2021.11.17 - Automating dependencies management for fun and profit

Virtual Tech Meetup - Romain Aviolat

Agenda

- About me
- Dependencies management / Software lifecycle / SSDLC
- Why it's important to do it properly
- How to automate using bots
- Demo
- How we're doing it at Kudelski Security
- Questions

Feel free to ask questions directly in the chat during the presentation !

KUDELSKI SECURITY

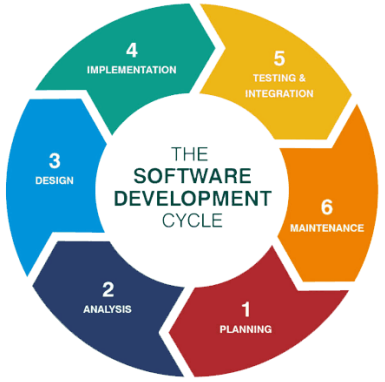
R&D Services

Me

- Romain Aviolat
- Cloud & Security Principal Kudelski-Security Eng dept.
- CloudSec / AppSec
- Love to automate stuff
- Open-source advocate

KUDELSKI SECURITY

R&D Services



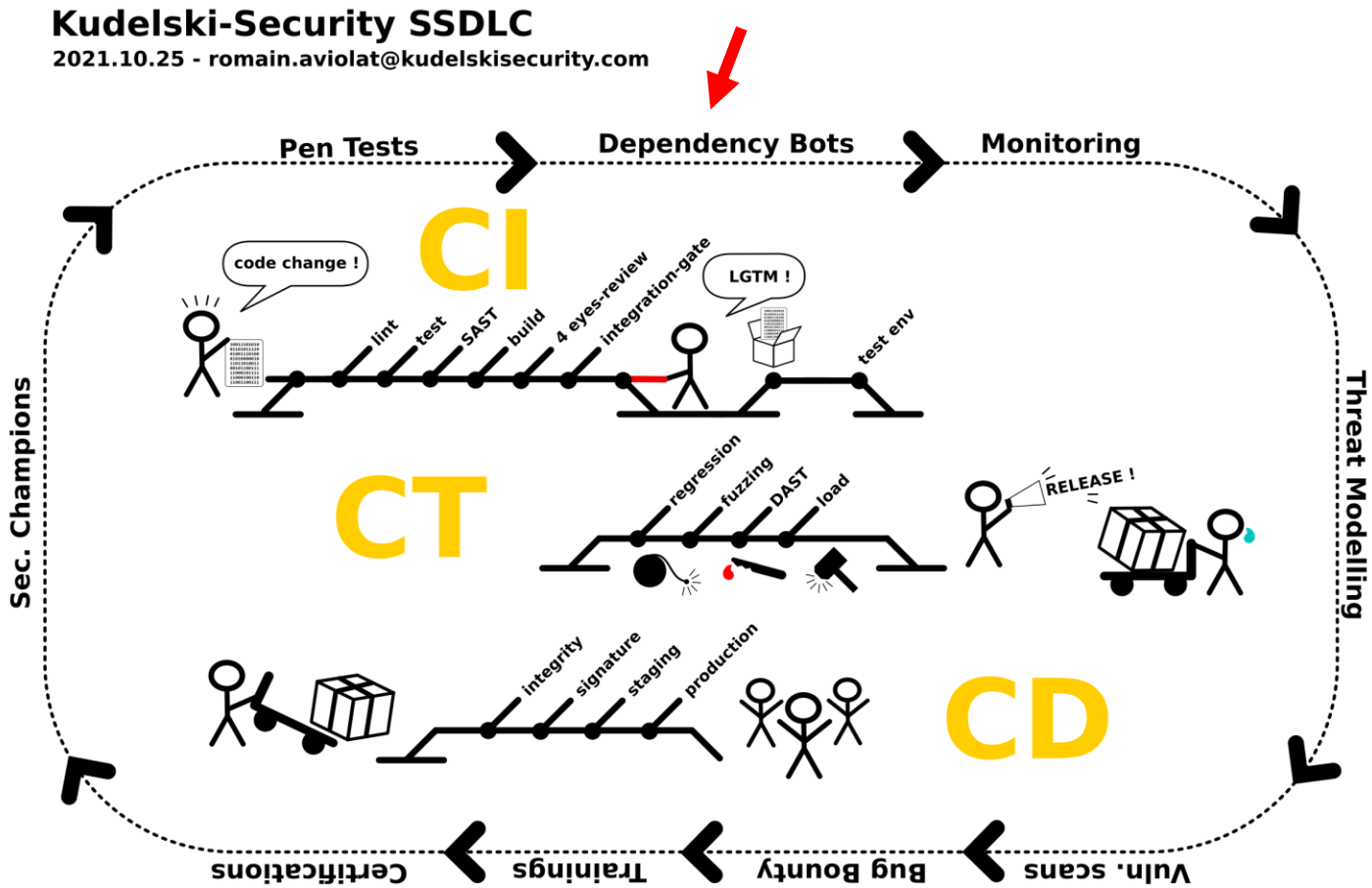
Secure Software Development Life-Cycle

KUDELSKI SECURITY

R&D Services

KS SSDLC

- 360° approach to security
- Still a WIP, but that's the target
- Integrate Security inside each phase of the software development
- Dependency management is just one part of our SSDLC strategy



KUDELSKI SECURITY

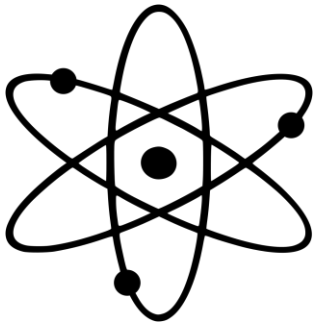
R&D Services

Just to make sure we're on the same page

- Dependencies management is about managing dependencies!
- In your code (software, infrastructure), or more generally between components.
- Maybe you are not doing it today, and that's fine!
- We're here to learn :)

KUDELSKI SECURITY

R&D Services



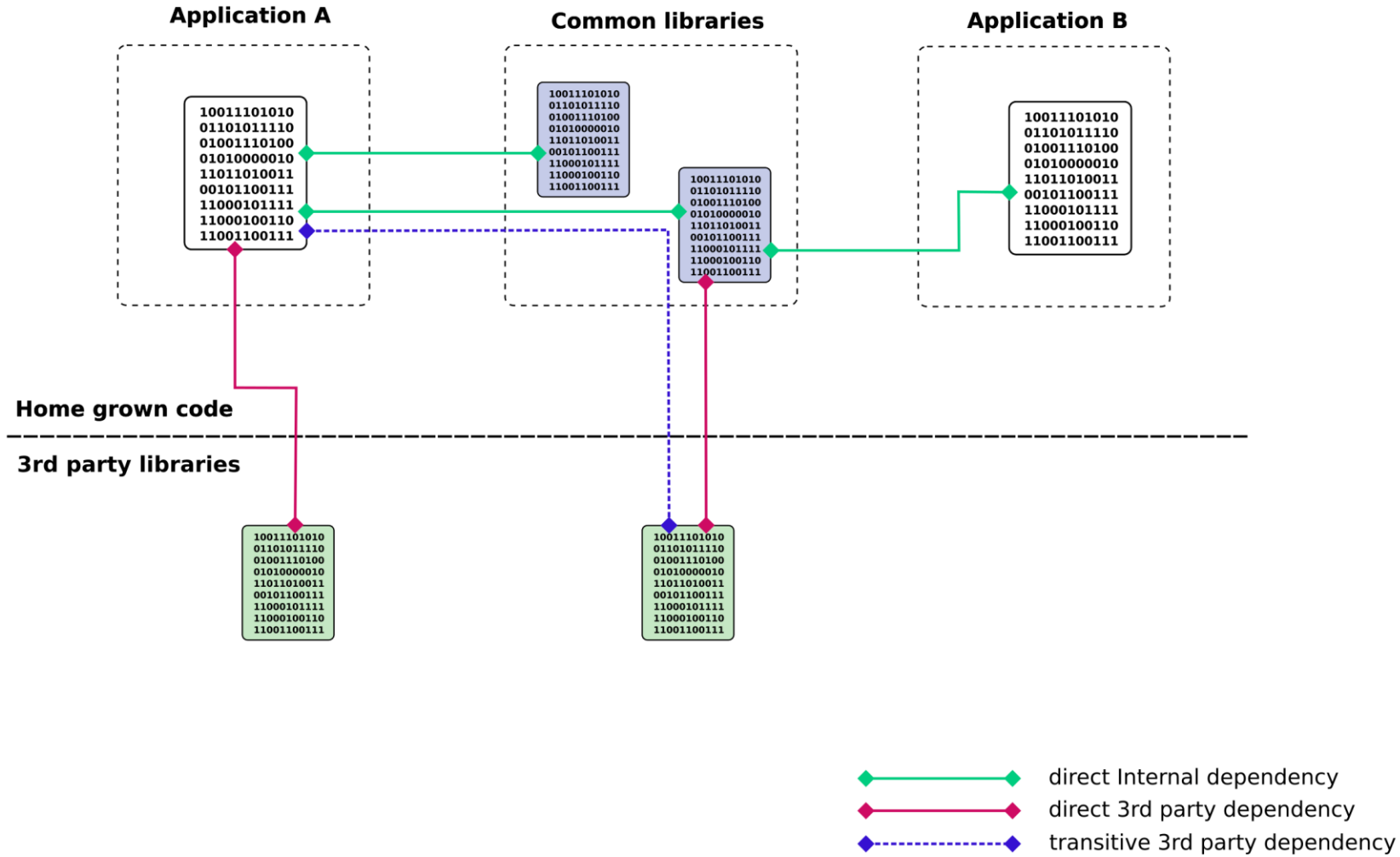
A bit of theory ...

