Maven Overview

You will learn how to …

You will learn how to …

- Create Maven projects with Eclipse

You will learn how to …

- Create Maven projects with Eclipse
- Add dependencies to Maven pom.xml file

You will learn how to …

- Create Maven projects with Eclipse
- Add dependencies to Maven pom.xml file
- Build and run Maven projects

Practical Results

Practical Results

- Cover the most common Maven tasks that you will need on daily projects

Practical Results

- Cover the most common Maven tasks that you will need on daily projects

- Not an A to Z reference … for that you can see Maven Reference Manual

Practical Results

- Cover the most common Maven tasks that you will need on daily projects
- Not an A to Z reference … for that you can see Maven Reference Manual
- But don't worry, I will give you links to

Practical Results

- Cover the most common Maven tasks that you will need on daily projects
- Not an A to Z reference … for that you can see Maven Reference Manual
- But don't worry, I will give you links to
  - Maven Reference Manual

Free Maven Resources

Free Maven Resources

- **Maven Reference Manual**
  - www.luv2code.com/mavenreferencemanual

What is Maven?

What is Maven?

- Maven is a Project Management tool

What is Maven?

- Maven is a Project Management tool
- Most popular use of Maven is for build management and dependencies

What Problems Does Maven Solve?

What Problems Does Maven Solve?

- When building your Java project, you may need additional JAR files
  - For example: Spring, Hibernate, Commons Logging, JSON etc…

What Problems Does Maven Solve?

- When building your Java project, you may need additional JAR files
  - For example: Spring, Hibernate, Commons Logging, JSON etc…

- One approach is to download the JAR files from each project web site

What Problems Does Maven Solve?

- When building your Java project, you may need additional JAR files
  - For example: Spring, Hibernate, Commons Logging, JSON etc…
- One approach is to download the JAR files from each project web site
- Manually add the JAR files to your build path / classpath

