warmup

Naviguer dans le labyrinthe de la Gestion de Dépendances

Hervé Boutemy Louis Jacomet

sonatype Gradle

summary

/ Agenda

1. Introduction
   Qui, pourquoi, comment?
2. C’est quoi une dépendance
   Et oui, c’est quoi?
3. Déclaration et résolution de dépendances
   Maven, Gradle et NPM
4. Gestion des mises à jour
   Introspection, Automatisation
5. SBOM, la solution?
   kezako, xBOM

Introduction

sonatype

Maven™

/ Hervé Boutemy

Architecte Solutions, Sonatype, Software Supply Chain
Mainteneur Maven, Membre de la Fondation Apache
Mainteneur des plugins Maven pour SBOM CycloneDX & SPDX

Gradle

/ Louis Jacomet

- Senior Lead Software Engineer chez Gradle
- Dependency management, user support, product thinking

- name = "Louis Jacomet"
- mastodon = "@ljacomet@foojay.social"
- github = "ljacomet"
- twitter = "@ljacomet"

Introduction

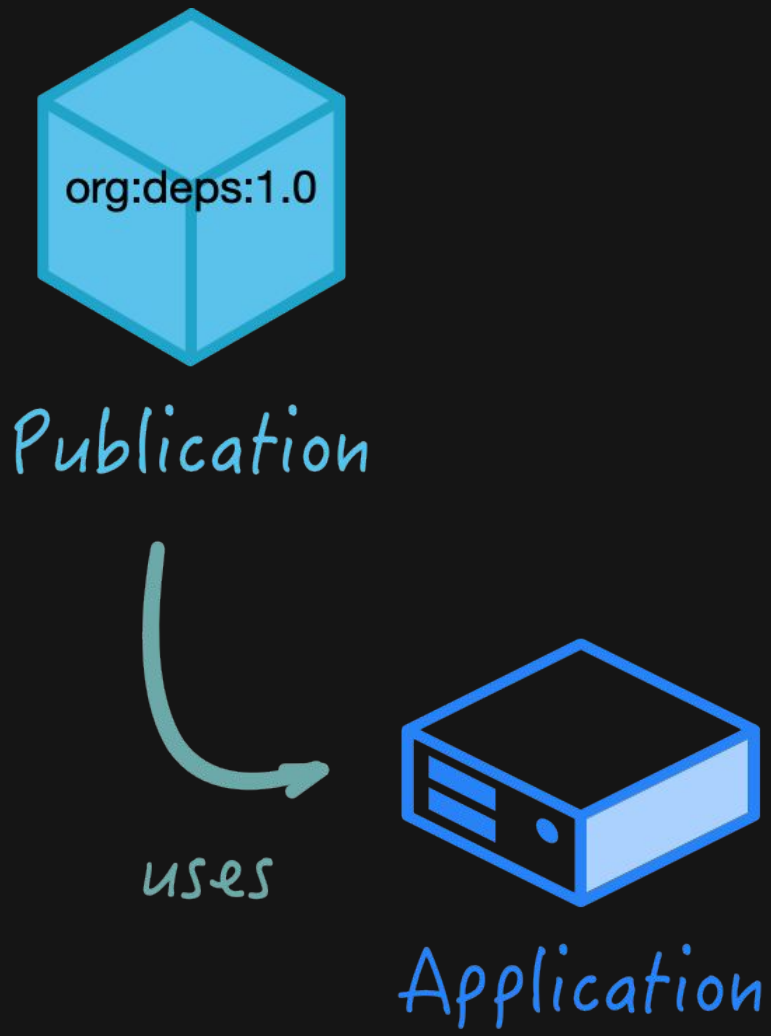
/ Questions ?

- C’est quoi une dépendance ?
- C’est quoi pour vous “gérer ses dépendances” ?

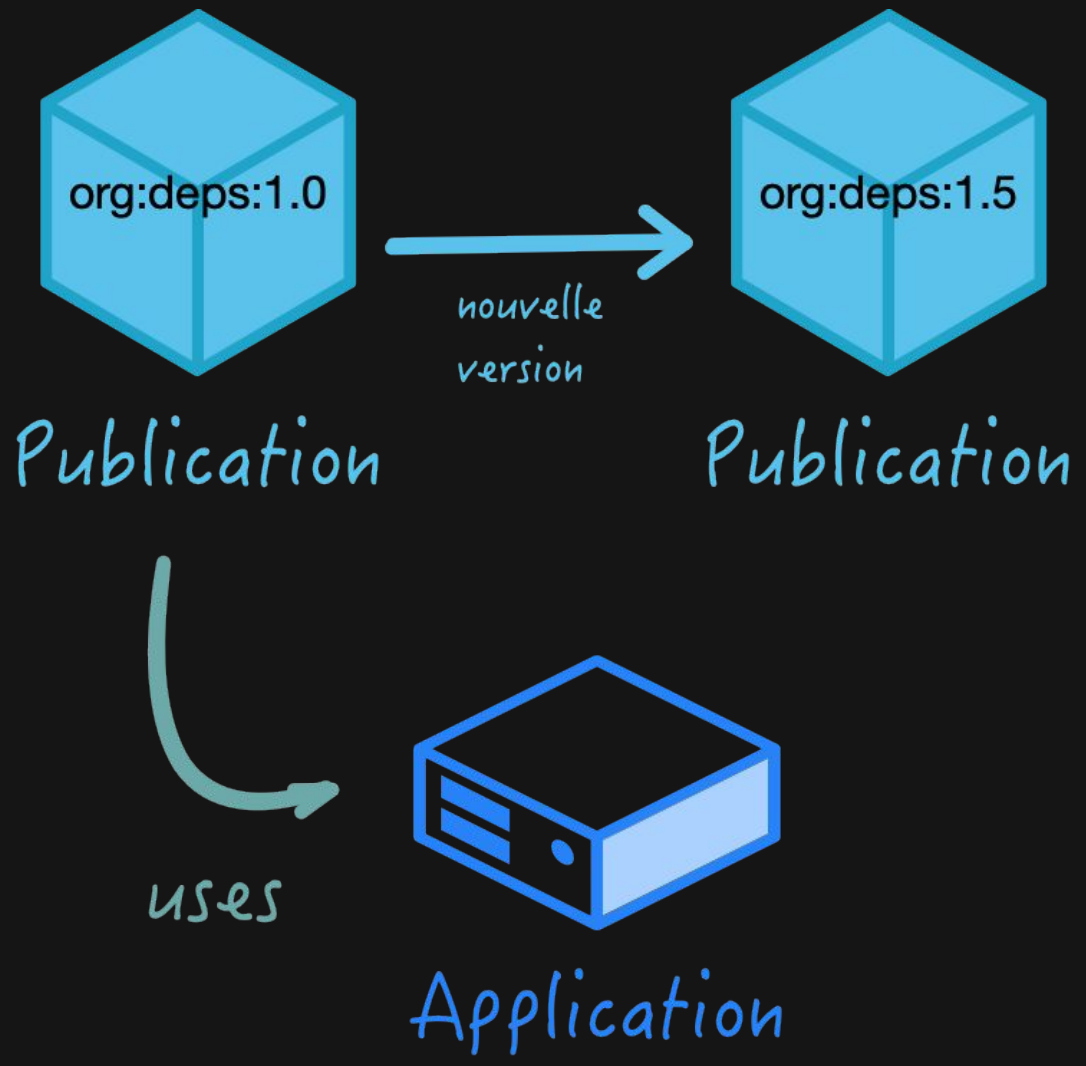
C’est quoi une dépendance ??

