

**ORACLE<sup>®</sup>**

## **Maven, Spring and Coherence**

Jon Hall - Oracle UK Consulting  
[jonathan.hall@oracle.com](mailto:jonathan.hall@oracle.com)

# About me

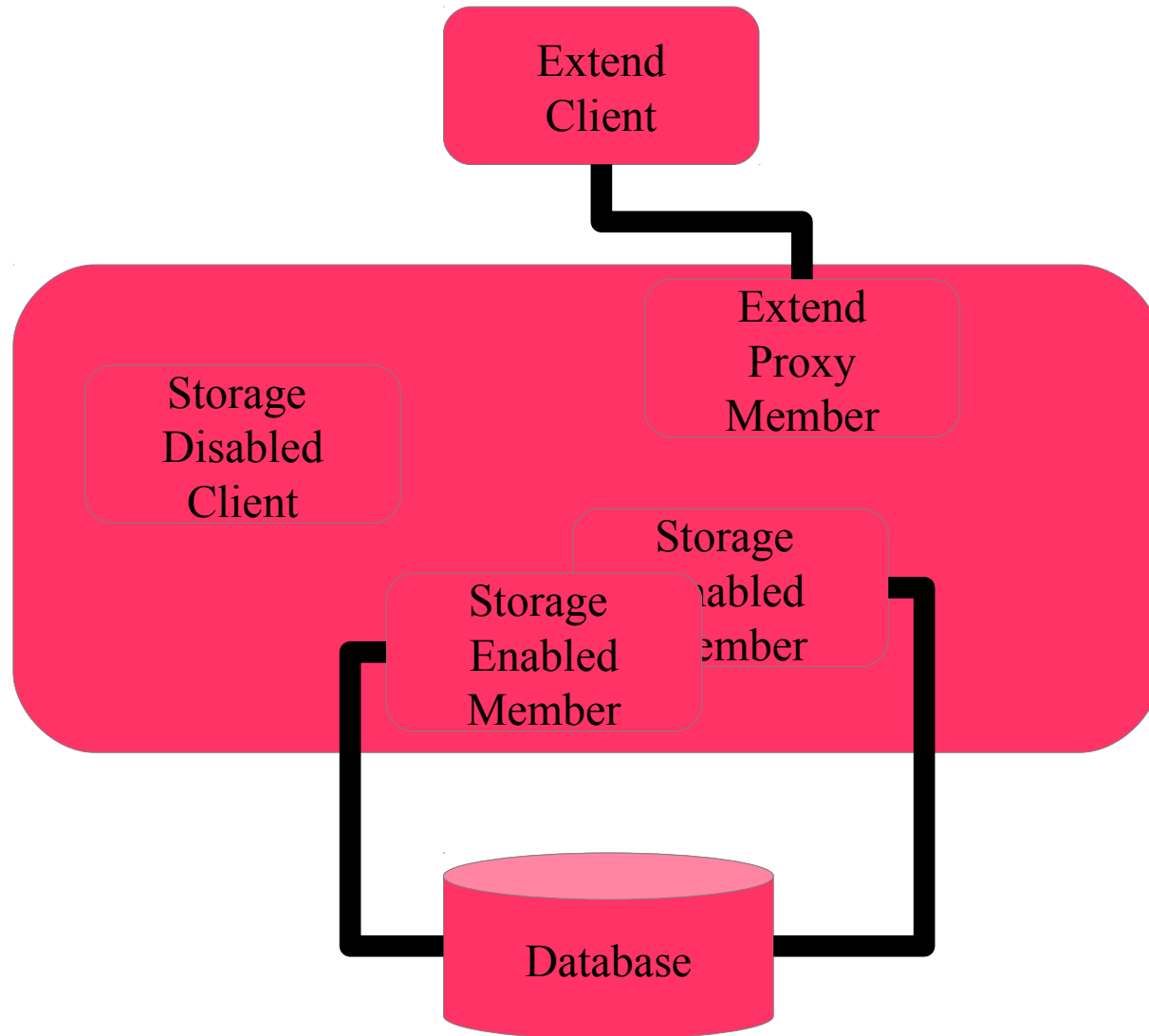
- Pragmatic, hands-on, delivery focused Enterprise Developer/Architect with over 21 years IT experience
- Spring since 2005 and Maven since 2008 – like KISS
- Have worked for Oracle previously (part of Tangosol acquisition) – Coherence
- littlegrid Coherence test-support framework Developer
- <http://www.linkedin.com/jhall>