My Project without Maven

My Project without Maven

My Super Cool App

My Project without Maven

My Super Cool App

My Project without Maven

My Super Cool App

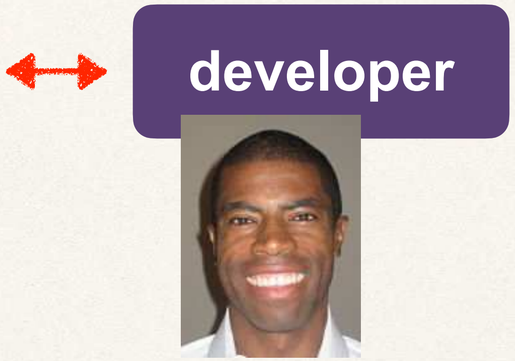
Spring
Hibernate
Commons Logging
JSON
…

developer

My Project without Maven

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
…

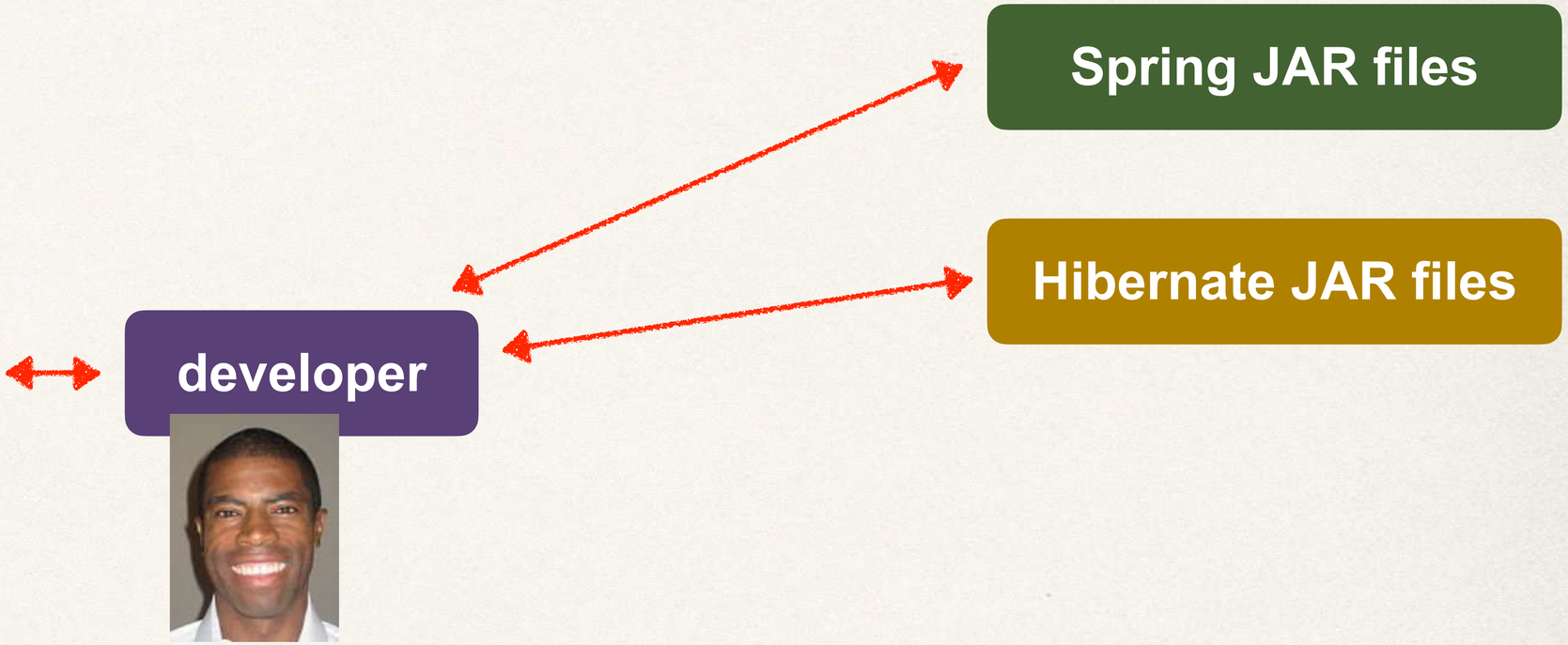


Spring JAR files

My Project without Maven

My Super Cool App

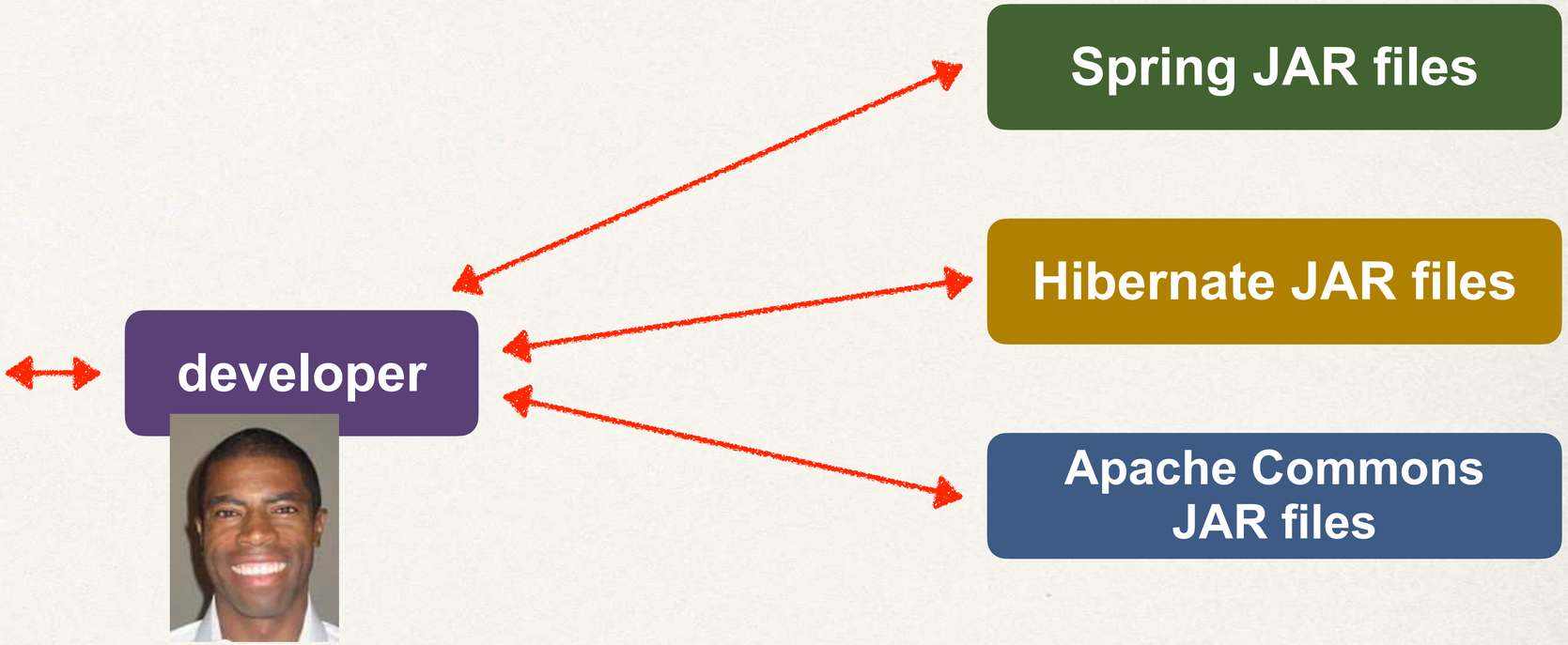
Spring
Hibernate
Commons Logging
JSON
…



My Project without Maven

My Super Cool App

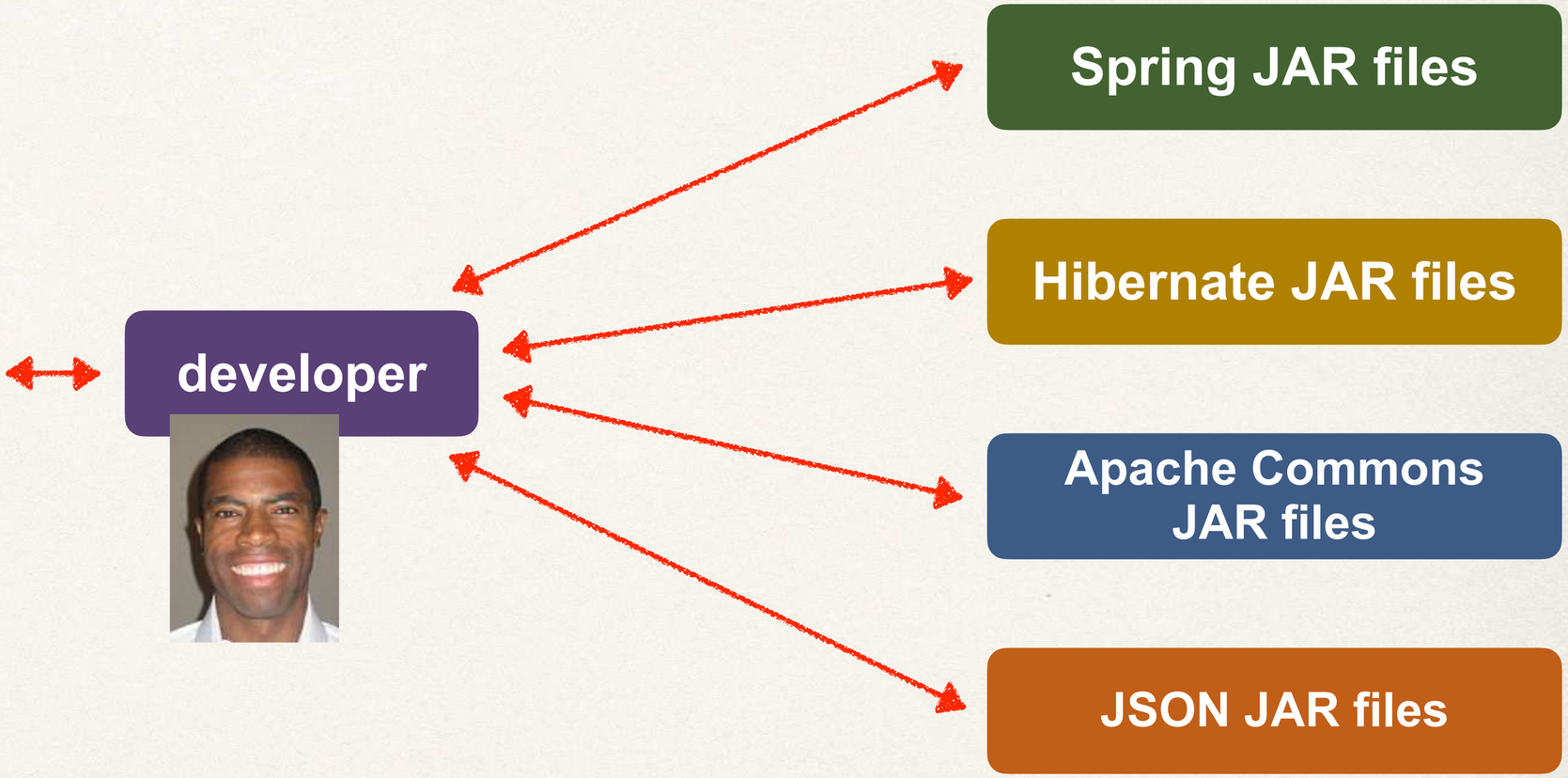
Spring
Hibernate
Commons Logging
JSON
…



My Project without Maven

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
…



Maven Solution

Maven Solution

- Tell Maven the projects you are working with (dependencies)
  - Spring, Hibernate etc ….

Maven Solution

- Tell Maven the projects you are working with (dependencies)
  - Spring, Hibernate etc ….
- Maven will go out and download the JAR files for those projects for you

Maven Solution

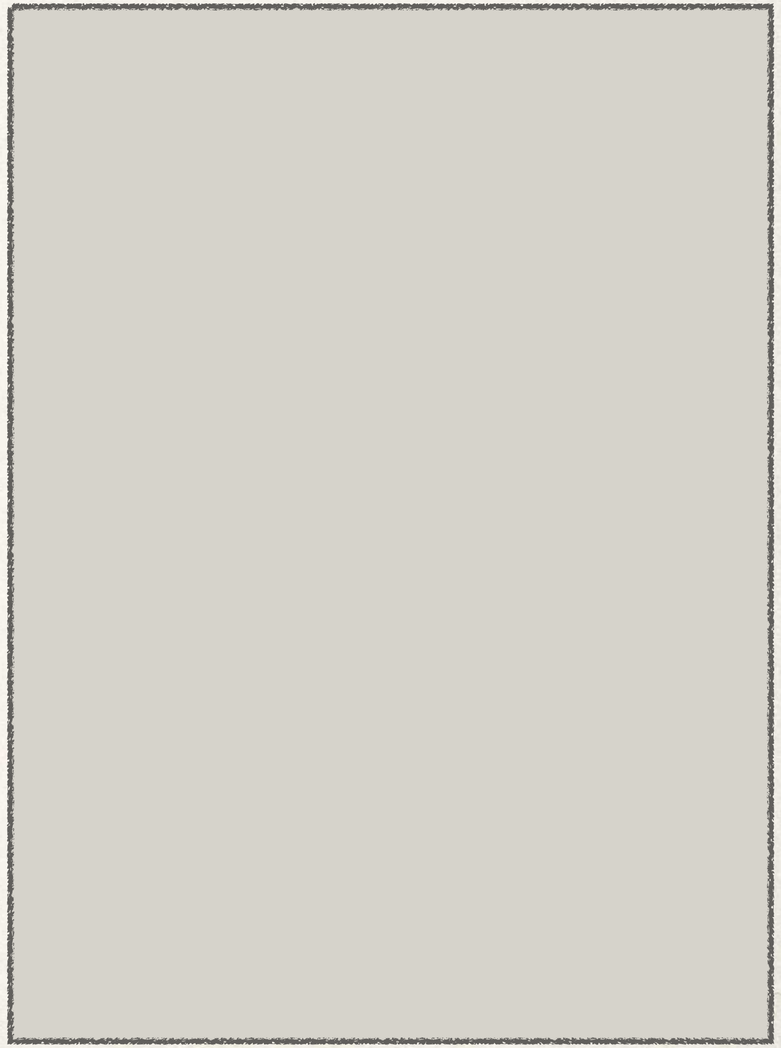
- Tell Maven the projects you are working with (dependencies)
  - Spring, Hibernate etc ….
- Maven will go out and download the JAR files for those projects for you
- And Maven will make those JAR files available during compile/run

Maven Solution

- Tell Maven the projects you are working with (dependencies)
  - Spring, Hibernate etc ….
- Maven will go out and download the JAR files for those projects for you
- And Maven will make those JAR files available during compile/run
- Think of Maven as your friendly helper / personal shopper :-)

My Project with Maven

Maven Central Repository (remote)

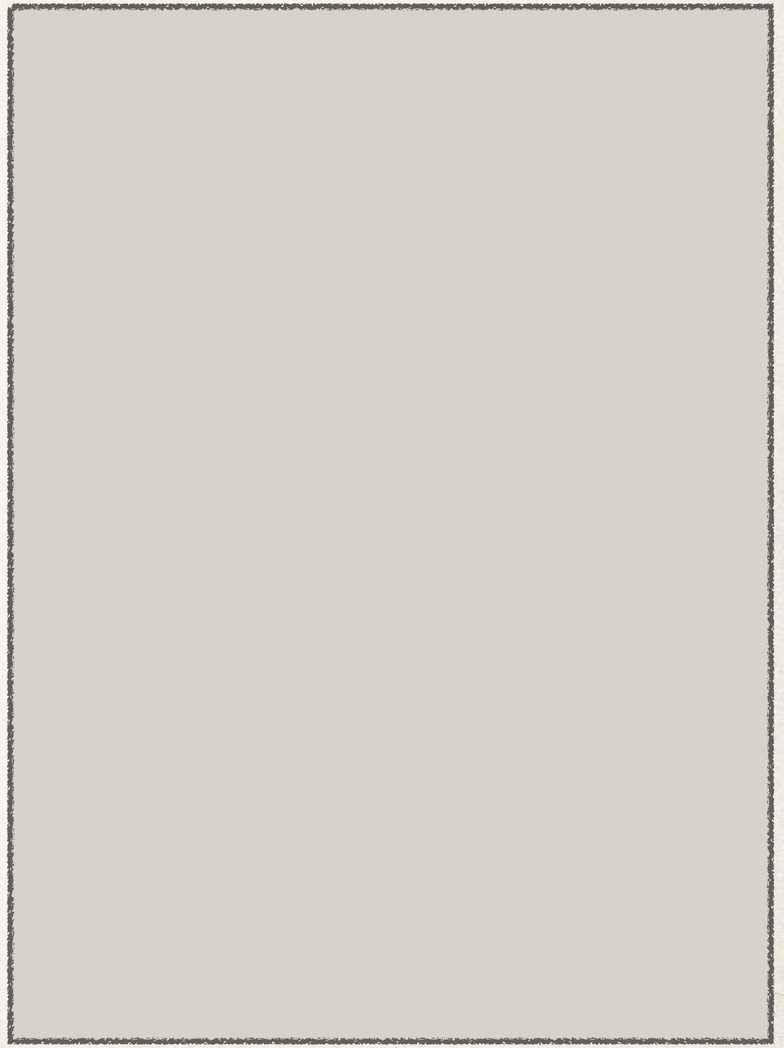


My Project with Maven

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
…

Maven Central Repository (remote)



My Project with Maven

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven Central Repository (remote)

My Project with Maven

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven

Maven Central Repository (remote)

My Project with Maven

Maven Central Repository (remote)

Spring JAR files

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven

My Project with Maven

Maven Central Repository (remote)

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven

Spring JAR files

Hibernate JAR files

My Project with Maven

Maven Central Repository (remote)

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven

Spring JAR files

Hibernate JAR files

Apache Commons
JAR files

My Project with Maven

Maven Central Repository (remote)

My Super Cool App

Spring
Hibernate
Commons Logging
JSON
...