Publication

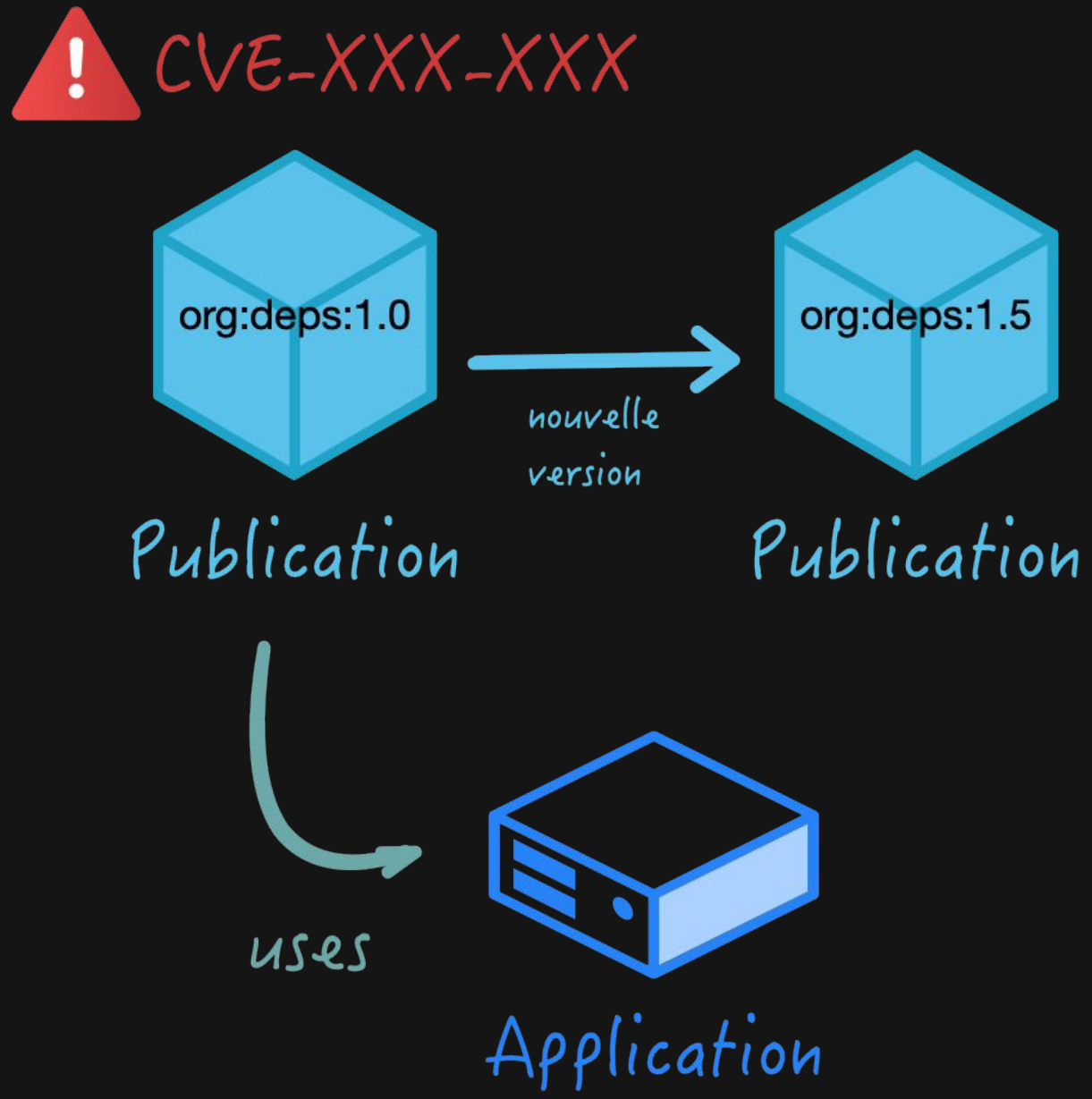
CODE



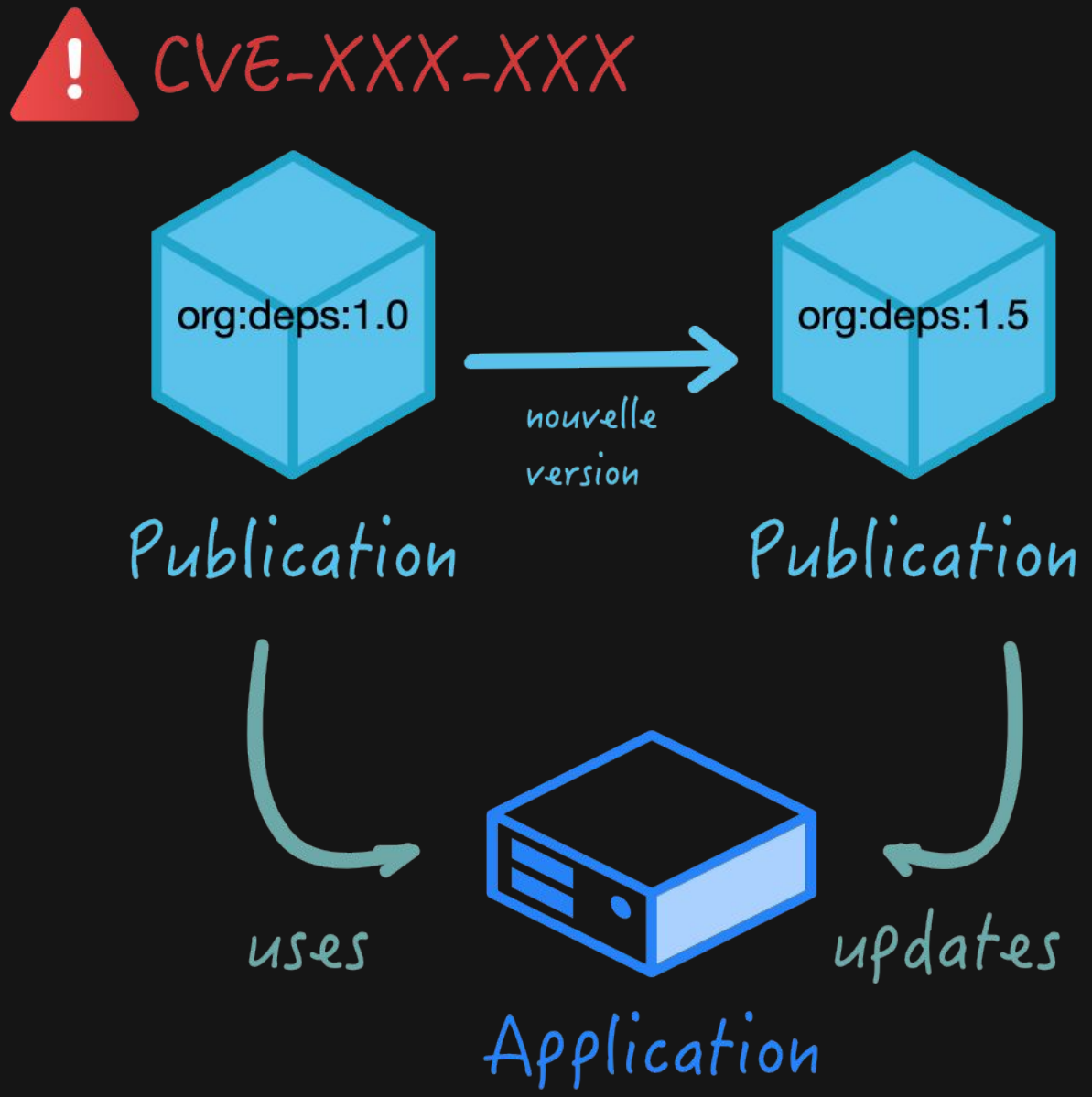
CODE


org:deps:1.0
nouvelle
version
org:deps:1.5
Publication
Publication
uses
Application

