NETAPSYS
ingénierie informatique

Maven

« Convention over configuration »

IUT de Nantes - Licence Professionnelle

Benoît COTINAT - benoit.cotinat@netapsys.fr

Construire son application

Comment ?

NETAPSYS
ingénierie informatique

Construire son application

- ✓ Problématiques
  - Librairies utilisées ? Partage au sein de l'équipe ?
  - Paramétrage ?
  - Assemblage ?
  - Déploiement ?

NETAPSYS
ingénierie informatique

Construire son application

- ✓ À la main
  - Pas d'exécution automatique des tests
  - Nombreuses manipulations
  - Pas reproductible
  - Source d'erreurs
  - Long

NETAPSYS
ingénierie informatique

Construire son application

- ✓ Ant
  - Scripts pour automatiser
  - Reproductible
  - Exécution des tests possible
  - Long à écrire
  - Peu réutilisable

NETAPSYS
ingénierie informatique

Construire son application

- ✓ Maven
  - Description du projet
  - Exécution des tests automatique
  - Reproductible
  - Extensible

Maven

pom

NETAPSYS
ingénierie informatique

pom.xml

```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>fr.dgac.oceane</groupId>
    <artifactId>oceane-parent</artifactId>
    <version>6.0.0-RC6</version>
    <packaging>pom</packaging>
    <name>oceane</name>

    <dependencies>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.4</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-resources-plugin</artifactId>
                <version>2.7</version>
                <configuration>
                    <encoding>${project.build.sourceEncoding}</encoding>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

NETAPSYS
ingénierie informatique

pom.xml

- « Project Object Model »
- Descripteur du projet
  - Identifiants
  - Dépendances
  - Plugins
- Définit un « artefact »

NETAPSYS
ingénierie informatique

Identification

- `groupId`
  - nom de l'organisation
- `artifactId`
  - nom du projet
- `version`
  - SNAPSHOT
- `packaging` :
  - type de construction (pom, jar, war, ear, …)

NETAPSYS
ingénierie informatique

Dépendances

- ✓ Liste des artefacts nécessaires à la construction du projet
  - librairie, driver, projet, …
- ✓ Transitivité
  - LibA → LibB → LibC
- ✓ Scope d'utilisation
  - test, compile (défaut), provided, …

NETAPSYS
ingénierie informatique

Plugins

- ✓ Cœur de l'exécution de Maven
- ✓ Définit des **goals** à exécuter
- ✓ Etend la construction du projet (reporting, …)
- ✓ Exécution :
  - mvn plugin:goal

NETAPSYS
ingénierie informatique

pom.xml

```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>fr.dgac.oceane</groupId>
    <artifactId>oceane-parent</artifactId>
    <version>6.0.0-RC6</version>
    <packaging>pom</packaging>
    <name>oceane</name>

    <dependencies>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-core</artifactId>
            <version>2.4</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-resources-plugin</artifactId>
                <version>2.7</version>
                <configuration>
                    <encoding>${project.build.sourceEncoding}</encoding>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

Maven

Conventions

NETAPSYS
ingénierie informatique

Conventions

- ✓ Nommage
- ✓ Emplacement des fichiers applicatifs
  - *<directory>*/src/main/java
  - *<directory>*/src/main/resources
  - *<directory>*/src/main/filters
  - *<directory>*/src/main/webapp

NETAPSYS
ingénierie informatique

Conventions

- ✓ Emplacement des fichiers liés aux tests
  - *<directory>*/src/test/java
  - *<directory>*/src/test/resources
  - *<directory>*/src/test/filters

