container-default, plexus-containers, JRE System Library [J2SE-1.5], Maven Dependencies, plexus-component-annotations, plexus-component-javadoc, plexus-component-metadata, plexus-container-default, src, target, pom.xml, Team, Devoxx, ErrorReporting, m2eclipse, Maven, apache-maven, maven, maven-compat, maven-core, JRE System Library [J2SE-1.5], src/main/java, org.apache.maven, execution, lifecycle, mapping, DefaultLifecycleExecutor.java, Lifecycle.java, LifecycleExecutionException.java

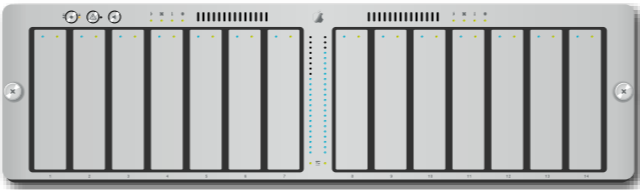
Editor: LifecycleExecutor..., LifecycleExecutor..., plugin/pom.xml, MavenSession.java, AbstractCoreMaven..., org.codehaus.plex... »4

Welcome

- Links
 - Plexus project Web Page
 - Writing Plexus Components
- News and Updates
 - Sonatype Blogs
 - Maven 2.1.0 Released
 - Migrating from Artifactory to Nexus
 - Maven Virtual Versions: Let's Fix this Mess!
 - Installing Nexus on a Solaris Zone
 - Sonatype Maven Meetup on March 19th & 20th
 - More...
 - Plexus Users Mailing List
 - Loading...
 - Plexus Dev Mailing List
 - Loading...
- Cheat Sheets
 - Sample Cheatsheet
- Tasks
 - Issues
 - [PLX-414] Lack of documentation of Plexus component lifecycle and in
 - [PLX-413] DefaultConverterLookup is not thread safe
 - [PLX-410] 5min tutorial doesn't work due to missing artifact
 - [PLX-409] Gleaned components.xml is saved to a wrong place in case
 - [PLX-400] Consider using implementor simple (or fqcn) class name as r
 - More...

Bottom Bar: Welcome Overview Dependencies Repositories Build Plugins Reporting Profiles Team Dependency Hierarchy Dependency Graph pom.xml 343M of 832M

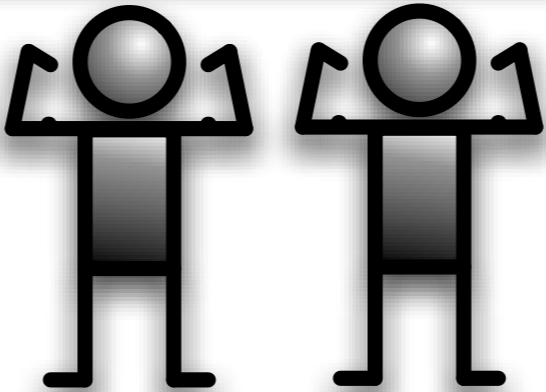
m2eclipse Configuration Framework



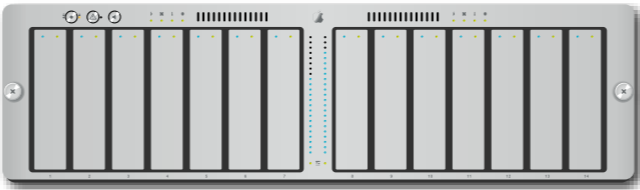
Turn Maven mumbo-jumbo into Eclipse talk



