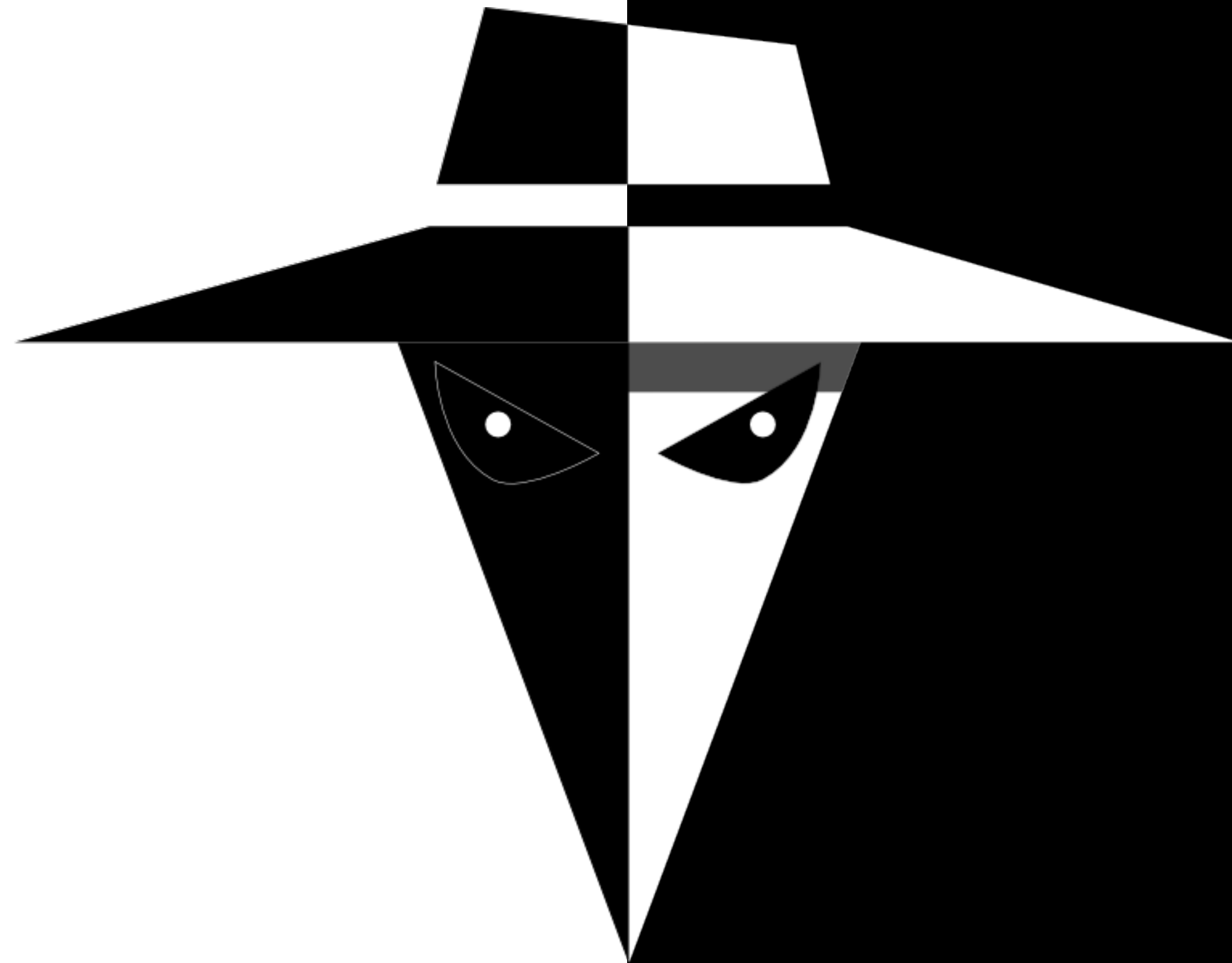
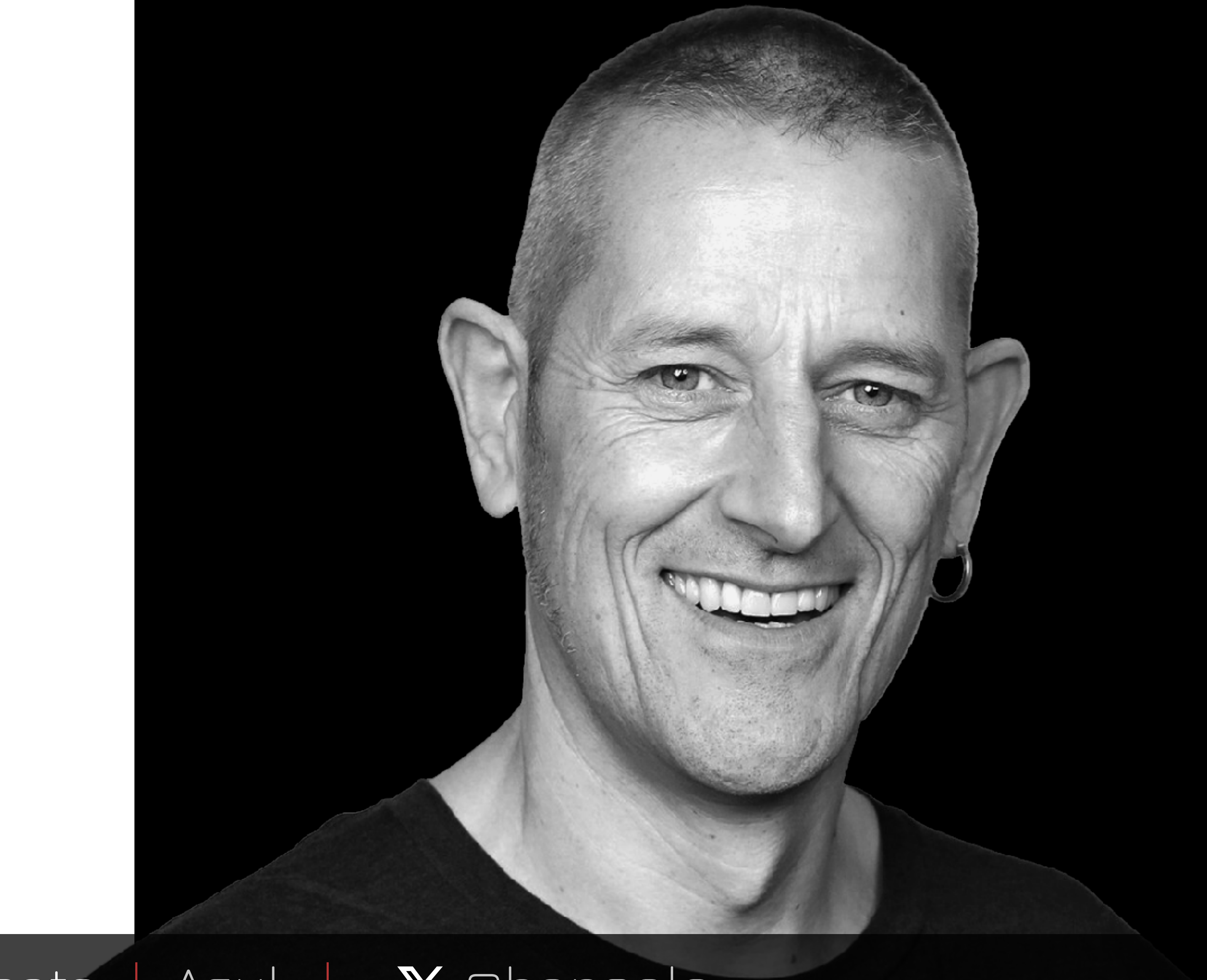


WAR GAMES



JAVA VULNERABILITIES AND WHY YOU SHOULD CARE

ABOUT ME.



Gerrit Grunwald | Developer Advocate | Azul | X @hansolo_

I.A.M.A.

