

An Intro to Maven

{ Anuj Sehgal (s.anuj@jacobs-university.de)

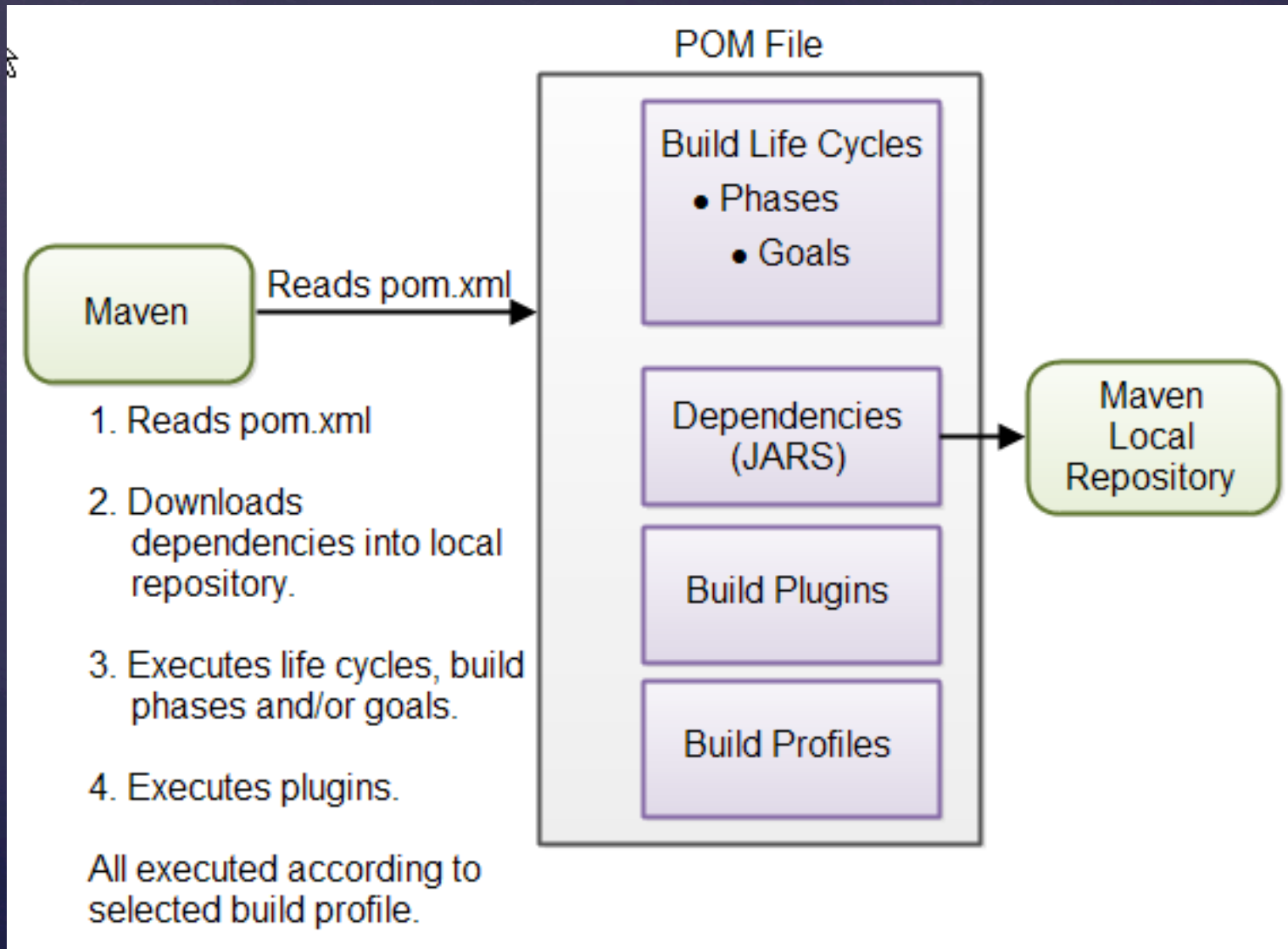
What is Maven?

- ⌘ Project management tool.
- ⌘ Provides a build lifecycle framework.
- ⌘ Automation of the build, test, deployment infrastructure.

- ⌘ Maven uses *Convention* not *Configuration*.
 - ⌘ Specify what to build.
 - ⌘ Define dependencies, tasks.
 - ⌘ No need to configure all these steps.

- ⌘ Can manage many aspects of the project lifecycle.
 - ⌘ Builds, documentation, reporting, dependencies, releases, distribution, mailing lists, etc.

What is Maven?



Plugins

- ⌘ Plugins add new behavior to the project's build.
 - ⌘ Can be retrieved from remote repositories.
- ⌘ Surefire is responsible for unit tests.
 - ⌘ JUnit is used here.
 - ⌘ TestNG support was added as well.
- ⌘ Plugins for many things.
 - ⌘ Compiling code.
 - ⌘ Generating reports.
 - ⌘ Deploying on a server
- ⌘ Declare plugins in the POM file.

Project Object Model (POM)

- ⌘ *Some* configuration information.
 - ⌘ Dependencies, plugins, goals, build profiles, project version, developers, info, etc.
- ⌘ An XML file describes the important information about the project.

```
<project>  
  <modelVersion>4.0.0.</modelVersion>  
  <groupId>de.jacobs.university.cnds</groupId>  
  <artifactId>my-project</artifactId>  
  <version>1.0</version>  
</project>
```

Why a project model?

- ⌘ Dependency management via coordinates.
 - ⌘ Group ID, Artifact ID and Version.
- ⌘ Remote repositories.
 - ⌘ Specify remote locations to search for dependencies.
- ⌘ Plugins
 - ⌘ Contain logic to work on data/configuration in POM file.
- ⌘ Portability and integration.
 - ⌘ Different IDEs have standard way of describing projects now.
- ⌘ Search and filter project artifacts.
 - ⌘ Index and search projects, and create repositories.

The build process

- ⌘ Split into Cycles -> Phases -> Goals.
- ⌘ A build life cycle is a collection of Phases, each with a sequence of goals.
- ⌘ Start build by passing name of phase or goal.
 - ⌘ All preceding phases/goals executed.



Life Cycles, Phases, Goals

- ↳ Three built-in life cycles

- ↳ default, clean, site

- ↳ Build phases

- ↳ validate, compile, test, package, install, deploy

```
mvn <phase>
```


Create project

```
mvn archetype:generate
```

- ⌘ *Archetype* specifies that basic template to base things on.
- ⌘ The goal is *generate*.
- ⌘ What is the plugin? Which is the goal?

Package project

```
mvn package
```

⌘ Does it run?

⌘ What is it missing?

Create project

```
<build>
  ...
  <plugins>
    ...
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>>true</addClassPath>
            <mainClass>de.jacobs...</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
    ...
  </plugins>
</build>
```