developer

Maven

Spring JAR files

Hibernate JAR files

Apache Commons JAR files

JSON JAR files

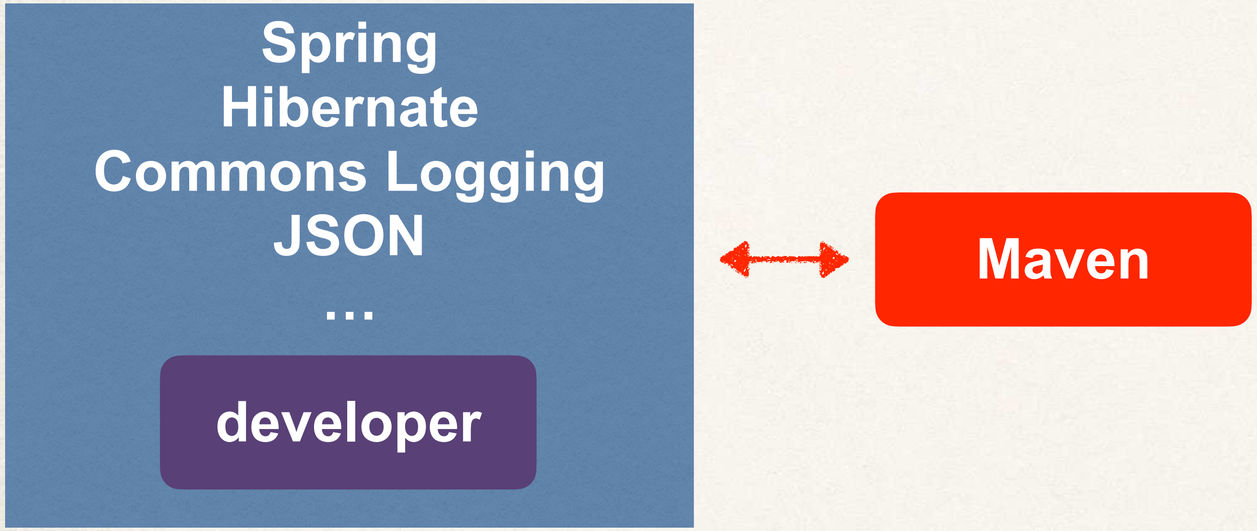
Maven - How It Works

Maven - How It Works

Project Config file

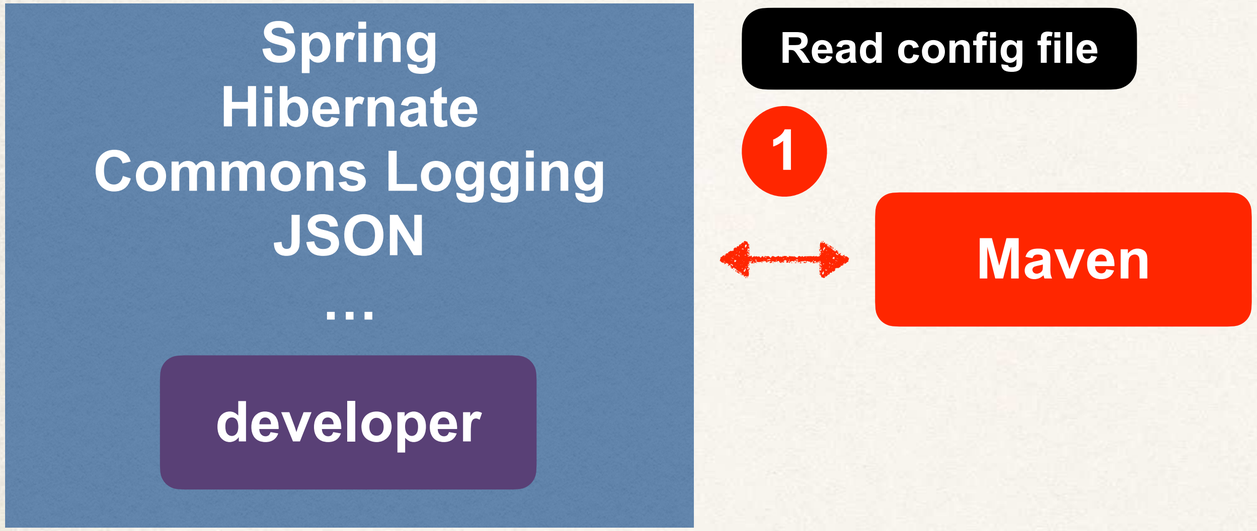
Maven - How It Works

Project Config file



Maven - How It Works

Project Config file



Maven - How It Works

Maven Local Repository

(your computer)

Project Config file

Spring
Hibernate
Commons Logging
JSON
...

