UNIVERSITÉ CÔTE D'AZUR

POLYTECH NICE-SOPHIA

Construction automatique avec Maven

Philippe Collet

Polytech'Nice Sophia – SI3

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

Du gros code ?

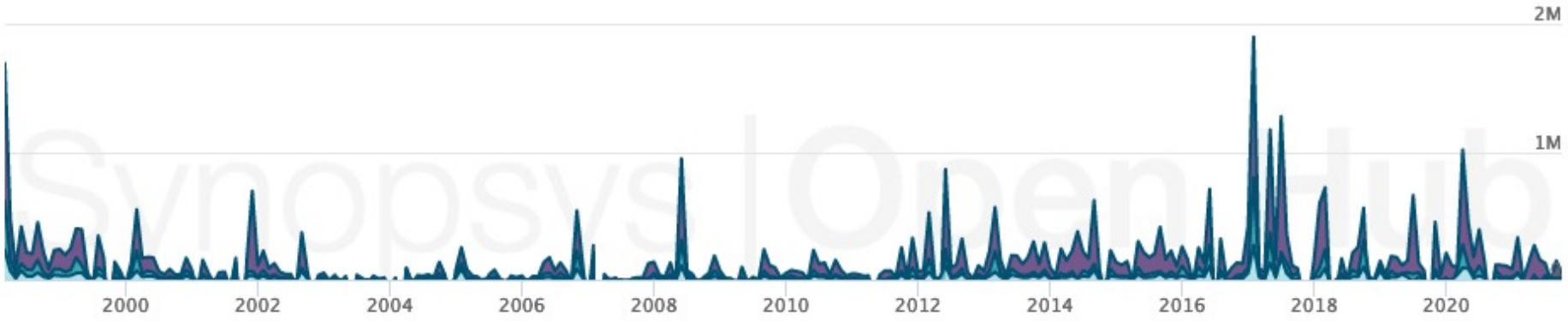
Languages

Analyzed 6 days ago. based on code collected 6 days ago.

Total Lines :	32,754,115	Code Lines :	23,899,066	Percent Code Lines :	73.0%
Number of Languages :	47	Total Comment Lines :	4,956,663	Percent Comment Lines :	15.1%
		Total Blank Lines :	3,898,386	Percent Blank Lines :	11.9%