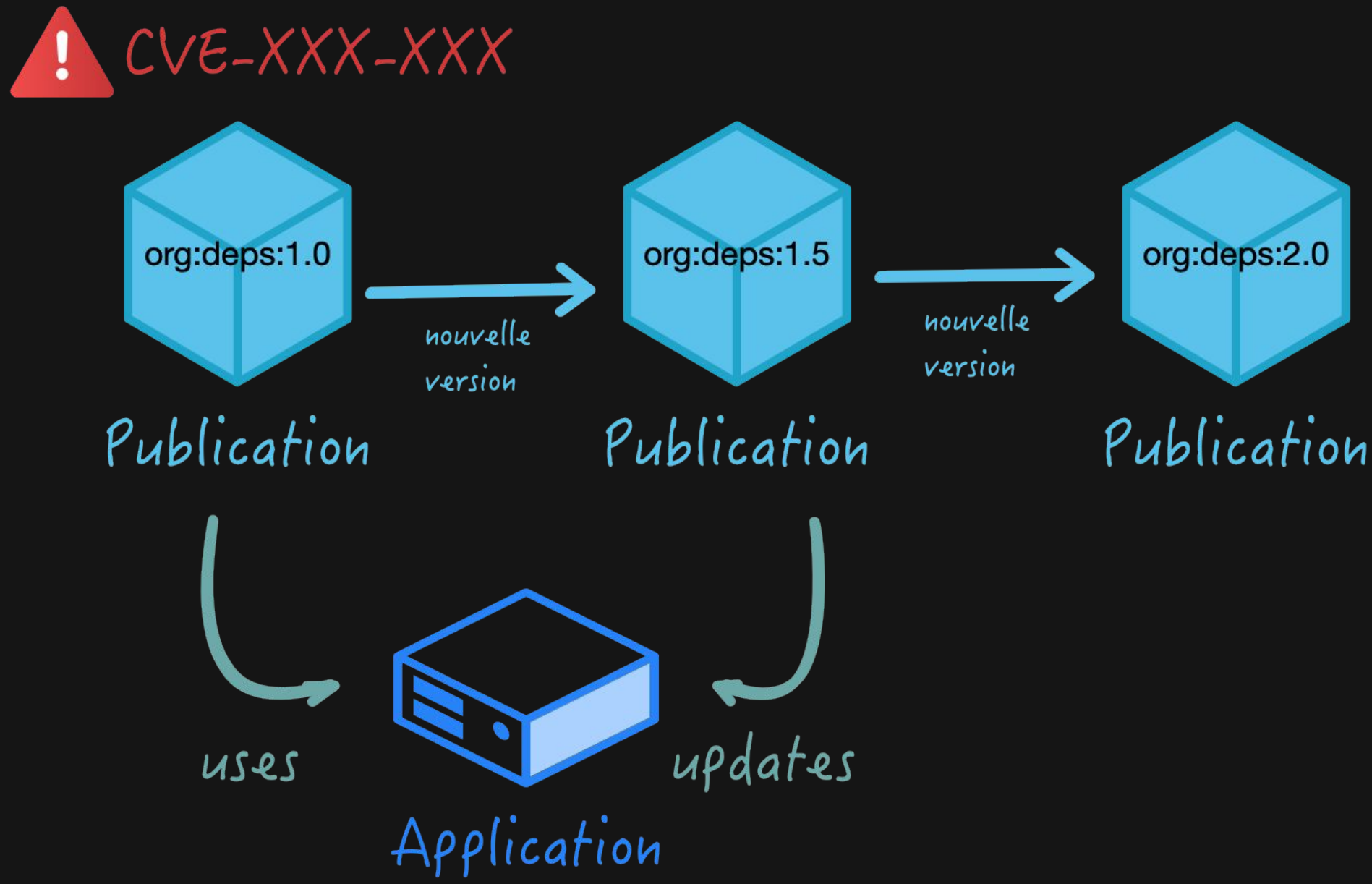
CODE

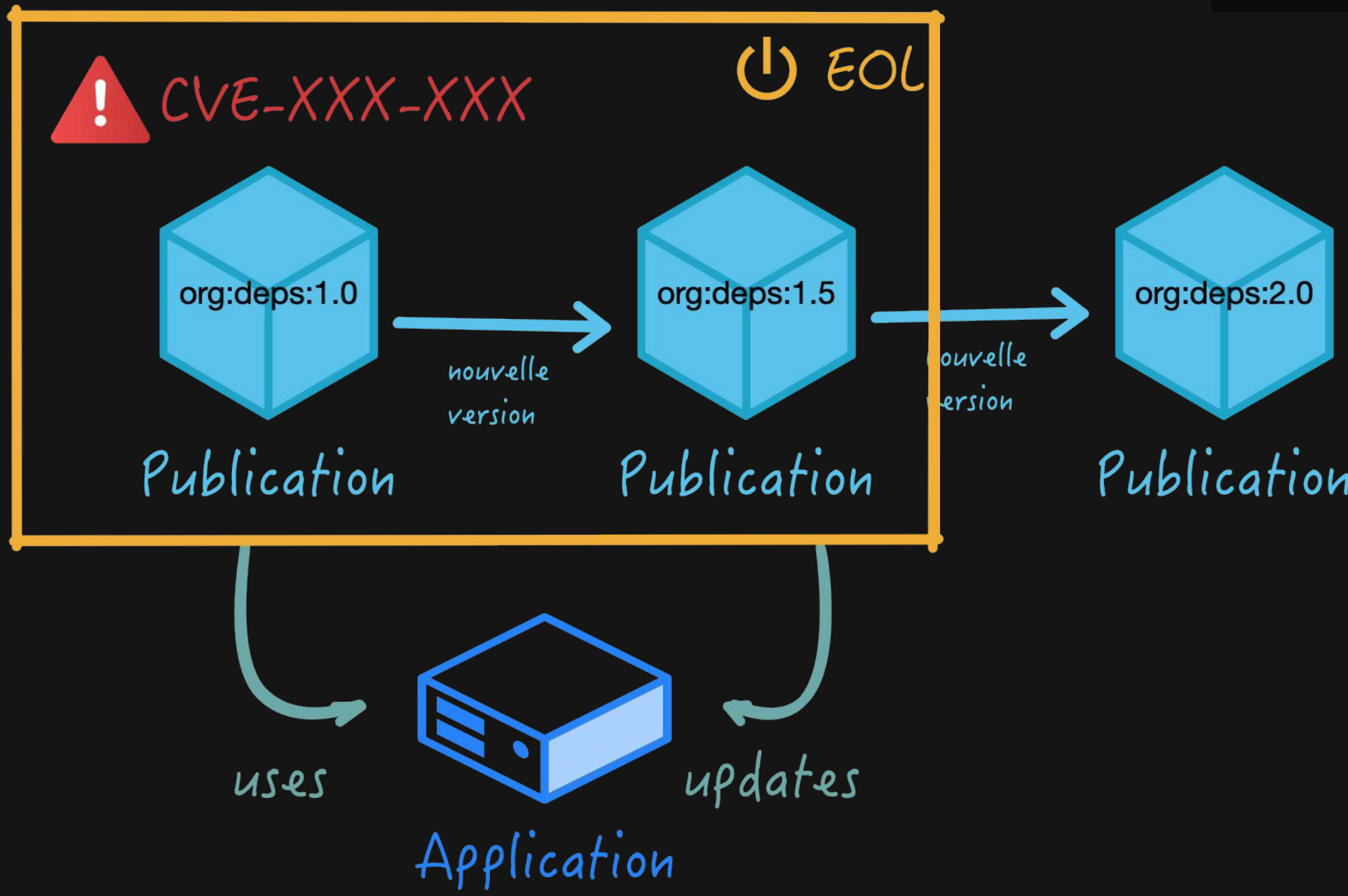

CVE-XXX-XXX
org:deps:1.0
org:deps:1.5
nouvelle version
Publication
Publication
uses
Application