DEVELOPER

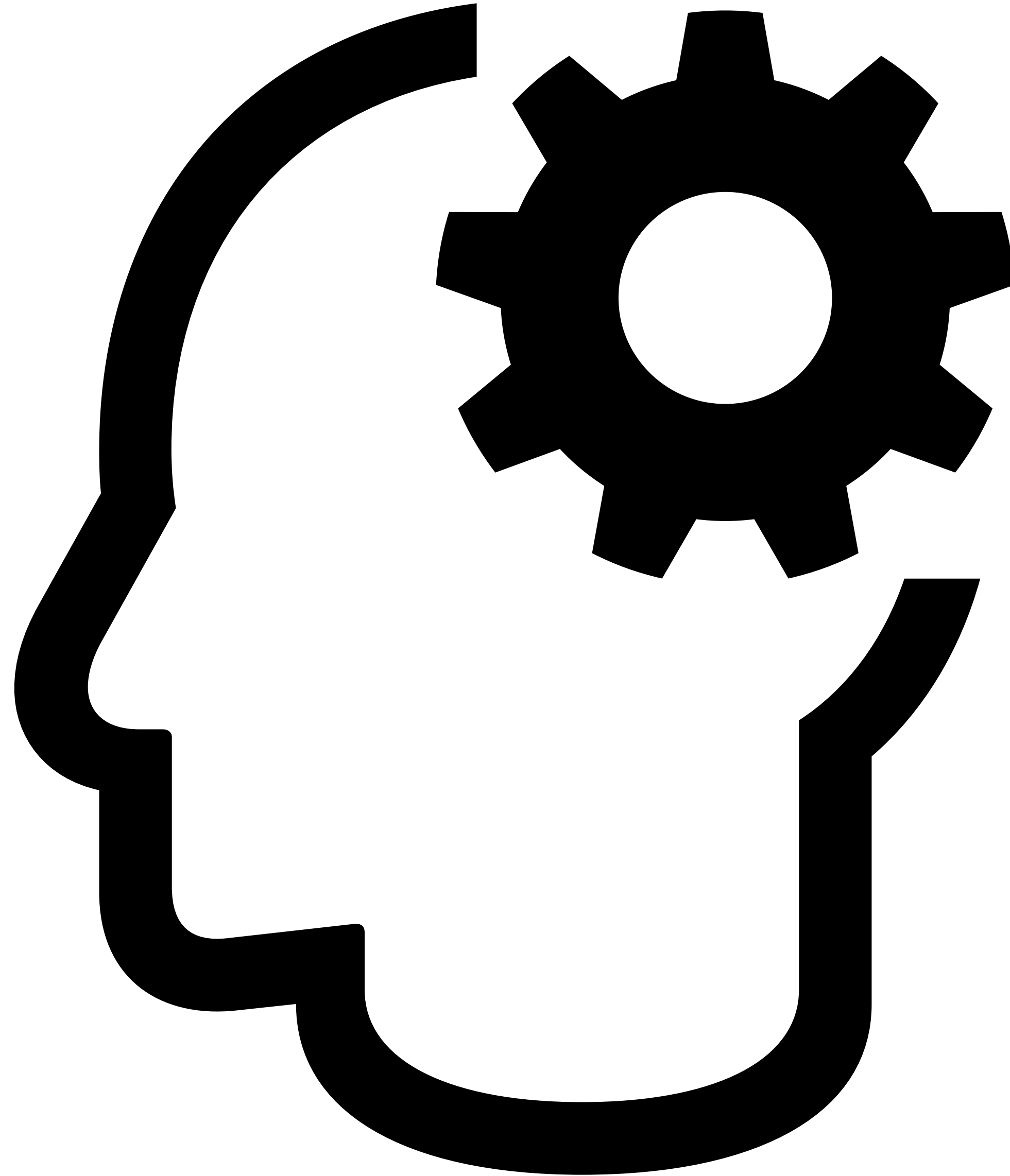
NOT A
SECURITY
EXPERT

24TH

NOVEMBER

2021

LOG₄SHELL



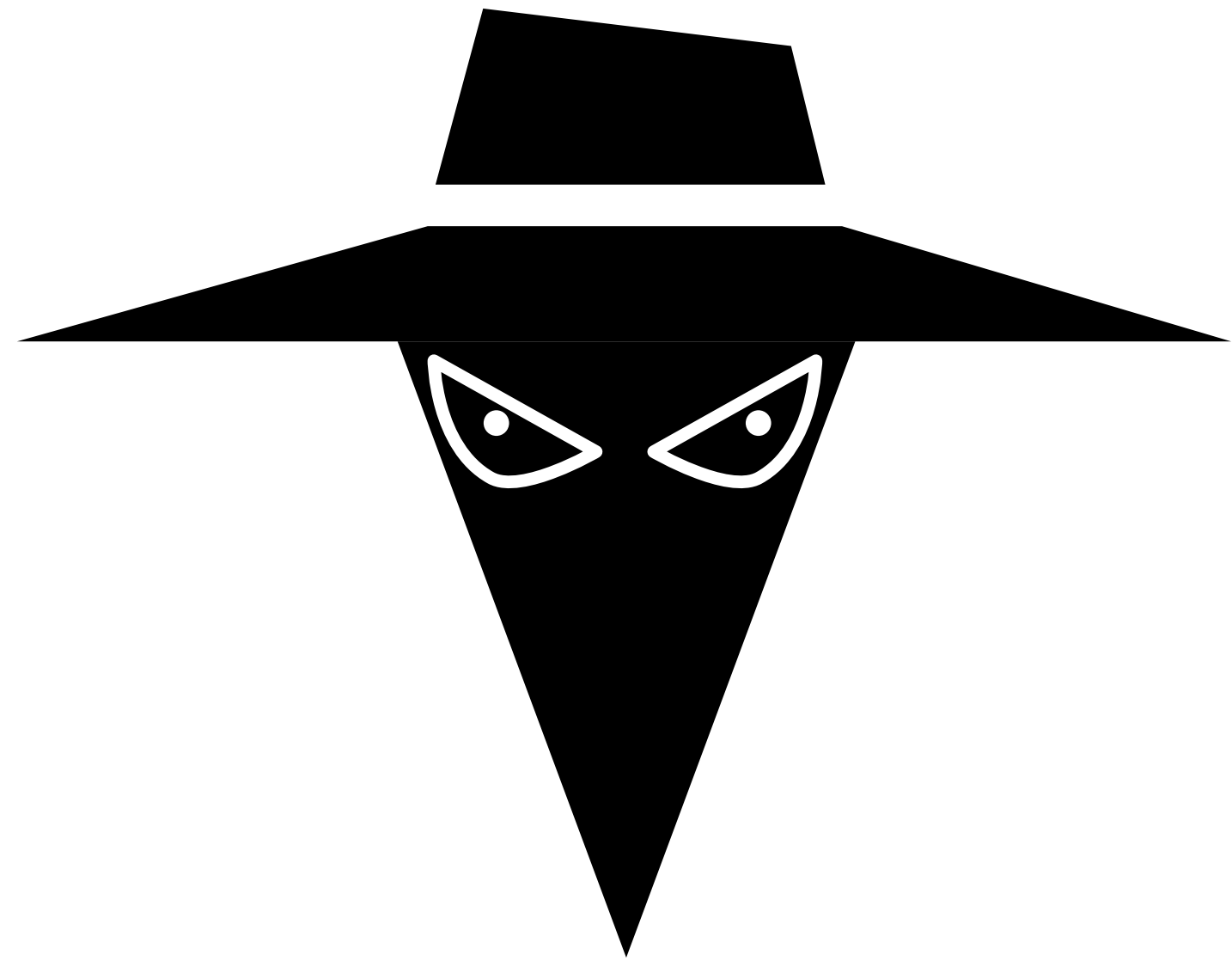
20TH CENTURY

Software landscape

20TH CENTURY

Software landscape

- ✦ Code was self written and closed source
- ✦ Source code was managed in a repository on a local server
- ✦ Manually build
- ✦ Delivered on hardware (CD, DVD, USB-Sticks)
- ✦ Ran on closed networks or local servers
- ✦ Large monolithic systems
- ✦ Connected systems only in government / banking / energy providers
- ✦ Full control over the source code

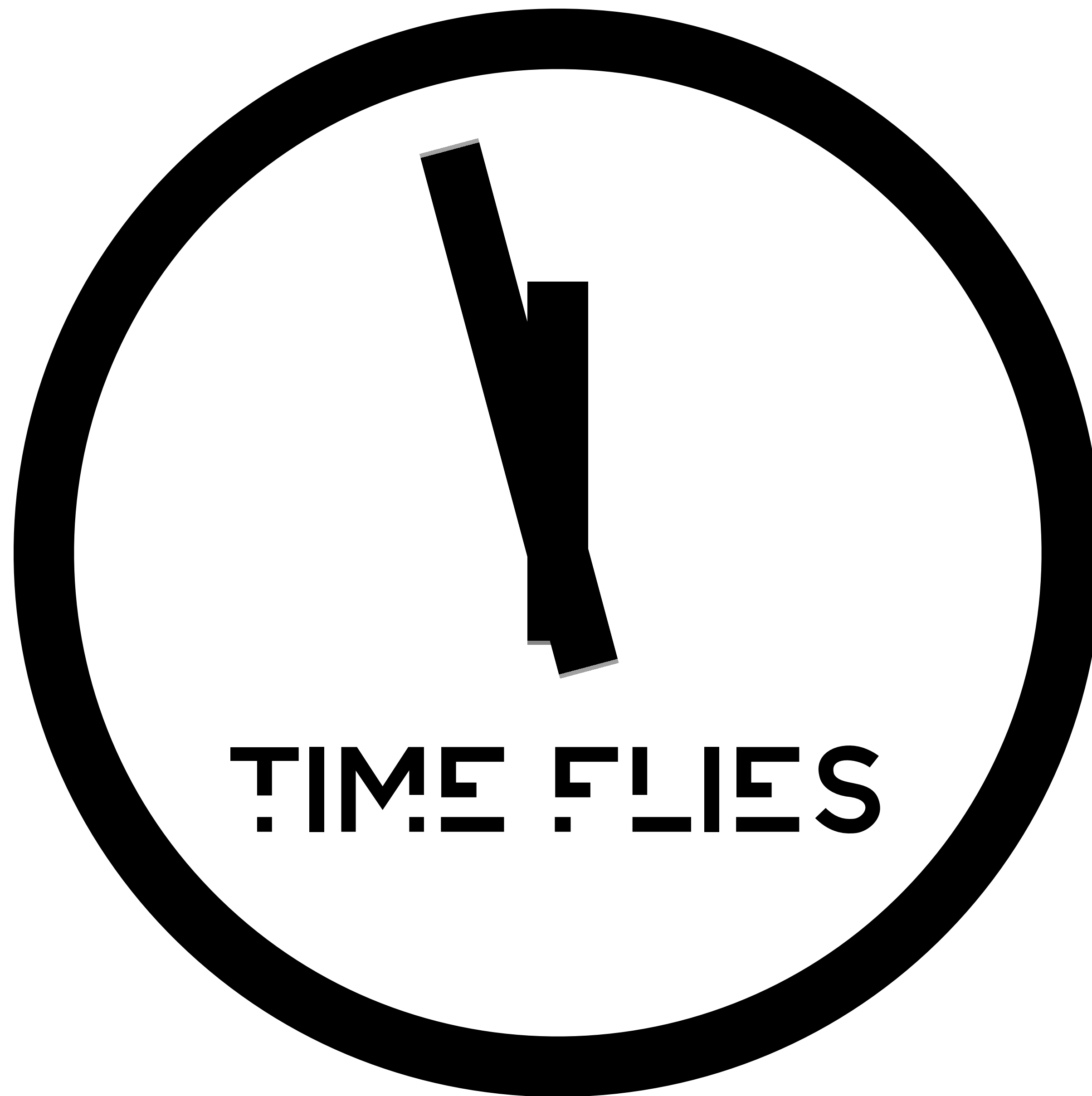


20TH CENTURY VULNERABILITIES

20TH CENTURY VULNERABILITIES

Vulnerabilities

- ✦ Password hacking / cracking
- ✦ Computer viruses (spread via floppy discs/usb sticks)
- ✦ Early days of hacking via internet