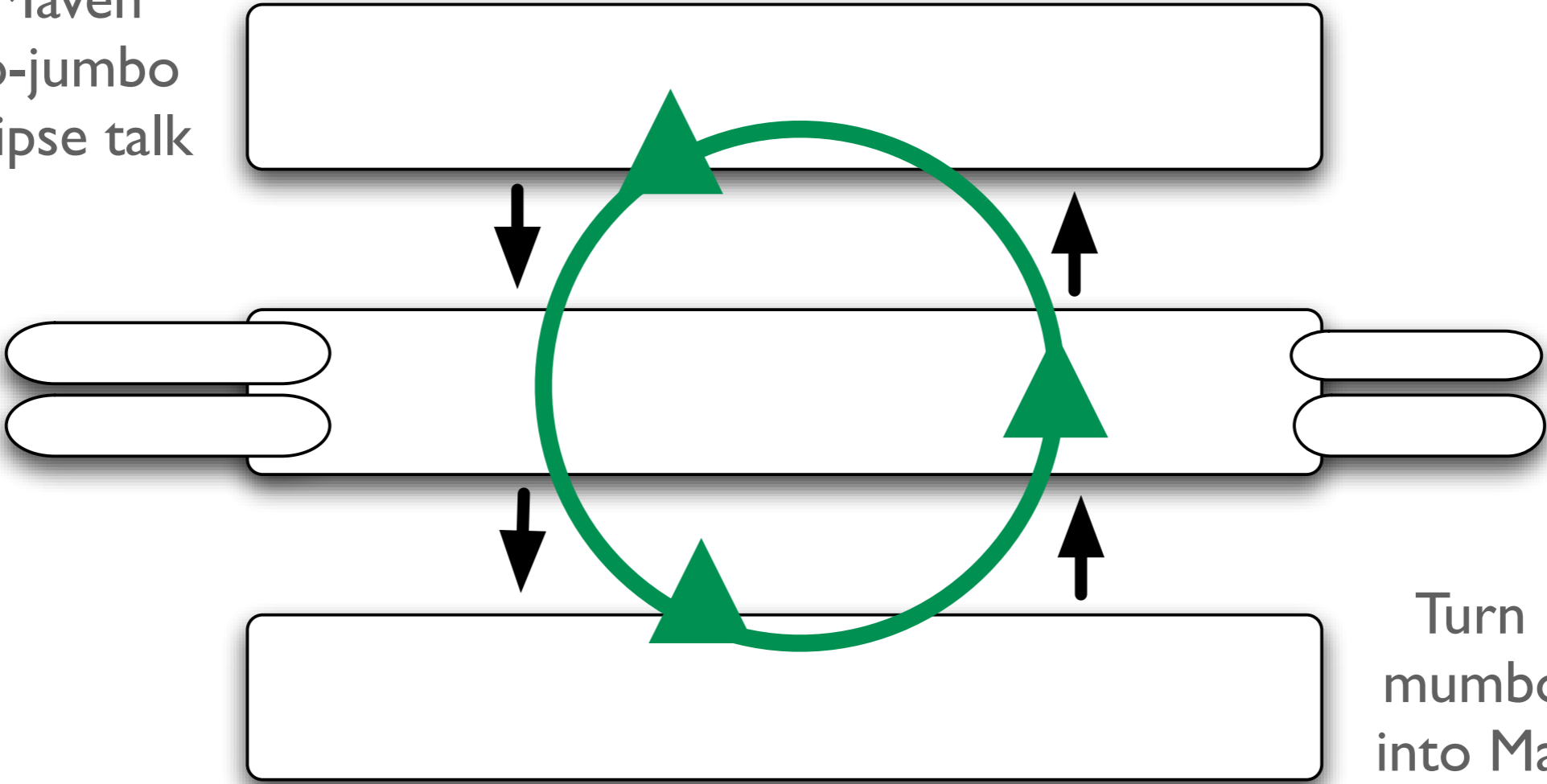
Turn Eclipse mumbo-jumbo into Maven talk



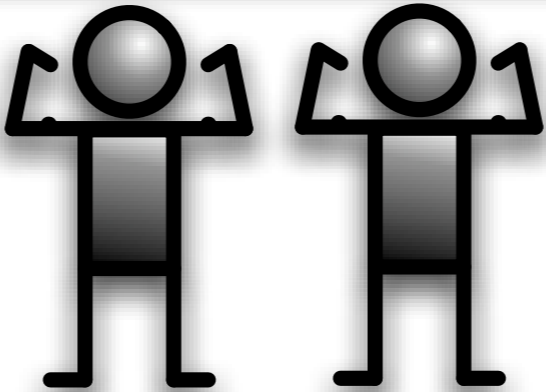
m2eclipse Configuration Framework



Turn Maven mumbo-jumbo into Eclipse talk



Turn Eclipse mumbo-jumbo into Maven talk



m2e configuration framework

✻ Open framework that supports

- ▶ Initial project configuration during import or update configuration operations
- ▶ Maven project model change event notification
- ▶ Maven dependencies classpath container content manipulation
- ▶ Custom maven lifecycle mapping and build integration (planned for 0.9.8)