KUDELSKI SECURITY

R&D Services

Software dependencies

- An application is composed of many external components (librari es, modules, …)
- Direct vs transitive dependencies
- Some dependencies are not under your control
- 3rd party libraries are not always Open-Sources libraries, but we'll use both terms interchangeably in those slides

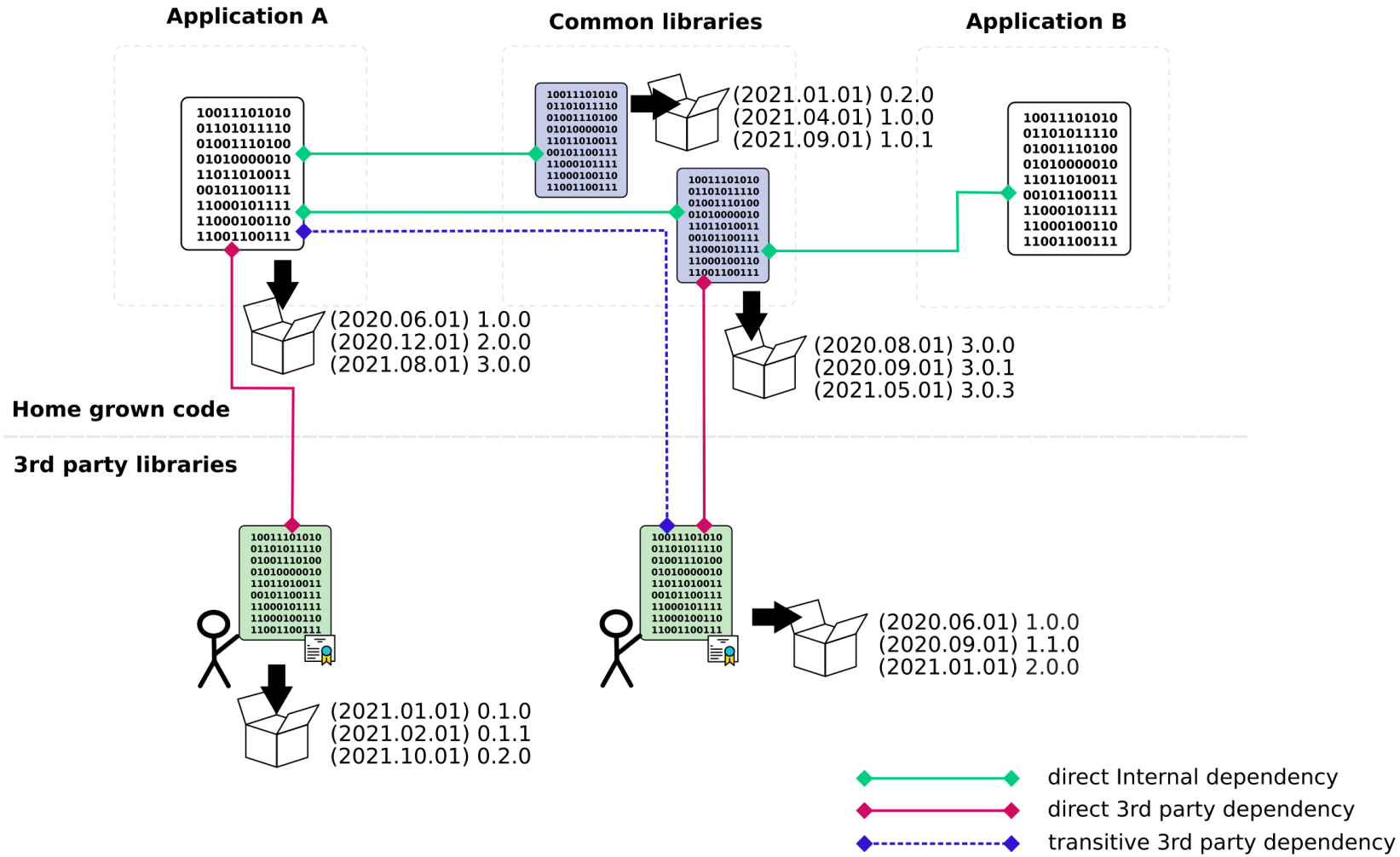


KUDELSKI SECURITY

R&D Services

Software dependencies

- Each component has its own life cycle
- SemVer used to track version
- 3rd party libraries are not under our control
  - Own maintainers
  - Own licenses

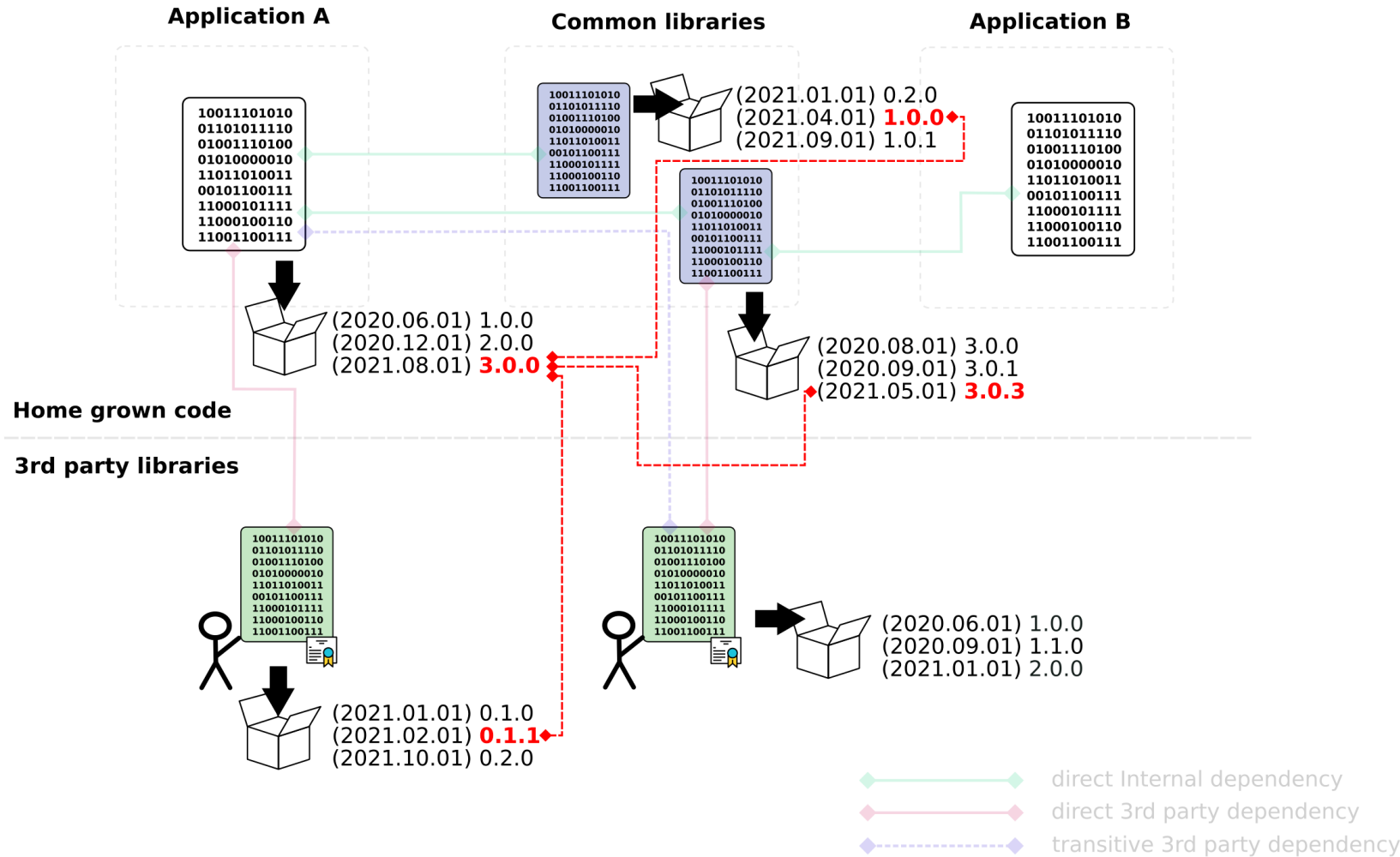


KUDELSKI SECURITY

R&D Services

Software dependencies

- The application Software Bill of Material is the sum of its dependencies and versions (and their licenses)
- Dependencies management is about managing the life cycle of those dependencies