NETAPSYS
ingénierie informatique

Conventions

- Répertoire de travail de Maven :
  - *<directory>*/target/

Maven

Construction

NETAPSYS
ingénierie informatique

Cycles de vie

- ✓ Inclus dans Maven
  - clean
  - build
  - site

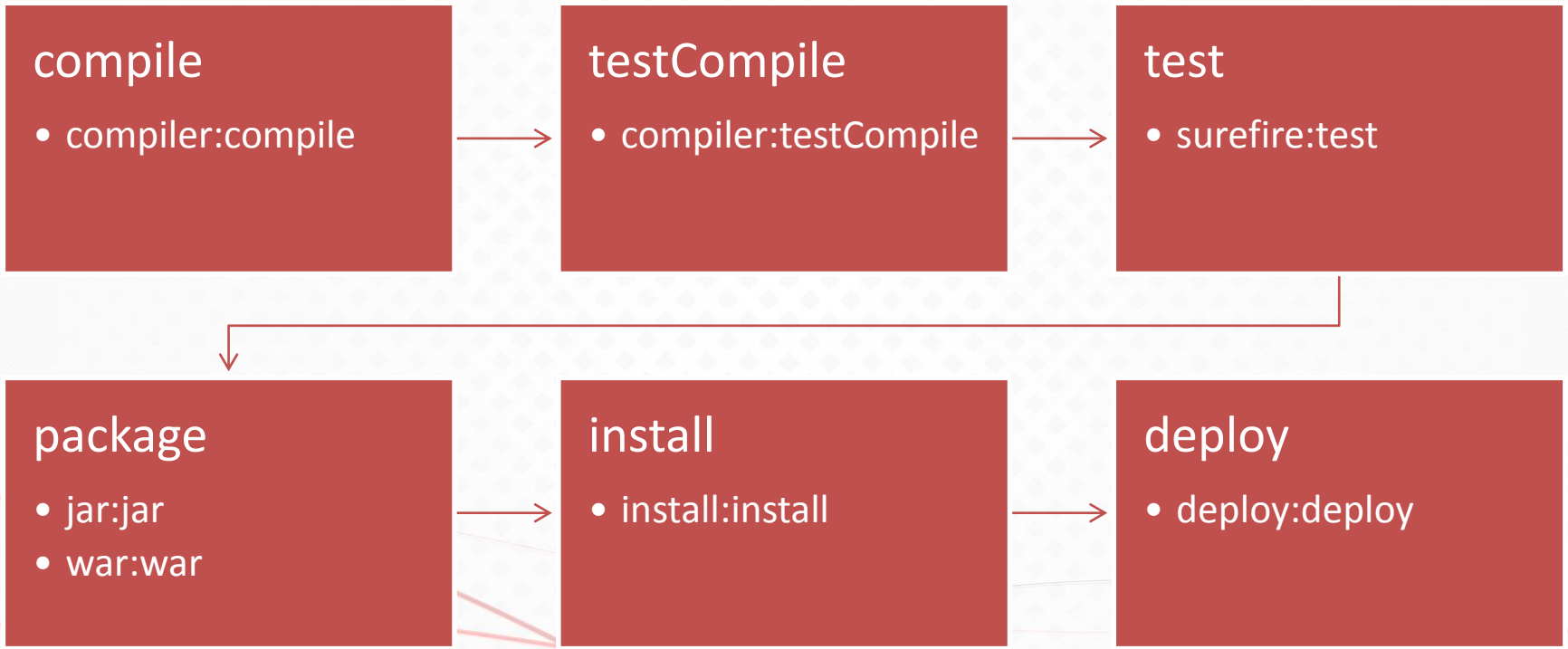
NETAPSYS
ingénierie informatique

Cycles de vie

- ✓ Constitué de phases
  - Etape du cycle
  - Transitives
  - Appel de goal de plugin
  - Dépendante du type de packaging

NETAPSYS
ingénierie informatique

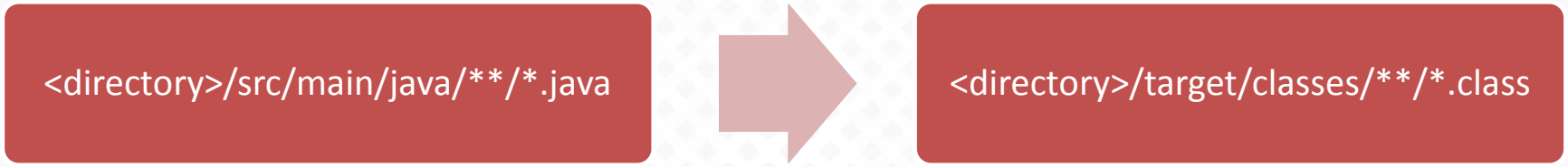
« Build lifecycle »