# Disclaimer

The following is intended to outline general product use and direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Session overview



# Session overview

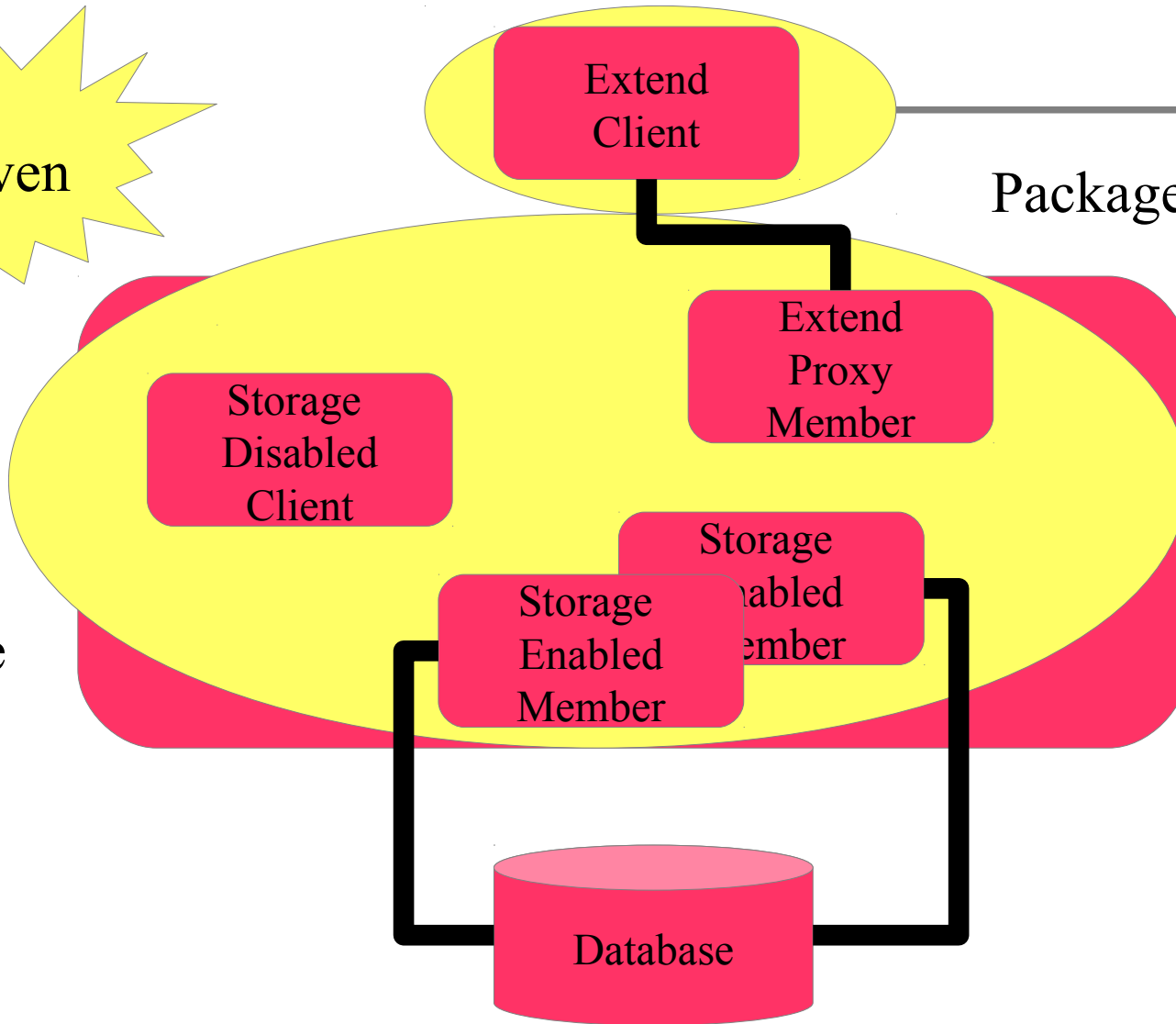


Maven

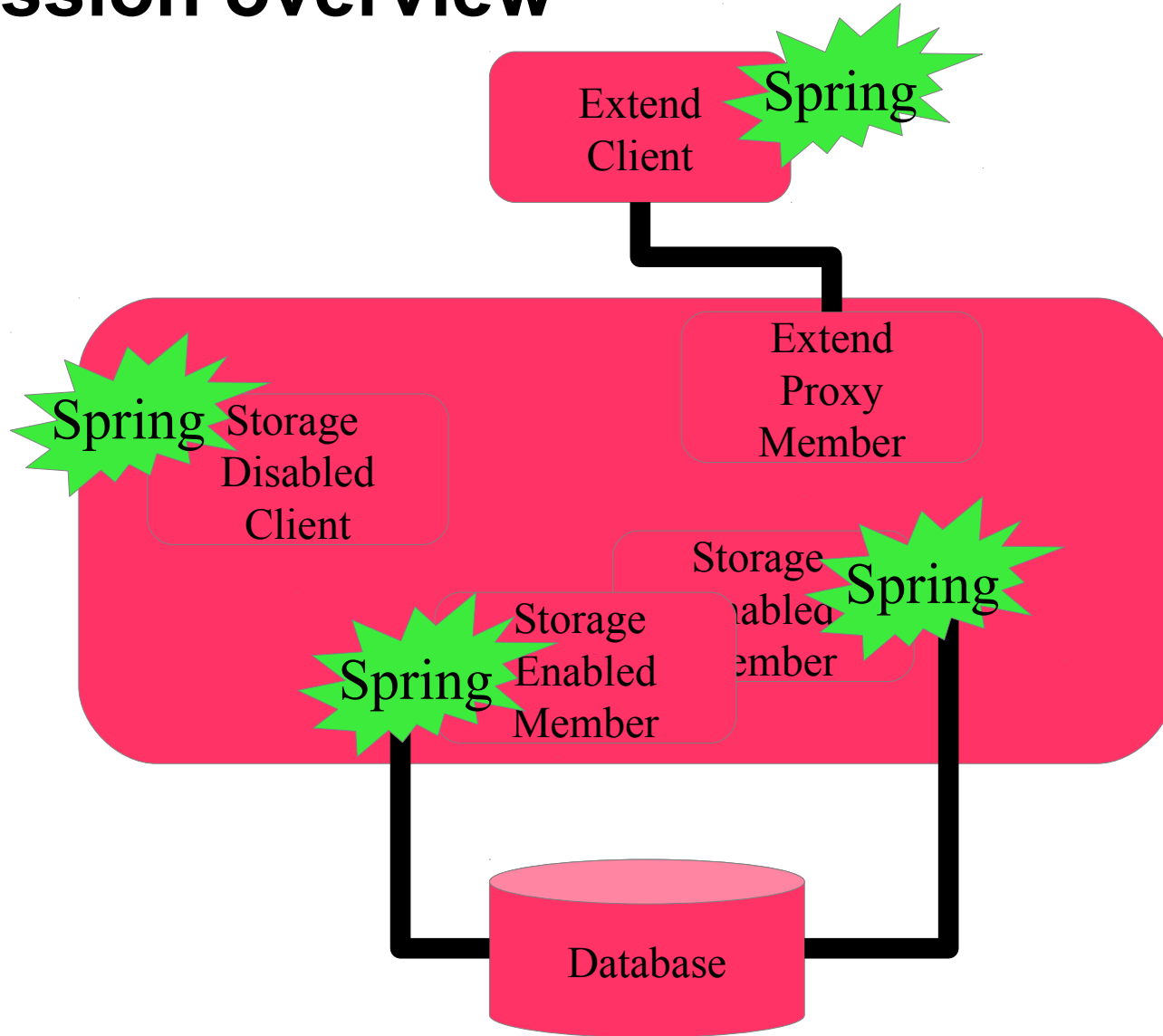
Package as a JAR

Package as tar.gz,  
Zip etc.