Code, Comments and Blank Lines

Zoom 1yr 3yr 5yr 10yr **All**

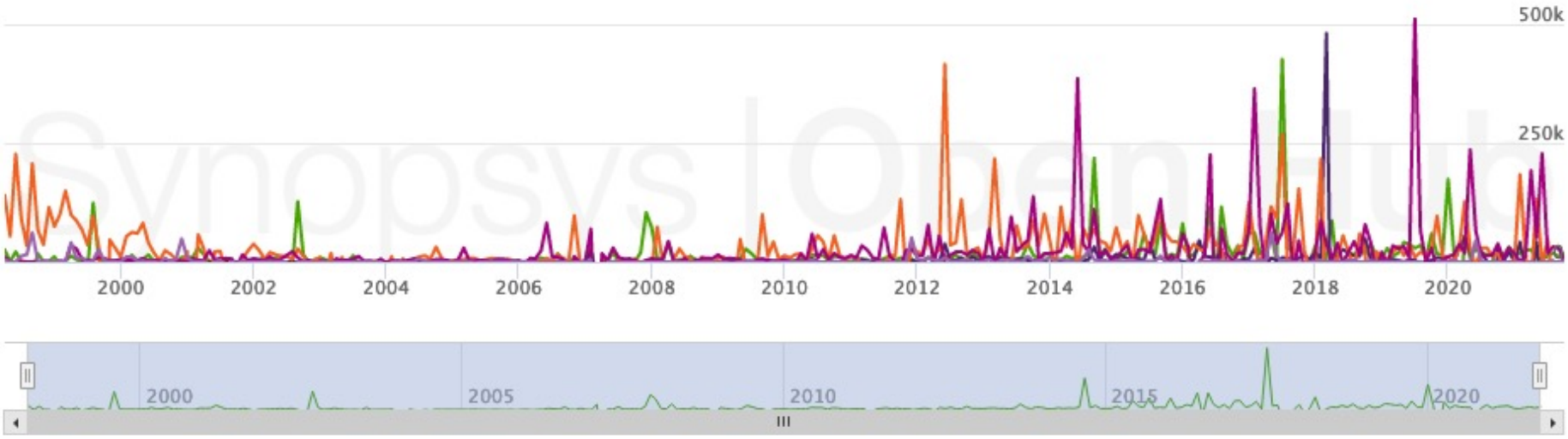


https://www.openhub.net/p/firefox

Du gros code ?

LOC by Language

Zoom 1yr 3yr 5yr 10yr **All**



Language Breakdown

Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	6,282,197	1,400,026	18.2%	1,224,755	8,906,978	27.2%
JavaScript	6,034,290	1,736,360	22.3%	1,164,450	8,935,100	27.3%
HTML	3,387,763	99,180	2.8%	360,787	3,847,730	11.7%
C	3,154,230	824,939	20.7%	469,231	4,448,400	13.6%
Rust	2,393,849	449,824	15.8%	245,211	3,088,884	9.4%
Python	898,780	254,735	22.1%	241,219	1,394,734	4.3%

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

Du gros code ?

Commits

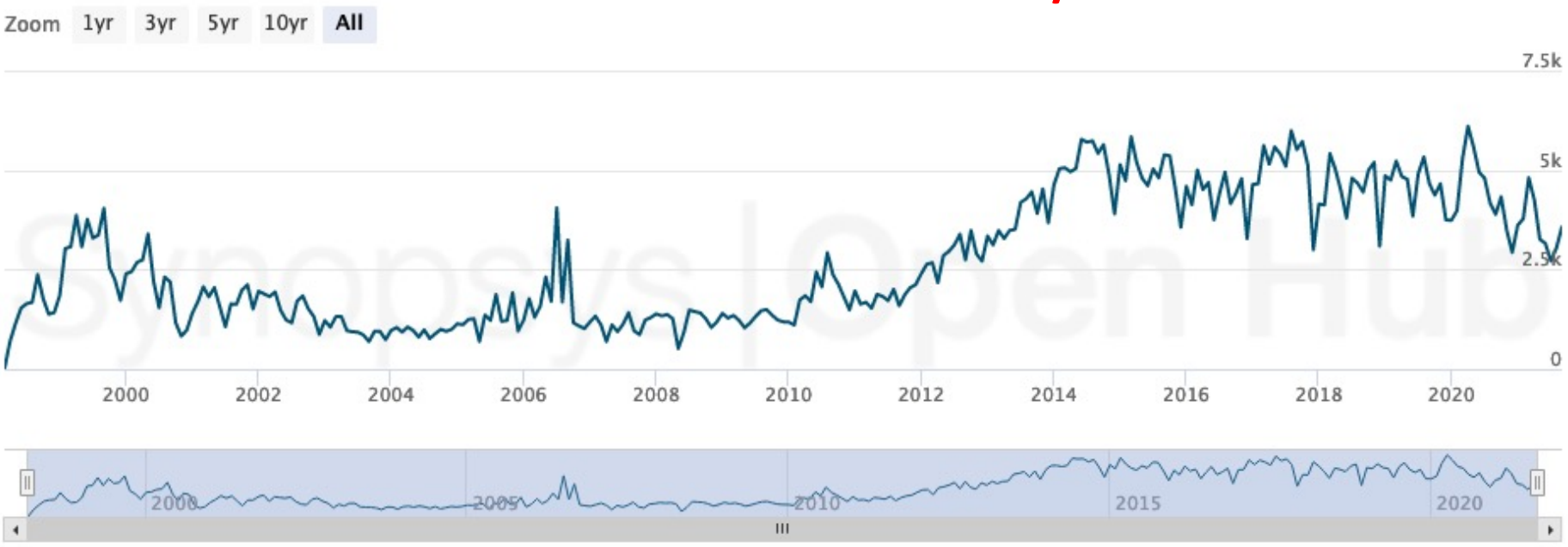
Analyzed 6 days ago. based on code collected 6 days ago.