NETAPSYS
ingénierie informatique

« Build lifecycle »

- ✓ compile

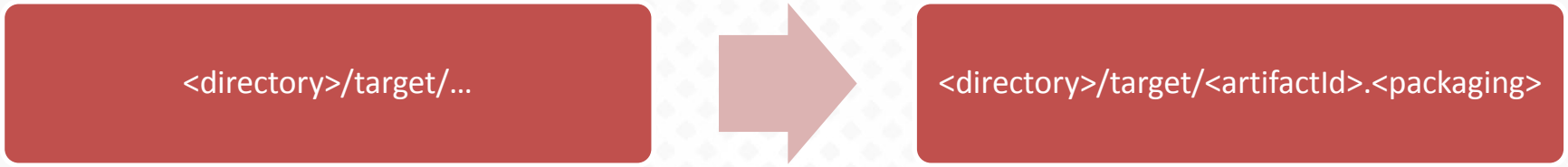


- ✓ test-compile

NETAPSYS
ingénierie informatique

« Build lifecycle »

- ✓ package



- ✓ install

Maven

Stockage

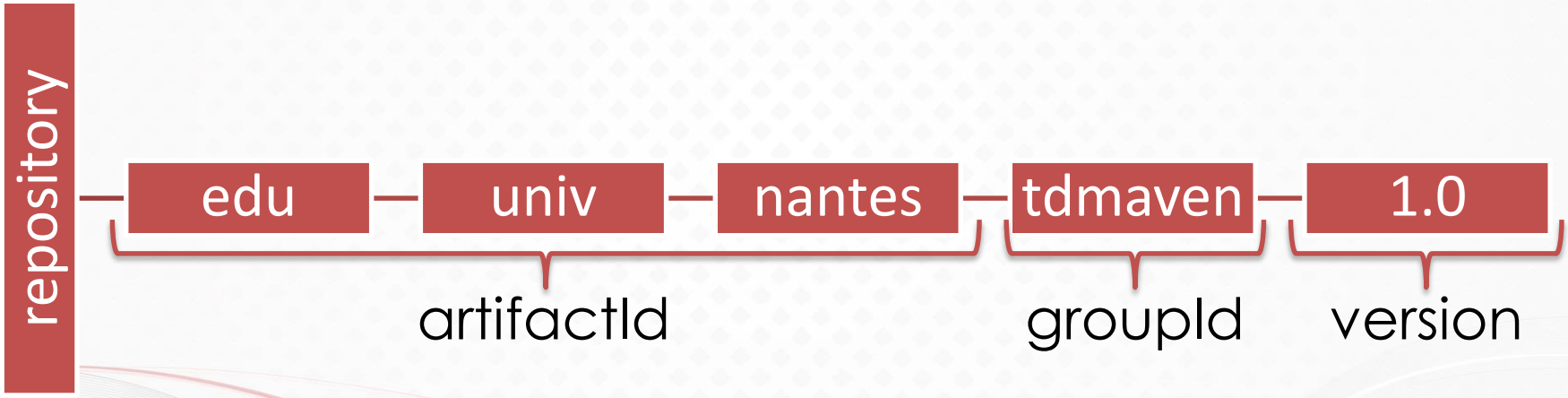
NETAPSYS
ingénierie informatique

Repository

- ✓ Stockage des artefacts
- ✓ Local
  - ~/.m2/repository
- ✓ Distants
  - Officiel : repo.maven.apache.org/maven2
  - Entreprise (Nexus, Artifactory, …)