Bundle repositories: Nexus

Sonatype Nexus™ Professional Edition Log In
Version 1.3.1 Professional

Sonatype™ Servers << Nexus

Artifact Search
Advanced Search

Views/Repositories
Repositories
System Feeds

Help

Welcome | Repositories

Refresh User Managed Repositories

Repository	Type	Format	Policy	Repository Status	Repository Path
Public Snapshot Repositories					
Uber P2 Repository	group	maven2		AJDT Update Site, Equinox Aspects...	http://osgi.sonatype.org/content/groups/all
3rd party	hosted	maven2	release	In Service	http://osgi.sonatype.org/content/repositories/thirdparty
AJDT Update Site	hosted	p2	release	In Service	http://osgi.sonatype.org/content/repositories/ajdt
Apache Snapshots	proxy	maven2	snapshot	In Service	http://osgi.sonatype.org/content/repositories/apache-snapshots
Central M1 shadow	virtual	maven1		In Service	http://osgi.sonatype.org/content/repositories/central-m1
Codehaus Snapshots	proxy	maven2	snapshot	In Service	http://osgi.sonatype.org/content/repositories/codehaus-snapshots
Equinox Aspects Update Site	proxy	p2	release	In Service	http://osgi.sonatype.org/content/repositories/equinox-aspects
Ganymede Update Site	proxy	p2	release	In Service	http://osgi.sonatype.org/content/repositories/ganymede
M2Eclipse Update Site	hosted	p2	release	In Service	http://osgi.sonatype.org/content/repositories/m2eclipse
M2Eclipse Update Site (without dependencies)	hosted	p2	release	In Service	http://osgi.sonatype.org/content/repositories/m2eclipse-simple
Maven Central	proxy	maven2	release	In Service	http://osgi.sonatype.org/content/repositories/central
Orbit Update Site	hosted	p2	release	In Service	http://osgi.sonatype.org/content/repositories/orbit
Releases	hosted	maven2	release	In Service	http://osgi.sonatype.org/content/repositories/releases
Releases P2	virtual	p2		In Service	http://osgi.sonatype.org/content/repositories/releases-p2
Snapshots	hosted	maven2	snapshot	In Service	http://osgi.sonatype.org/content/repositories/snapshots
Snapshots P2	virtual	p2		In Service	http://osgi.sonatype.org/content/repositories/snapshots-p2
Subclipse Update Site	hosted	p2	release	In Service	http://osgi.sonatype.org/content/repositories/subclipse