	All Time	12 Month	30 Day
Commits:	772987	42755	3708
Contributors:	8413	1086	402
Files Modified:	711988	105257	12172
Lines Added:	346646453	13245172	1298823
Lines Removed	256407152	11567281	1151913

Aout 2017 : 5194 contributeurs / 381 851 commits

Oct. 2021 : 8413 contributeurs / 772 987 commits

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…

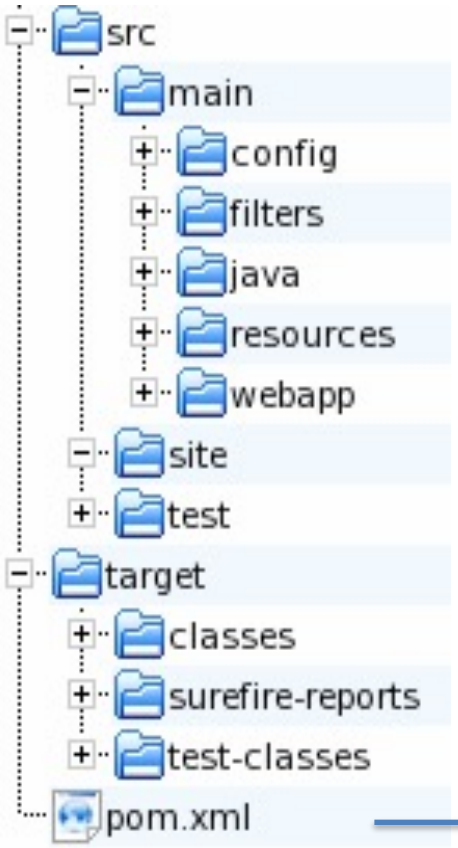
Maven

- 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

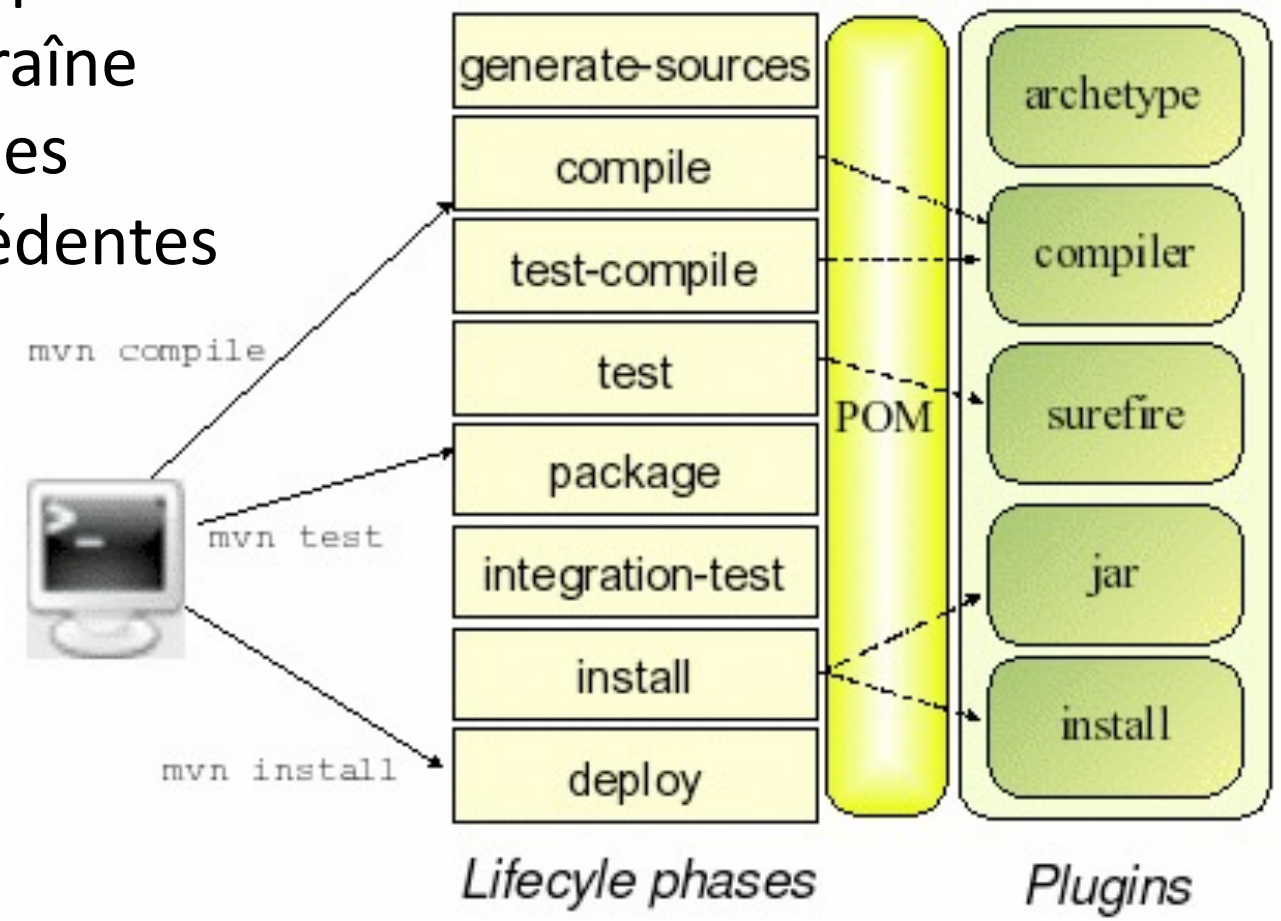
```
