Build

Christophe Demarey

Septembre 2014

1

Notions de base

Build process

« The process of converting source code files into standalone software artifact(s) that can be run on a computer, or the result of doing so »

Build process

Building software implies

- the management of the dependencies
- to ensure the portability
- automation


BUILD

Build process

Building software implies

- the management of the dependencies
- to ensure the portability
- automation
  - run with a single command line
  - the build process should be repeatable

Dependency management

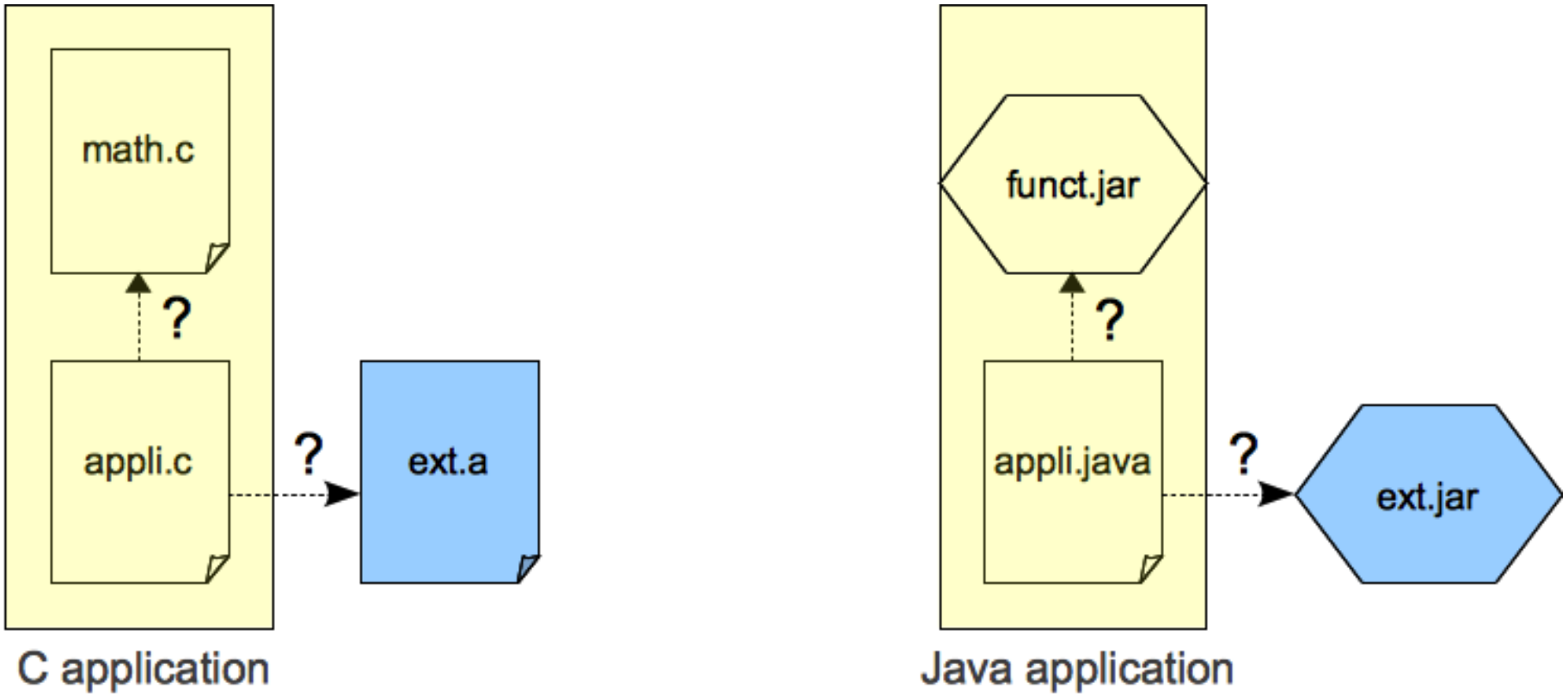
Why?

Any system needs some coupling between its parts and external ones

Dependency:

when any piece of code requires a binding with another piece of code or library

Dependency management



Dependency management

Dependency scope

- Compile
- Run time

The build process

- Expects a description of the dependencies
- Provides the way to solve them: make the links

Portability

Execute the same program on various execution environments

Environments ?