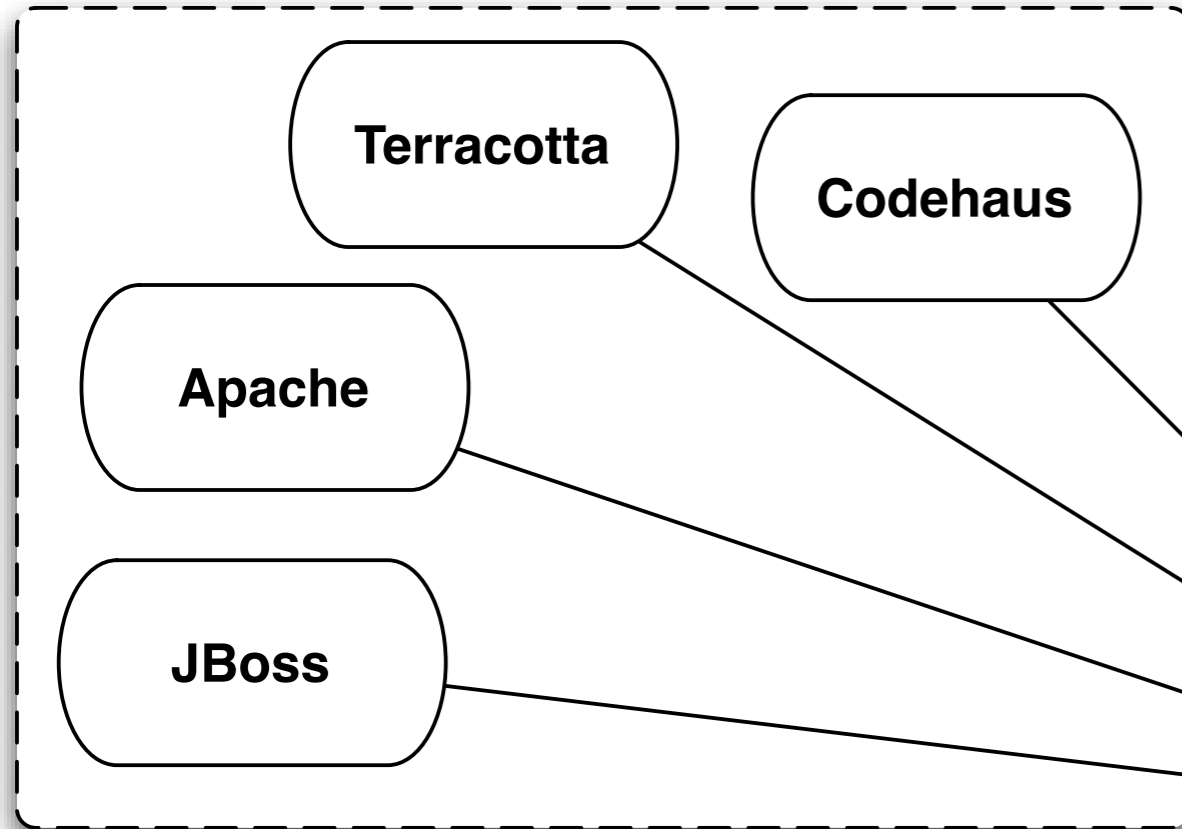
Orbit Update Site

Refresh Path Lookup: Browse Local Storage

- org.objectweb.asm.source_3.0.0.v200803061811.jar
- org.objectweb.asm.source_3.1.0.v200803061910.jar
- org.objectweb.asm_3.0.0.v200803061811.jar
- org.objectweb.asm_3.1.0.v200803061910.jar
- org.sat4j.core_2.0.0.v20080602.jar
- org.sat4j.pb_2.0.0.v20080602.jar
- org.uddi4j_2.0.5.v200805270300.jar
- org.w3c.css.sac.source_1.3.0.v200805290154.jar
- org.w3c.css.sac_1.3.0.v200805290154.jar
- org.w3c.dom.smil.source_1.0.0.v200806040011.jar
- org.w3c.dom.smil_1.0.0.v200806040011.jar
- org.w3c.dom.svg.source_1.1.0.v200806040011.jar
- org.w3c.dom.svg_1.1.0.v200806040011.jar