CODE

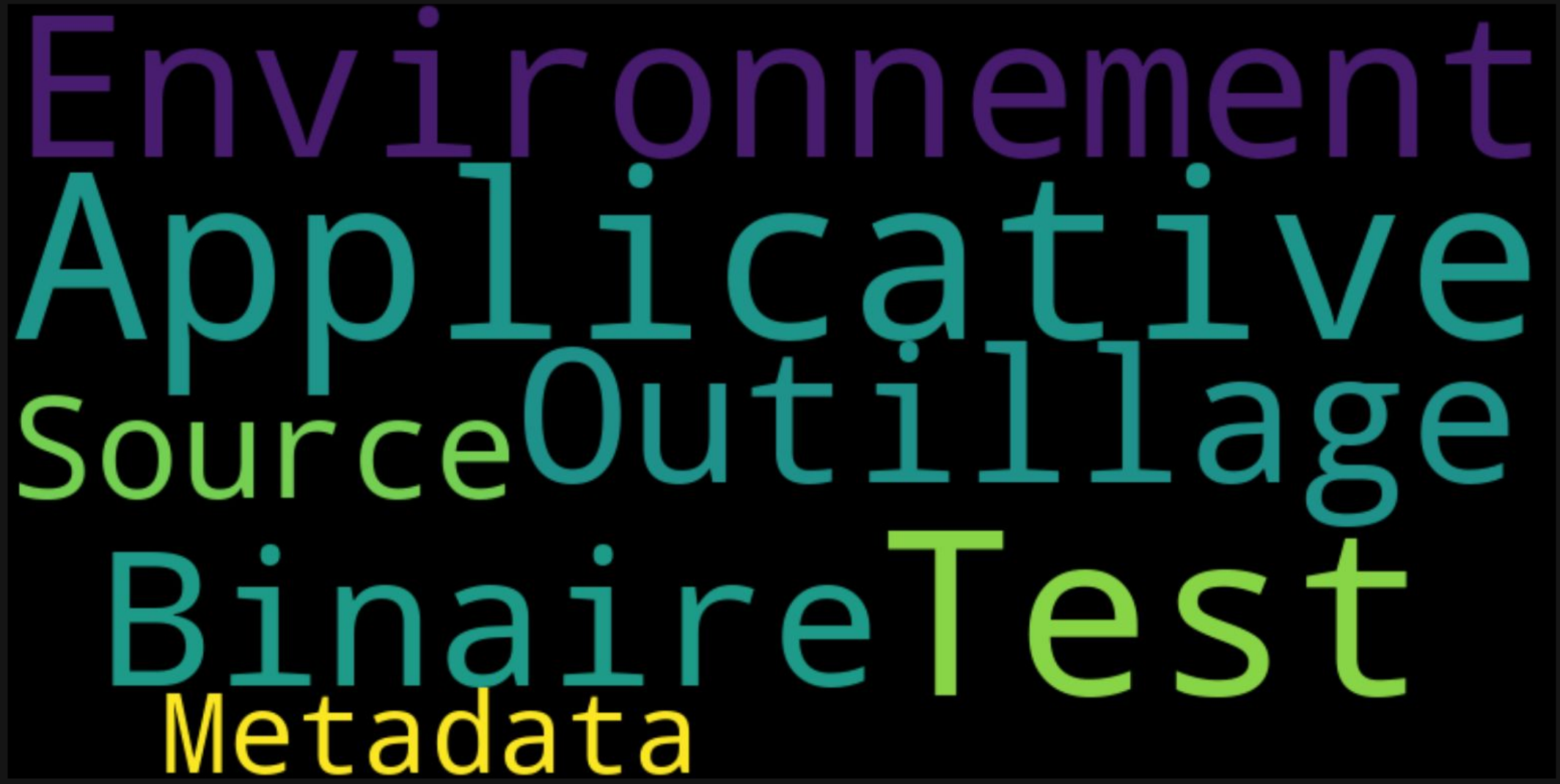

CVE-XXX-XXX
org:deps:1.0
org:deps:1.5
nouvelle version
Publication
Publication
uses
updates
Application

CODE


CVE-XXX-XXX
org:deps:1.0
nouvelle version
org:deps:1.5
nouvelle version
org:deps:2.0
Publication
Publication
Publication
uses
updates
Application


CVE-XXX-XXX
EOL
org:deps:1.0
org:deps:1.5
org:deps:2.0
nouvelle version
nouvelle version
Publication
Publication
Publication
uses
updates
Application

CODE


Environnement
Applicative
Source
Outillage
Binaire
Test
Metadata

CODE

Déclaration et résolution de dépendances

Déclaration et résolution de dépendances

/ Dépendance d’outils / environnement

- Version de Java
- Version de l’outil de build
  - Maven: Enforcer plugin et wrapper
  - Gradle: wrapper
- Environnement de production
  - OS et version
  - Container / Virtual / …

CODE

/ maven-core pom.xml

```
<dependencies>
 <dependency>
   <groupId>org.apache.maven</groupId>
   <artifactId>maven-model</artifactId>
 </dependency>
 <dependency>
   <groupId>org.apache.maven</groupId>
   <artifactId>maven-settings</artifactId>
 </dependency>
 ...
</dependencies>
```

/ maven pom.xml

```
<dependencyManagement>
 <dependencies>
   <dependency>
     <groupId>org.apache.maven</groupId>
     <artifactId>maven-model</artifactId>
     <version>${project.version}</version>
   </dependency>
   <dependency>
     <groupId>org.apache.maven</groupId>
     <artifactId>maven-settings</artifactId>
     <version>${project.version}</version>
   </dependency>
</.../>
```

\+ import BOM POM

CODE

/ build init

build.gradle.kts

```
dependencies {
    api(libs.math)
    api(libs.guava)

    implementation(project(":utils"))
}
```

/ build init

libs.versions.toml

```
[versions]
math = "3.6.1"
guava = "30.1.1-jre"

[libraries]
math = { module = "org.apache.commons:commons-math3", version.ref = "math" }
guava = { module = "com.google.guava:guava", version.ref = "guava" }
```

