Construction automatique avec Maven

Philippe Collet

Polytech'Nice Sophia – SI3

2017-2018

LA LIVRAISON DU KATA POKER

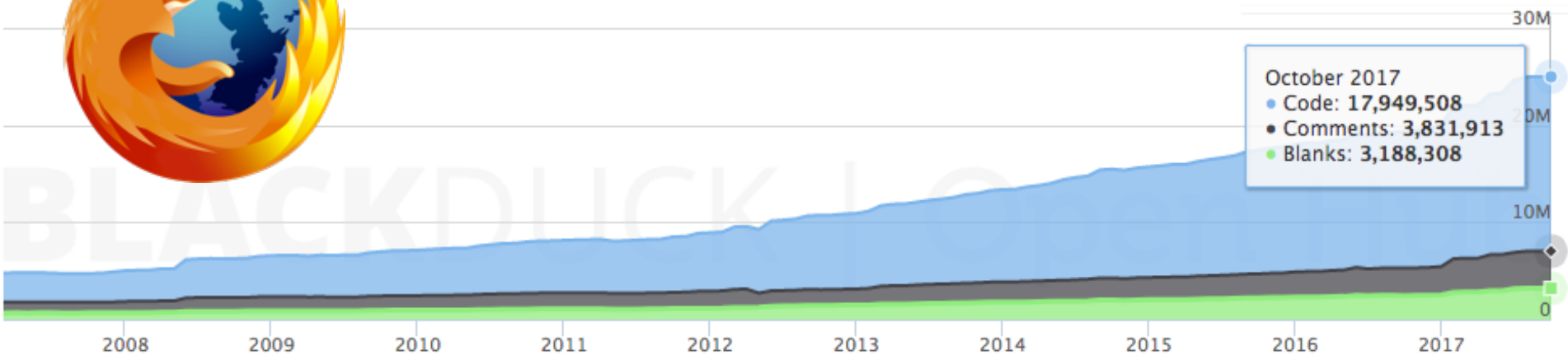
Parmi les 21 équipes…

- Pas mal de compilations Maven réussies
- 1 Makefile
- 1 script Python
- 1 script d'install Unix qui appelle des commandes Maven (hein ?)
- 4 pom.xml placés dans un sous-répertoire du git avec tout le projet
- 1 pom.xml placé dans un sous-package de test (hein ?)
- 4 commandes d'exécution Java pas compatibles avec mon système (\ windows partout)


I CLOSE MY EYES AND I CAN HEAR YOU SAY
YOU'RE NOT ALONE

https://www.openhub.net/p/firefox

Du gros code ?



Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	5,638,126	1,164,958	17.1%	1,068,420	7,871,504	31.5%
JavaScript	3,843,910	1,248,123	24.5%	861,445	5,953,478	23.8%
HTML	2,580,233	143,074	5.3%	341,763	3,065,070	12.3%
C	2,391,984	639,402	21.1%	400,702	3,432,088	13.7%
Rust	918,990	166,362	15.3%	100,608	1,185,960	4.7%
XML	775,965	19,065	2.4%	50,659	845,689	3.4%
Python	558,804	162,832	22.6%	141,870	863,506	3.5%
Java	328,587	122,226	27.1%	67,792	518,605	2.1%
CSS	241,678	14,184	5.5%	37,701	293,563	1.2%

5194 contributeurs / 381 851 commits

Commits

Analyzed 11 days ago. Based on code collected about 1 month ago.

	All Time	12 Month	30 Day
Commits:	381851	58802	5043
Contributors:	5194	1315	438
Files Modified:	383348	177982	14891
Lines Added:	229963211	47347912	2394363
Lines Removed	147230094	28104257	1377294

Commits per Month

Allez, on compile,on exécute les tests…

- 1 commit, 1 exécution des tests, 1 construction du binaire, 1 packaging des binaires, ça prend ????

137 heures (en 2012)

- 5842 commits en août 2017…
- (30 x 24 x 60) / 5842 ???