Compile  
Test



# Session overview



# Session overview

....from zero to using on a project in about 30 minutes

With a few of my opinions thrown in :-)

Will demo some example code

Possible future Coherence improvements/integration with  
Spring and Maven

# Maven



# Maven: WHAT? WHY?

- Maven's primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal there are several areas of concern that Maven attempts to deal with:
  - Making the build process easy
  - Providing a uniform build system
  - Providing quality project information
  - Providing guidelines for best practices development
  - Allowing transparent migration to new features
- Source: Maven website

# Maven: understanding the bits

pom.xml

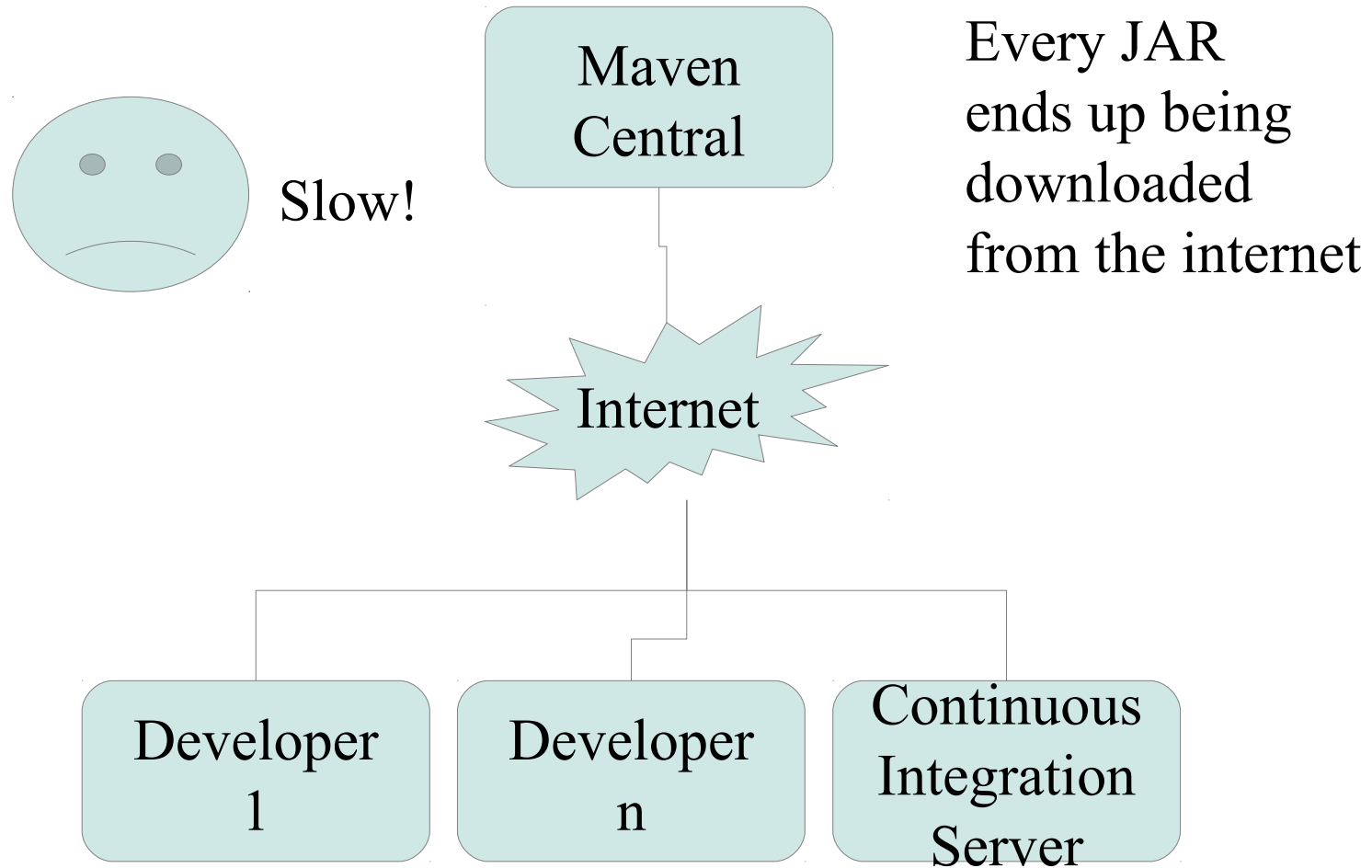
settings.xml

Plugins – compile, JAR, Surefire etc

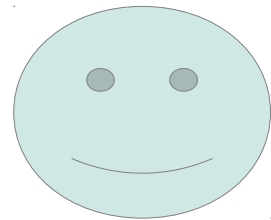
Local repository (on your machine)