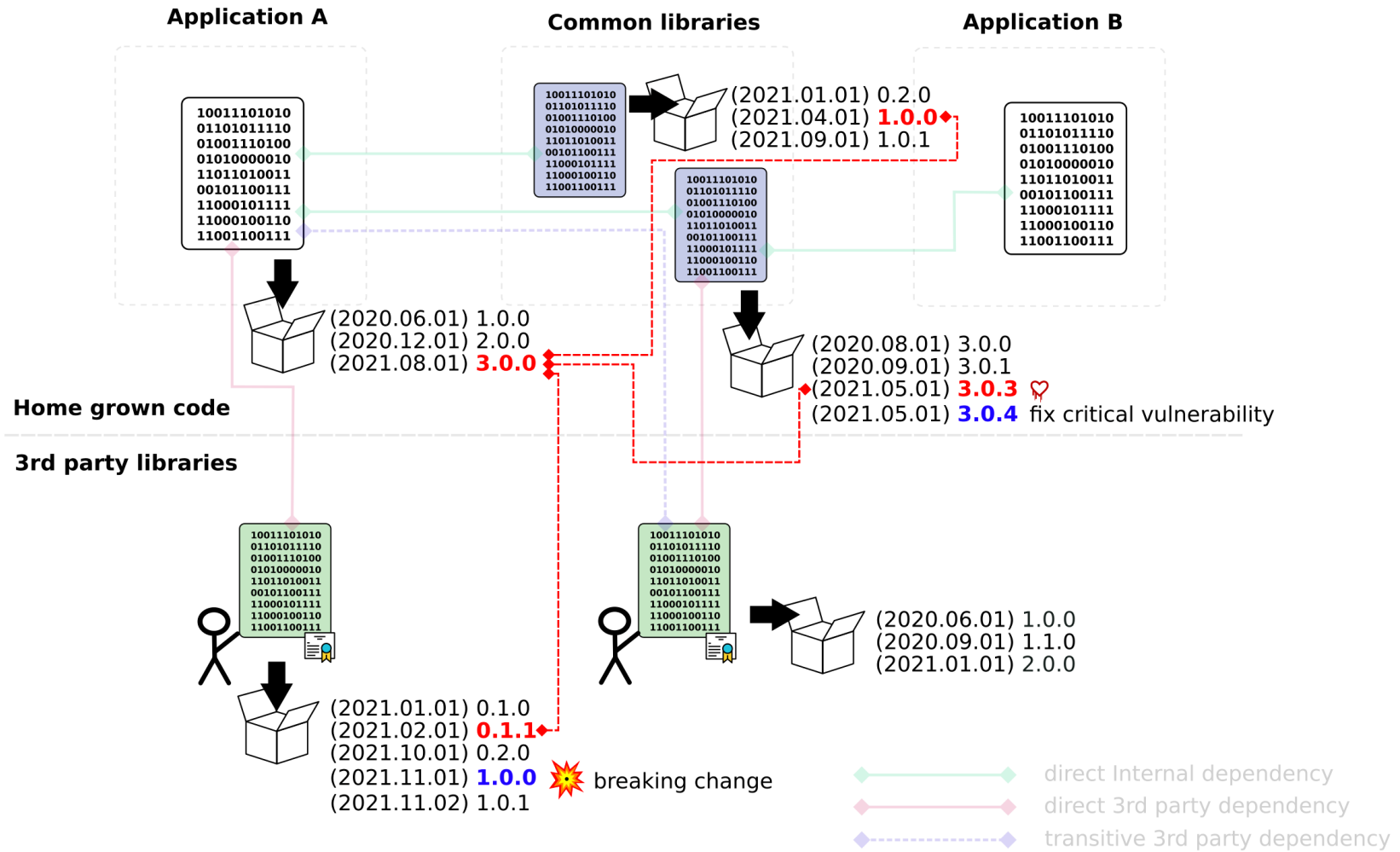


KUDELSKI SECURITY

R&D Services

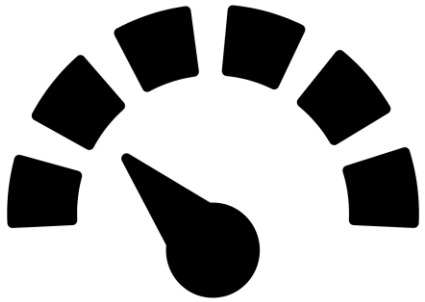
Software dependencies

- New releases will come all the time (sometimes at a crazy pace)
- Containing
  - security fixes
  - improvements / fixes
  - New features
  - breaking changes
  - (sometimes new bugs)
- Do it regularly otherwise the application will become harder to maintain and less secure over time



KUDELSKI SECURITY

R&D Services



Metrics !

KUDELSKI SECURITY

R&D Services

Metrics from AppSec vendors

- Data coming from AppSec Vendors (Veracode, WhuiteSource, Contrast, Snyk, …)
- Create statistics based on metrics coming from their tools (from clients).
- Gives good insights in terms of AppSec (links at the end of the slides)

snyk

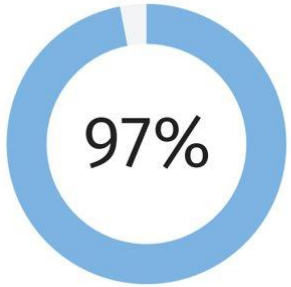
CONTRAST SECURITY

WhiteSource

VERACODE

118 | The average application contains **118** open-source libraries.

> Virtually no modern application can avoid including open source libraries that provide functionality that would be difficult or time-consuming to write from scratch.



OPEN SOURCE

Open source libraries can be a significant cause for concern. For example, 97 percent of the typical Java application is made up of open source libraries.

KUDELSKI SECURITY

R&D Services

Metrics from AppSec vendors

- Third Party libraries are not less secure than home-grown code
- There's just many more code coming from them
- Entropy increases over time
- Conclusion, release often and bump your dependencies versions!

2.6 The average library uses a version that is **2.6** years old.

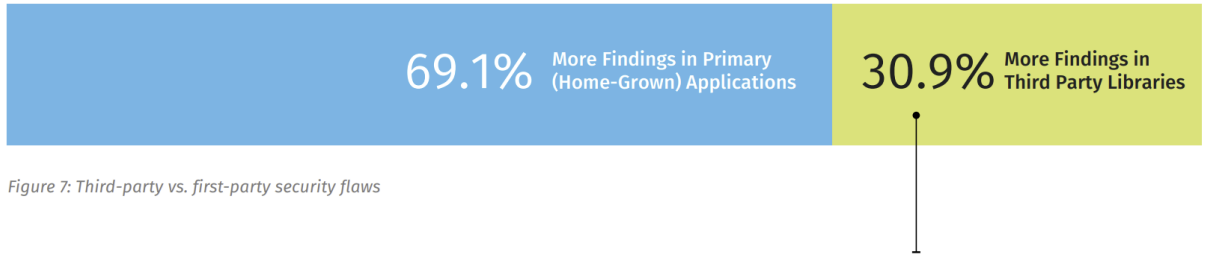


Figure 7: Third-party vs. first-party security flaws

KEY LESSON

Software security comes from getting the whole picture, and that means identifying and tracking the third-party open source libraries used in your applications.

50 The average Java application has **50** open-source vulnerabilities.

16% Java libraries in applications have a **16%** chance of having a Critical or Major vulnerability.

44% The odds of an application having a vulnerability in a Java library increase from **7%** to **44%** as the library age goes from 1 year to 4 years..

KUDELSKI SECURITY

R&D Services

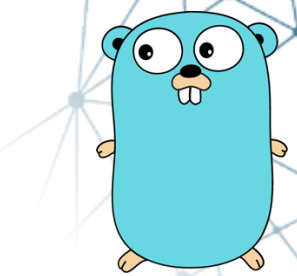
Kudelski Security Engineering

KUDELSKI SECURITY

R&D Services

Kudelski Security Engineering

- Building the products and infrastructure that supports the business
- 60 developers + SRE engineers (CH, IN, US)
- Multiple Cross-functional, product / pizza teams
- >160 Git repositories (Applications, Infrastructure code, OPs automation, security)
- Java / Kotlin, JS / TS REACT, Golang, Python
- Infra-as-Code (IaC), Terraform, Helm, Docker
- Agile / DevSecOps / Scrum
- Applications are all containerized and running on k8s

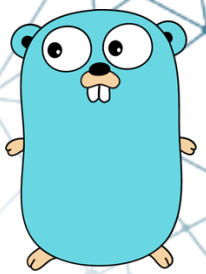


KUDELSKI SECURITY

R&D Services

Problems we wanted to solve

- Lots of dependencies to maintain + not properly tracked
- Outdated libraries on certain components
- With the rise of Infrastructure-as-Code and containerization the problem was amplified, more components, more dependencies, more teams, more projects, … .



KUDELSKI SECURITY

R&D Services

Dependency bots

KUDELSKI SECURITY

R&D Services

Dependency bots

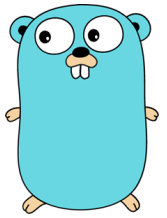
- Bot are everywhere (code, dependencies, tidy, grooming, …)
- Help you manage your software dependencies
- Integrate inside your workflow and tooling
- Submit Pull / Merge requests inside your projects
- **Won't** change the code but the code dependencies
- Renovate, Dependabot, Snyk (security only)
- Less Ops, more security !
- We'll focus on Renovate today

KUDELSKI SECURITY

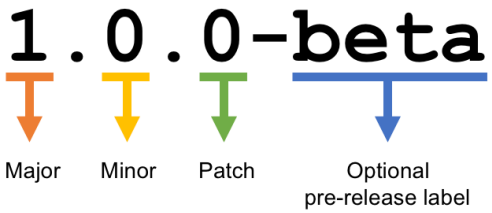
R&D Services

Renovate

- Open-source project (https://github.com/renovatebot) (WhiteSource)
- Very active (multiple releases **every** day)
- Support main DevOps platform
- Main languages / package managers
- Auto-detects the language
- Scheduling
- Highly configurable (regexp, languages, packages, …)
- State-less