1 commit toutes les 7 minutes

http://oduinn.com/blog/2012/08/21/137-hours-compute-hours-every-6-minutes/

Problèmes

- Vous ne pouvez pas le faire à la main
- Vous ne pouvez pas le faire dans votre IDE préféré
  - Le code est sur un serveur, etc.
- Mais ce ne sont que des dépendances entre des modules, des classes, des fichiers, des trucs…

Maven

- Projet de la fondation Apache
  - 2003 : Maven 1.x
  - 2005 : Maven 2.x
  - 2010 : Maven 3.x
- Outil de gestion de projet par construction automatisé
  - abstractions encourageant la standardisation des projets pour des problématiques récurrentes
  - Construction, test, distribution, documentation, collaborations…

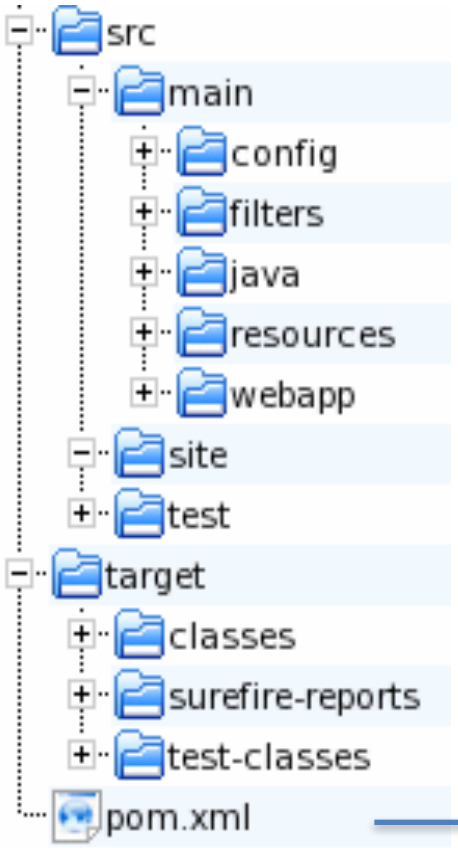
Maven

- Gestion de tâches comme des rapports, des dépendances, des configurations, des releases, des distributions, etc.
  - Approche entièrement déclarative non centrée sur les tâches à exécuter (comme ant ou make)
- Proposition de bonnes pratiques par défaut
  - Structuration du projet
  - Un seul livrable par projet Maven produit une seule sortie (Artefact maven)

CONVENTION OVER CONFIGURATION

Organisation

Structure



Fichier POM (Project Object Model)