Repository manager

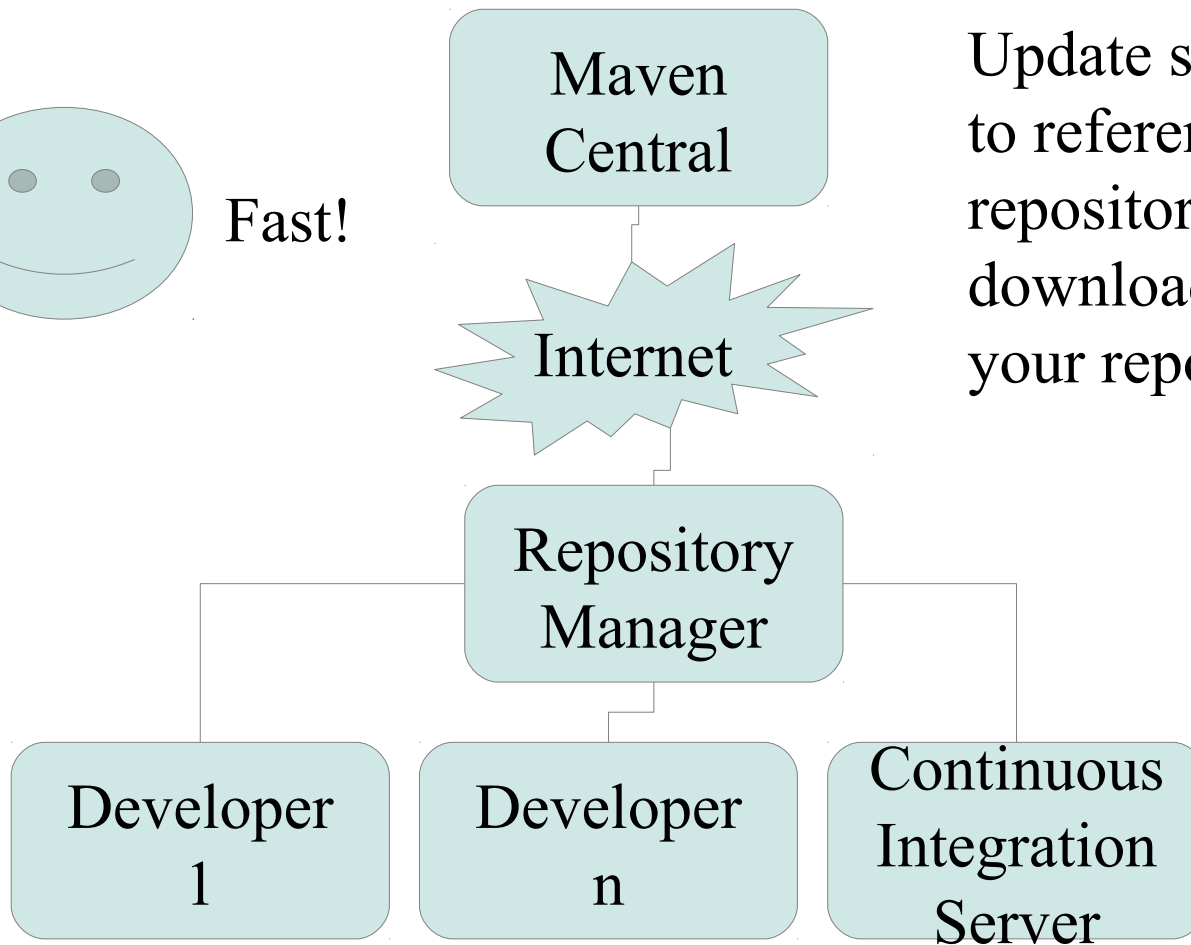
# Maven: publicly available JARs without a repository manager



# Maven: publicly available JARs with a repository manager



Fast!