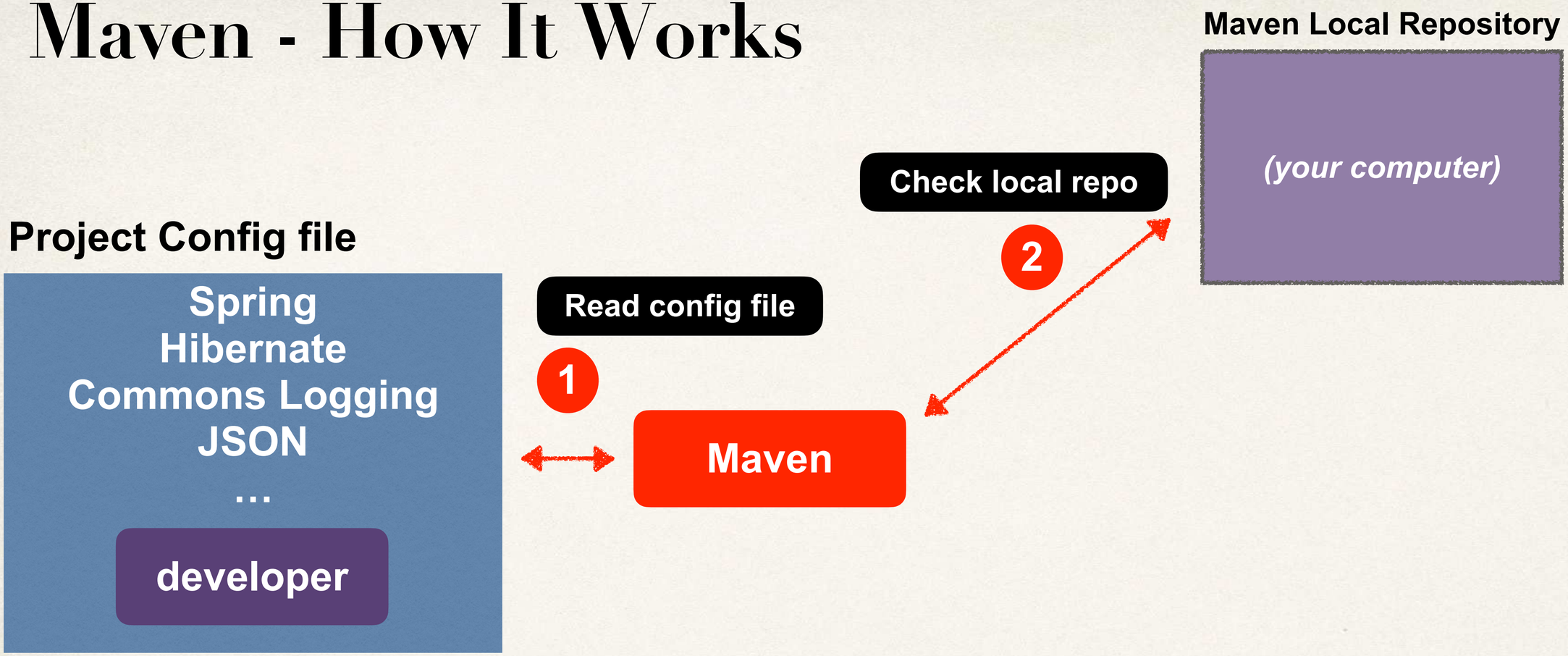
developer

Read config file

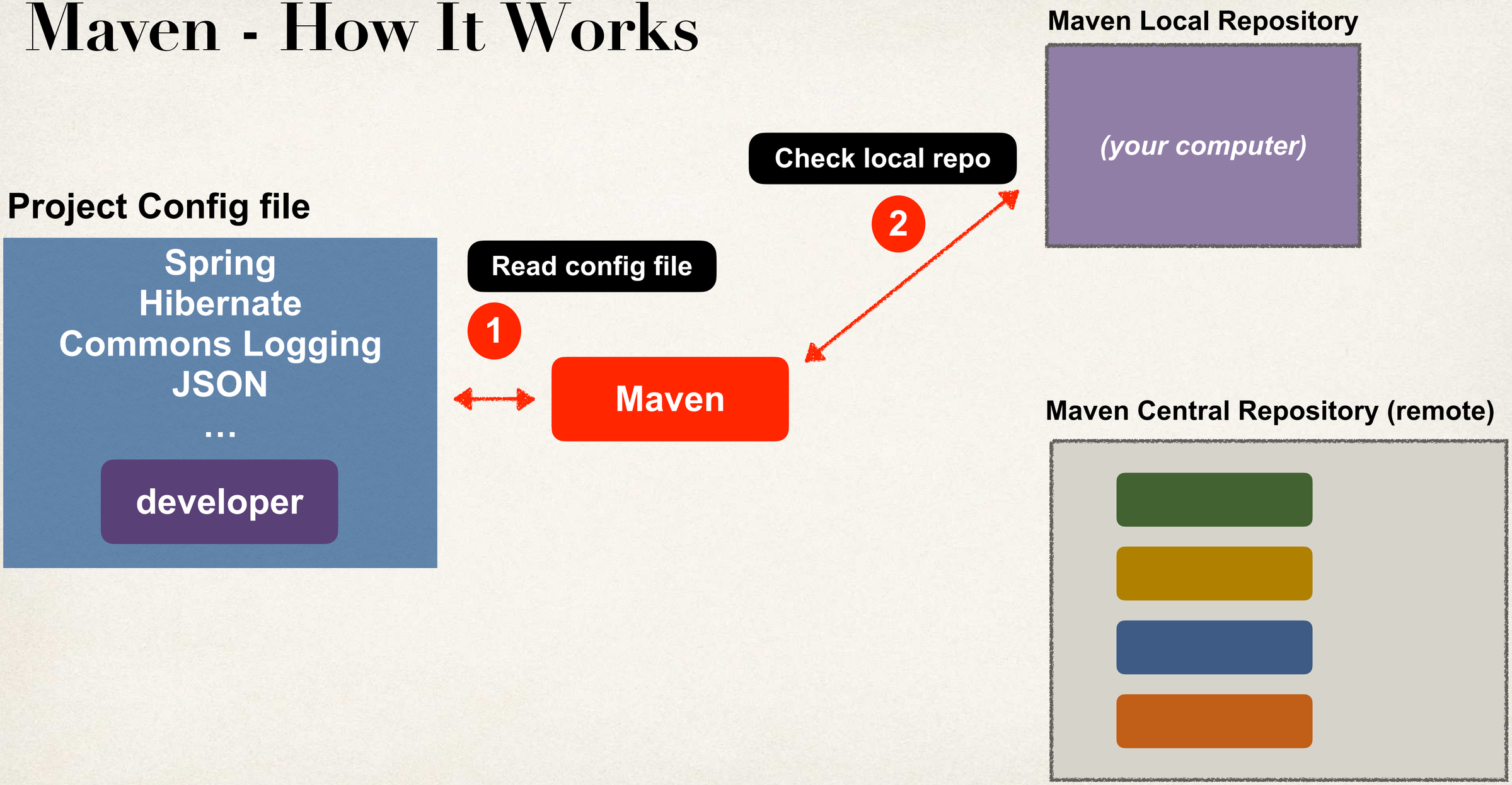
1

Maven

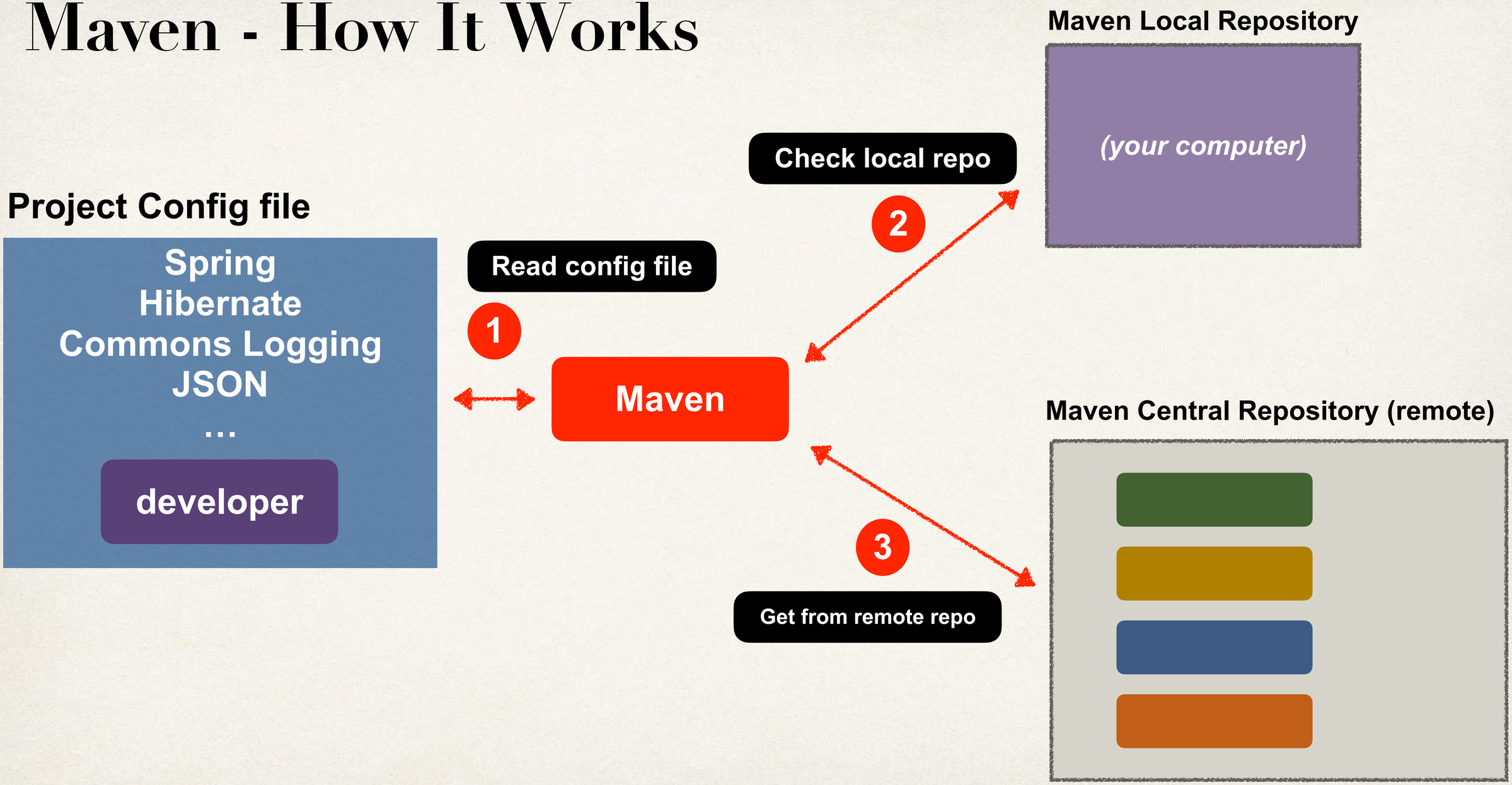
Maven - How It Works



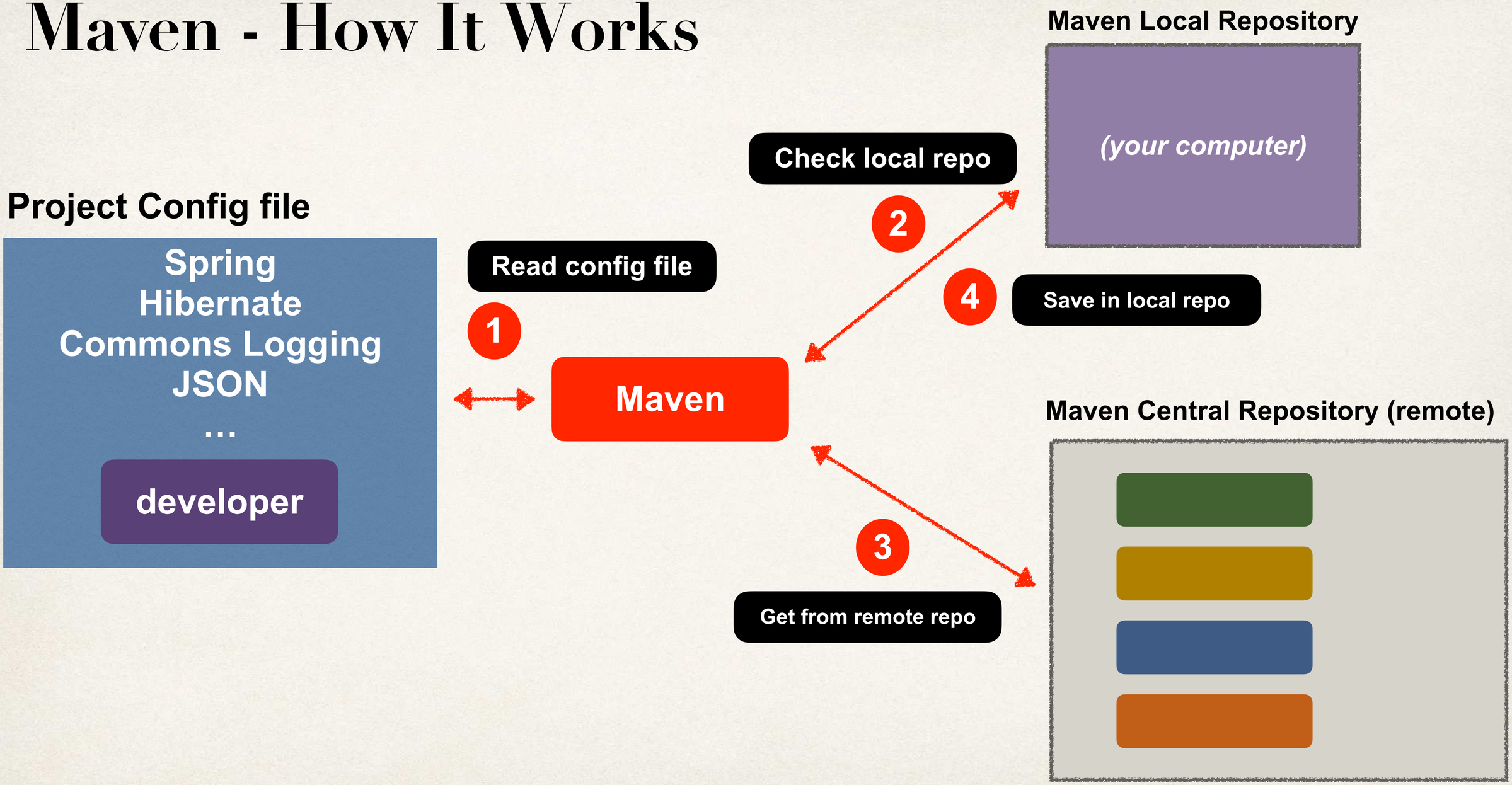
Maven - How It Works



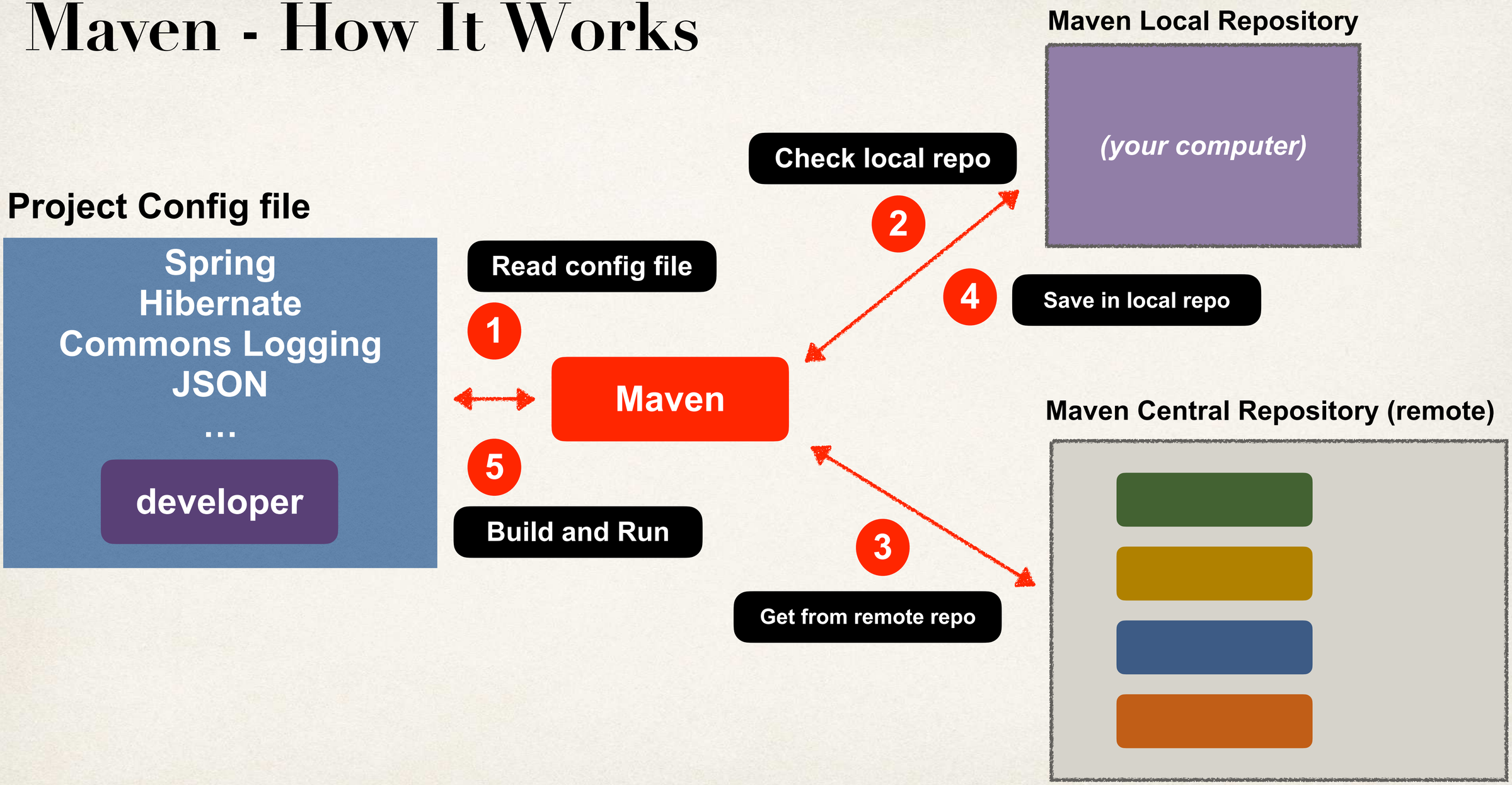
Maven - How It Works



Maven - How It Works



Maven - How It Works



Handling JAR Dependencies

Handling JAR Dependencies

- When Maven retrieves a project dependency
  - It will also download supporting dependencies
  - For example: Spring depends on commons-logging …