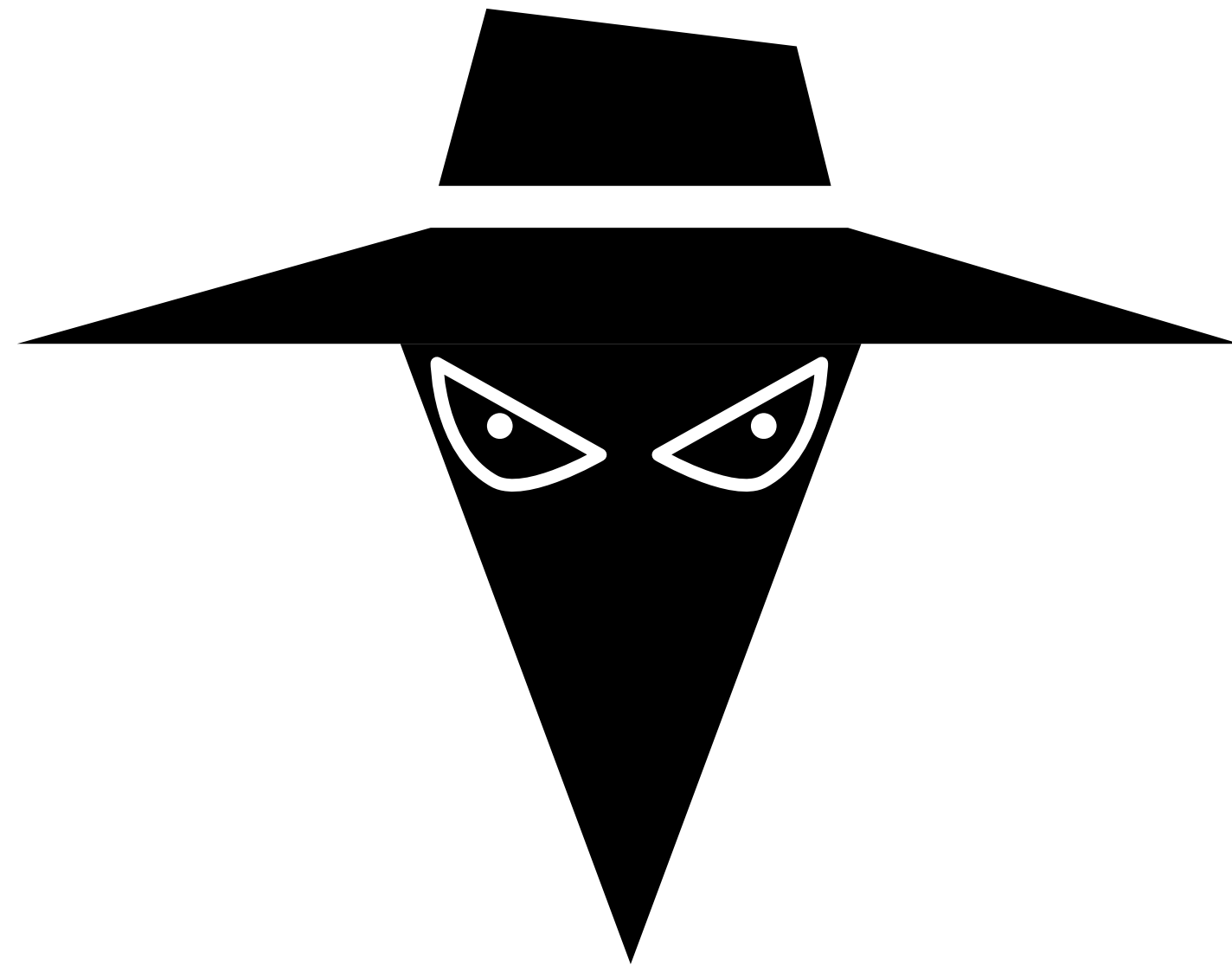
21ST CENTURY

Software landscape

21ST CENTURY

Software landscape

- ✦ A lot of open source software used
- ✦ Distributed source code management systems
- ✦ Automated builds by CI / CD systems
- ✦ Hosted in artifact repositories
- ✦ Running on public networks
- ✦ Accessible via browsers or api's
- ✦ "Everything" is connected
- ✦ No full control over the source code
- ✦ Today we have a whole software supply chain



21ST CENTURY VULNERABILITIES

21ST CENTURY VULNERABILITIES

Vulnerabilities

- ✦ Danger through Social Engineering (SIM swapping etc.)
- ✦ Malware / Ransomware (spread via mail / websites)
- ✦ Everything that is connected, will be hacked
- ✦ Spreading malicious code is way easier
- ✦ The whole software supply chain is target of attacks

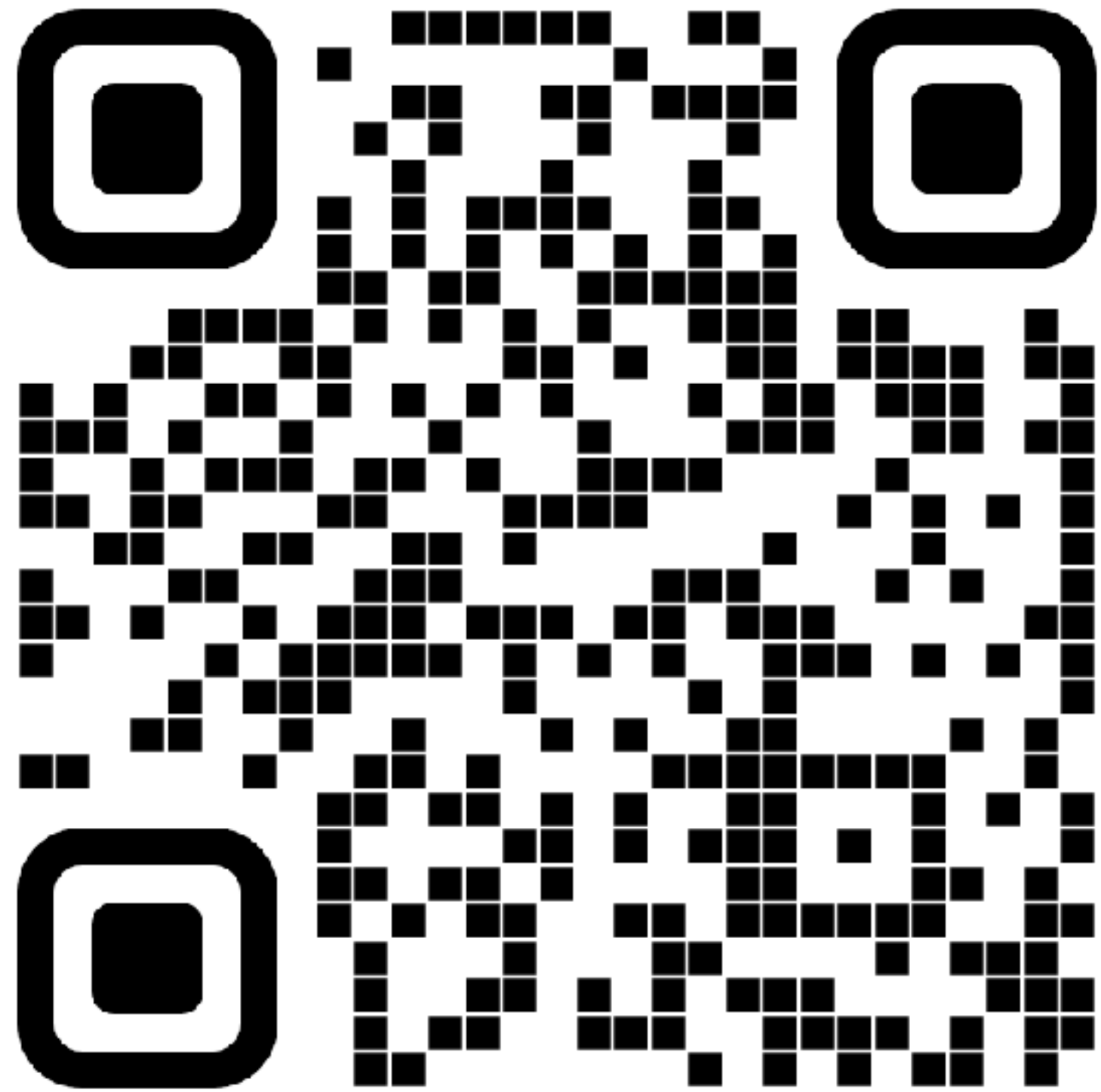
SOME DEFINITIONS

CWE

Common Weakness Enumeration

CWE

Common Weakness Enumeration



<https://cwe.mitre.org/>

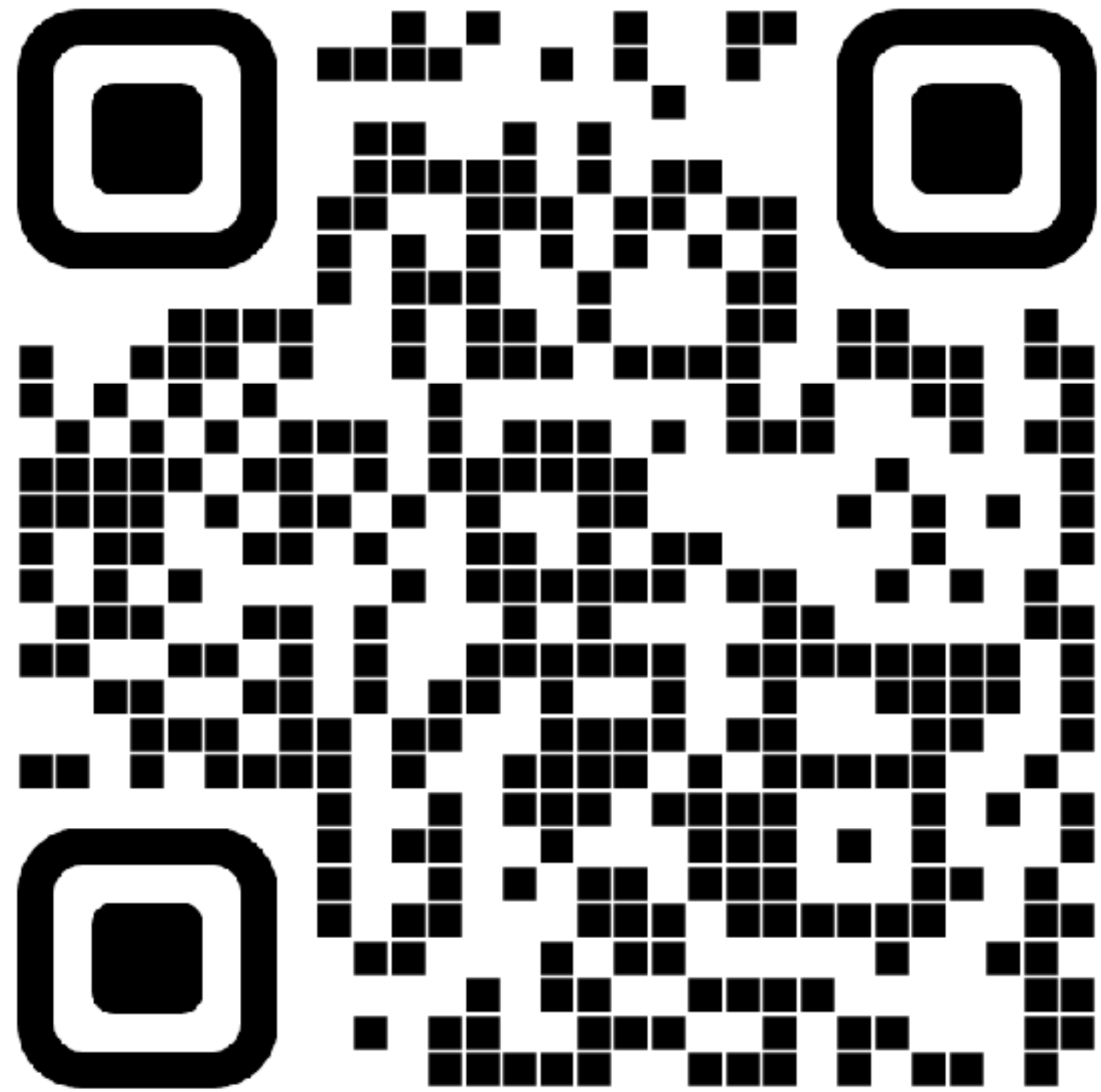
Community developed list
of software and hardware
weakness types.