- OS for compiled languages
- Interpreter for interpretated languages
- Virtual machines (for Java, C#)
- Browser for web applications
- Hardware for operating systems

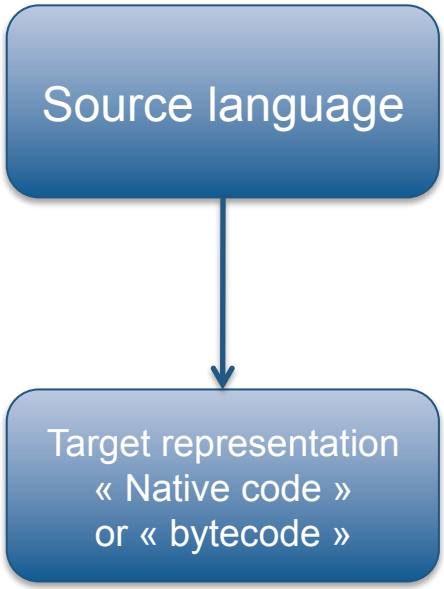
Portability

Most common problem:

compile the same code and make it work under

- Linux,
- Windows,
- and MacOS

What is part of a build?



Compilation

Tests

- Compile and run
- Generate reports

Documentation

- User, technical, source code (ex: javadoc)

Packaging

- Source and binaries
- Install and setup packages (rpm, .deb, installer...)

2

Maven

What is Maven?