<project xmlns="http://maven.apache.org/POM/4.0.0" ... >
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 ...">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven 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

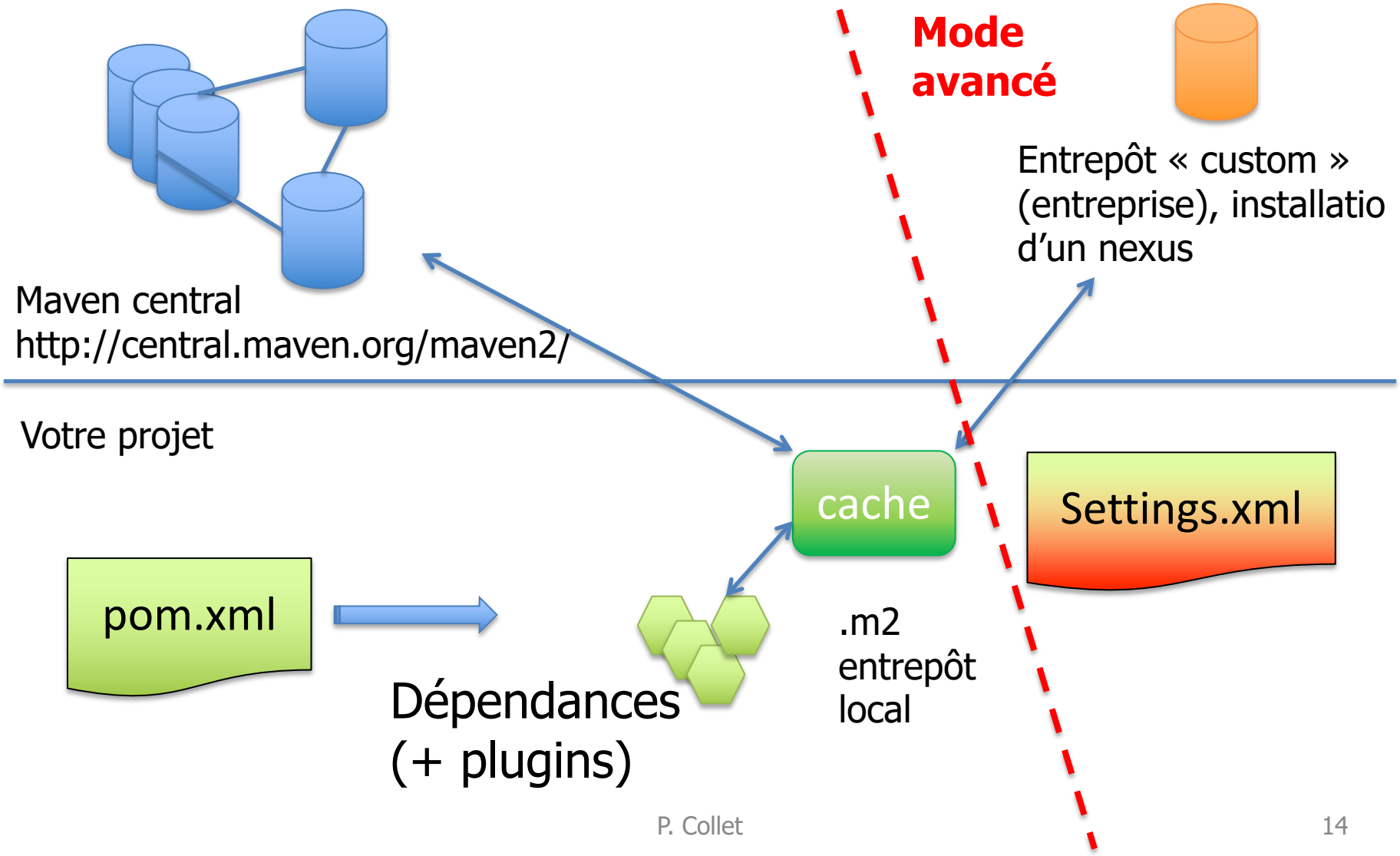
- Toutes les dépendances ne sont pas utiles tout le temps
  - Certaines uniquement pour les tests
  - D'autres sont fournies par les serveurs d'applications…
- Maven propose 4 portées de dépendance:
  - Compile: (par défaut) dispo dans toutes les phases