29.1.0 published 10 days ago
29.2.0 published 10 days ago
29.2.1 published 9 days ago
29.2.2 published 9 days ago
29.2.3 published 9 days ago
29.2.4 published 9 days ago
29.2.5 published 9 days ago
29.2.6 published 8 days ago
29.3.0 published 7 days ago
29.3.1 published 7 days ago
29.3.2 published 6 days ago
29.3.3 published 6 days ago
29.3.4 published 6 days ago
29.3.5 published 5 days ago
29.4.0 published 5 days ago
29.4.1 published 5 days ago
29.4.2 published 5 days ago
29.4.3 published 4 days ago
29.5.0 published 3 days ago
29.6.0 published 3 days ago
29.6.1 published 3 days ago
29.7.0 published yesterday
29.7.1 published 23 hours ago
29.7.2 published 18 hours ago
29.8.0 published 18 hours ago
29.8.1 published 14 hours ago
29.8.2 published 12 hours ago
29.8.3 published 7 hours ago



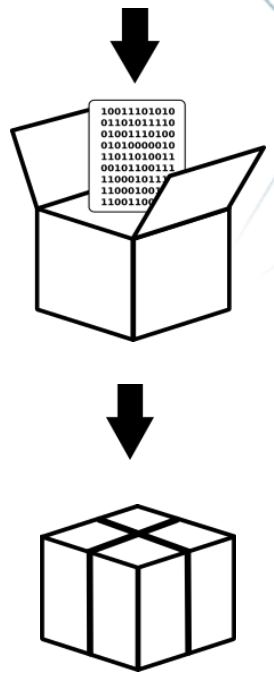
Pre-requisites

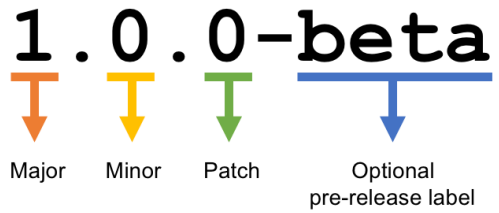
KUDELSKI SECURITY

R&D Services

Automation requirements

- Releases must be properly versioned to automate this process
- Out-of-the box for public repositories
- (if you have a private artifact registry, Artifactory / Nessus).
- SemVer is the DeFacto way to do it https://semver.org, human + Machine readable
- Major, Minor, Patch





KUDELSKI SECURITY

R&D Services

Versioning or not?

- Is it mandatory? What if I always include the latest version of all my dependencies without tracking them?

- If you don't you can't guarantee that your application will work the same way or that you're not introducing issues every time you rebuild it.
- Valid for everything that is packaged or compiled.

- It's a question of process maturity. Being able to create "Reproducible" builds should be the target.

```
1  FROM alpine:latest
2
3  LABEL maintainer "Romain Aviolat"
4  LABEL comment    "Versioning is for cowards !"
5
6  RUN apk add --update --no-cache python3 bash curl git file
7  RUN pip install --upgrade pip
8  RUN pip install awscli boto3
9  COPY ./my-app.py /my-app.py
10
11 ENTRYPOINT [ "python3", "/my-app.py"]
12
```

Reproducible	Re-running the build steps with identical input artifacts results in bit-for-bit identical output. Builds that cannot meet this MUST provide a justification why the build cannot be made reproducible. "○" means that this requirement is "best effort". The user-provided build script SHOULD declare whether the build is intended to be reproducible or a justification why not. The build service MAY blindly propagate this intent without verifying reproducibility. A consumer MAY reject the build if it does not reproduce.				○

SLSA ("salsa") is Supply-chain Levels for Software Artifacts

SLSA (pronounced "salsa") is security framework from source to service, giving anyone working with software a common language for increasing levels of software security and supply chain integrity.

The best way to read about SLSA is to visit slsa.dev.

The fun way to get a taste of SLSA is to check out the Operation SLSA videos.

What's in this repo?

The primary content of this repo is the docs/ directory, which contains the core SLSA specification and sources to the slsa.dev website.

You can read SLSA's documentation here:

- Levels (Defining the framework)
- Requirements (How to attain compliance)
- Use cases
- Our roadmap

KUDELSKI SECURITY

Pinning versions

- Use language-specific package managers
  - go modules
  - python pip
  - java maven
  - …)
- That's what the Bots will read
- Generates your Software Bill Of Material (SBOM)

go.mod 944 Bytes

```
module k8s-tunnels

go 1.16

require (
        github.com/AlecAivazis/survey/v2 v2.3.1
        github.com/Azure/azure-sdk-for-go v55.8.0+incompatible
        github.com/Azure/go-autorest/autorest v0.11.18 // indirect
        github.com/Azure/go-autorest/autorest/azure/auth v0.5.8
        github.com/Azure/kubelogin v0.0.10
        github.com/aws/aws-sdk-go v1.40.38
        github.com/aws/aws-sdk-go-v2/config v1.6.1 // indirect
        github.com/cloudflare/cloudflared v0.0.0-20210113175915-a2109e4a7894
        github.com/hashicorp/vault-plugin-auth-jwt v0.10.1
        github.com/hashicorp/vault/api v1.1.1
                                                    1.0
                                                    3.0 // indirect
                                                    1.0.3 // indirect
```

pom.xml 9.12 KB

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/200
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd,
  <modelVersion>4.0.0</modelVersion>

  <properties>
    <cfc-report-client.version>5.9.1</cfc-report-client.version>
    <snow-dsl.version>3.9.4</snow-dsl.version>
    <cfc-iam-internal-api.version>5.1.4</cfc-iam-internal-api.version>
    <cfc-edr-client.version>1.2.1</cfc-edr-client.version>
    <cfc-user-data-client.version>1.2.0</cfc-user-data-client.version>
    <cfc-vulnerability-client.version>1.2.8</cfc-vulnerability-client.version>
    <customer-stats-api.version>3.3.11</customer-stats-api.version>
    <java-jwt.version>3.18.2</java-jwt.version>
    <cfc-grpc-util.version>1.3.6</cfc-grpc-util.version>
    <grpc-spring-boot.version>4.5.8</grpc-spring-boot.version>
    <assertk-jvm.version>0.25</assertk-jvm.version>
    <springmockk.version>3.0.1</springmockk.version>
    <reactor-kotlin-extensions.version>1.1.5</reactor-kotlin-extensions.version>
  </properties>
```

requirements.txt 147 Bytes