NVD

National Vulnerability Database

NVD

National Vulnerability Database



<https://nvd.nist.gov/>

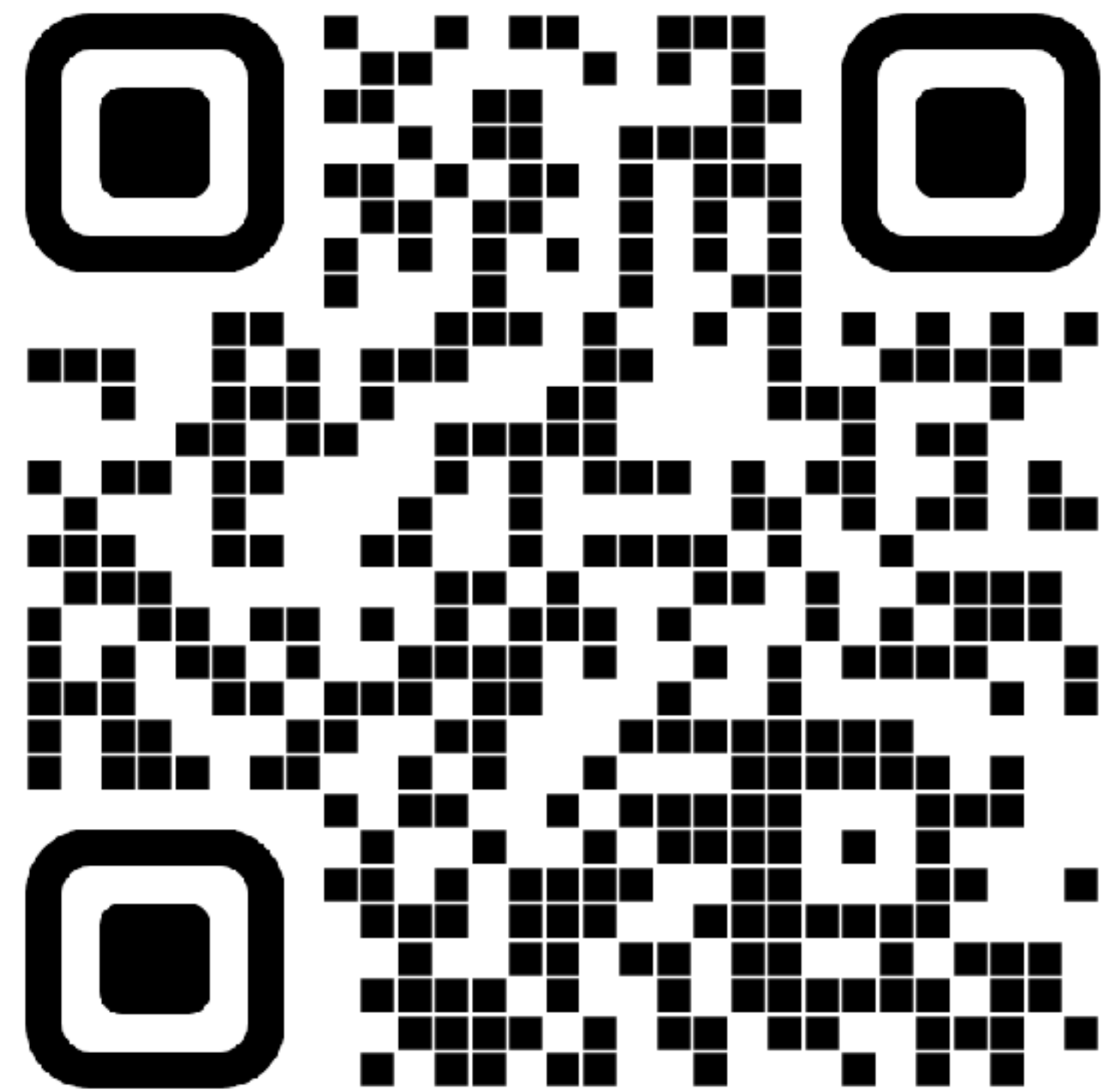
U.S. government repository
of standards based
vulnerability management
data, represented using
the Security Content
Automation Protocol (SCAP)

CVE

Common Vulnerability + Exposure

CVE

Common Vulnerability + Exposure



<https://cve.org/>

CVE Program Mission

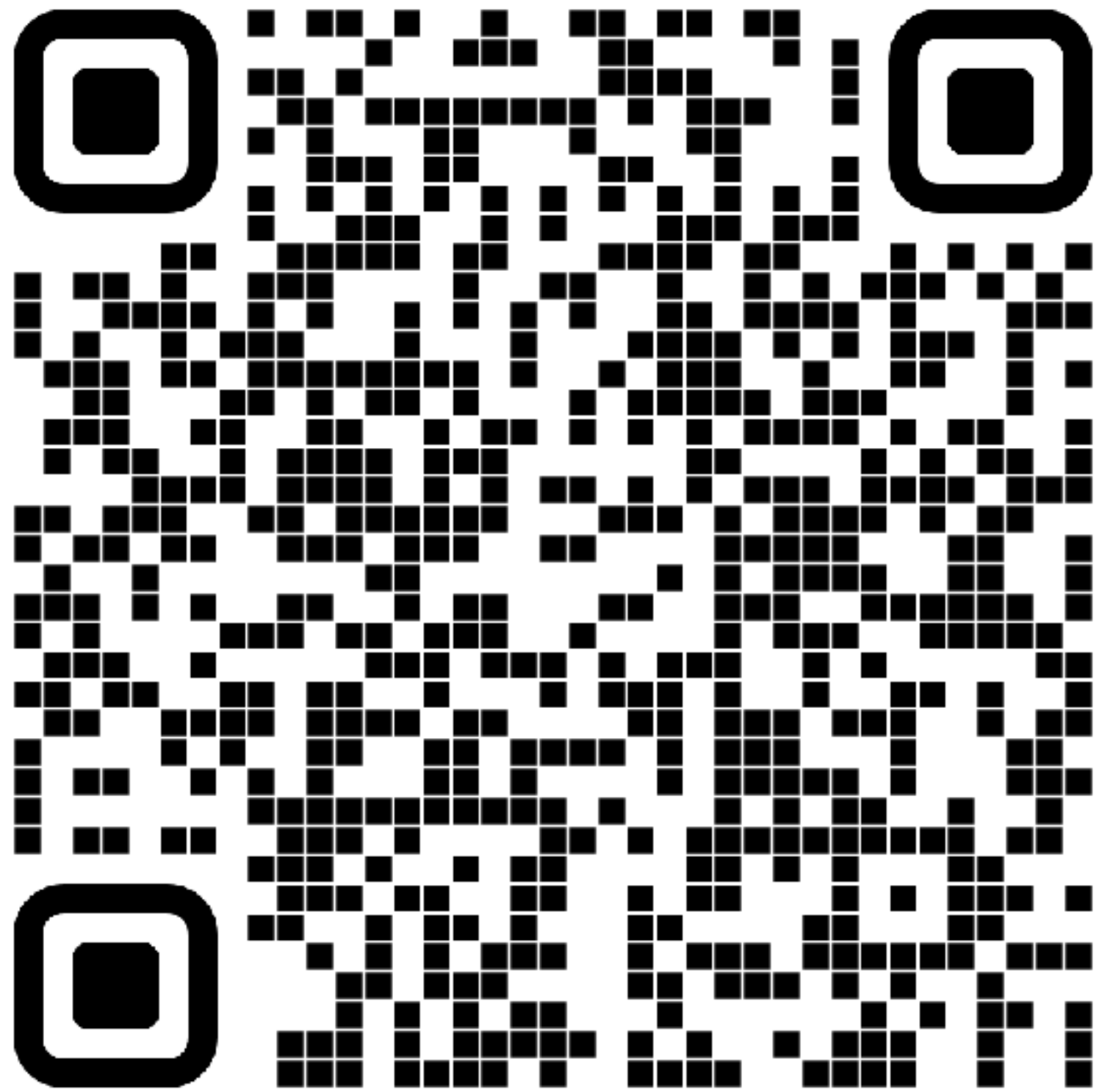
"Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities"