Handling JAR Dependencies

- When Maven retrieves a project dependency
  - It will also download supporting dependencies
  - For example: Spring depends on commons-logging …
- Maven will handle this for us automagically

Building and Running

Building and Running

- When you build and run your app …

Building and Running

- When you build and run your app …
- Maven will handle class / build path for you

Building and Running

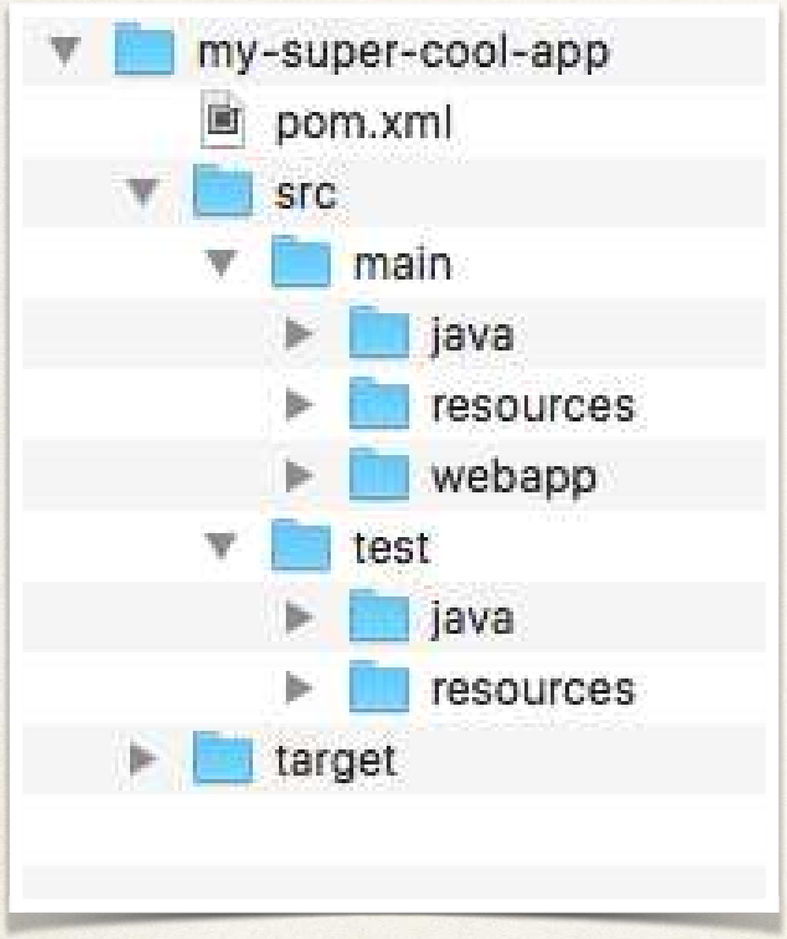
- When you build and run your app …
- Maven will handle class / build path for you
- Based on config file, Maven will add JAR files accordingly

Standard Directory Structure

- Normally when you join a new project
  - Each development team dreams up their own directory structure
  - Not ideal for new comers and not standardized

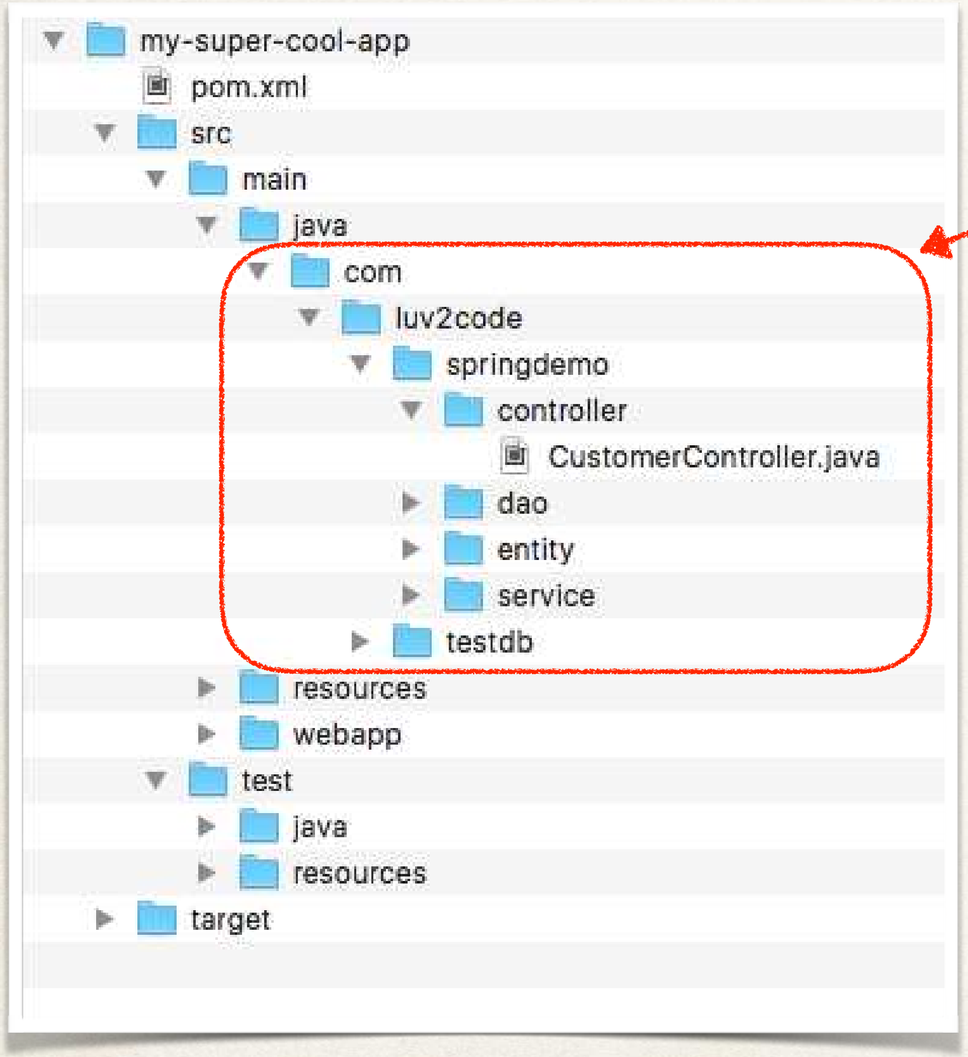
- Maven solves this problem by providing a standard directory structure

Standard Directory Structure



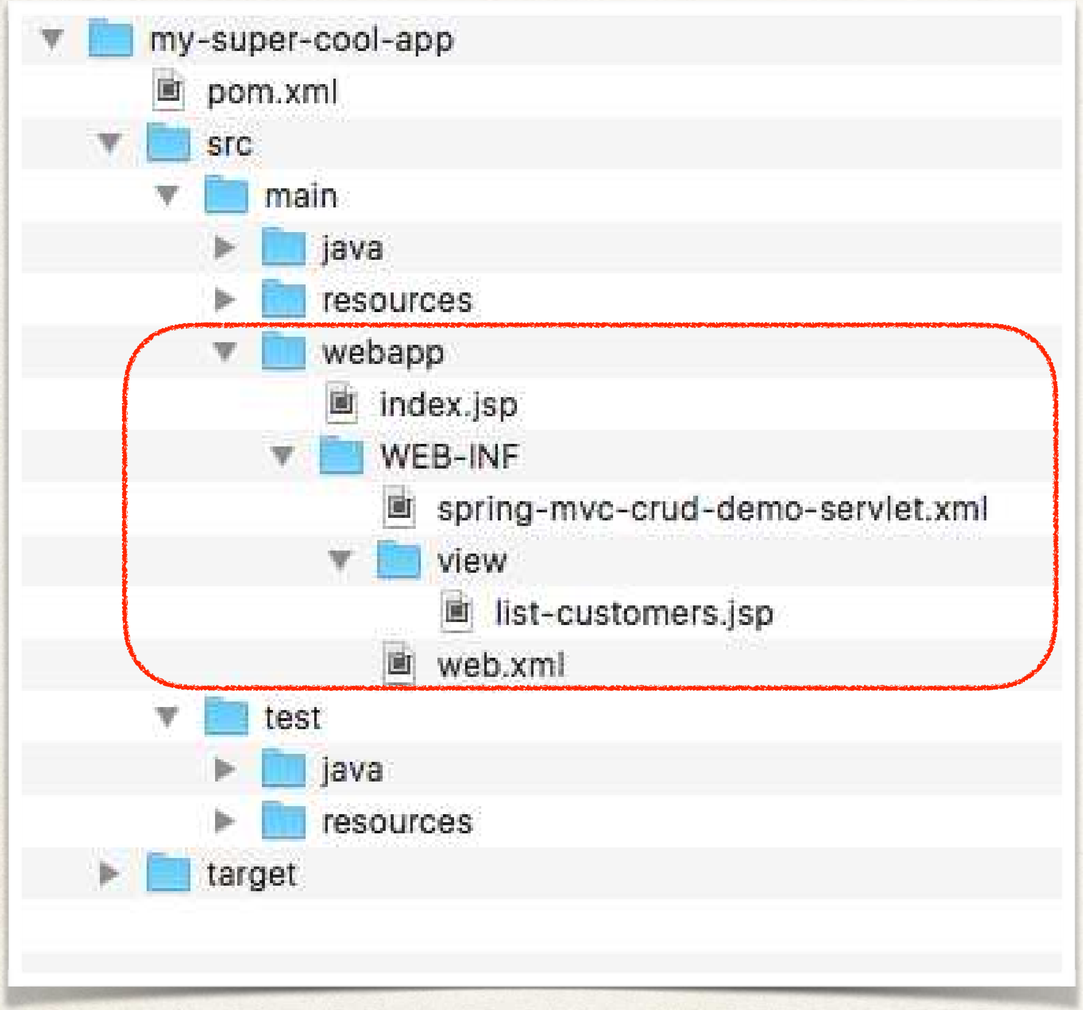
Directory	Description
src/main/java	Your Java source code
src/main/resources	Properties / config files used by your app
src/main/webapp	JSP files and web config files other web assets (images, css, js, etc)
src/test	Unit testing code and properties
target	Destination directory for compiled code. Automatically created by Maven