Update settings.xml to reference your repository - downloaded from your repository

# Maven and Coherence

# Group, Artifact and version (GAV) parameters

The below is what the GAV \*might possibly\* look like:

```
<groupId>com.oracle.coherence</groupId>  
<artifactId>coherence</artifactId>  
<version>3.7.1.4</version>
```

<http://maven.apache.org/guides/mini/guide-naming-conventions.html>

# Maven: 3<sup>rd</sup> party JARs – like Coherence

Downloading:

<http://repo1.maven.org/maven2/com/oracle/coherence/coherence/3.7.1.4/coherence-3.7.1.4.pom>

[WARNING] The POM for

com.oracle.coherence:coherence:jar:3.7.1.4 is missing,  
no dependency information available

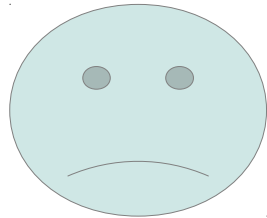
Downloading:

[public/com/oracle/coherence/coherence/3.7.1.4/coherence-3.7.1.4.jar](http://repo1.maven.org/maven2/com/oracle/coherence/coherence/3.7.1.4/coherence-3.7.1.4.jar)

Downloading:

<http://repo1.maven.org/maven2/com/oracle/coherence/coherence/3.7.1.4/coherence-3.7.1.4.jar>

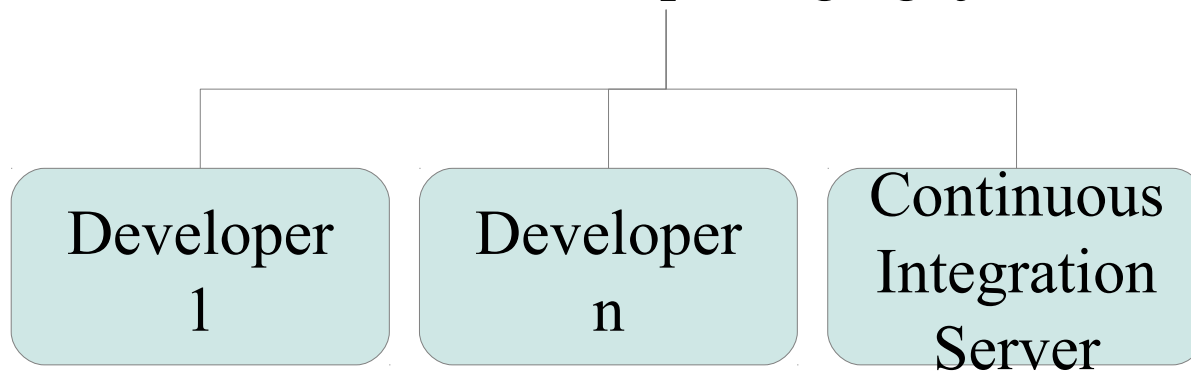
# Maven: 3<sup>rd</sup> party JARs without a repository manager



Arrh!

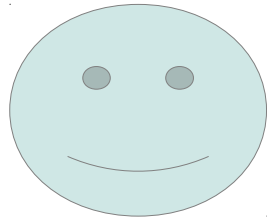
Every 3<sup>rd</sup> party JAR  
needs to be installed  
manually

```
mvn install:install-file -Dfile=coherence.jar -DgroupId=com..  
-DartifactId=.. -Dversion=.. -Dpackaging=jar
```