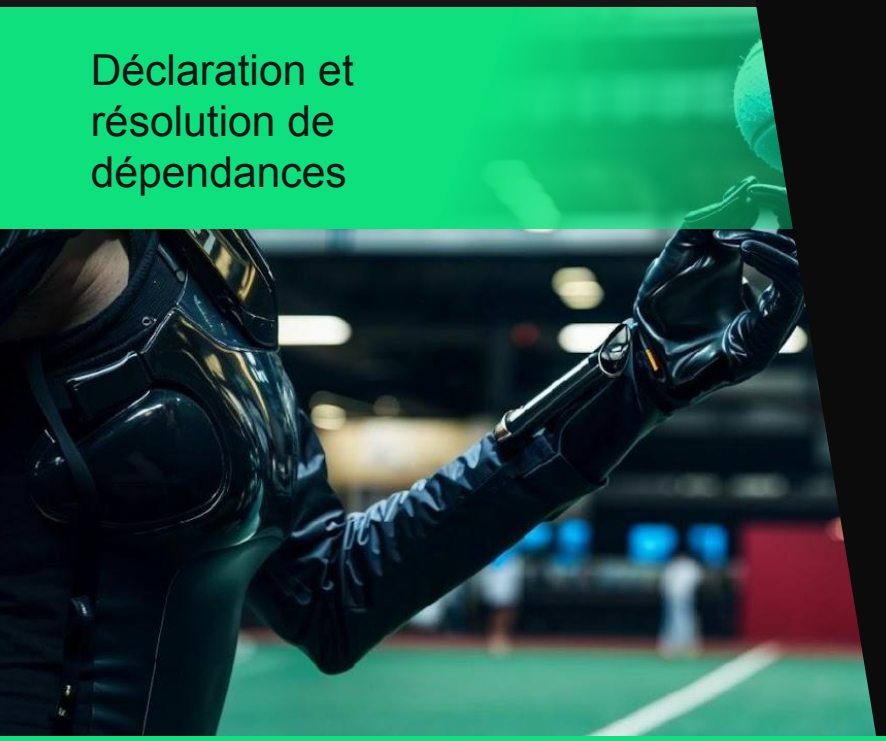
CODE

/ Github action

package.json

```
"dependencies": {
  "@actions/core": "^1.10.0",
  "@actions/github": "^5.1.1",
  "@octokit/action": "^4.0.10"
},
"devDependencies": {
  "@vercel/ncc": "^0.33.3",
  "eslint": "^8.9.0",
  "eslint-config-google": "^0.14.0"
}
```

Déclaration et résolution de dépendances

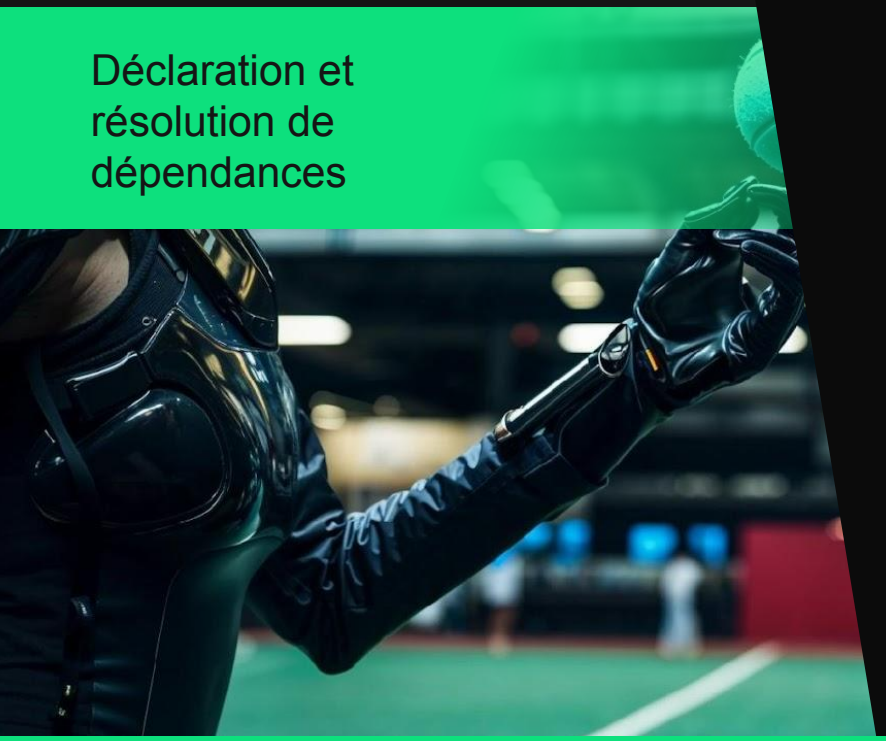


/ Range & locks

- NPM
  - Use range notation heavily
  - Resolution locks a specific version
- Maven
  - Discourage range notations
- Gradle
  - Discourage range notations
  - Optional dependency locking