```
attrs==19.1.0
configparser~=5.0.2
fusion-tools==2.*
jsonpath-ng==1.5.2
pytest==4.6.11
pytest-forked==1.0.2
pytest-xdist==1.26.0
requests~=2.25.1
```

KUDELSKI SECURITY

R&D Services

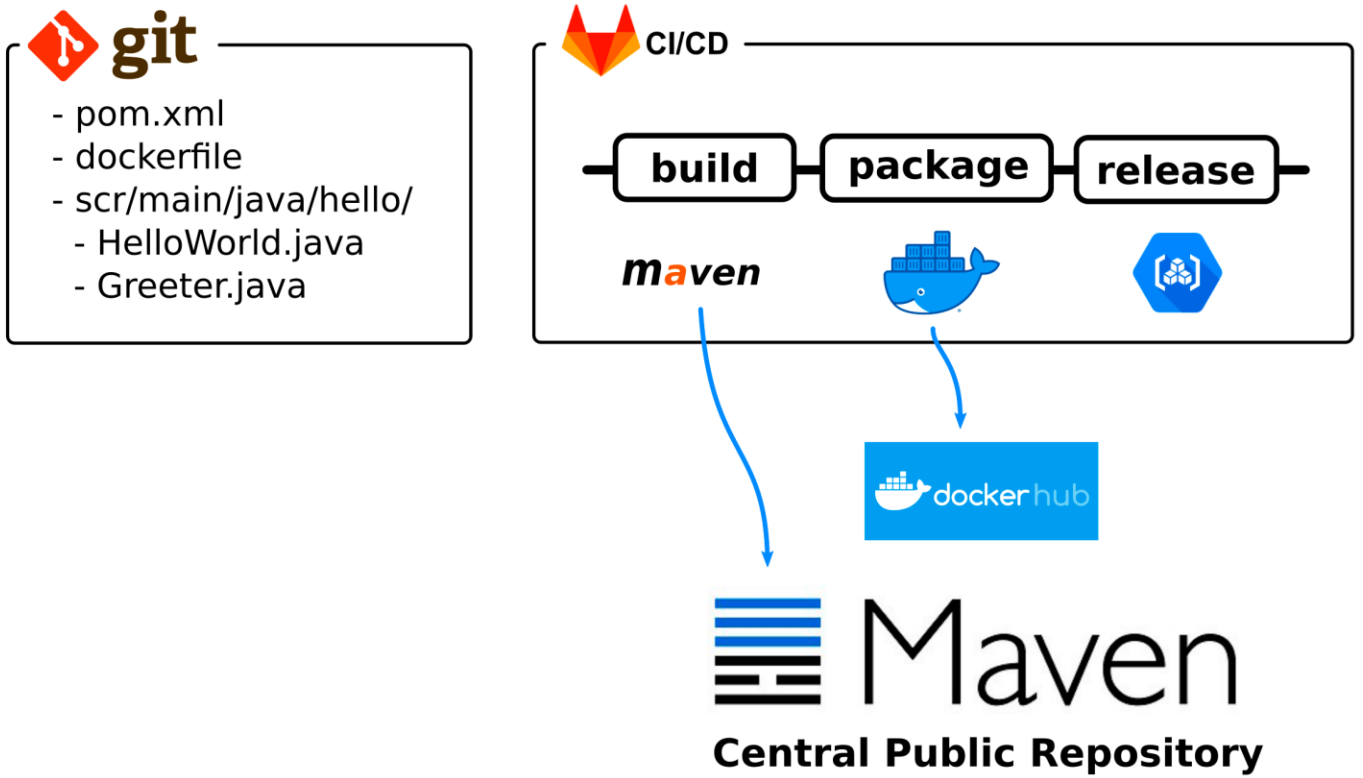
Use-case 1 – Software dependencies

KUDELSKI SECURITY

R&D Services

Use-case 1

- Java application
- Uses Maven for packaging
- Maven Central public repository to fetch open-source libraries
- Docker to package the application
- Simple build / release pipeline

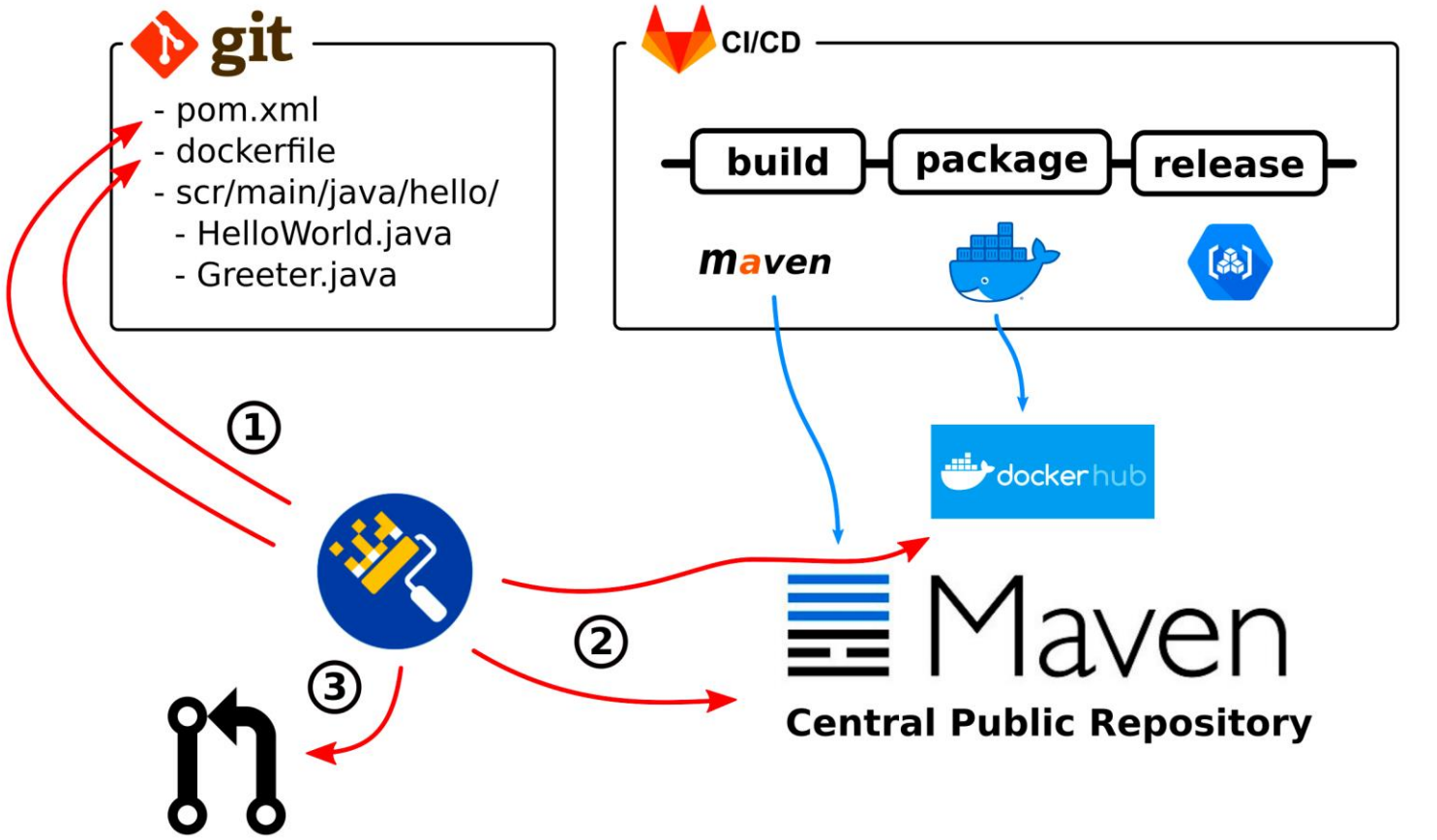


KUDELSKI SECURITY

R&D Services

Use-case 1

- Renovate bot configured on the Git repo
- It'll scan for known package-managers (or configured ones)
- Check if new versions are present
- Submit a pull/merge-request
- Trigger sub-sequent CI/CD pipelines



KUDELSKI SECURITY

R&D Services

Renovate output

pom.xml 1012 Bytes Edit

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.springframework</groupId>
    <artifactId>hello-maven</artifactId>
    <packaging>jar</packaging>
    <version>0.1.0</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <build>
        <plugins>
            <plugin>
                <groupId>com.rabbitmq.jms</groupId>
                <artifactId>rabbitmq-jms</artifactId>
                <version>2.1.1</version>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-shade-plugin</artifactId>
                <version>3.2.4</version>
            </plugin>
        </plugins>
    </build>
</project>
```

Direct dependency in our code

Open Created 18 minutes ago by svc-renovate-bot Developer 0 of 1 task completed Edit Mark as draft

chore(deps): update dependency com.rabbitmq.jms:rabbitmq-jms to v2.3.0

Overview 0 Commits 1 Pipelines 1 Changes 1

This MR contains the following updates:

Package	Update	Change
com.rabbitmq.jms:rabbitmq-jms (source)	minor	2.1.1 -> 2.3.0

Renovate detected it

Fetched the releases info from Maven central

Release Notes

▼ rabbitmq/rabbitmq-jms-client

`v2.3.0`

Compare Source

Changes between 2.2.0 and 2.3.0

This is a maintenance release that includes a new feature and dependency upgrades. It is backward-compatible with 2.2.0. All users are encouraged to upgrade.

Add option to requeue message if processing takes too long

GitHub MR: #137

Bump dependencies

GitHub issue: #138

`v2.2.0`

Compare Source

Changes between 2.1.0 and 2.2.0

This is a maintenance release that includes a security fix and dependency upgrades. It is backward-compatible with 2.1.0. All users are encouraged to upgrade. User of 1.x should especially consider upgrading, as 1.x extended support period ends on 31 December 2020.

Included the release note that mentions a vulnerability fix

Direct Vulnerabilities

Known vulnerabilities in the com.rabbitmq.jms:rabbitmq-jms package. This does not include vulnerabilities belonging to this package's dependencies.

Report new vulnerabilities

VULNERABILITY	VULNERABLE VERSIONS	SNYK PATCH	PUBLISHED
C Deserialization of Untrusted Data	[,1.15.2),[2.0.0,2.2.0)	Not available	12 Mar, 2021
M Deserialization of untrusted data	(,1.5.0)	Not available	18 Jul, 2016

Versions

Show all versions

VERSION	PUBLISHED	LICENSES	DIRECT VULNERABILITIES
com.rabbitmq.jms:rabbitmq-jms 2.3.0 LATEST	05 May, 2021	Apache-2.0 OR MPL-2.0	0 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 1.15.2	03 Nov, 2020	Apache-2.0 OR MPL-1.1	0 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 2.2.0	03 Nov, 2020	Apache-2.0 OR MPL-2.0	0 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 2.1.1	05 Jun, 2020	Apache-2.0 OR MPL-1.1	1 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 1.15.1	05 Jun, 2020	Apache-2.0 OR MPL-1.1	1 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 2.1.0	20 May, 2020	Apache-2.0 OR MPL-1.1	1 C 0 H 0 M 0 L
com.rabbitmq.jms:rabbitmq-jms 1.15.0	20 May, 2020	Apache-2.0 OR MPL-1.1	1 C 0 H 0 M 0 L

Our version contains a critical vulnerability

KUDELSKI SECURITY

R&D Services

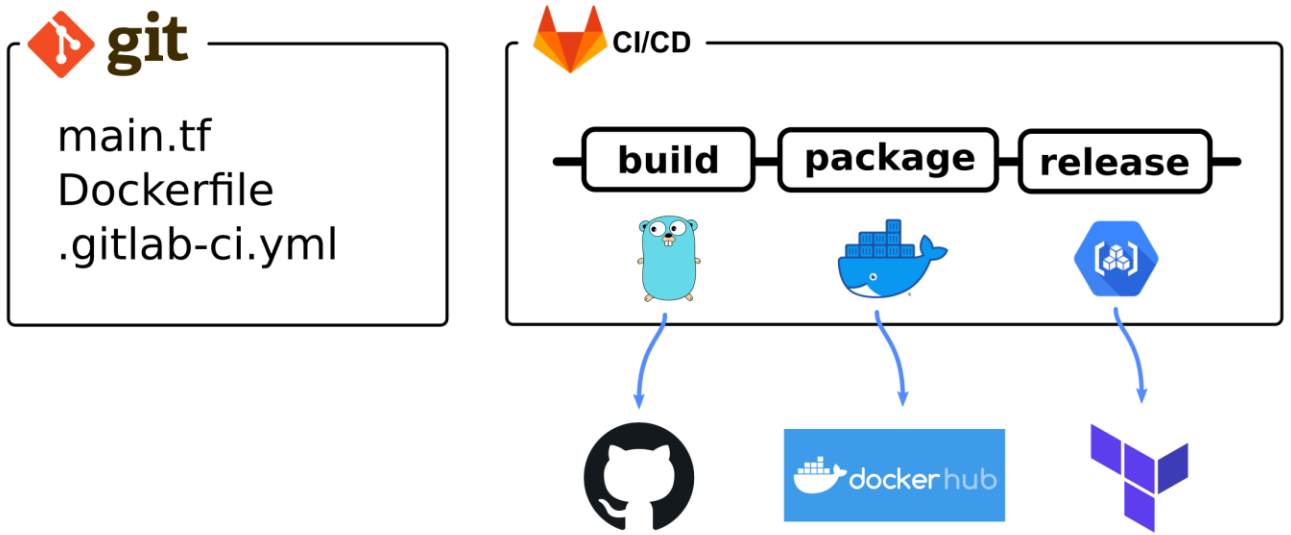
Use-case 2 – Infrastructure dependencies