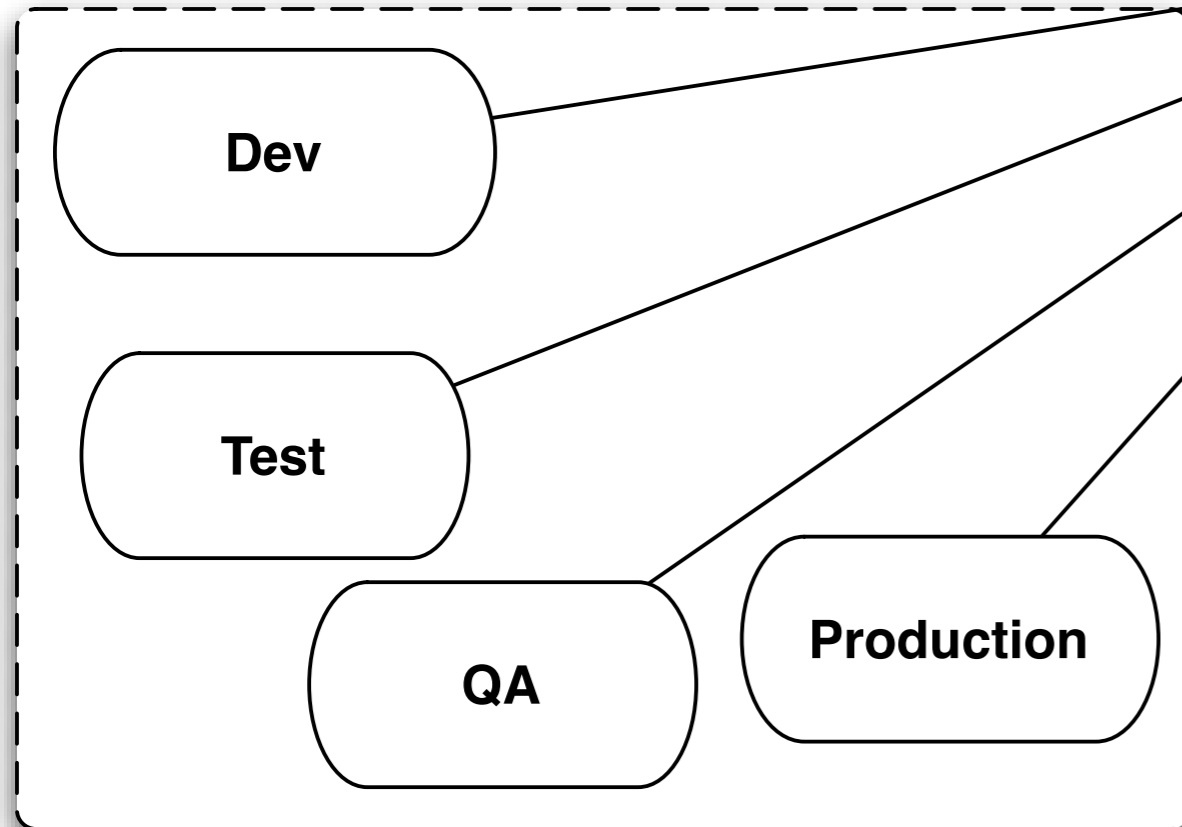
artifacts.xml
content.xml

Repository & Configuration Management



Open Source Ghetto

- bad POMs
- repositories in POMs (bad dog!)
- mixed snapshot and release repositories
- bad optional dependencies
- screwed up metadata

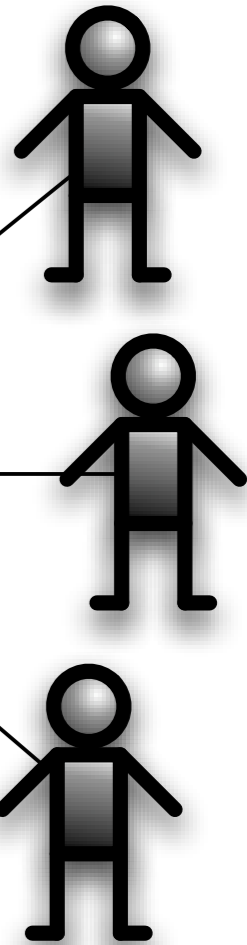


Nexus

One repository configuration!