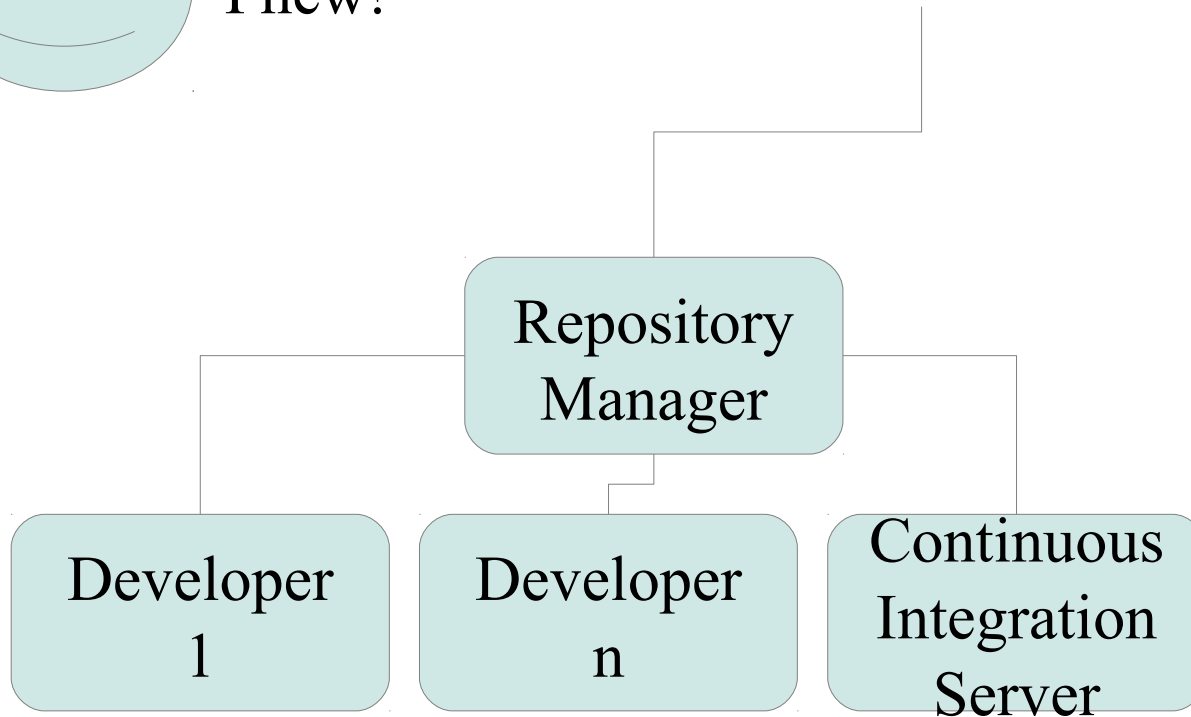


# Maven: 3<sup>rd</sup> party JARs with a repository manager



Phew!

Maven deploy or upload to  
your repository manager



# Maven repository manager summary

Downloads and caches publicly available JARs that would normally need to be downloaded from Maven central

Place to deploy or upload 3<sup>rd</sup> party JARs that are not available from Maven central

Place to deploy your own artifacts (JARs, ZIPs, tar.gz) when building and releasing your own projects – for sharing binary releases

# Maven repository manager summary

If you're using Maven, then for your own sanity use a repository manager ;-)

For those with their own OSS projects that want to make it available via Maven central: <http://oss.sonatype.org>

# Maven argLine: defining in POM file

Required if you want to pass system properties to the Surefire plugin

```
<build>  
  <plugins>  
    <plugin>  
      <groupId>org.apache.maven.plugins</groupId>  
      <artifactId>maven-surefire-plugin</artifactId>  
      <configuration>  
        <argLine>-Xmx768m -Xms768m</argLine>  
      </configuration>
```

# Maven argLine: technique for appending additional parameters

```
<profile>
  <id>customArgLineToBeUsed</id>
  <activation>
    <property>
      <name>argLine</name>
    </property>
  </activation>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <configuration>
          <argLine>-Xmx768m -Xms768m ${argLine}</argLine>
        </configuration>
      </plugin>
    </plugins>
  </build>
</profile>
```

# Maven argLine: passing additional parameters to custom argLine profile

mvn clean install

```
-DargLine="-Dtangosol.coherence.wka.port=40011"
```

Here, the previous defined memory settings have been used and additional the parameter(s) are also passed and used

# Maven assembly plugin: specifying in the POM

```
<plugins>
  <plugin>
    <artifactId>maven-assembly-plugin</artifactId>
    <executions>
      <execution>
        <id>package</id>
        <goals>
          <goal>single</goal>
        </goals>
        <phase>package</phase>
      </execution>
    </executions>
    <configuration>
      <descriptors>
        <descriptor>src/main/assembly/data-grid-server-deploy.xml</descriptor>
      </descriptors>
    </configuration>
  </plugin>
</plugins>
```