KUDELSKI SECURITY

R&D Services

Use-case 2

- Terraform code used to deploy some costly resources on Azure
- Application that we package internally to do SAST on k8s deployments
- How to leverage Renovate to maintain the Terraform provider up-to-date and
  - Test new versions
  - Catch Breaking changes

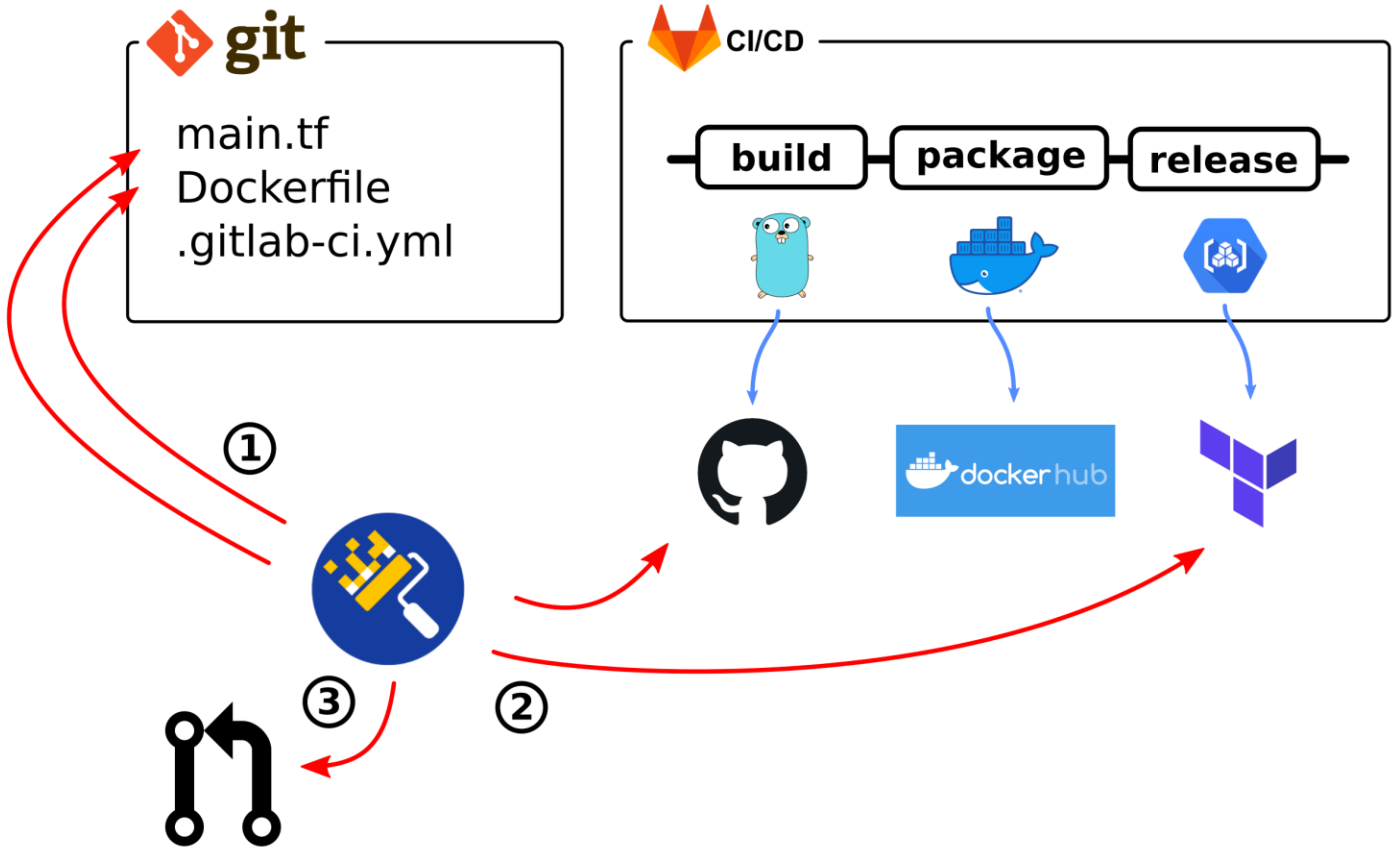


KUDELSKI SECURITY

R&D Services

Use-case 2

- Renovate bot configured on the Git repo
- It'll scan for known package-managers (or configured ones)
- Check if new versions are present
- Submit a pull/merge-request
- Trigger sub-sequent CI/CD pipelines



KUDELSKI SECURITY

R&D Services

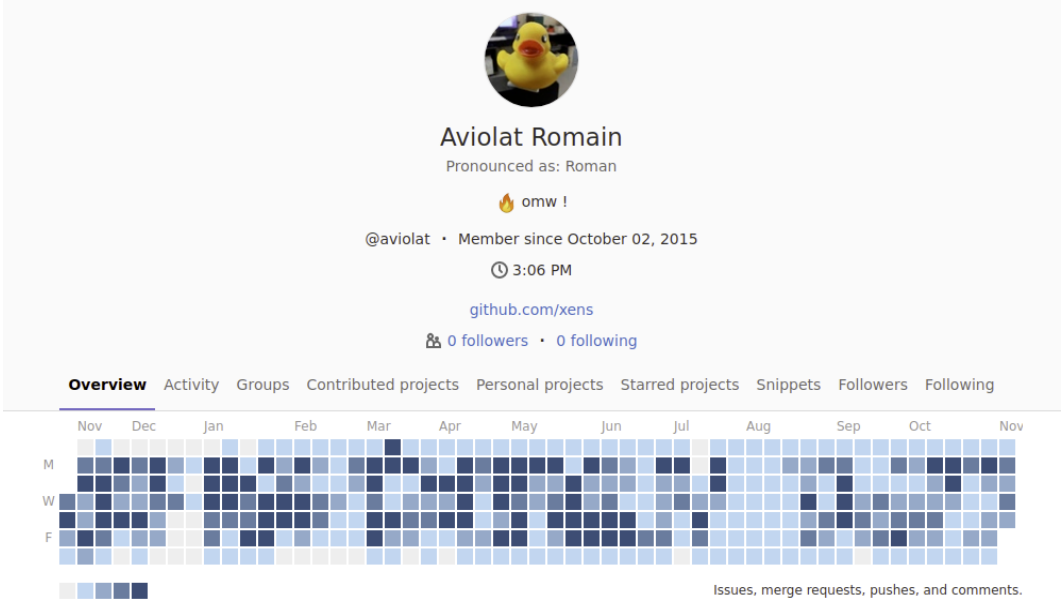
Our setup at Kudelski Security

KUDELSKI SECURITY

R&D Services

Renovate-as-a-service

- Started to use a dependency bot to manage all our software dependencies >1Y ago.
- Goal is to have up-to-date libraries as much as we can to lower the number of security issues in our code.
- Bot has sent **7400** Pull Requests in a year, **96%** have been merged in production
- < OPs, > Security