- Description du projet
- A la racine du projet

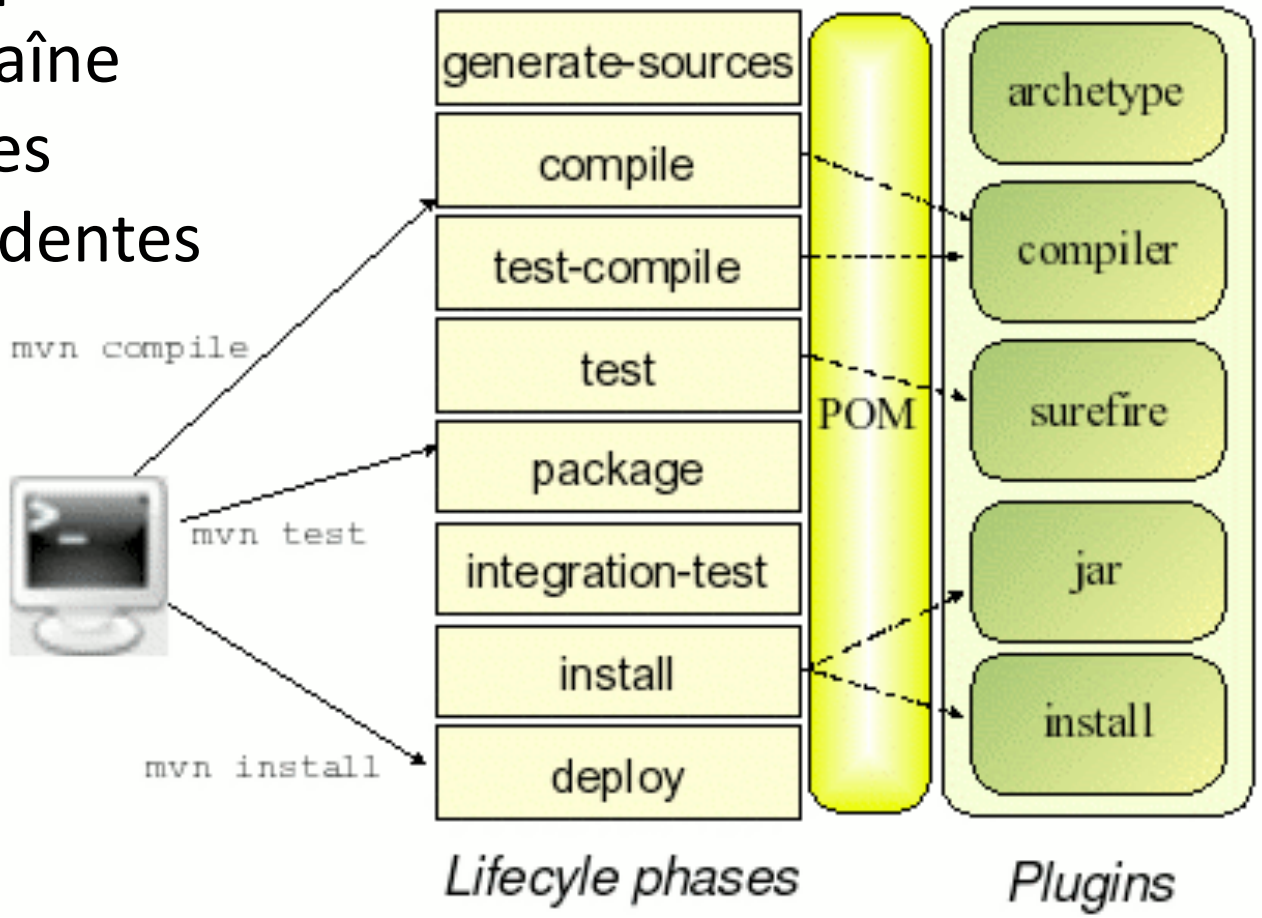
```
<project xmlns="http://maven.apache.org/POM/4.0.0" ... >
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 ...">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Maven : pom.xml

- Le fichier central de toute configuration, qui contient la majorité des informations sur le projet
  - Informations de versions
  - Gestion des configurations
  - Dépendances
  - Ressources de l'application
  - Tests
  - Membres de l'équipe, …
- Suivre les conventions et l'organisation standard
  - Réduit la taille du fichier pom.xml
  - Rend le projet plus simple à comprendre et à étendre par plugins

Cycle de vie du projet

- Appeler une phase du cycle entraîne l'exécution des phrase précédentes



Gestion des dépendances

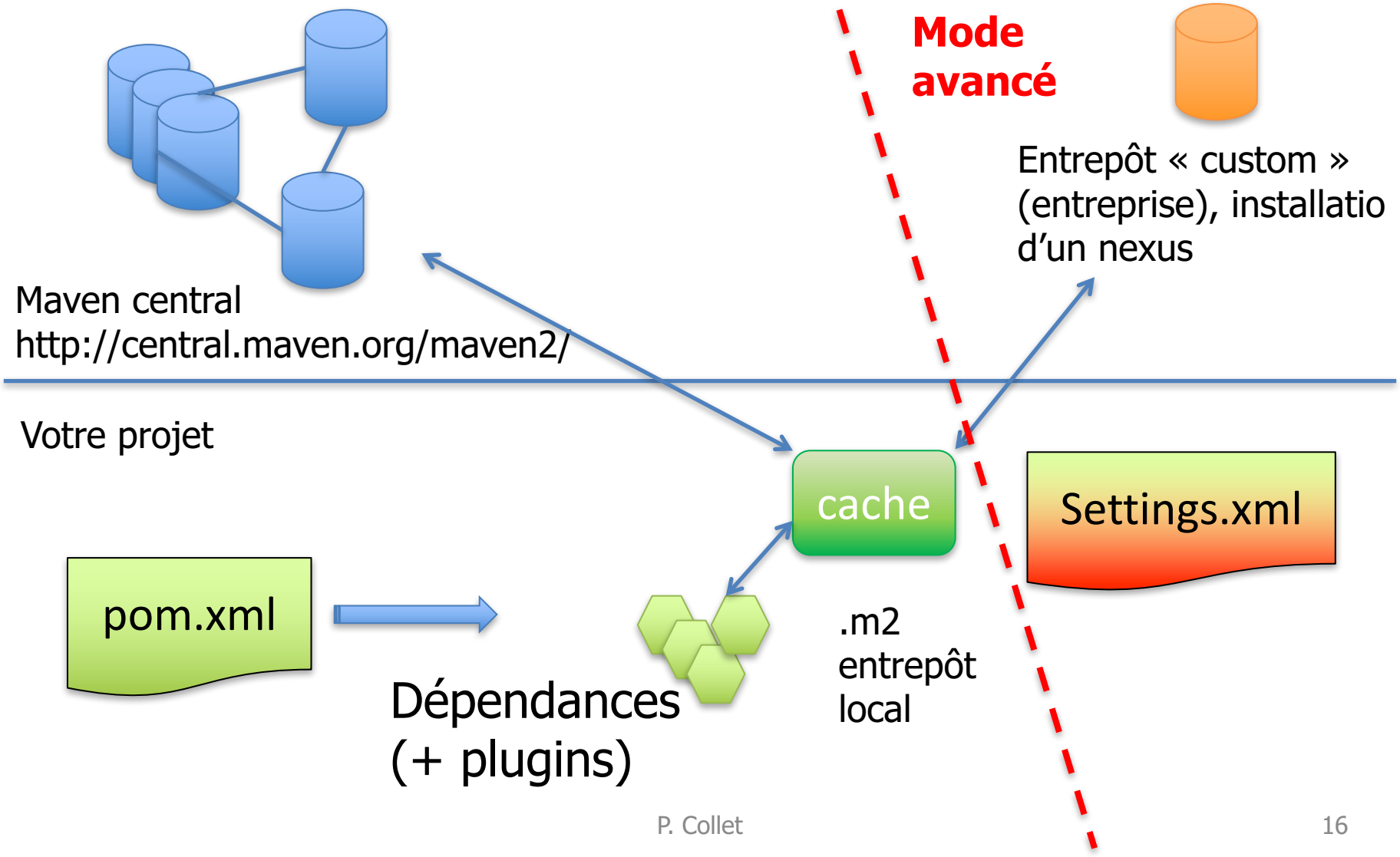
- Toutes les dépendances ne sont pas utiles tout le temps
  - Certaines uniquement pour les tests