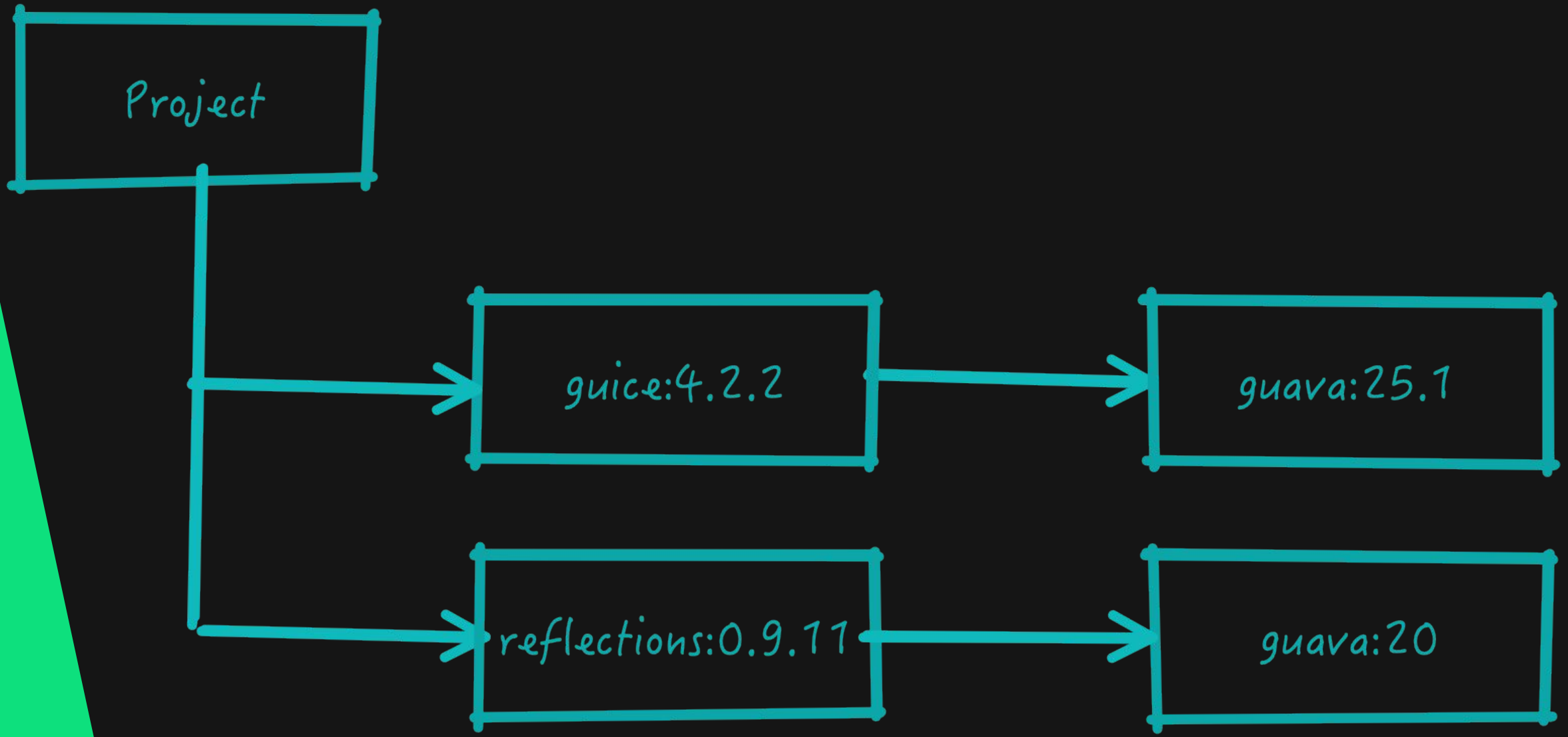
/ GLOBAL GUIDE

Déclaration et résolution de dépendances

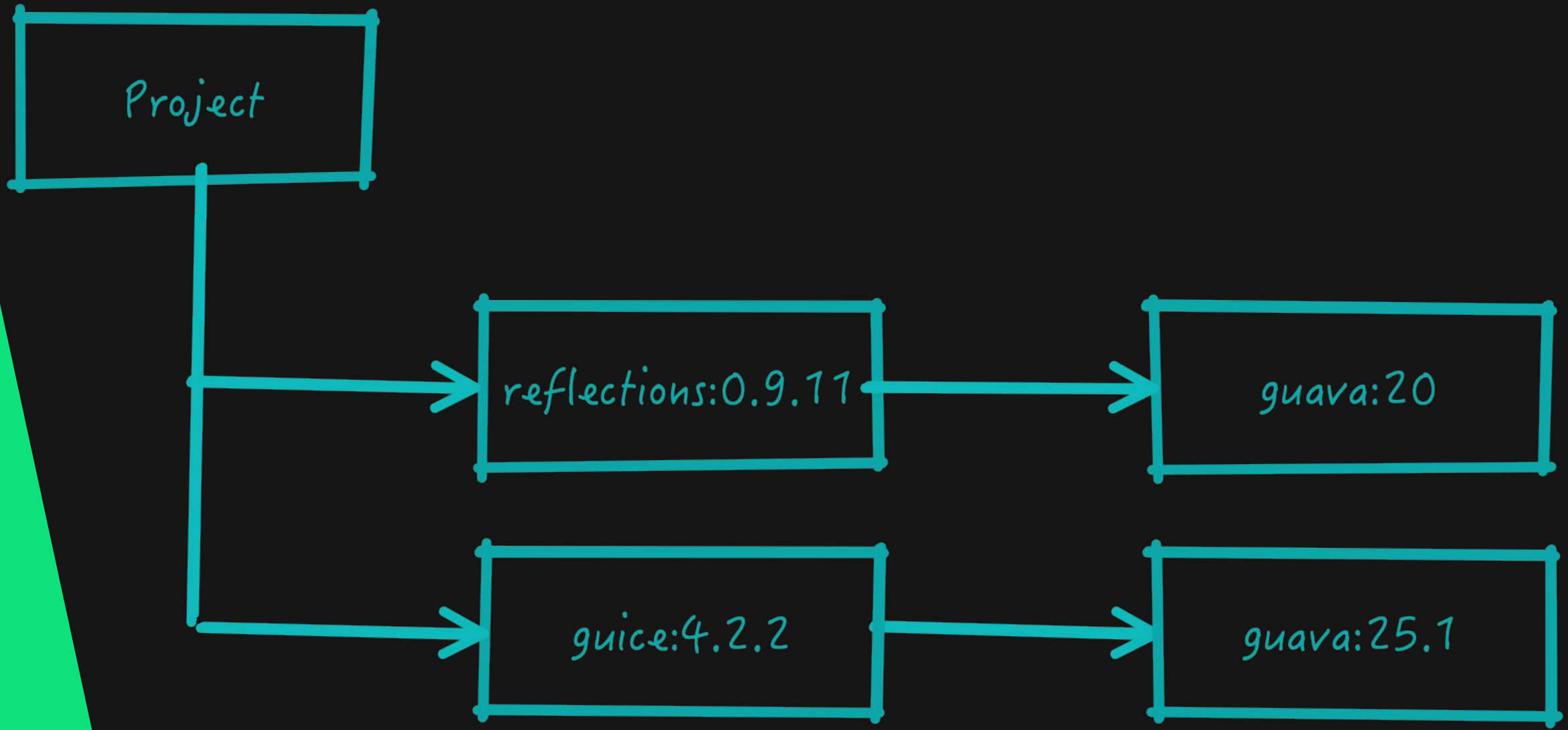


/ Résolution et conflits

CODE



CODE



/ GLOBAL GUIDE

Déclaration et résolution de dépendances

/ Node / NPM

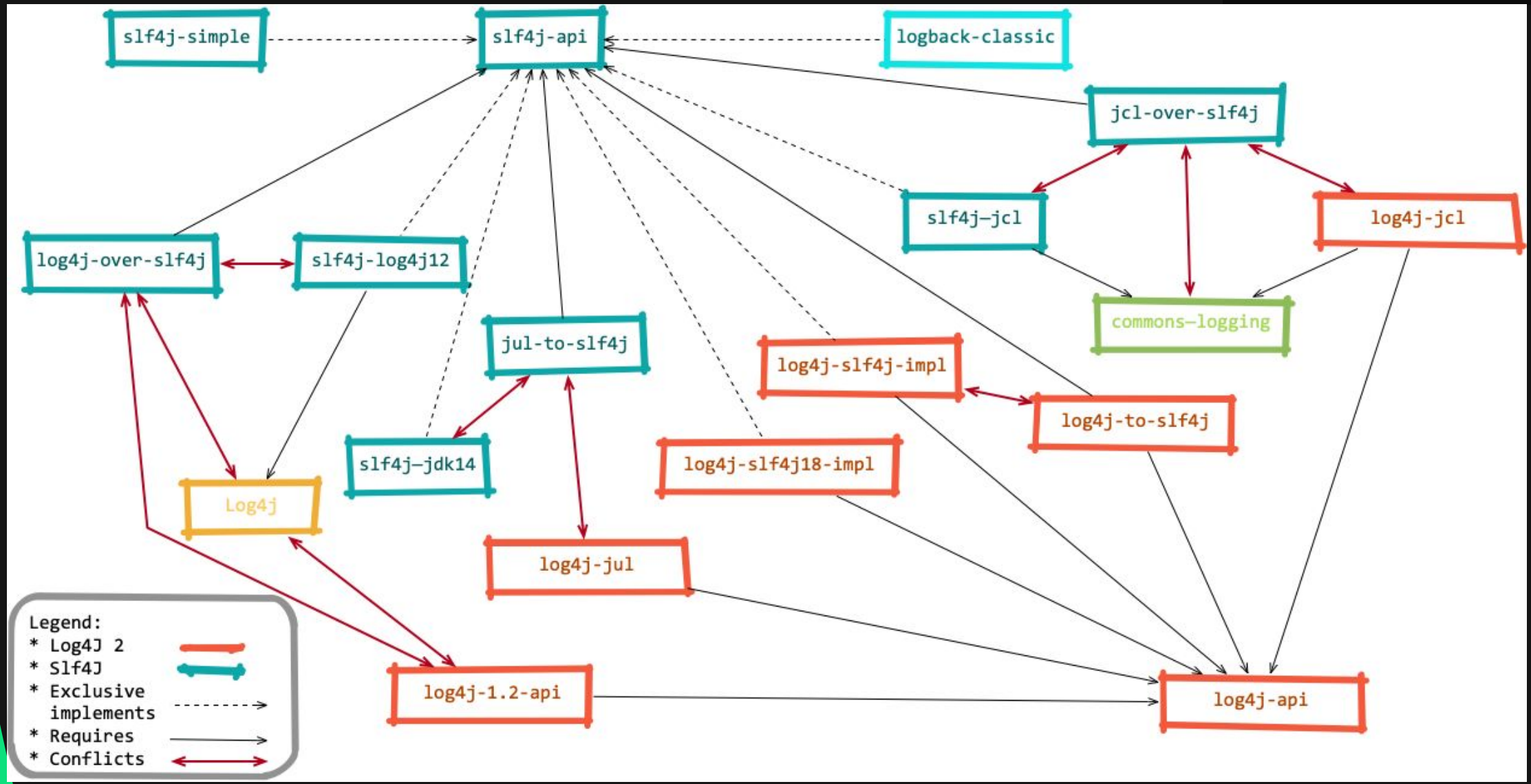
- Supporte plusieurs versions d'un package
  - Pas d' "effet classpath"
- Déduplication avec dedupe