Standard Directory Structure



Place your Java source code here

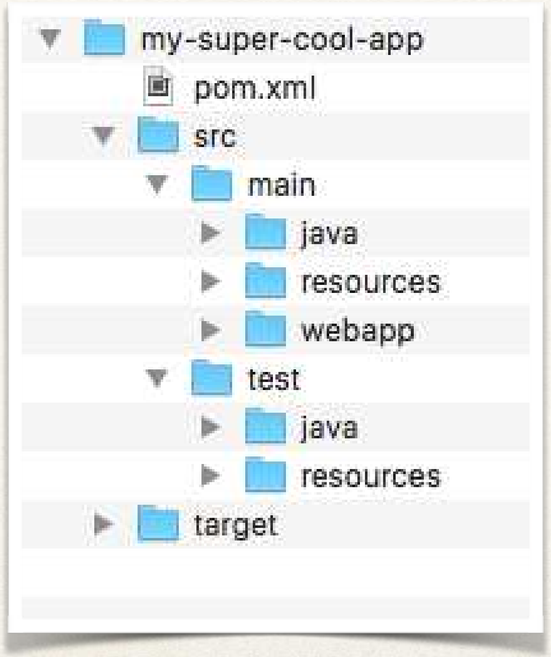
Standard Directory Structure



Place your Web assets here

Standard Directory Structure Benefits

- For new developers joining a project
  - They can easily find code, properties files, unit tests, web files etc …



Standard Directory Structure Benefits

- Most major IDEs have built-in support for Maven
  - Eclipse, IntelliJ, NetBeans etc
  - IDEs can easily read/import Maven projects
- Maven projects are portable
  - Developers can easily share projects between IDEs
  - No need to fight about which IDE is the best LOL!

Advantages of Maven

- Dependency Management
  - Maven will find JAR files for you
  - No more missing JARs
- Building and Running your Project
  - No more build path / classpath issues
- Standard directory structure

My Personal Best Maven Benefit(s)

- Once you learn Maven, you can join a new project and be productive

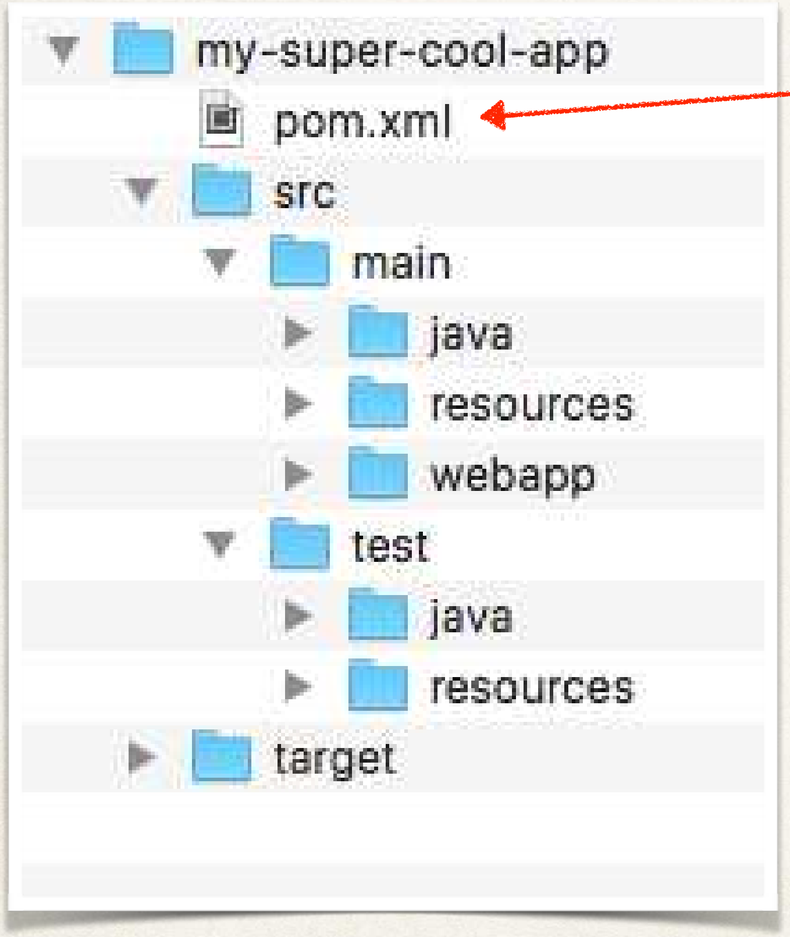
- You can build and run a project with minimal local configuration

Maven Key Concepts

Maven Key Concepts

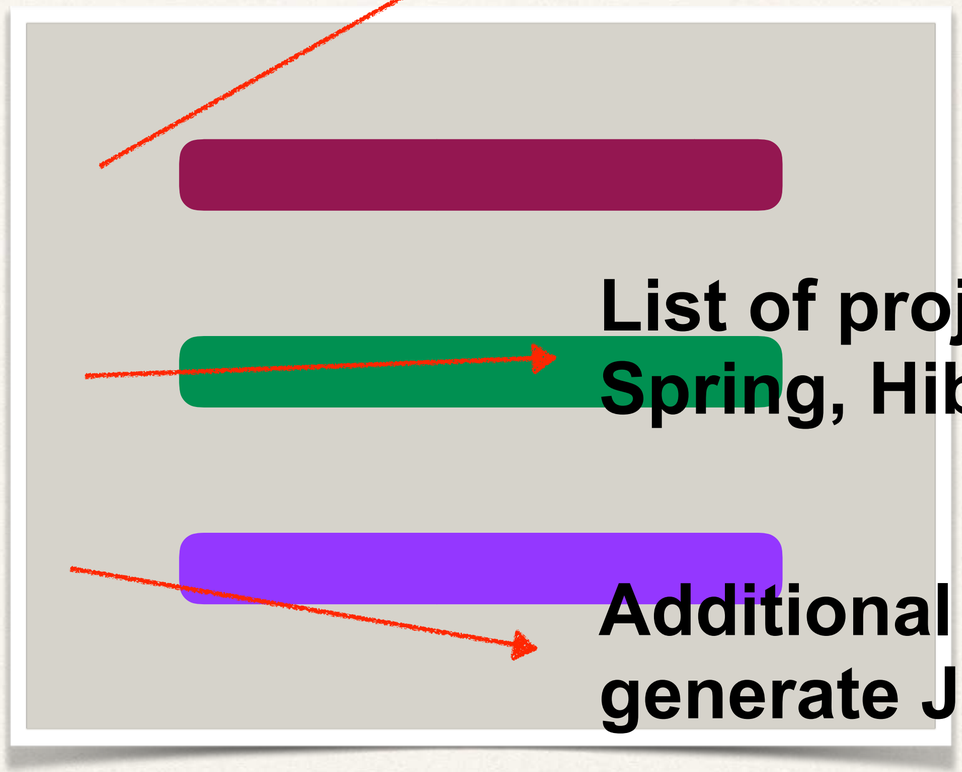
- POM File - pom.xml
- Project Coordinates

POM File - pom.xml



- Project Object Model file: POM file
- Configuration file for your project
  - Basically your “shopping list” for Maven :-)
- Located in the root of your Maven project

POM File Structure



Project name, version etc
Output file type: JAR, WAR, …

List of projects we depend on
Spring, Hibernate, etc…

Additional custom tasks to run:
generate JUnit test reports etc…

Simple POM File

```
<project ...>

 <modelVersion>4.0.0</modelVersion>

 <groupId>com.luv2code</groupId>
 <artifactId>mycoolapp</artifactId>
 <version>1.0.FINAL</version>
 <packaging>jar</packaging>

 <name>mycoolapp</name>

 <dependencies>
   <dependency>
     <groupId>junit</groupId>
     <artifactId>junit</artifactId>
     <version>3.8.1</version>
     <scope>test</scope>
   </dependency>
 </dependencies>

    <!-- add plugins for customization -->

</project>
```

Project name, version etc
Output file type: JAR, WAR, …

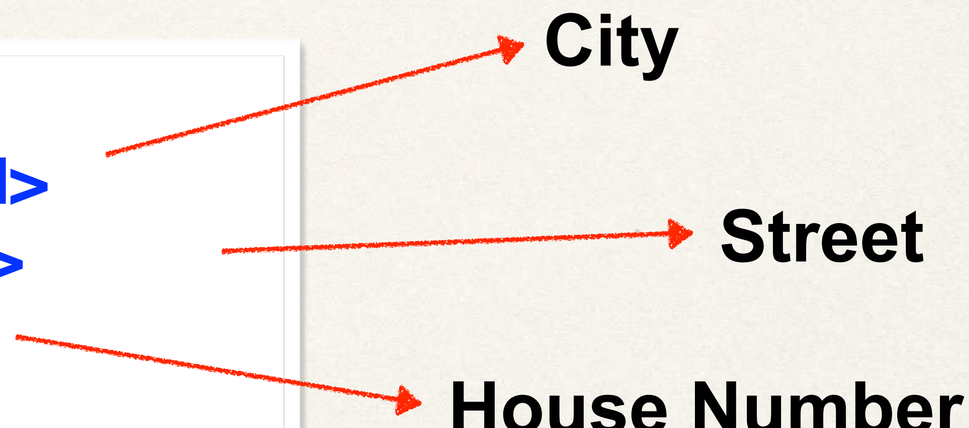
List of projects we depend on
Spring, Hibernate, etc…

Additional custom tasks to run:
generate JUnit test reports etc…

Project Coordinates

- Project Coordinates uniquely identify a project
  - Similar to GPS coordinates for your house: latitude / longitude
  - Precise information for finding your house (city, street, house #)

```
<groupId>com.luv2code</groupId>
<artifactId>mycoolapp</artifactId>
<version>1.0.FINAL</version>
```