  - D'autres sont fournies par les serveurs d'applications…
- Maven propose 4 portées de dépendance:
  - Compile: (par défaut) dispo dans toutes les phases
  - Provided: utilisé pour compiler mais pas au déploiement
  - Runtime: pas nécessaire à la compilation mais uniquement à l'exécution (driver JDBC par exemple)
  - Test : uniquement pour compiler et exécuter les tests

Gestion des dépendances (2)

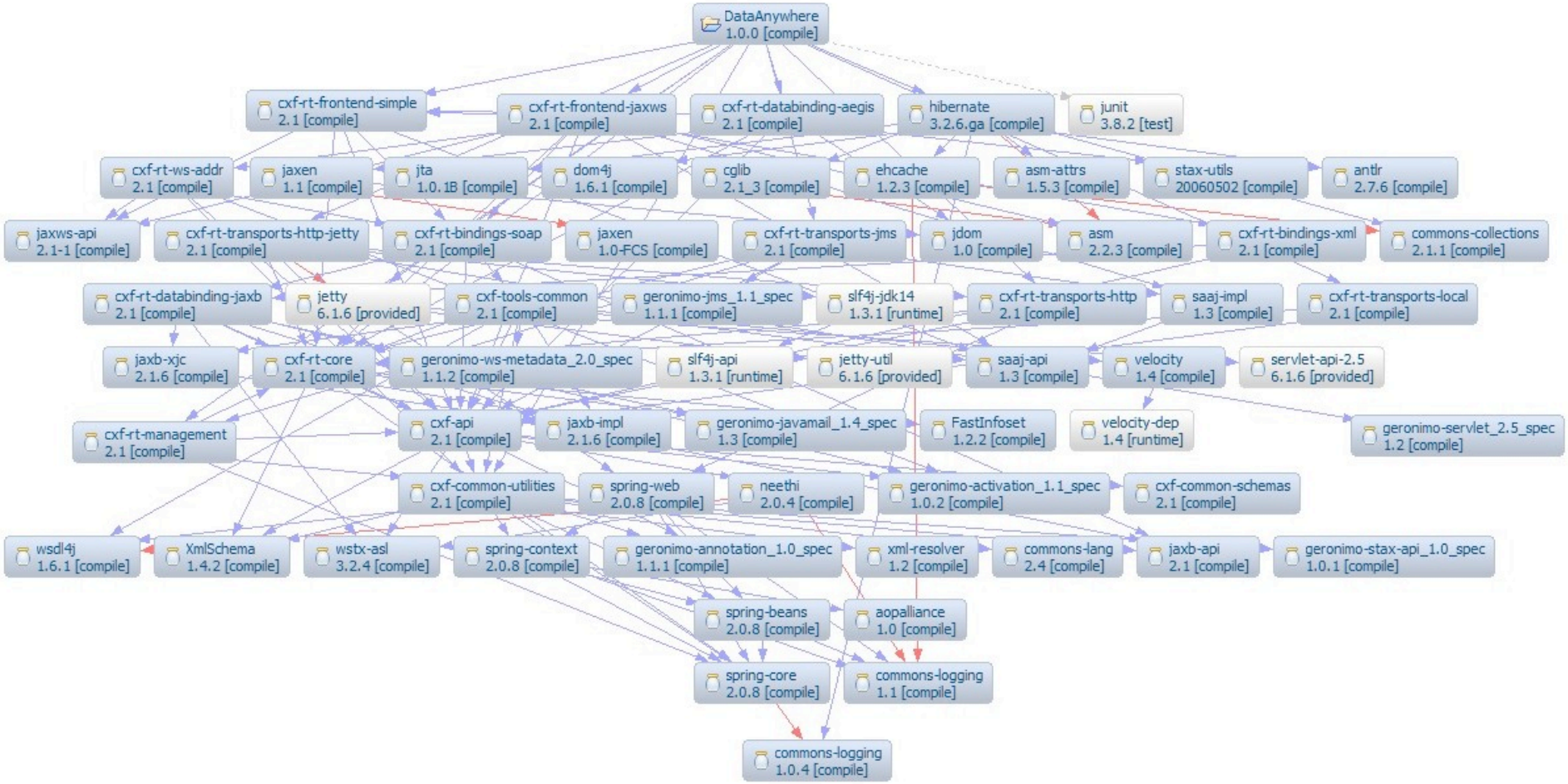
```
<project>
[...]
<dependencies>
 <dependency>
   <groupId>log4j</groupId>
   <artifactId>log4j</artifactId>
   <version>1.2.14</version>
 </dependency>
 <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>3.8.1</version>
   <scope>test</scope>
 </dependency>
</dependencies>
[...]
</project>
```

- Télécharge le jar (automatiquement !)
- Place le classpath (spécifiquement pour les tests dans l'exemple)
- Les dépendances sont maintenues à jour

Architecture des entrepôts



Gros projet… Bcp de dépendances !



Quelques commandes de base

- Nettoyage