  - Provided: utilisé pour compiler mais pas au 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)

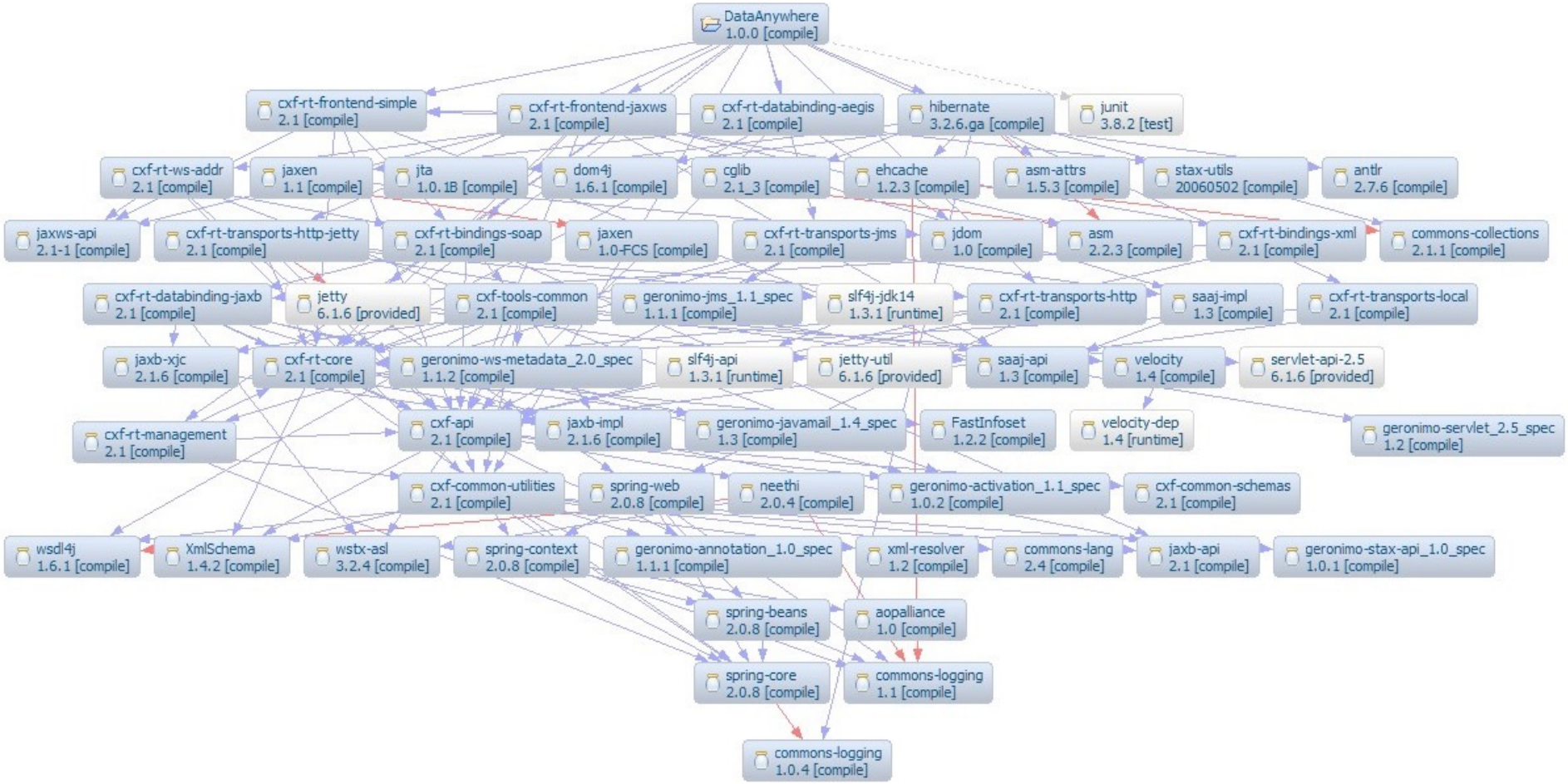
```
<project>
[...]
<dependencies>
 <dependency>
   <groupId>log4j</groupId>
   <artifactId>log4j</artifactId>
   <version>1.2.14</version>
 </dependency>
 <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>3.8.1</version>
   <scope>test</scope>
 </dependency>
</dependencies>
[...]
</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 pour le 2nd projet https://github.com/collet/ps5-template