KUDELSKI SECURITY

R&D Services

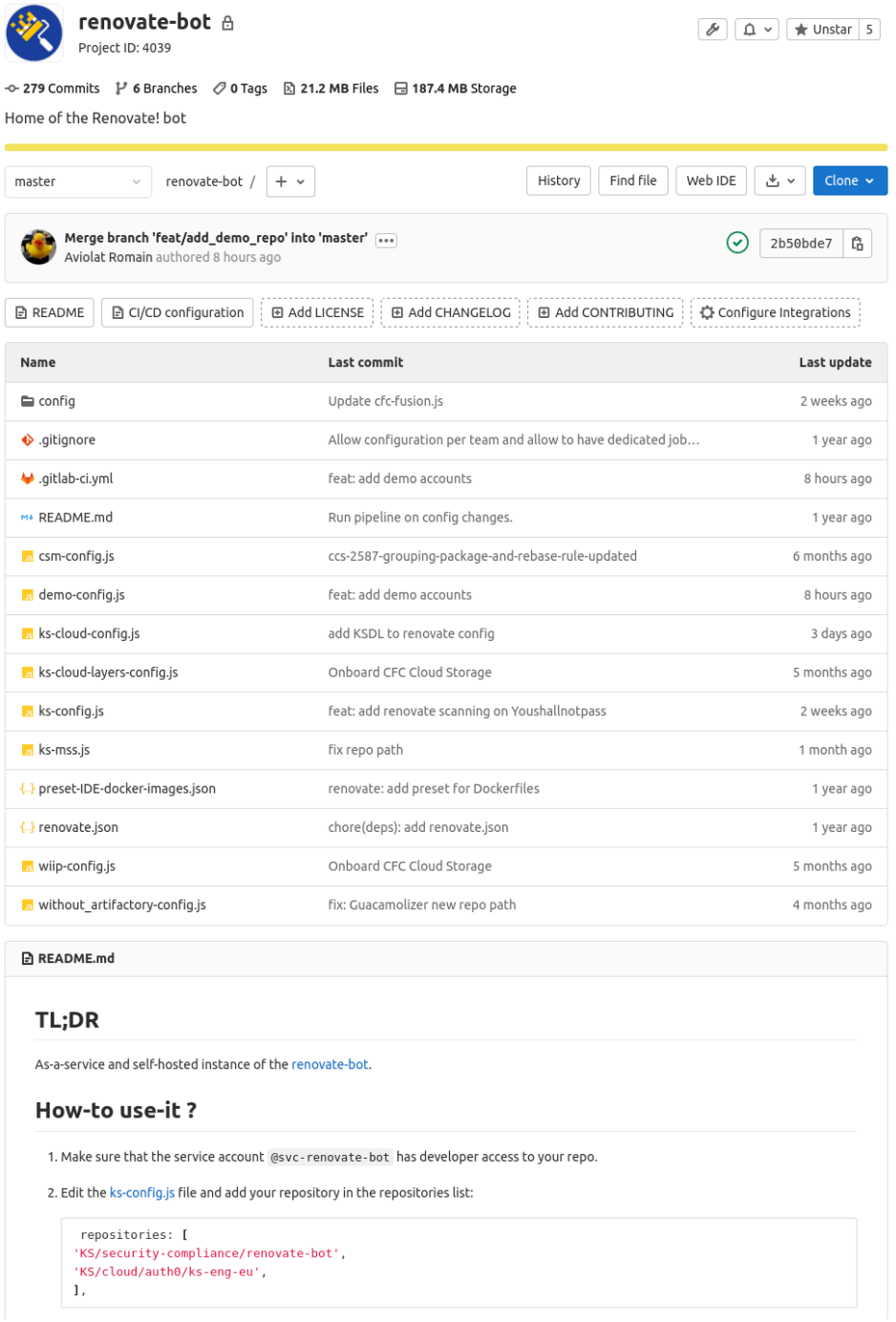
We use it to maintain

- Applications and infrastructure dependencies
- Containers base images
- CI/CD templates .gitlab-ci.yml docker images references
- Packaging of specific tool in KS contain factory (RegExpr matcher)
- Terraform modules versions
- Git submodules between projects
- Helm / Docker-Compose releases

R&D Services

Renovate-as-a-service

- Single Renovate bot for all the development teams.
- The bot has "developer" accesses to all our Git repositories (can pull and send PRs)
- Bot is maintained by infra (updated ~ every month)
- Single Git repo to host the bot configs, one config file per team, with its own settings
- Different schedules per teams



KUDELSKI SECURITY

Things to know

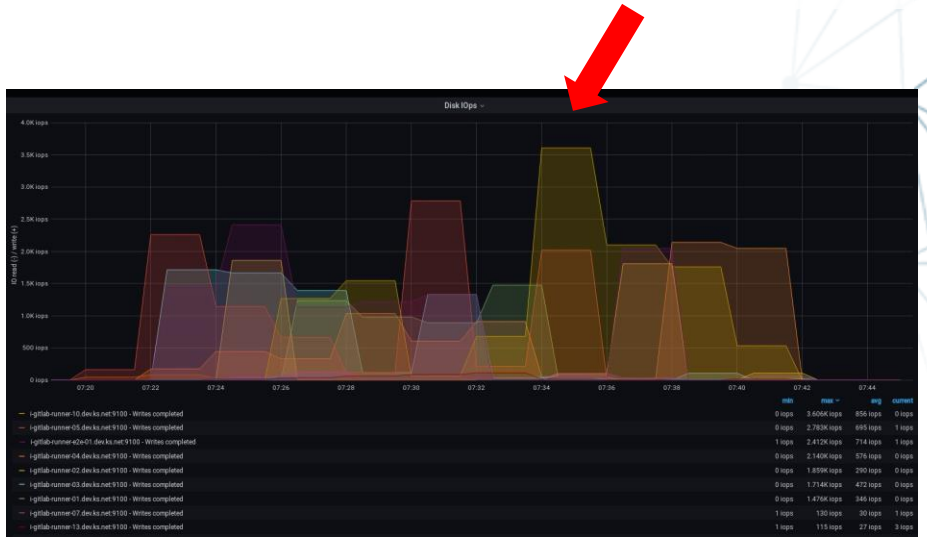
KUDELSKI SECURITY

R&D Services

Things to know

- If the bot triggers one pipeline per pull request across all your git repos you might put a lot of load on your infra, you might adapt your workflow
- Start by onboarding a couple of repo, see how it goes and adapt
- Adapt your CI/CD pipelines (specific jobs for Renovate branches ?)
- Beware of PR fatigue !
- Configure Renovate to fit your needs ("Noise reduction")
  - Schedule
  - Grouping upgrades per language ?
- Renovate will **hammer** your APIs (artifact repo, gitlab, github)

Also known as a PDOS (Pipeline-induced Deny of Service)



It's time to give love to your Gitlab admin

429 Too Many Requests

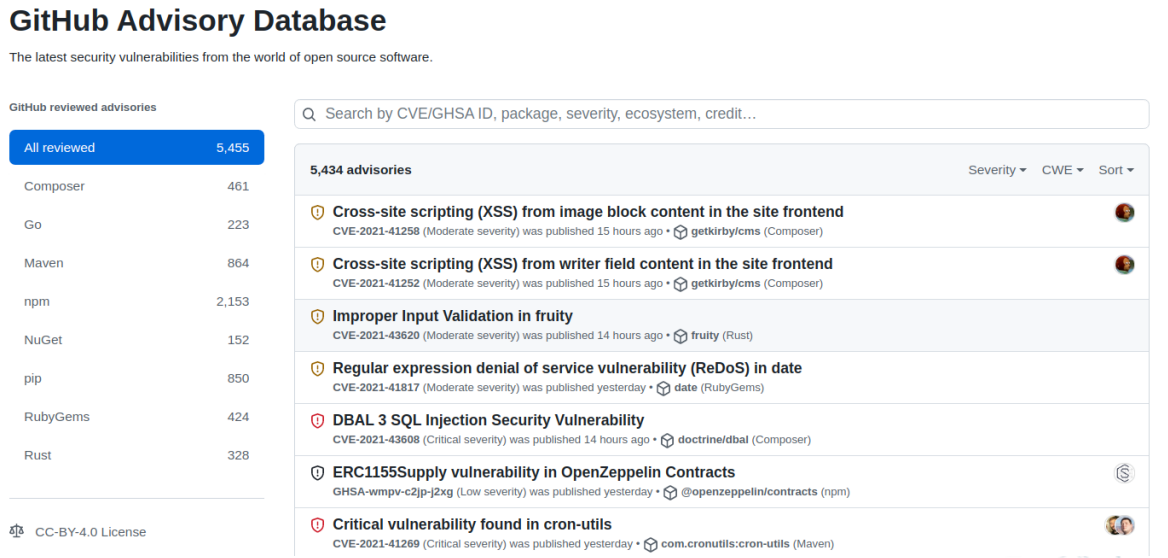
The HTTP `429 Too Many Requests` response status code indicates the user has sent too many requests in a given amount of time ("rate limiting").

KUDELSKI SECURITY

R&D Services

Things to know

- Renovate will notify for security vulnerabilities in your dependencies **only** if you're using Github
- Leverages Github Advisory Database
- If you're on Github, you might also give a try to Dependabot, it has less functionalities and supports less languages but it's built-in

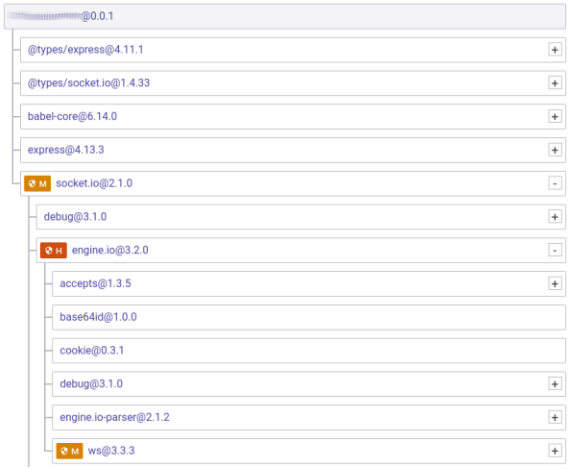


KUDELSKI SECURITY

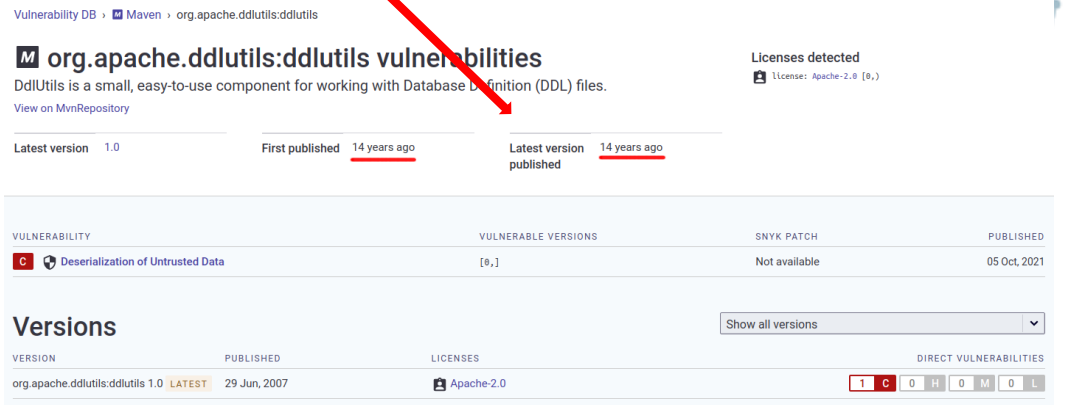
R&D Services

Things to know (if you're not on GH)