```
$ mvn clean
```

- Exécution des tests unitaires

```
$ mvn test
```

- Packaging = compilation + test + création d'un jar

```
$ mvn package
```

- Install = package + déploiement dans l'entrepôt Maven local

```
$ mvn install
```

- **A la livraison :**

```
$ mvn clean package
```

Et pour exécuter ?

- Exec Maven Plugin
  - http://www.mojohaus.org/exec-maven-plugin/
- **exec:exec** pour exécuter des programmes (Java ou pas) dans un processus séparé
- **exec:java** pour exécuter des programmes Java dans la même machine virtuelle
- Avec plein plein d'options pour tout faire de manière indépendante de la plateforme

```
$ mvn exec:java
```

Pom.xml fourni

```
    <groupId>TeamName</groupId> <!-- CHANGE ME -->
    <artifactId>JarName</artifactId> <!-- CHANGE ME -->
    <version>0.1-SNAPSHOT</version> <!-- CHANGE ME -->

  <properties>
   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> <!--
CHANGE ME IF NEEDED, other : ISO-8859-1 -->
   <maven.compiler.source>1.8</maven.compiler.source>
   <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <build>
    <!-- USE ME TO OVERRIDE DEFAULT STRUCTURE
    <sourceDirectory>src</sourceDirectory> <!-- CHANGE ME -->
    <testSourceDirectory>test</testSourceDirectory> <!-- CHANGE ME -->
    -->
```

```
    <plugins>

      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>1.6.0</version>
        <executions>
          <execution>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>Complete.Qualified.Name.Of.The.Main.Class</mainClass> <!--
CHANGE ME -->
<!--           <arguments>
            <argument>argument1</argument>
          </arguments>
          <systemProperties>
            <systemProperty>
              <key>myproperty</key>
              <value>myvalue</value>
            </systemProperty>
          </systemProperties>
-->
        </configuration>
      </plugin>

    </plugins>
```

```
<dependencies>
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.0.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.0.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.vintage</groupId>
    <artifactId>junit-vintage-engine</artifactId>
    <version>4.12.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.platform</groupId>
    <artifactId>junit-platform-launcher</artifactId>
    <version>1.0.0</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.junit.platform</groupId>
    <artifactId>junit-platform-runner</artifactId>
    <version>1.0.0</version>
    <scope>test</scope>
</dependency>
</dependencies>
```