Project Coordinates - Elements

Name	Description
Group ID	Name of company, group, or organization. Convention is to use reverse domain name: **com.luv2code**
Artifact ID	Name for this project: **mycoolapp**
Version	A specific release version like: **1.0, 1.6, 2.0** … If project is under active development then: **1.0-SNAPSHOT**

```
<groupId>com.luv2code</groupId>
<artifactId>mycoolapp</artifactId>
<version>1.0.FINAL</version>
```

Example of Project Coordinates

```
<groupId>com.luv2code</groupId>
<artifactId>mycoolapp</artifactId>
<version>1.0.RELEASE</version>
```

```
<groupId>org.springframework</groupId>
<artifactId>spring-context</artifactId>
<version>5.0.0.RELEASE</version>
```

```
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.2.11.Final</version>
```

Adding Dependencies

```
<project ...>
 ...
 <dependencies>

      <dependency>
         <groupId>org.springframework</groupId>
         <artifactId>spring-context</artifactId>
         <version>5.0.0.RELEASE</version>
      </dependency>

      <dependency>
         <groupId>org.hibernate</groupId>
         <artifactId>hibernate-core</artifactId>
         <version>5.2.11.Final</version>
      </dependency>
      ...

 </dependencies>

</project>
```

Dependency Coordinates

- To add given dependency project, we need
  - **Group ID**, **Artifact ID**
  - **Version** is optional …
    - Best practice is to include the version (repeatable builds)

- May see this referred to as: **GAV**
  - **G**roup ID, **A**rtifact ID and **V**ersion

How to Find Dependency Coordinates

- Option 1: Visit the project page (spring.io, hibernate.org etc)

- Option 2: Visit http://search.maven.org (easiest approach)

Maven - How to Find Dependencies

How to Find Dependency Coordinates

- Option 1: Visit the project page (spring.io, hibernate.org etc)

- Option 2: Visit http://search.maven.org (easiest approach)

Maven Archetypes

Maven Archetypes

- Archetypes can be used to create new Maven projects
- Contains template files for a given Maven project
- Think of it as a collection of “starter files” for a project
  - Java Project, Web Project, etc

archetype := starter project

Common Archetypes

Archetype Artifact ID	Description
maven-archetype-quickstart	An archetype to generate a sample Maven project.
maven-archetype-webapp	An archetype to generate a sample Maven Webapp project.
… others …	…

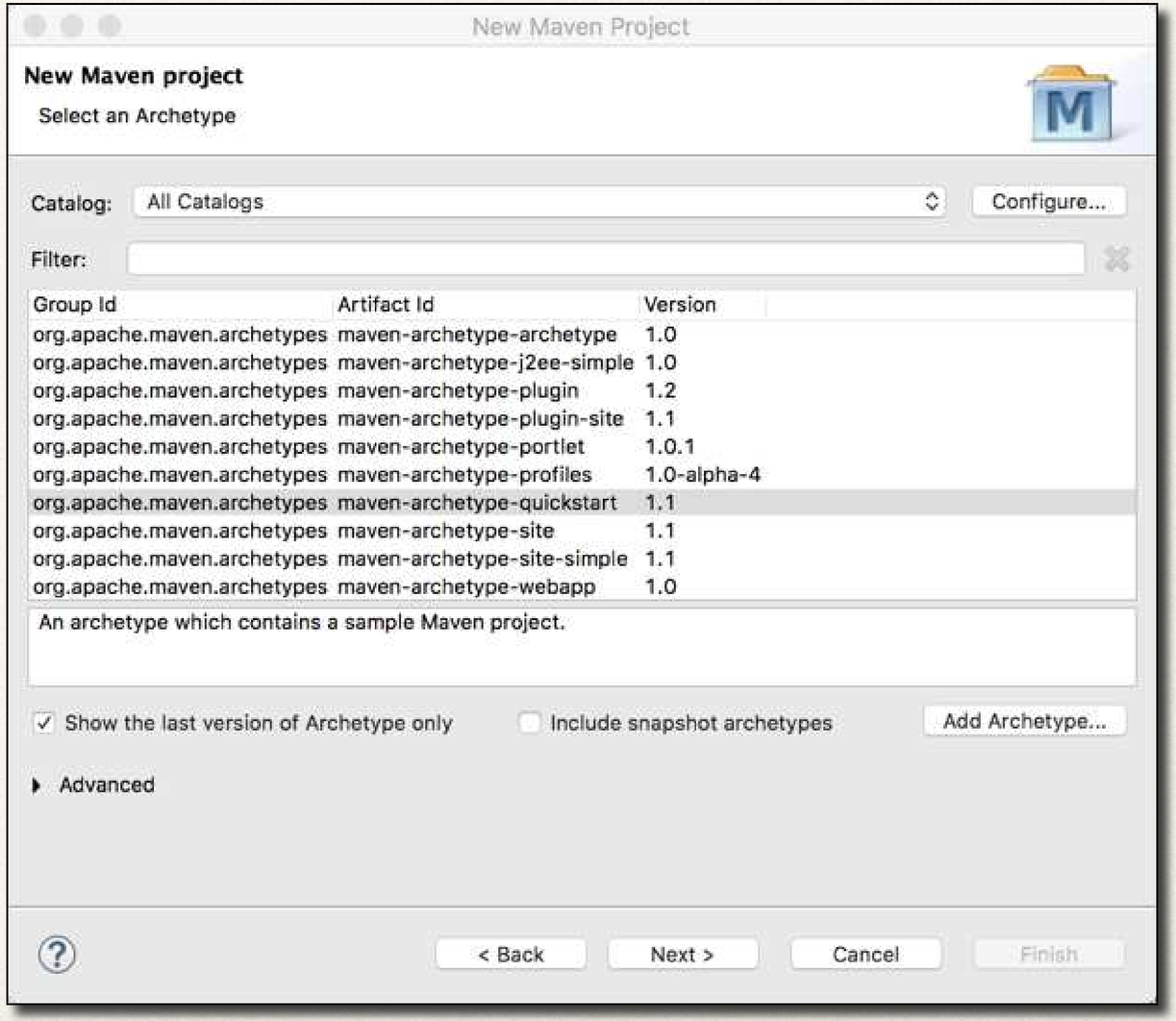
http://maven.apache.org/archetypes

Archetypes

- You can create new projects using Maven Archetypes (starter project)
  - From the command-line with Maven
  - From an IDE
    - Eclipse, IntelliJ, NetBeans etc…

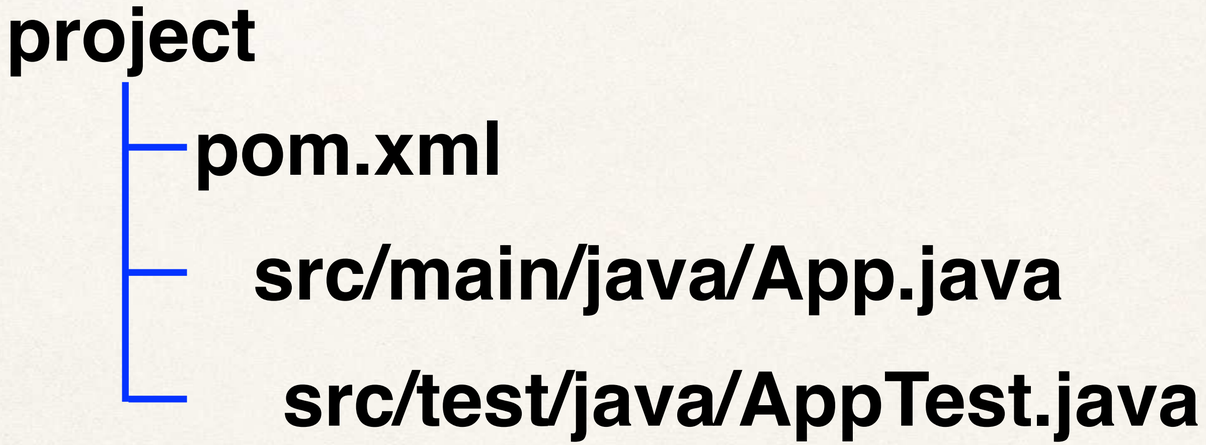
List of Archetypes - Eclipse

File > New Maven Project



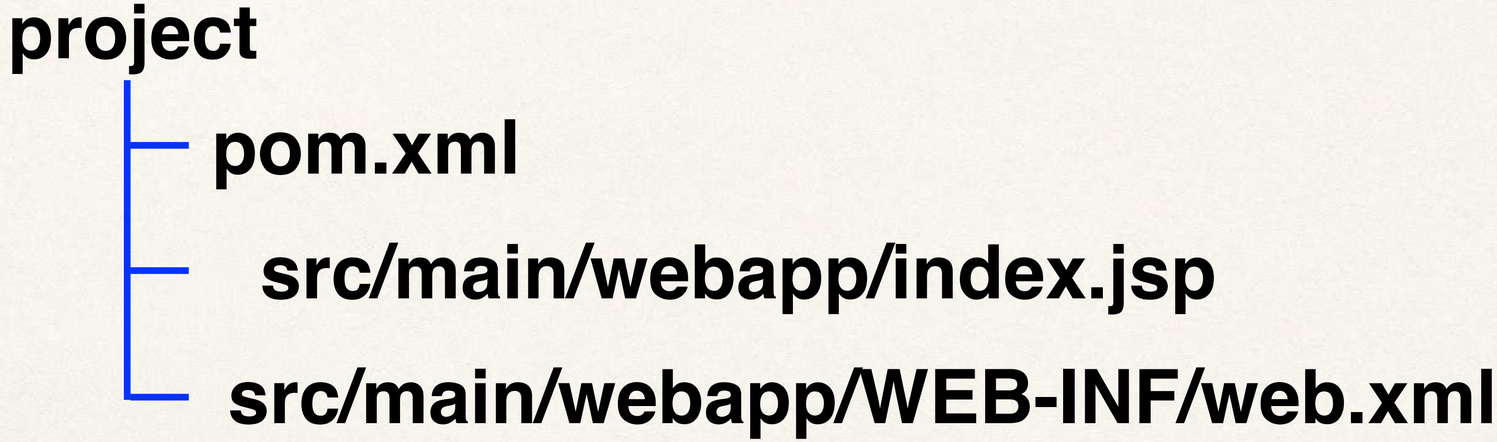
Quickstart Archetype

- maven-archetype-quickstart contains a sample Maven project



Webapp Archetype

- maven-archetype-webapp contains a sample Maven webapp project



Creating a Simple Maven Project

The Game Plan

- Learn about Eclipse's support for Maven
- Create a simple Maven Project with Eclipse
- Explore the generated files and directories
- Run our App
- Update the Java version settings