# Maven assembly plugin: specifics in assembly descriptor

```
<assembly>
  <id>deploy</id>
  <formats>
    <format>tar.gz</format>
  </formats>
  <dependencySets>
    <dependencySet>
      <outputDirectory>lib</outputDirectory>
      <scope>runtime</scope>
    </dependencySet>
  </dependencySets>
  <fileSets>
    <fileSet>
      <directory>src/main/deploy/bin</directory>
      <outputDirectory>bin</outputDirectory>
      <fileMode>555</fileMode>
    </fileSet>
  </fileSets>
</assembly>
```



**Spring**

# Spring: WHAT? WHY?

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

Source: Spring framework website

# SpringAwareCacheFactory

Supported method of integration between Coherence and Spring

[http://docs.oracle.com/cd/E24290\\_01/coh.371/e22621/integratespring.htm](http://docs.oracle.com/cd/E24290_01/coh.371/e22621/integratespring.htm)

Coherence 3.4.x (or higher)

Spring 2.x (or higher)

# Cache Store: Spring on server-side

Configurable-cache-factory-config in override file

```
<cachestore-scheme>
```

```
  <class-scheme>
```

```
    <class-name>spring-bean:accountCacheStore</class-  
name>
```

```
  </class-scheme>
```

```
<bean id="accountCacheStore"
```

```
  class="...account.repository.impl.AccountCacheStoreImpl"
```

```
  scope="prototype">
```

# Spring: a simple, but useful configuration technique that I use

```
<bean id="basePropertiesConfigurer"  
      class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  
  
    <property name="ignoreResourceNotFound" value="true"/>  
    <property name="systemPropertiesModeName"  
value="SYSTEM_PROPERTIES_MODE_OVERRIDE"/>  
  
    <property name="locations">  
      <list>  
        <value>classpath:/properties/projectname-default.properties</value>  
        <value>file:${LOCAL_PROPERTIES_OVERRIDE_HOME}/projectname-  
local.properties</value>  
      </list>  
    </property>  
</bean>
```

# Backing Map: Spring on server-side

Code and demo

# Named Cache: Spring on client-side

Coherence's named cache is typically wired into a repository or service

```
<bean id="accountCache"  
      class="com.tangosol.net.CacheFactory"  
      factory-method="getCache">  
  <constructor-arg value="account"/>  
</bean>
```

# Named Cache and Cache Store

Code and demo



# Typical patterns

Typically wired into a Repository (or DAO) Service:

```
<bean id="accountRepository"  
class="..account.repository.impl.AccountRepositoryCoher  
enceImpl">
```

```
    <property name="accountCache"  
ref="accountCache"/>  
</bean>
```

# Anti-pattern

Don't do this in your code – checking for dependency and doing dependency 'pull' if not already initialised:

```
if (cache == null) {  
    NamedCache cache = CacheFactory.getCache("account");  
}
```

If you're worried about your dependencies not being injected (which you shouldn't be as it is your code), then you could use `@Required` to check dependencies

# Spring AOP and Coherence side cache

Code and demo

# Maven, Spring and Coherence - possible future

# Possible future integration of Maven and Coherence

Could have officially documented GAV parameters

Could have officially released POM to allow for easy deploying/uploading of Coherence to your Repository manager

Could be possible to request Coherence from an official Oracle Maven Repository

Could be possible to install Coherence into enterprise repository from an official Oracle Maven Repository

# Possible future integration of Spring and Coherence

Could imagine how, using the namespace technique introduced into the Coherence commons a while ago, it would be possible to have a Spring namespace that allows the referencing of beans in a Spring context to be used by objects defined within the cache configuration, e.g. a cache store etc. and how Spring would look after the objects defined within its context (instead of Coherence).

# Coherence Incubator and Maven

Coherence Incubator is moving over to be entirely Maven-based

Hopefully Brian will be joining us virtually later to talk about this and what else is changing in the Incubator



# Summary

Thanks!

Questions? Thoughts?

How do you do it?