Déclaration et résolution de dépendances

/ Gradle

- Résolution de conflits d'implémentation
- Possibilité de dire qu'une dépendance a la même `capability` qu'une autre
  - Détection de conflits de loggers
  - Détection de conflits d'implémentation d'APIs
  - Et plus ...
  - Example: https://github.com/gradlex-org/jvm-dependency-conflict-resolution

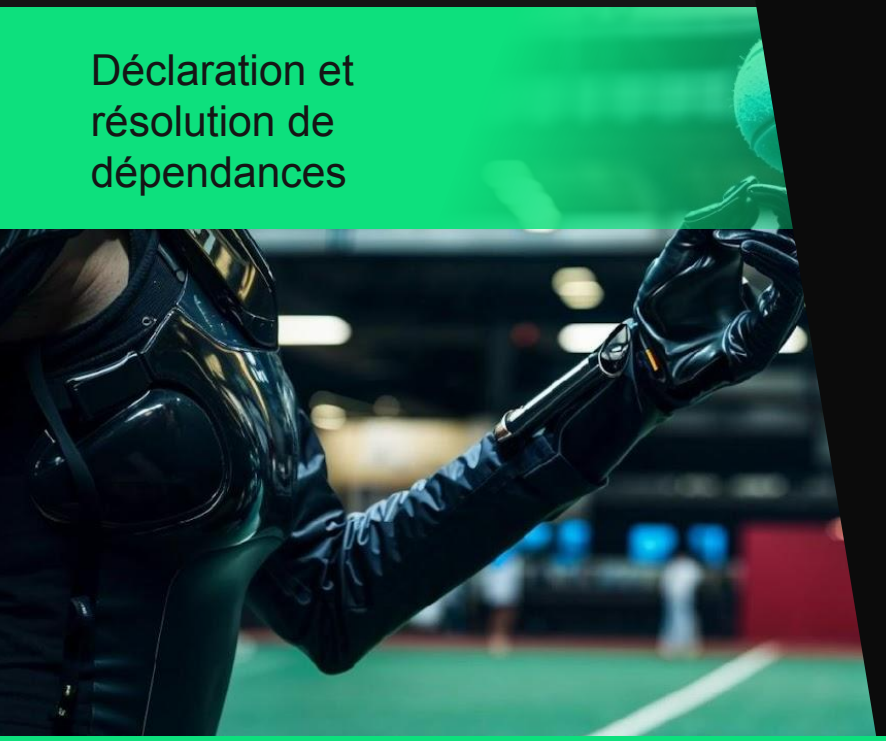
CODE


slf4j-simple
slf4j-api
logback-classic
jcl-over-slf4j
slf4j-jcl
log4j-jcl
log4j-over-slf4j
slf4j-log4j12
commons-logging
jul-to-slf4j
log4j-slf4j-impl
log4j-to-slf4j
slf4j-jdk14
log4j-slf4j18-impl
Log4j
log4j-jul
log4j-1.2-api
log4j-api
Legend:
* Log4J 2
* Slf4J
* Exclusive implements
* Requires
* Conflicts

Gestion des mises à jour

/ GLOBAL GUIDE

Déclaration et résolution de dépendances



/ Introspection et analyse

/ GLOBAL GUIDE

Déclaration et résolution de dépendances

/ Maven

- `dependency:list`, `dependency:tree` (`-Dverbose`)
- `dependency:analyze`

/ Gradle

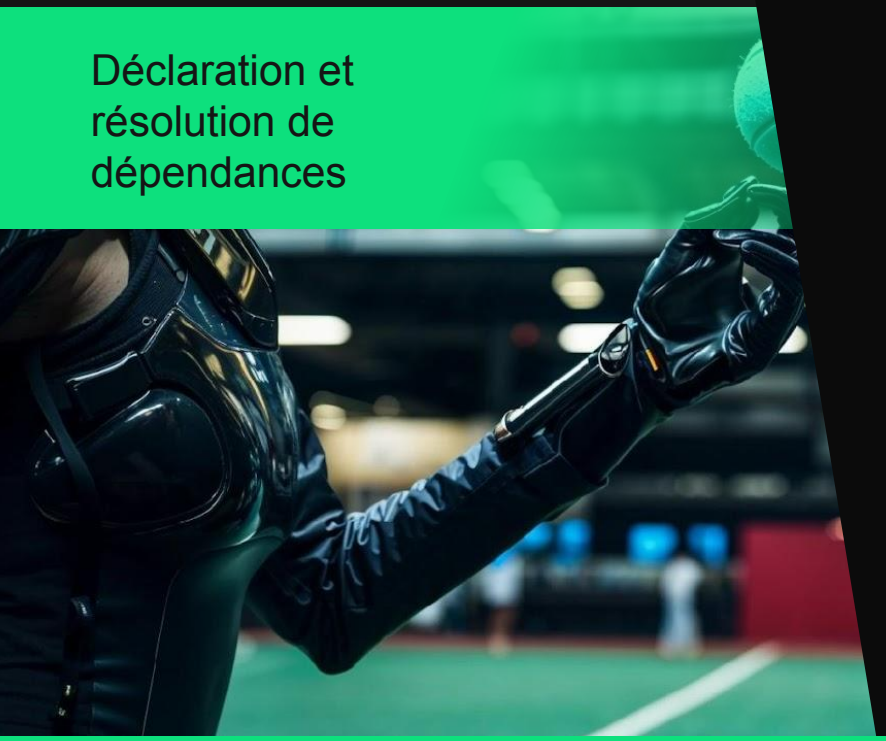
- `dependencies`, `dependencyInsight`
- Plugin externe pour l'analyse (`com.autonomousapps.dependency-analysis`)

/ Npm

- `ls` (`--all`)
- Plugin externe pour l'analyse (`depcheck`)

/ GLOBAL GUIDE

Déclaration et résolution de dépendances



/ Mises à jour

/ GLOBAL GUIDE

Déclaration et résolution de dépendances

/ Node / NPM

- le fichier de lock référence les dépendances effectives du projet
  - range déclaré
  - version sélectionnée
- Facilite le travail les outils de mise à jour tiers

Permet de l'automatisation poussée, avec un outillage performant et efficace:

- Dependabot
- Renovate
- et autres ...