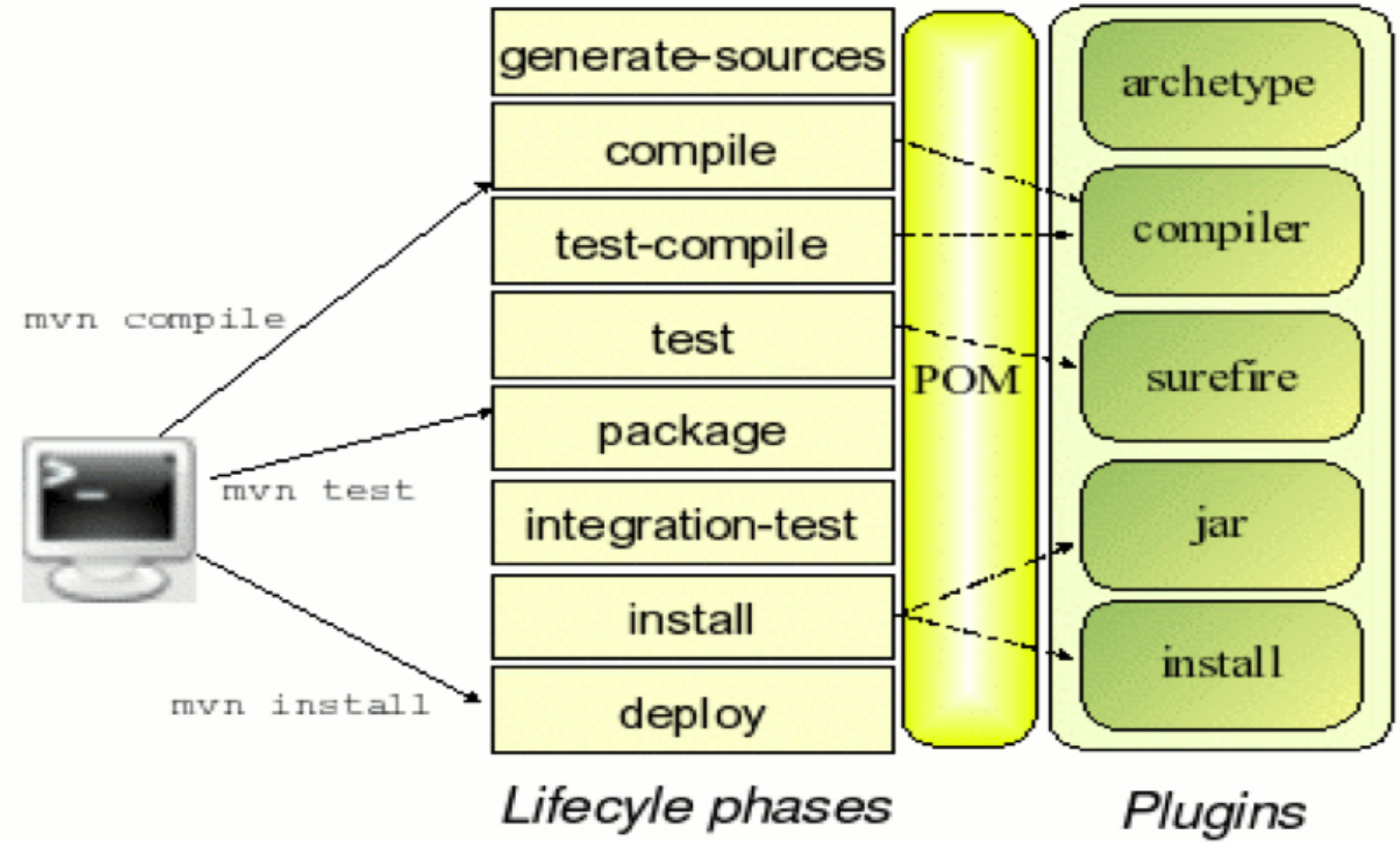
A build automation tool

Maven describes

1. how software is built,
2. its dependencies.

Maven uses **conventions** for the build procedure, and only exceptions need to be written down

Maven
> Phases and plugins

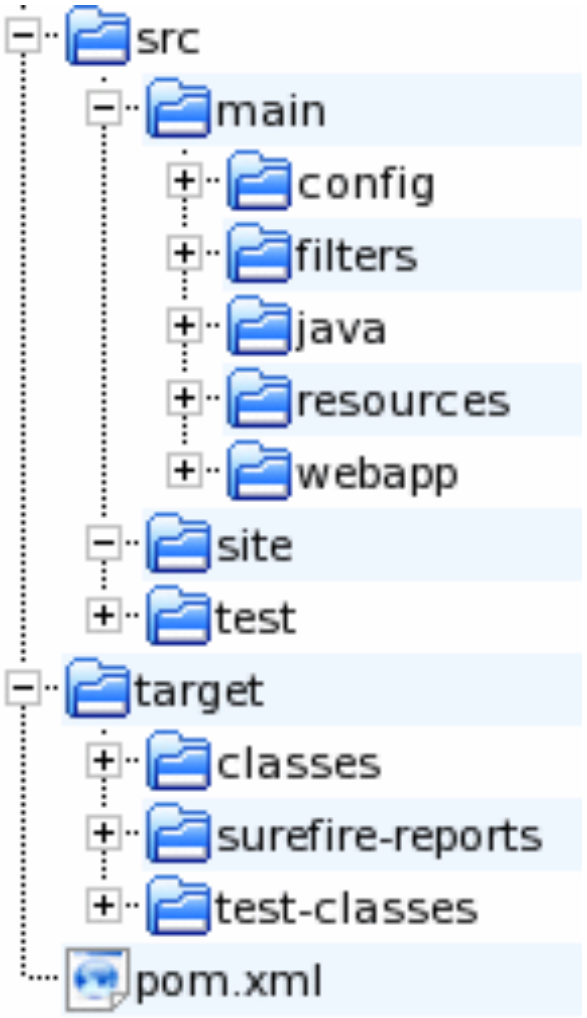


Maven
> Standard directory layout

- Great/very simple
- Lack of flexibility

  for unusual build structure

```
mvn archetype:create \
  -DarchetypeGroupId=org.apache.maven.archetypes \
  -DgroupId=com.mycompany.app \
  -DartifactId=my-app
```



Maven
> Dependencies

Dynamically downloads Java libraries and plugins from

- the central repository : http://central.sonatype.org/
- other repositories (possible to define your own)

Local cache of downloaded artefacts

Transitive dependencies management

```
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-io</artifactId>
    <version>1.3.2</version>
</dependency>
```

Maven
> pom file

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.myorg</groupId>
  <artifactId>myapp</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Full application name</name>
  <url>http://myapp.myorg.org</url>
  ...
<dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Maven

Web site : http://maven.apache.org/

Licence : Apache License Version 2.0

Latest release : 3.2.3

Language : Java

Links :

- An introduction to Maven,

  http://www.javaworld.com/javaworld/jw-12-2005/jw-1205-maven.html