LOG₄SHELL

CVE-2021-44228

CVE-2021-44228

Log4Shell

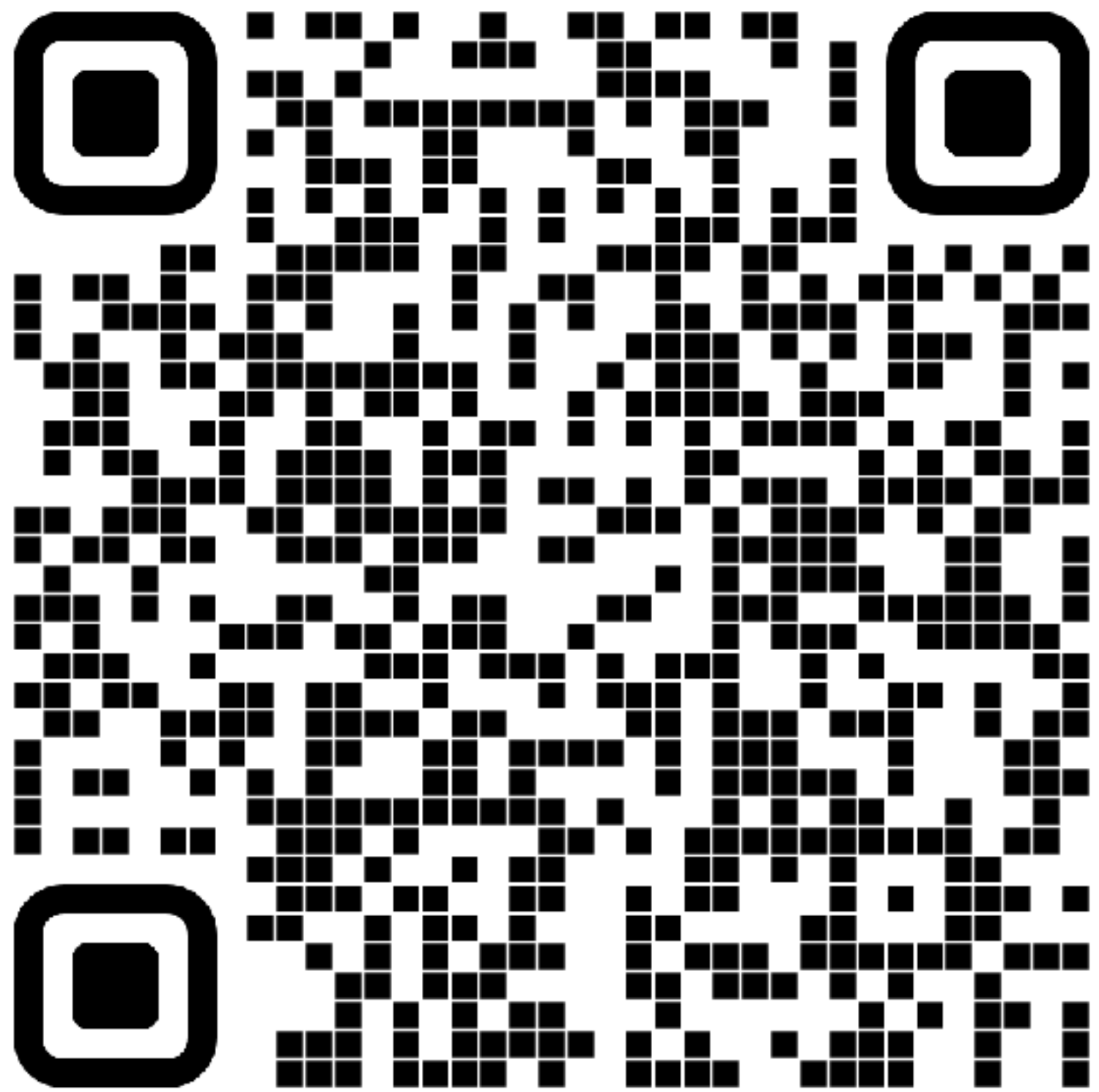


<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

CVE-2021-44228

Log4Shell



<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

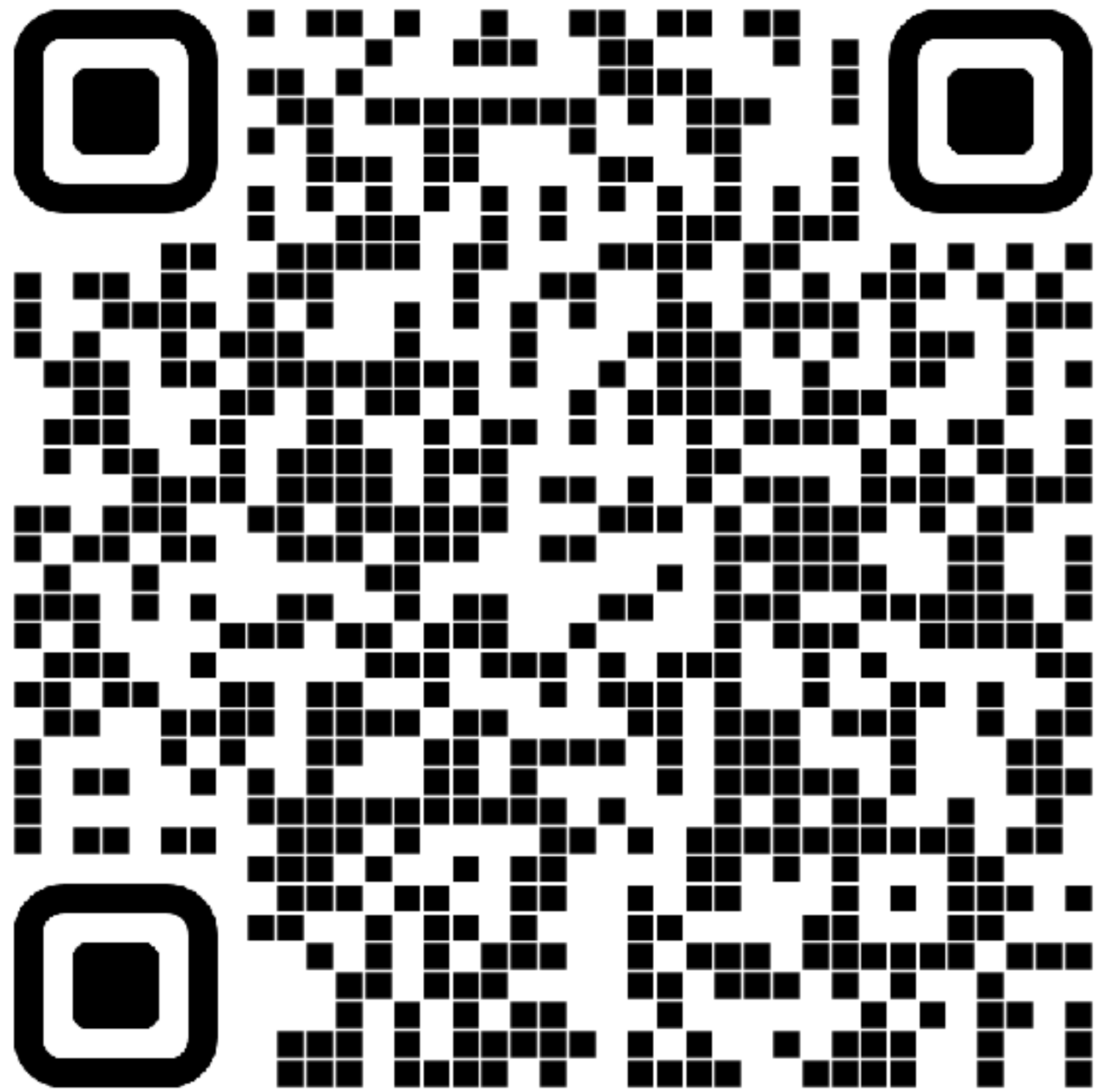
Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

Versions affected



CVE-2021-44228

Log4Shell



<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

Apache Log4j2 2.0-beta9 through 2.15.0

(excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log

messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints.

An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled.

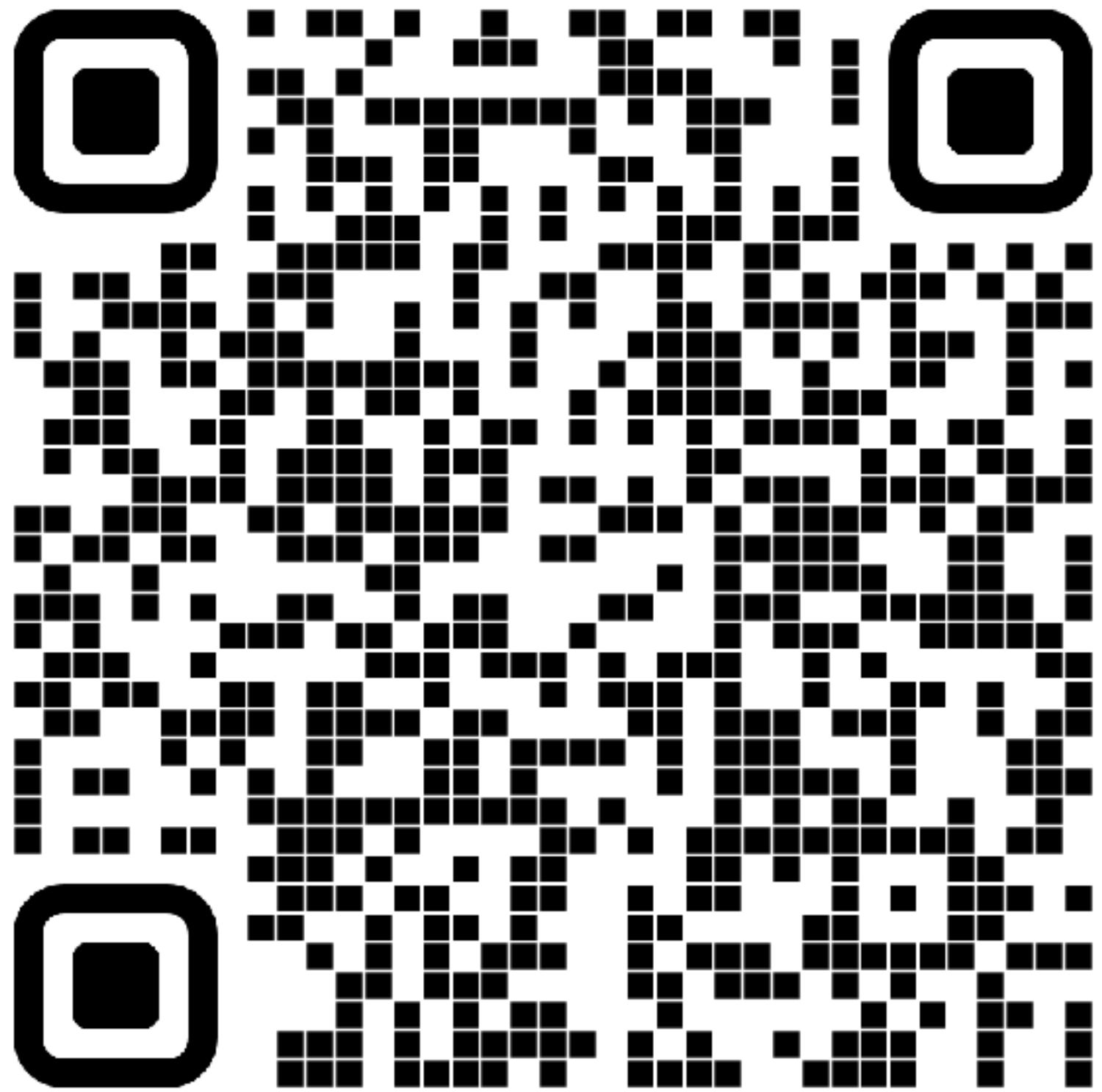
From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

The vulnerability



CVE-2021-44228

Log4Shell



<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled.