Group

Unified indices for m2e

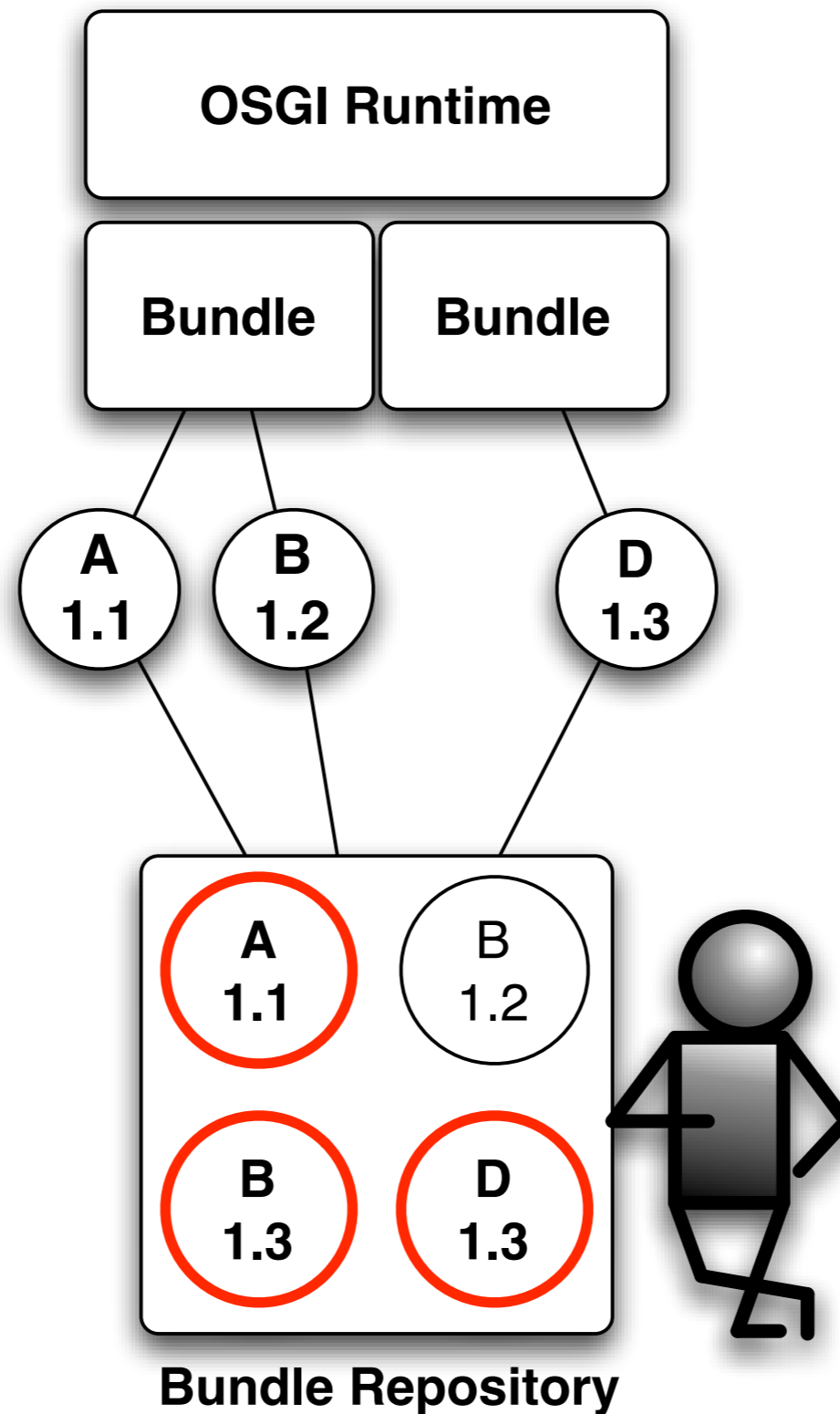


Precious Developers

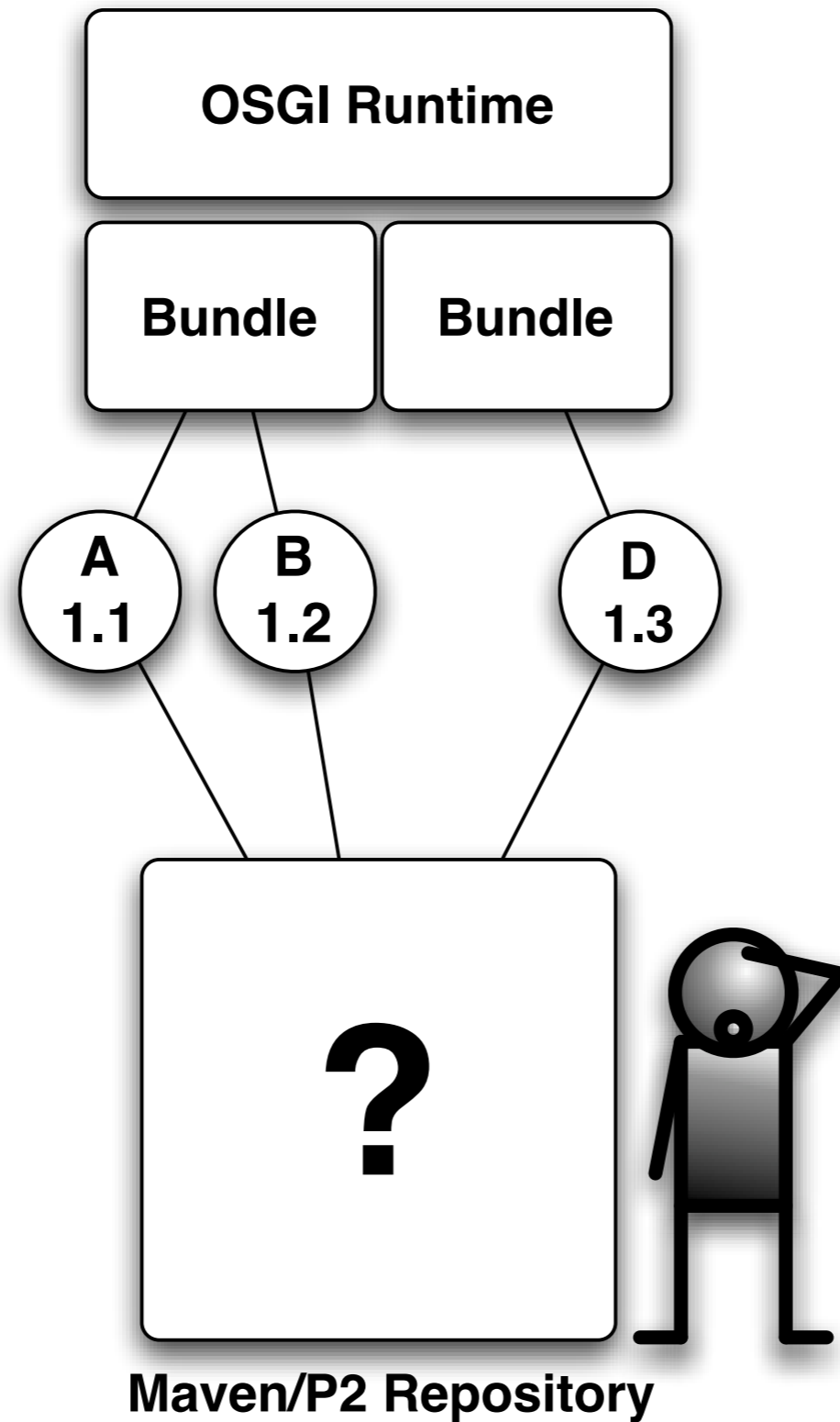
Your Ghetto World

Just kidding, your dev environment is probably perfect

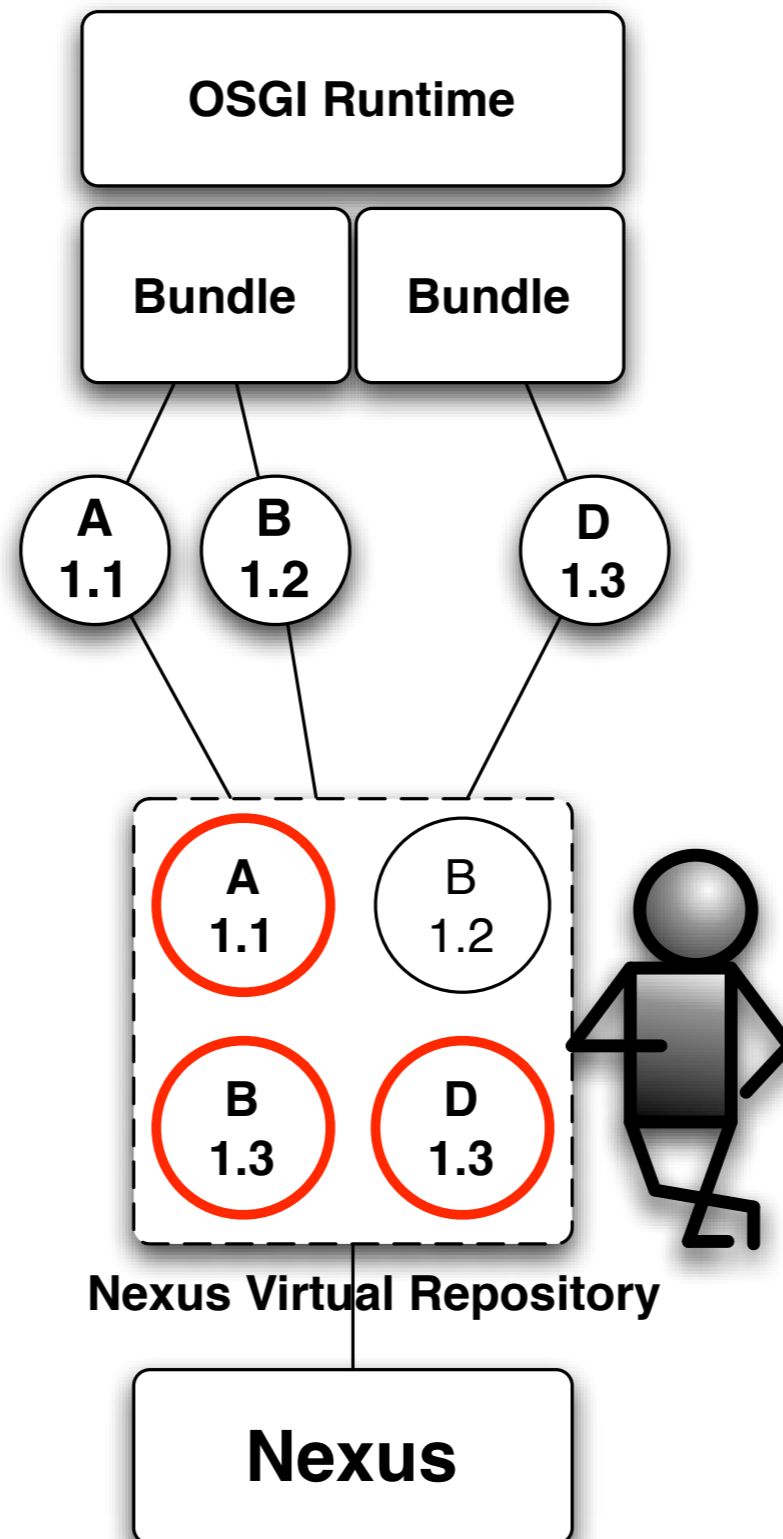
Typical way people deal with OSGi



OSGi is greedy & it's a problem



Resolving against a virtual Nexus repository



OSGi server & configuration model

- * There are solutions like SpringDM but they seem fairly proprietary and custom build frameworks are just not scalable.
- * Sonatype has ASF licensed Equinox runtime templates created to work with Maven tooling
 - * Standard Equinox distribution
 - * Java Service Wrapper
 - * NexusInstall Bundle for retrieving and installing bundles
- * There is still a need for a configuration model
 - * Instance archives (ITARs) and Configuration archives (CARs)
 - * An ITAR and CAR are combined and processed to make a running instance
 - * ITARs and CARs are versioned artifacts like everything else in a good artifact repository
 - * ITARs need to be pure binary distributions devoid of all configuration and startup/shutdown scripts
 - * CARs container parameterized (Velocity templates) which can be interpolated against any number of sources like a properties file, a database, LDAP, system properties, or ENVARs
 - * With this model you can create sets of production nodes for any type of task or use

Provisioning tooling: p2

- ✿ The p2 provisioning framework is the most advanced provisioning system available for OSGi and it's unlikely anything else will catch up.
- ✿ What about OBR? The supposedly standard provisioning specification from the OSGi Alliance? In short I don't think there is anything useful there yet.
- ✿ Although p2 has focused on the Update Manager inside Eclipse it is a general provisioning tool for OSGi
- ✿ p2 + Nexus == very power full provisioning system

QA

✻ What questions to you have?