Déclaration et résolution de dépendances

/ Gradle

- Différents DSLs, options de notation
  - seul Gradle comprend vraiment
- Fichier de build et catalogue de version
  - Approximation

Complique l'automatisation

- Mais des solutions existent pour la détection!
  - https://blog.gradle.org/gradle-github-partnership-supply-chain-security
- La mise à jour par PR automatique reste un challenge

CODE

/ Dependency Graph
gradle/gradle

Dependency graph

Dependencies | Dependents | Dependabot

Export SBOM

Search all dependencies

634 Total

com.esotericsoftware.kryo:kryo `2.24.0`
Detected by **GitHub Dependency Graph Gradle Plugin** on Apr 18, 2024 (Maven) · settings.gradle.kts · BSD-2-Clause

com.esotericsoftware.minlog:minlog `1.2`
Detected by **GitHub Dependency Graph Gradle Plugin** on Apr 18, 2024 (Maven) · settings.gradle.kts · BSD-2-Clause

com.fasterxml.jackson.core:jackson-annotations `2.12.7`
Detected by **GitHub Dependency Graph Gradle Plugin** on Apr 18, 2024 (Maven) · settings.gradle.kts · Apache-2.0

com.fasterxml.jackson.core:jackson-annotations `2.16.1`
Detected by **GitHub Dependency Graph Gradle Plugin** on Apr 18, 2024 (Maven) · settings.gradle.kts · Apache-2.0

com.fasterxml.jackson.core:jackson-core `2.12.7`
Detected by **GitHub Dependency Graph Gradle Plugin** on Apr 18, 2024 (Maven) · settings.gradle.kts · Apache-2.0

https://github.com/gradle/gradle/network/dependencies

Déclaration et résolution de dépendances

/ Maven

- XML, format déclaratif, facile à parser
- `dependencies`, `dependencyManagement`, parent POM download & heritage, interpolation `${...}`, …
  - Seul Maven comprend vraiment!
  - dans votre environnement

Automatisation

- Souvent fait à base de parsing direct `pom.xml`
- Attention aux approximations!
  - Pour la détection
  - Et la mise à jour automatique

CODE

/ Dependency Graph
apache/maven

Dependency graph

Dependencies | Dependents

Export SBOM

Search all dependencies

936 Total

com.github.siom79.japicmp:japicmp-maven-plugin
Detected automatically on Apr 12, 2024 (Maven) · maven-model/pom.xml

org.apache.maven:maven-api-impl
Detected automatically on Apr 12, 2024 (Maven) · maven-model/pom.xml

org.apache.maven:maven-api-model
Detected automatically on Apr 12, 2024 (Maven) · maven-model/pom.xml

org.apache.maven:maven-xml-impl
Detected automatically on Apr 12, 2024 (Maven) · maven-model/pom.xml

org.codehaus.modello:modello-maven-plugin `4.1.0`
Detected automatically on Apr 12, 2024 (Maven) · maven-model/pom.xml

https://github.com/apache/maven/network/dependencies

SBOM, la solution?

/ GLOBAL GUIDE

SBOM

/ SBOM: Software Bill of Materials

Inventaire des composants logiciels

National Telecommunications and Information Administration
United States Department of Commerce

A “Software Bill of Materials” (SBOM) is a nested inventory for software, a list of ingredients that make up software components.

Objectives:

- Software Transparency
  - dev/ops/sec/management/business/...
  - client/fournisseur
  - expertise silos: Maven/Gradle/npm/Docker/Linux/...
- Software Management Best Practices
  - regulation/policy/compliance
  - formats for automation

CycloneDX SPDX

/ GLOBAL GUIDE

CycloneDX

/ CycloneDX format overview

Metadata	Supplier	Authors	Component		
	Manufacturer	Tools	Lifecycles		
Components	Supplier	Identity	Pedigree	Provenance	Evidence
	Component Type	Licenses	Hashes	Release Notes	Relationships
Services	Provider	Data Classification	Trust Zone		
	Endpoints	Data Flow	Relationships		
Dependencies	Components	Services			
Compositions	Completeness of:				
	Components	Services	Dependencies	Vulnerabilities	
Vulnerabilities	Details	Source	Exploitability (VEX)	Targets Affected	Proof of Concept
	Advisories	Risk Ratings	Evidence	Version Ranges	Recommendations
Formulation	Declared	Formulas	Tasks	Components	
	Observed	Workflows	Steps	Services	
Annotations	Per Person	Per Organization	Per Tool		
	Details	Timestamp	Signature		
Definitions	Standards	Requirements	Levels		
Declarations	Attestations	Evidence	Conformance	Mitigation Strategies	Assessors
	Claims	Counter Evidence	Confidence	Signatories	Signatures
Extensions	Properties	Per Organization	Per Team		
	Formal Taxonomy	Per Industry	...		

- XML
- json
- protobuf

Release History

VERSION	RELEASE DATE
CycloneDX 1.6	09 April 2024
CycloneDX 1.5	26 June 2023
CycloneDX 1.4	12 January 2022
CycloneDX 1.3	04 May 2021
CycloneDX 1.2	26 May 2020
CycloneDX 1.1	03 March 2019
CycloneDX 1.0	26 March 2018
Initial Prototype	01 May 2017

/ GLOBAL GUIDE

CycloneDX

/ CycloneDX for SBOM

Metadata	Supplier	Authors	Component		
	Manufacturer	Tools	Lifecycles		
Components	Supplier	Identity	Pedigree	Provenance	Evidence
	Component Type	Licenses	Hashes	Release Notes	Relationships
Vulnerabilities	Details	Source	Exploitability (VEX)	Targets Affected	Proof of Concept
	Advisories	Risk Ratings	Evidence	Version Ranges	Recommendations

Inventaire par build tool ou autre (attention aux approximations)

Inventaire par security tool

Regulation: “no vulnerability higher than ?”, VEX

/ GLOBAL GUIDE

xBOM

/ xBOM: x Bill of Materials

Inventaire des x

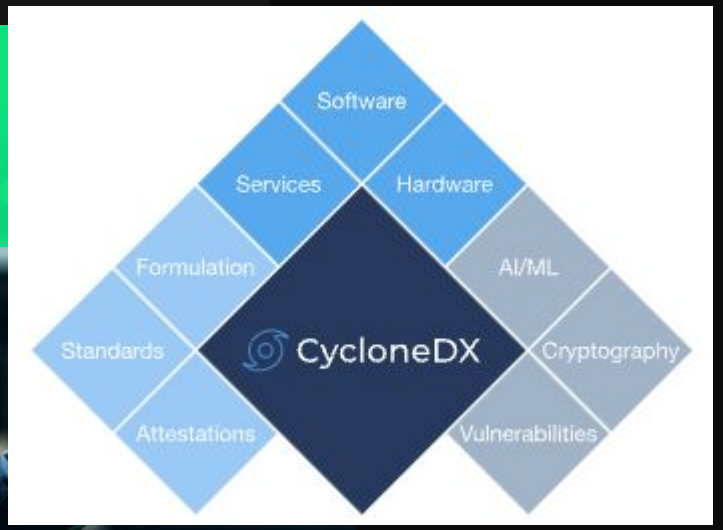
Objectives:

- x Transparency
- x Management Best Practices

/ GLOBAL GUIDE

CycloneDX xBOM

/ xBOM by CycloneDX



- SBOM: Software
- BOV: Vulnerabilities
- SaaSBOM: services, endpoints, data flows, ...
- OBOM: Operations, runtime, configurations, ...
- ML-BOM: Machine Learning, model, dataset, privacy, ...
- CBOM: Cryptography, prepare for quantum...
- ...

Gestion des **dépendances**

formats / outils

github = “ljacomet”

<github>hboutemy</github>

Merci

pour votre attention

DEVOXX FRANCE 2024