Description



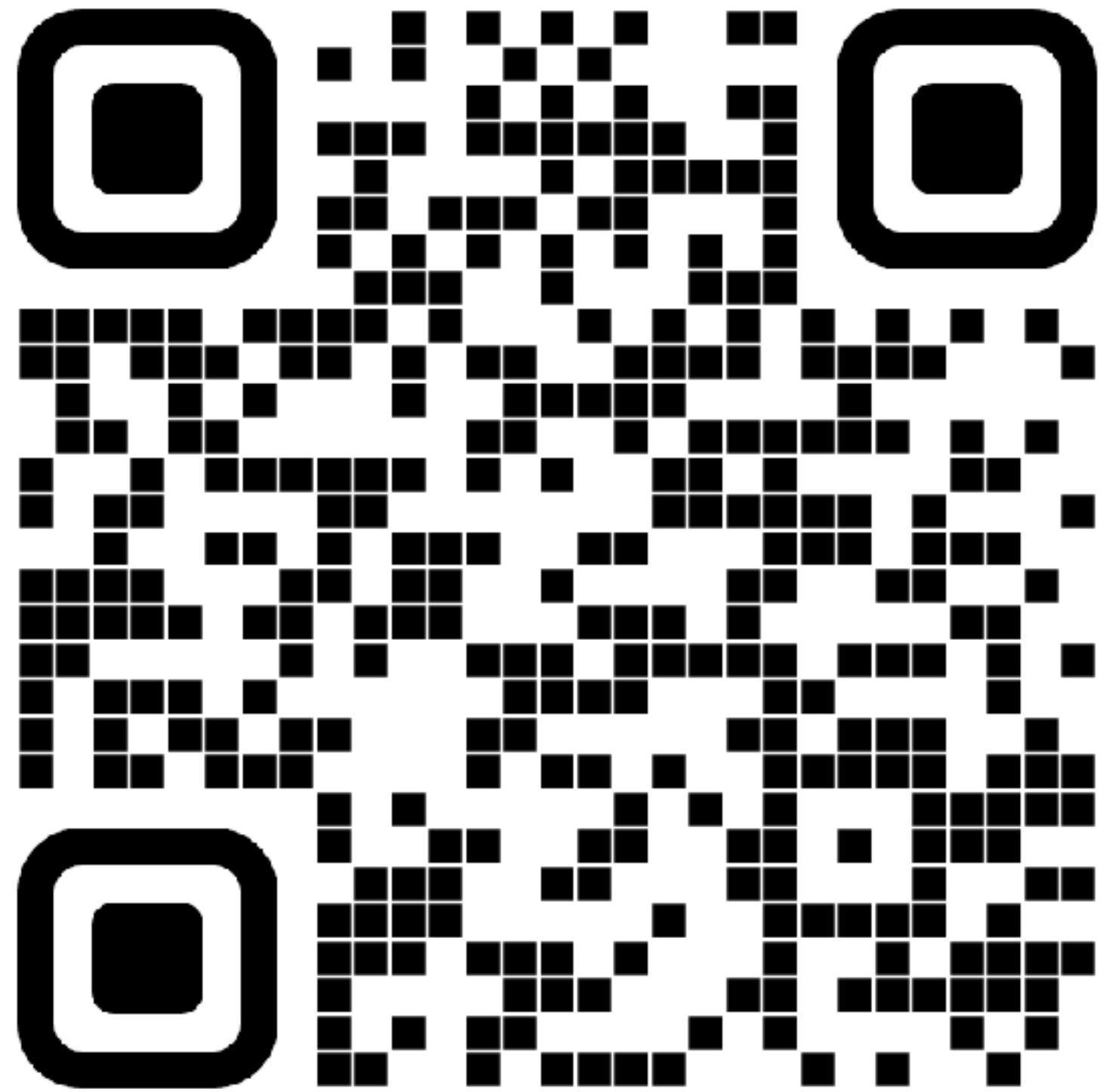
From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

CVSS

Common Vulnerability Severity Score

CVSS

Common Vulnerability Severity Score



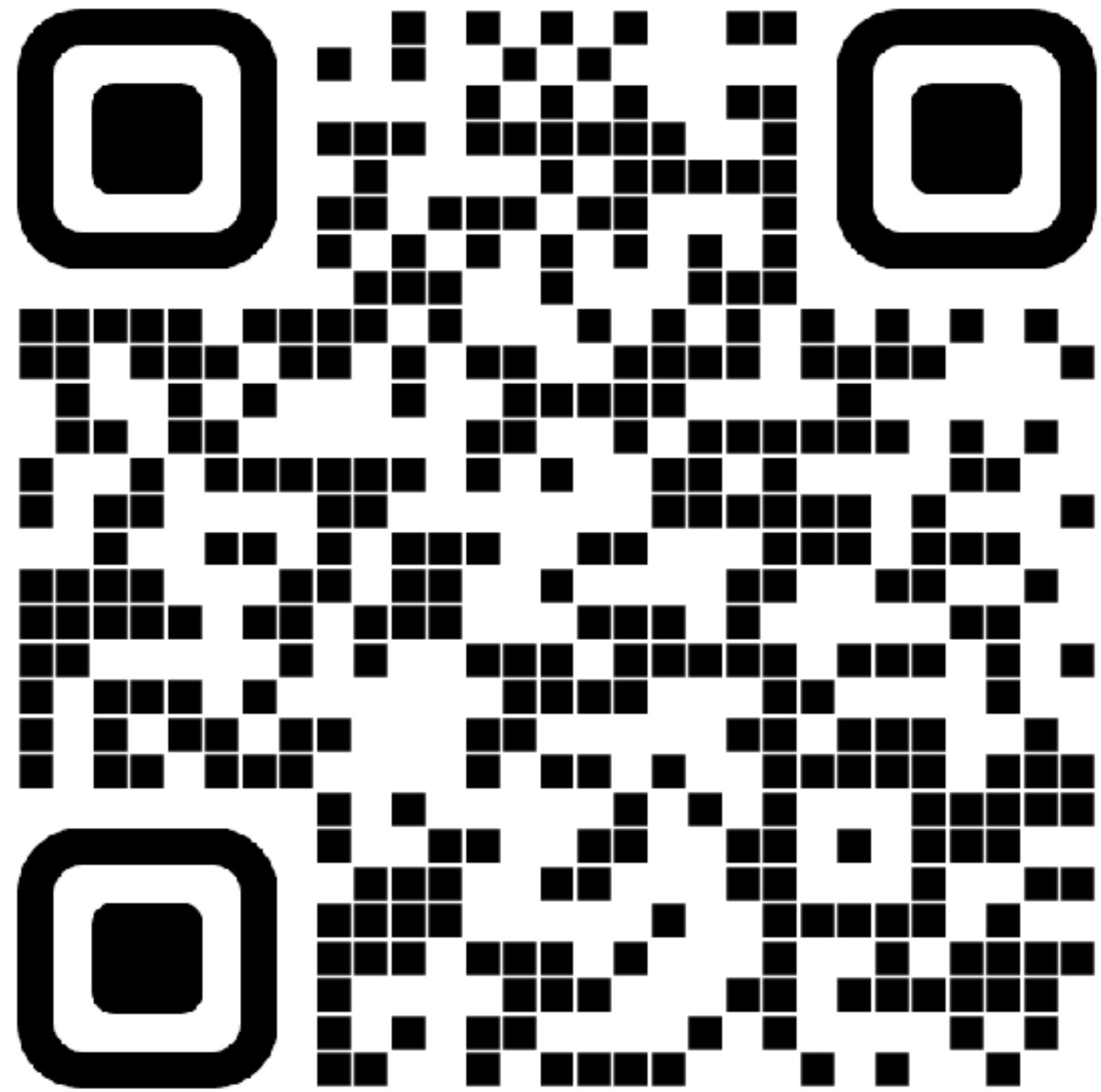
<https://nvd.nist.gov/vuln-metrics/cvss>

Vulnerability Severity Ratings (CVSS v2.0)

| Severity | | Score |
|----------|---|------------|
| Low | ● | 0.0 – 3.9 |
| Medium | ● | 4.0 – 6.9 |
| High | ● | 7.0 – 10.0 |

CVSS

Common Vulnerability Severity Score



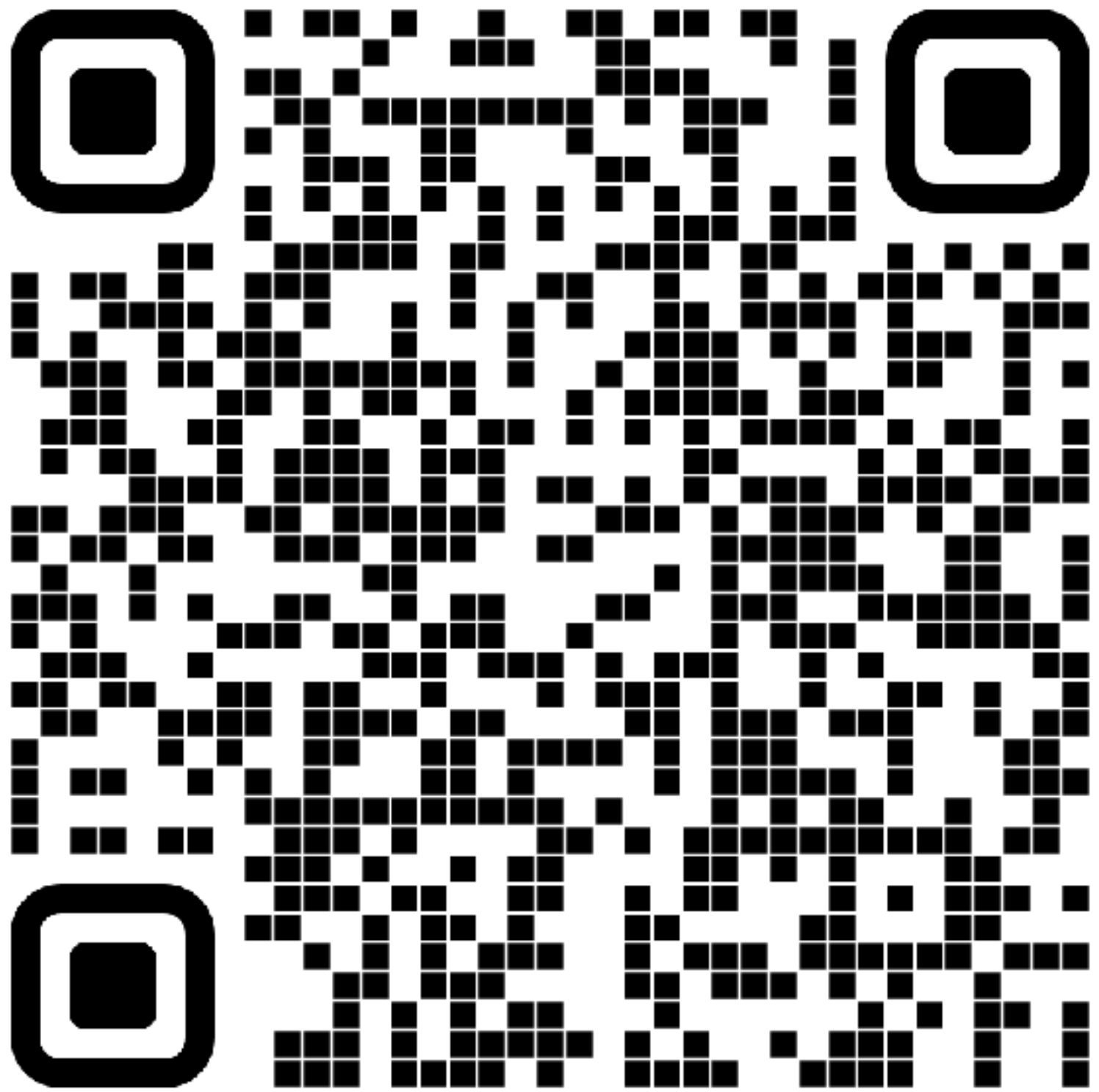
<https://nvd.nist.gov/vuln-metrics/cvss>