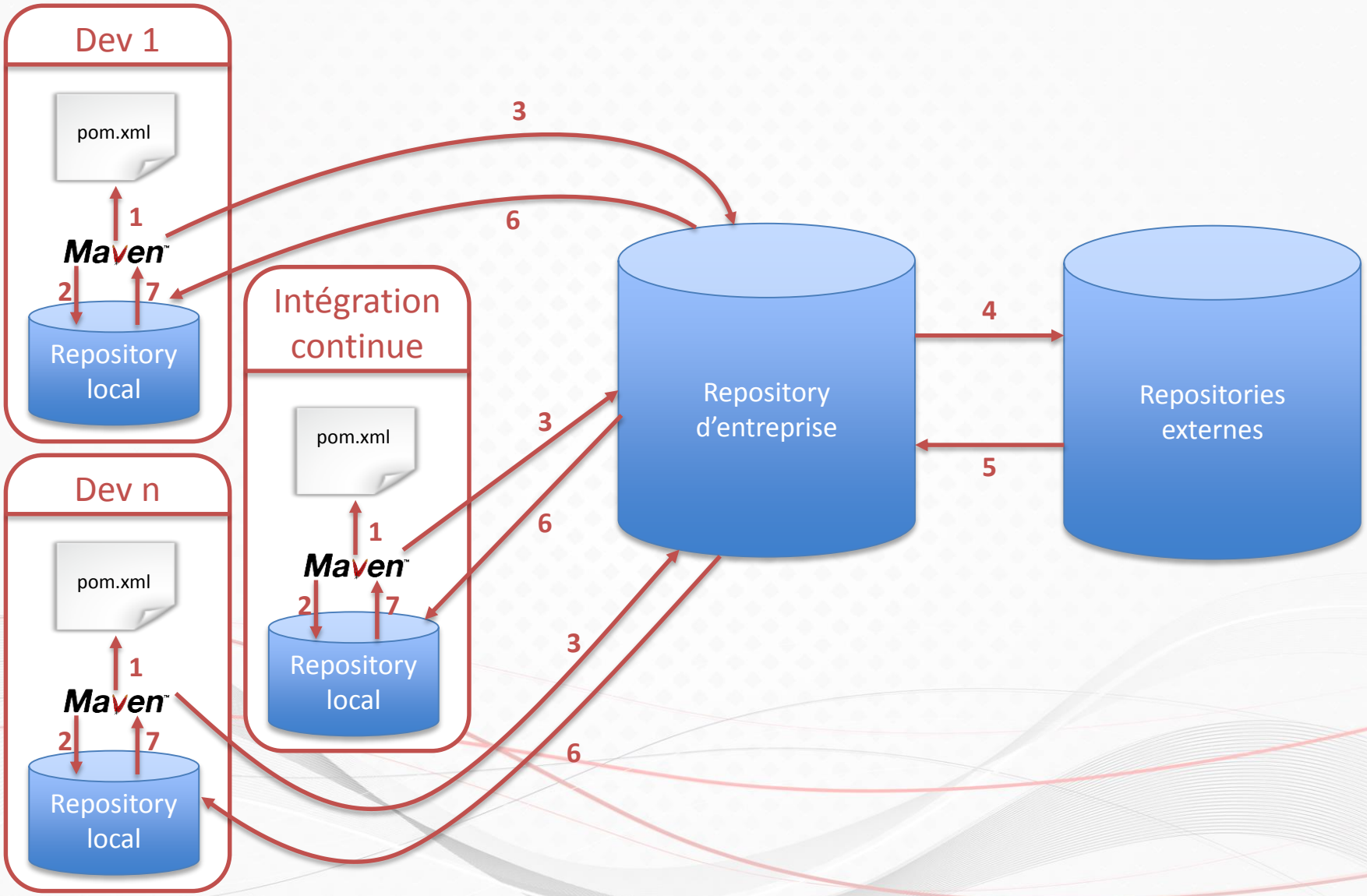
NETAPSYS
ingénierie informatique

Repository



NETAPSYS
ingénierie informatique

Repository



NETAPSYS
ingénierie informatique

Et aussi

- ✓ Modules
- ✓ Profiles
- ✓ Properties
- ✓ Filtering

NETAPSYS
ingénierie informatique

Réussir son projet

- ✓ Suivre les conventions
- ✓ Ecrire le moins de choses dans le pom
- ✓ Rester indépendant de l'environnement
- ✓ Toujours jouer les tests
- ✓ Utiliser les repositories