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

    <groupId>TeamName</groupId> <!-- CHANGE ME -->
    <artifactId>JarName</artifactId> <!-- CHANGE ME -->
    <version>0.3-SNAPSHOT</version> <!-- CHANGE ME -->

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> <!-- CHANGE ME IF NEEDED, other : ISO-8859-
        <java.version>17</java.version>
        <maven.compiler.source>${java.version}</maven.compiler.source>
        <maven.compiler.target>${java.version}</maven.compiler.target>
        <junit.jupiter.version>5.8.0</junit.jupiter.version>
    </properties>

  <build>
    <plugins>
        <!-- JUnit 5 requires Surefire version 2.22.2 / compiler 3.8.1 -->
        <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.8.1</version>
        </plugin>

      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.22.2</version>
      </plugin>
```

```
      <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>3.0.0</version>
        <executions>
          <execution>
            <goals>
              <goal>java</goal>
            </goals>
          </execution>
        </executions>
        <configuration>
          <mainClass>fr.unice.polytech.startingpoint.Main</mainClass> <!-- CHANGE ME -->
<!--          <arguments>
            <argument>argument1</argument>
          </arguments>
          <systemProperties>
            <systemProperty>
              <key>myproperty</key>
              <value>myvalue</value>
            </systemProperty>
          </systemProperties>
-->
        </configuration>
      </plugin>

    </plugins>
  </build>
```

```
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>${junit.jupiter.version}</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```