Vulnerability Severity Ratings (CVSS v3.1)

| Severity | Score |
|----------|------------|
| None | 0.0 |
| Low | 0.1 – 3.9 |
| Medium | 4.0 – 6.9 |
| High | 7.0 – 8.9 |
| Critical | 9.0 – 10.0 |

CVSS

CVE-2021-44228 (Log4Shell)



<https://nvd.nist.gov/vuln/detail/CVE-2021-44228>

CVSS v2.0

Severity

High

Score

9.3

CVSS v3.1

Severity

Critical

Score

10.0

IS JAVA
SECURE ?

OPENJDK VULNERABILITY GROUP

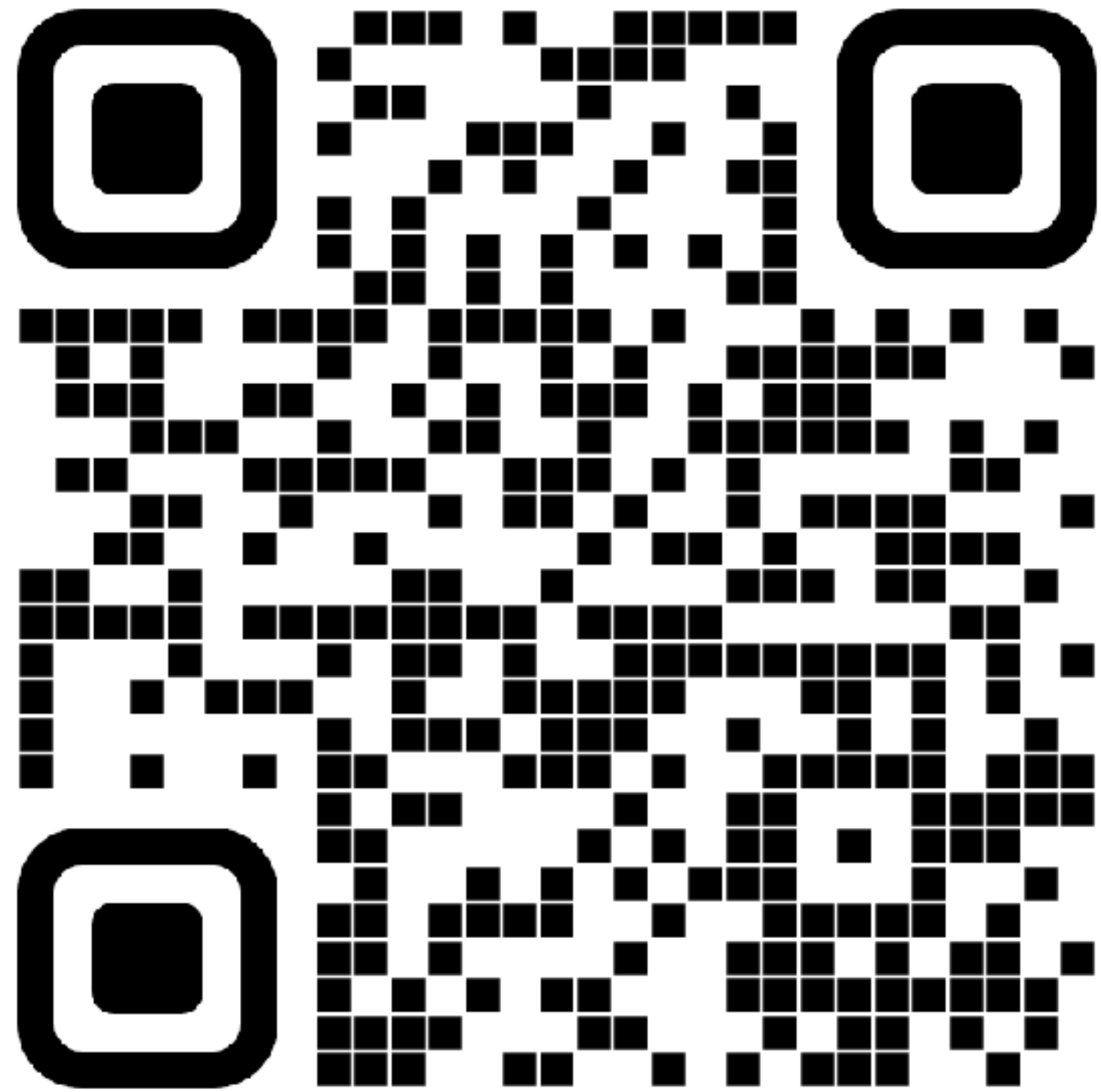
OPENJDK VULNERABILITY GROUP

What is it...?

- ✦ Private forum (trusted members of the OpenJDK community)
- ✦ Receives/reviews reports of vulnerabilities in the OpenJDK code base
- ✦ Collaborates on fixing the issues
- ✦ Coordinates the release of such fixes
- ✦ Maintains list of CVE's patched for each release
- ✦ Tracks CVE's by component (not all Java users leverage every component)
- ✦ Discusses OpenJDK security related issues
- ✦ Does not actively test the OpenJDK source code

VULNERABILITY

ADVISORIES



<https://openjdk.org/groups/vulnerability/advisories/>

OpenJDK Vulnerability Advisories

Published 4x a year

Describing

- Severity
- Area
- Affected versions

OPENJDK VULNERABILITY ADVISORY

Example 17th of October 2023

OpenJDK Risk matrix

| CVE ID | Component | CVSSv3.1 Vector | Affects ... | | | |
|--------------------------------|---------------------------------|--------------------|-------------|----|----|----|
| | | | 8 | 11 | 17 | 21 |
| CVE-2023-22067 | other-libs/ corba | 5.3 NLNNUNLN | • | | | |
| CVE-2023-22081 | security-libs/ javax.net.ssl | 5.3 NLNNUNNL | • | • | • | • |
| CVE-2023-22025 | hotspot/ compiler | 3.7 NHNNUNLN | | | • | • |

OpenJFX Risk matrix

| CVE ID | Component | CVSSv3.1 Vector | Affects ... | | |
|--------|-----------|--------------------|-------------|----|----|
| | | | 11 | 17 | 21 |
| None | | | | | |

Acknowledgements

We acknowledge the following parties for their reports and contributions: Carter Kozak, and Dinglijie.

We also thank the Leads of the [JDK 8 Updates](#), [JDK 11 Updates](#), [JDK 17 Updates](#), and [OpenJFX](#) Projects for providing the risk-matrix information for their releases.

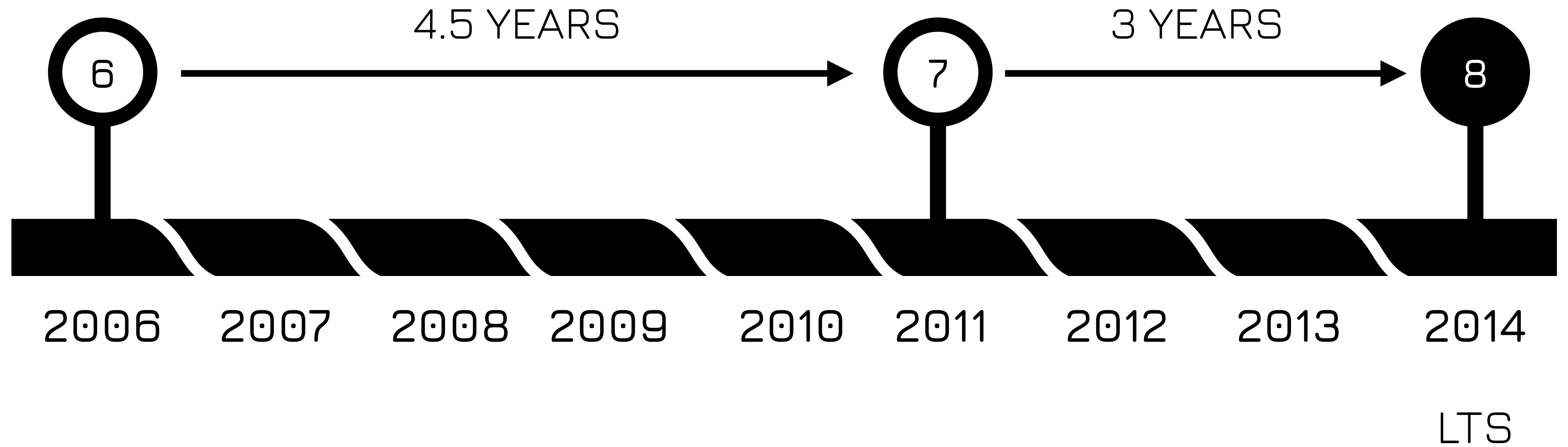
How to report a vulnerability

Please see the [reporting instructions](#) for information about how to report a vulnerability.