- Renovate **will not** monitor for security issues in your libraries
- Renovate will not highlight outdated libraries that are not maintained anymore
- It'll not highlight transitive vulnerabilities in your dependencies
- You need to use another tool to detect outdated / insecure &| transitive libraries (Snyk, Bolt, ...)
- For transitive vulns you need another tool whatever the platform is




```
<build>
    <plugins>
        <plugin>
            <groupId>com.rabbitmq.jms</groupId>
            <artifactId>rabbitmq-jms</artifactId>
            <version>2.1.1</version>
            <groupId>com.apache.ddlutils</groupId>
            <artifactId>ddlutils</artifactId>
            <version>1.0</version>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-shade-plugin</artifactId>
            <version>3.2.4</version>
        </plugin>
    </plugins>
</build>
</project>
```



KUDELSKI SECURITY

R&D Services

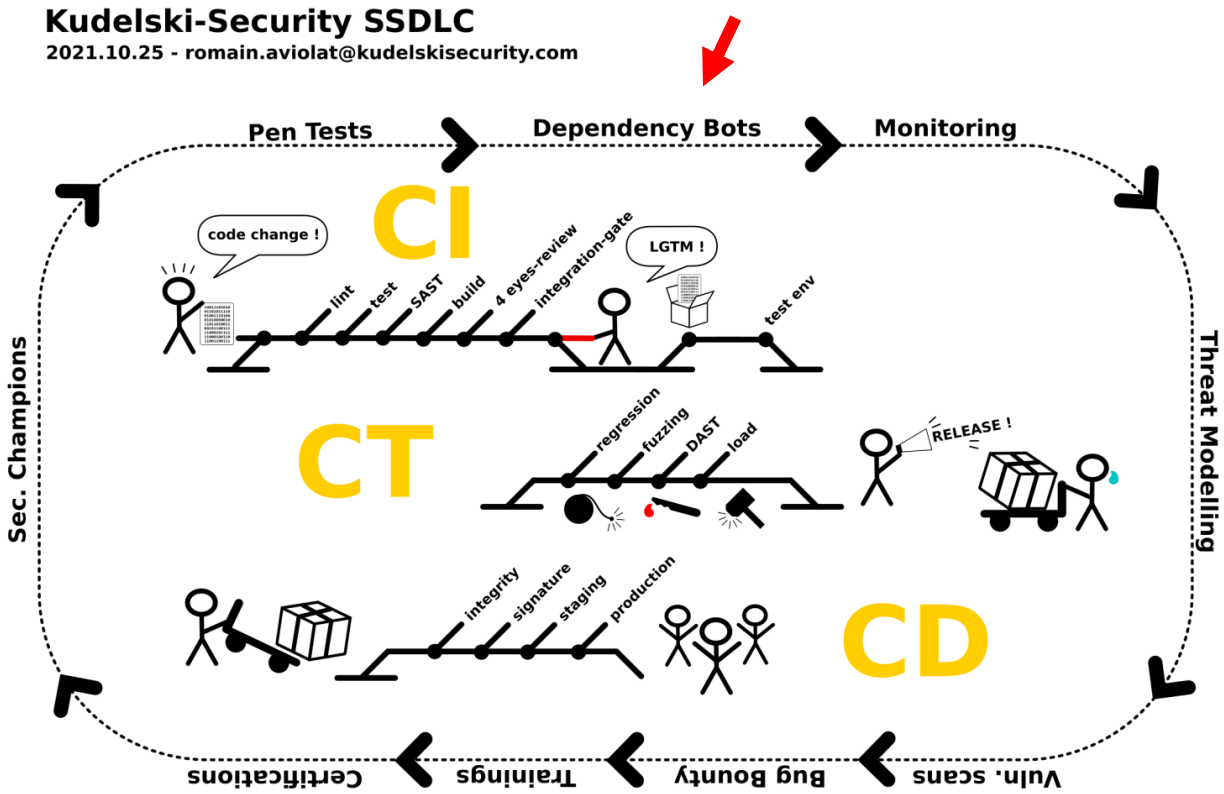
Takeaways

KUDELSKI SECURITY

R&D Services

Takeaways

- Renovate is great tool in the Developer toolbox
- Not an AppSec silver bullet, part of a bigger whole
- Helping us to better manage our software dependencies at KS
- Less OPs more Security using Bots
- Give love to your 3rd party dependencies, don't randomly choose them (properly maintained, license fit, security response policy, release cyclce, …)



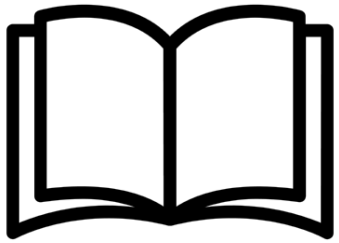
KUDELSKI SECURITY

R&D Services

Questions ?

KUDELSKI SECURITY

R&D Services



References

KUDELSKI SECURITY

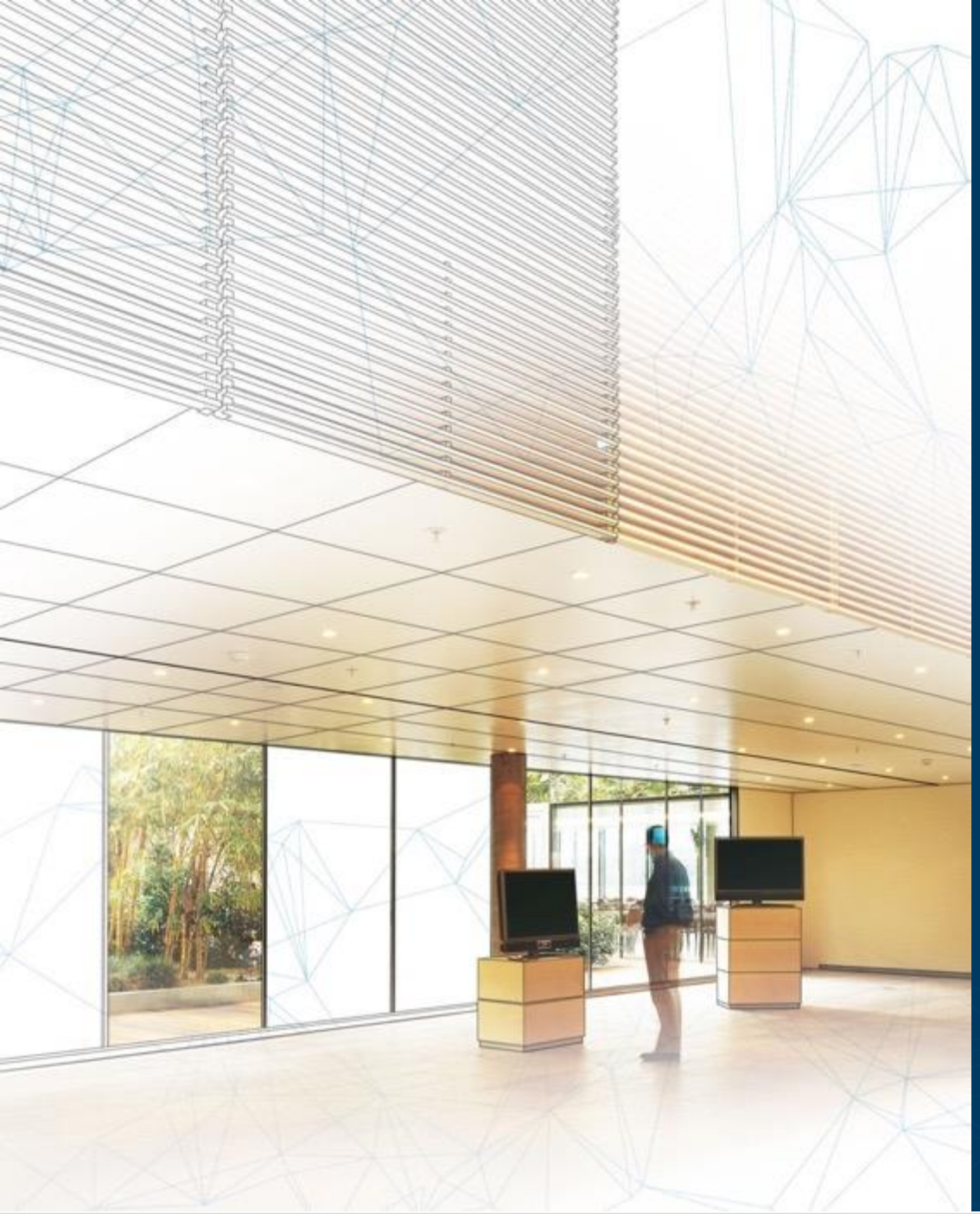
R&D Services

References

- Blog post from Github Engineering about Dependabot
  - https://github.blog/2020-06-01-keep-all-your-packages-up-to-date-with-dependabot
- Renovate bot
  - https://github.com/renovatebot
- SLSA ("salsa") Supply-chain Levels for Software Artifacts
  - https://github.com/slsa-framework/slsa

R&D Services

KUDELSKI SECURITY



KUDELSKI SECURITY

Thank You