Eclipse and Maven

- Most recent versions of Eclipse have built-in support for Maven
  - Use the **m2eclipse** plugin
- There is **no need** to download/install Maven separately
- Can create Maven projects, add dependencies all inside of Eclipse

Create a WebApp Maven Project

The Game Plan

- Create a WebApp Maven Project with Eclipse
- Explore the generated files and directories
- Add dependencies to our POM file
- Run our WebApp

Maven Repositories

Maven - How It Works

Maven Local Repository

(your computer)

Check local repo

2

4

Save in local repo

Project Config file (pom.xml)

Spring
Hibernate
Commons Logging
JSON
...

developer

Read config file

1

Maven

5

Build and Run

3

Get from remote repo

Maven Central Repository (remote)

Repository Types

- Local Repository
- Central Repository

Local Repository

- Located on developer's computer
  - MS Windows: **c:\Users\<users-home-dir>\.m2\repository**
  - Mac and Linux: **~/.m2/repository**

- Maven will search this local repository first
  - Before going to Maven Central Repository
  - Your local cache

Central Repository

- By default, Maven will search Maven's Central Repository (remote)
  - https://repo.maven.apache.org/maven2/
- Requires an Internet connection
- Once files are downloaded, they are stored in local repository

Maven Repositories - Additional

Maven - How It Works

Maven Local Repository

(your computer)

Check local repo

2

4

Save in local repo

Project Config file (pom.xml)

Spring
Hibernate
Commons Logging
JSON
...

developer

Read config file

1

Maven

5

Build and Run

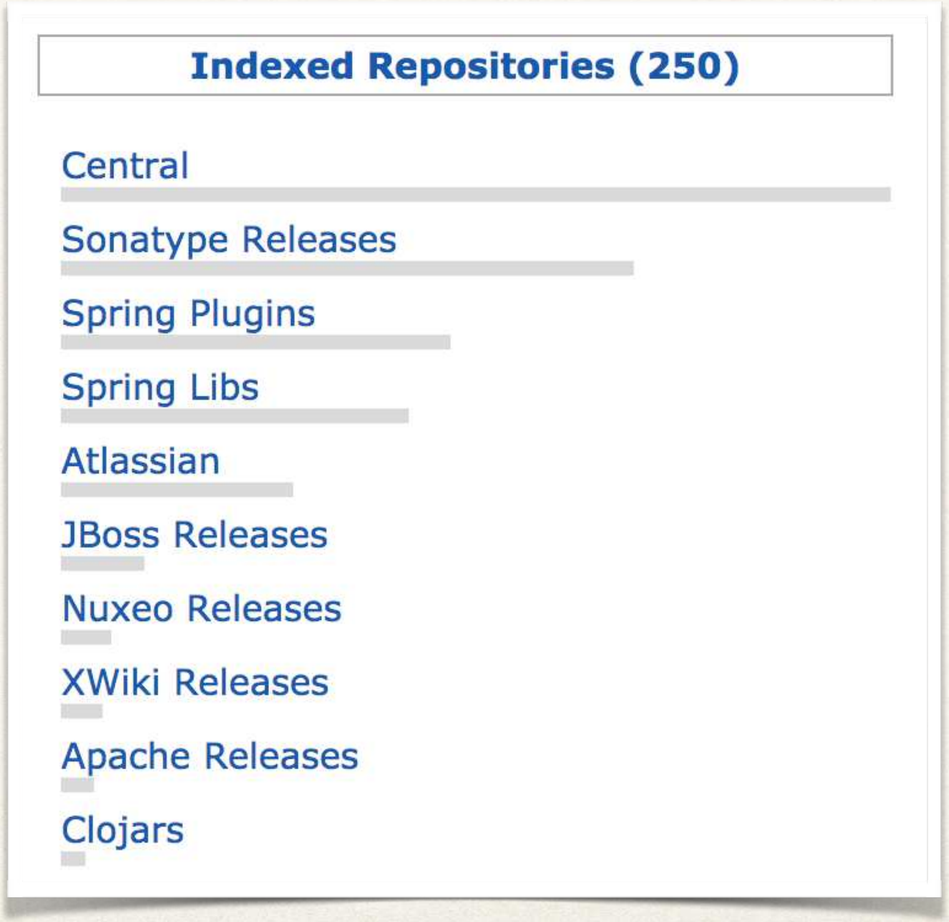
3

Get from remote repo

Maven Central Repository (remote)

Additional Repositories

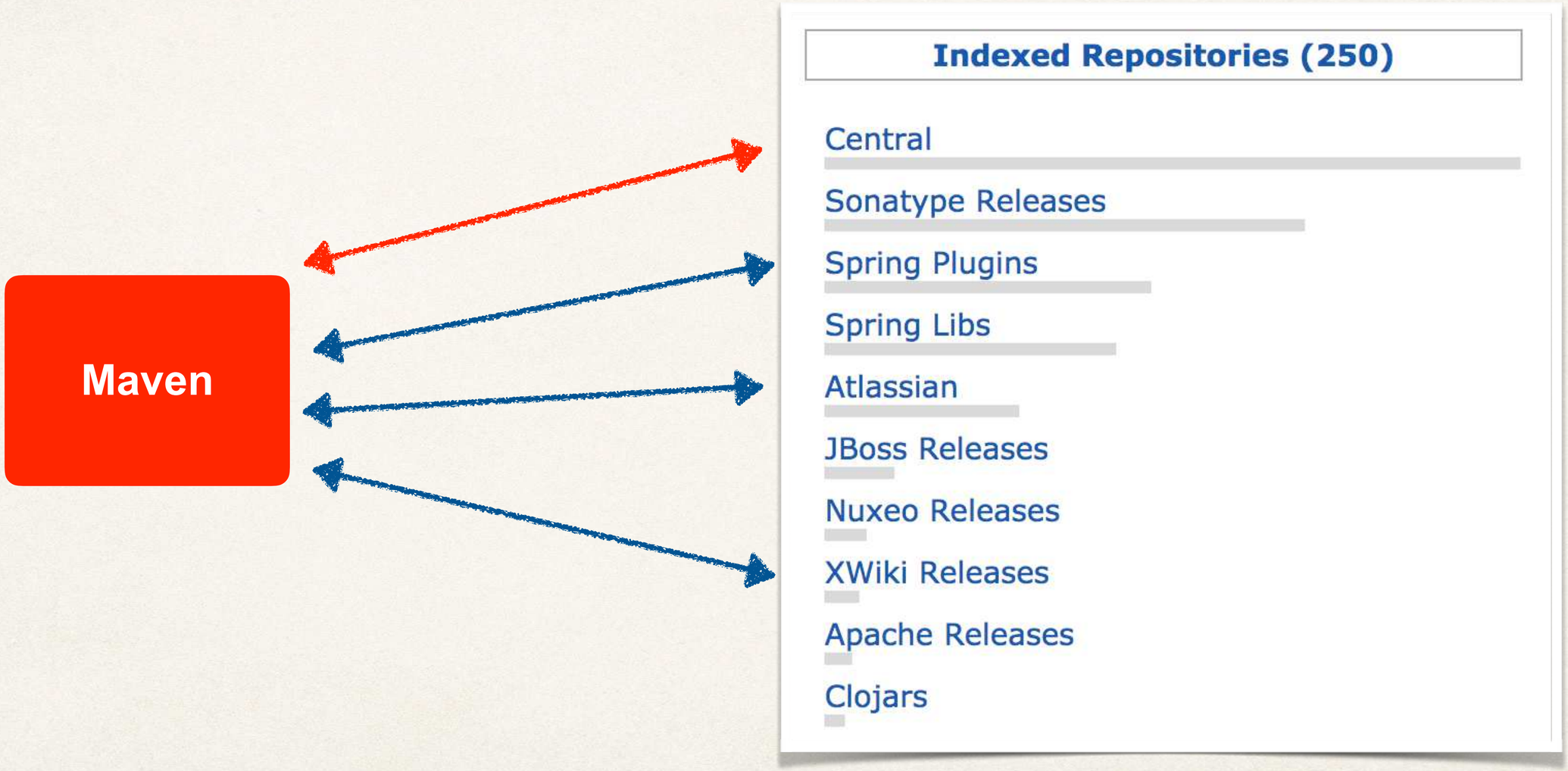
- Remember how mvnrepository.com can index additional repositories?



Why are there Additional Repositories?

- Some development teams may self-host their Maven projects / JARs
  - Alpha/Beta releases
  - More control over deployments etc
  - *Other reasons …*

Accessing Additional Repositories



Task: Configure Additional Repositories

Step-By-Step

- Step 1: Add **atlassian-mail** dependency to our project
  - This is only available in the Atlassian repository

- Step 2: Add the Atlassian repository to our **pom.xml** file

Step 1: Adding atlassian-mail Dependency

File: pom.xml

```
<dependencies>

    <dependency>
     <groupId>com.atlassian.mail</groupId>
     <artifactId>atlassian-mail</artifactId>
     <version>4.0.4</version>
    </dependency>

</dependencies>
```

Step 2: Adding Repositories

File: pom.xml

```
<repositories>

  <repository>
    <id>atlassian</id>
    <name>Atlassian Repository</name>
    <url>https://maven.atlassian.com/content/repositories/atlassian-public/</url>
  </repository>

  <repository>
        ...
   </repository>

</repositories>
```

Maven Repositories - Private

Maven - How It Works

Maven Local Repository

(your computer)

Check local repo

2

4

Save in local repo

Project Config file (pom.xml)

Spring
Hibernate
Commons Logging
JSON
...

developer

Read config file

1

Maven

5

Build and Run

3

Get from remote repo

Maven Central Repository (remote)

Private Repositories - Use Case

- Your company has created super-top-secret code modules
- Would like to share with other development teams at your company
- But let's keep it private
- NOT AVAILABLE TO THE PUBLIC

Private Maven Repository

- You can set up your own private Maven Repository
  - Secure it with credentials: id / password
- Create your super-top-secret projects and publish on private repository
- Your development teams can access your private repository

Maven Repository Manager Products

Product	Company	Website
Archiva	Apache	archiva.apache.org
Artifactory	JFrog	www.jfrog.com
Nexus	Sonatype	www.sonatype.com

Cloud Solutions

- If you don't need to self-host internally at your company
- Cloud hosted solutions are available
- Check for:
  - www.packagecloud.io, www.mymavenrepo.com
  - Google: maven cloud hosts