JAVA RELEASE CYCLE

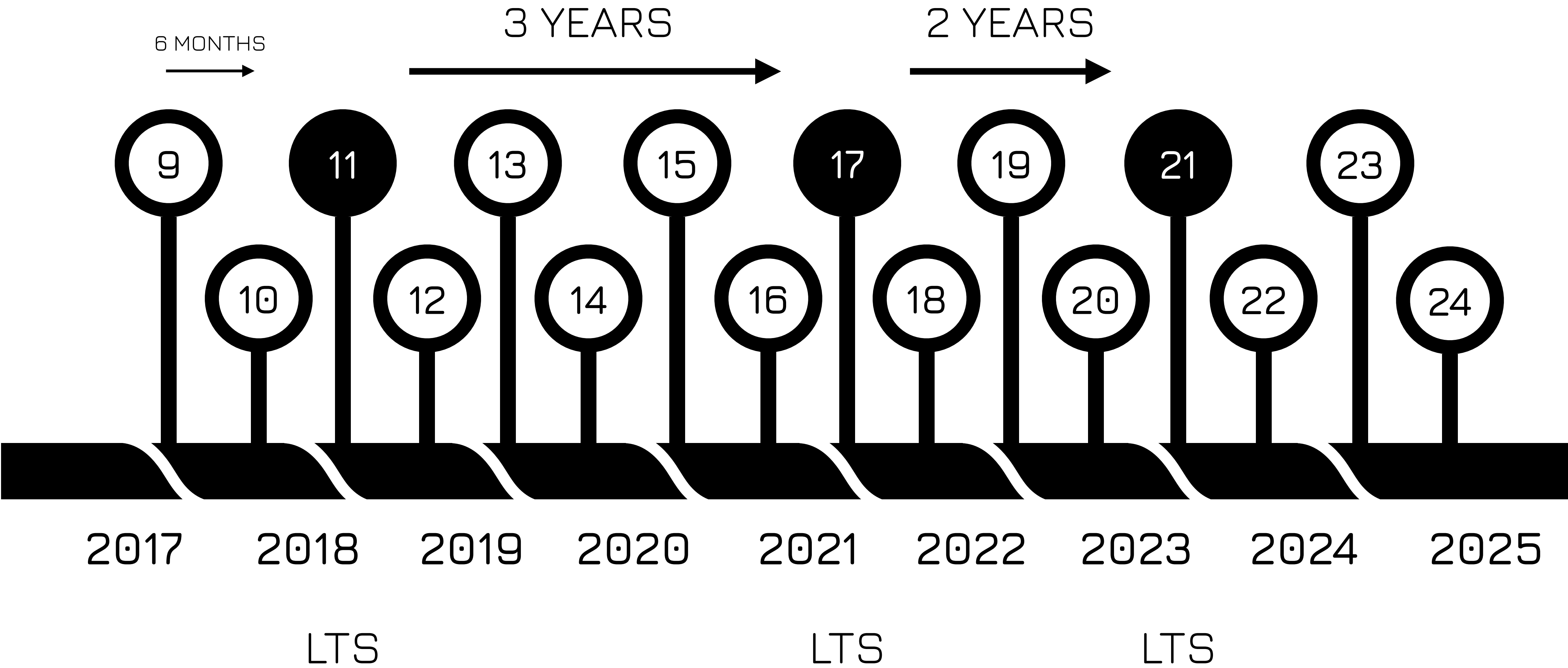
JAVA RELEASE CYCLE

Old Cadence



JAVA RELEASE CYCLE

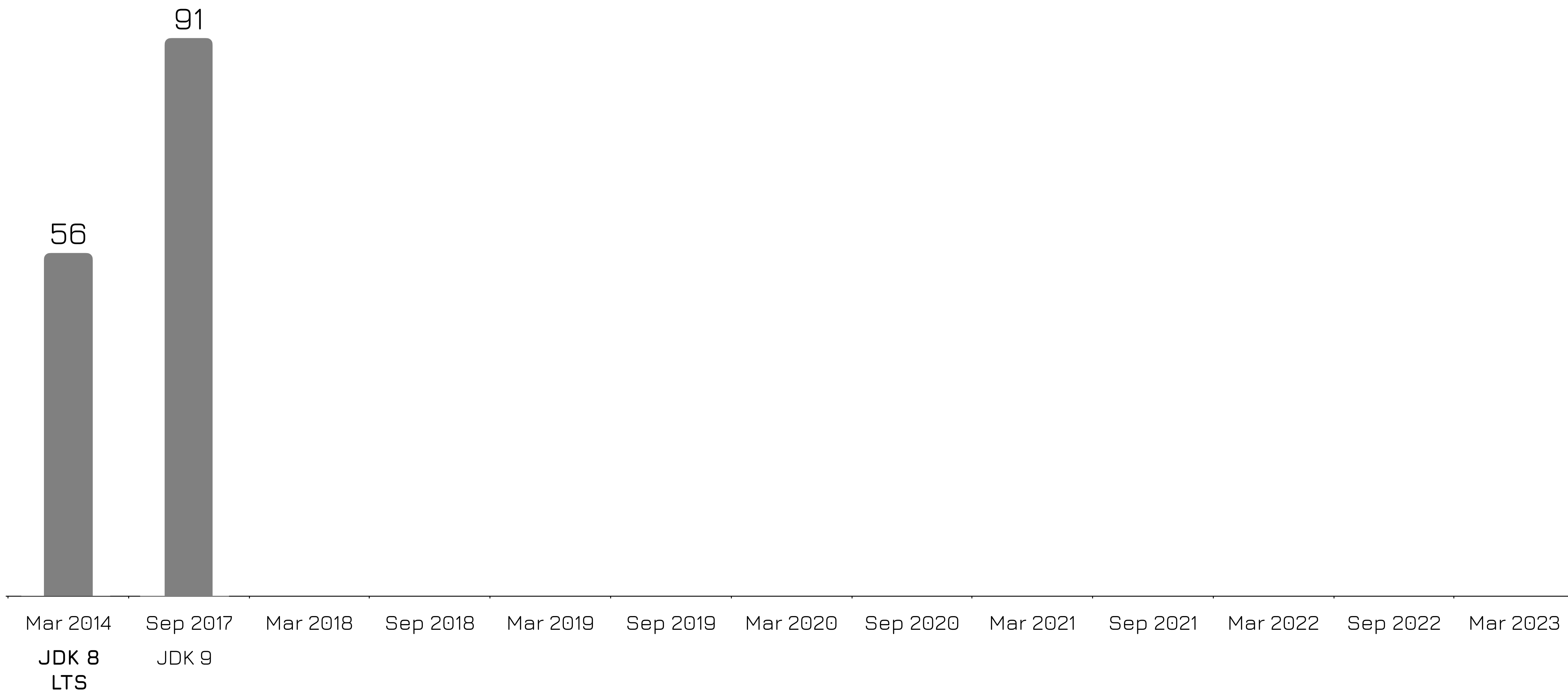
New Cadence



**BUT HOW DOES
THAT HELP ?**

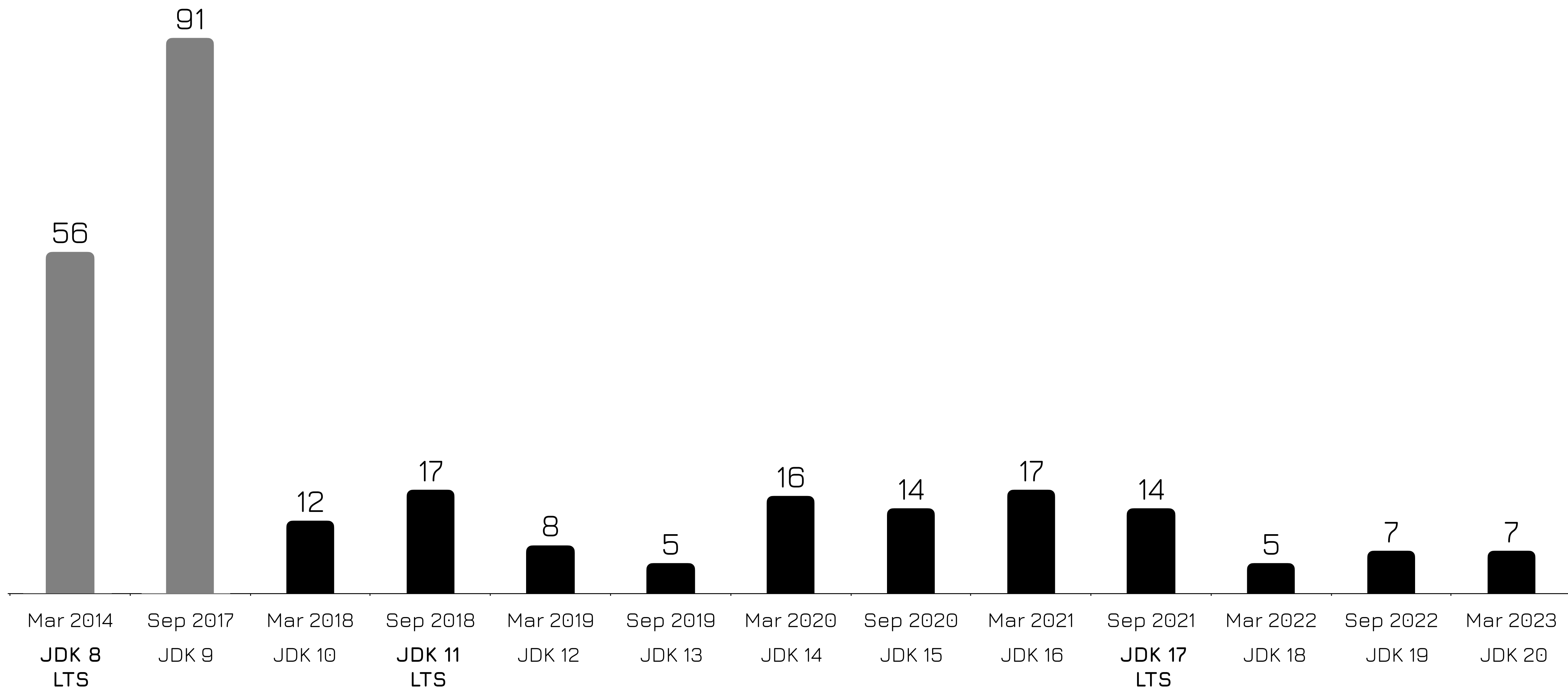
JAVA RELEASE CYCLE

Features per release



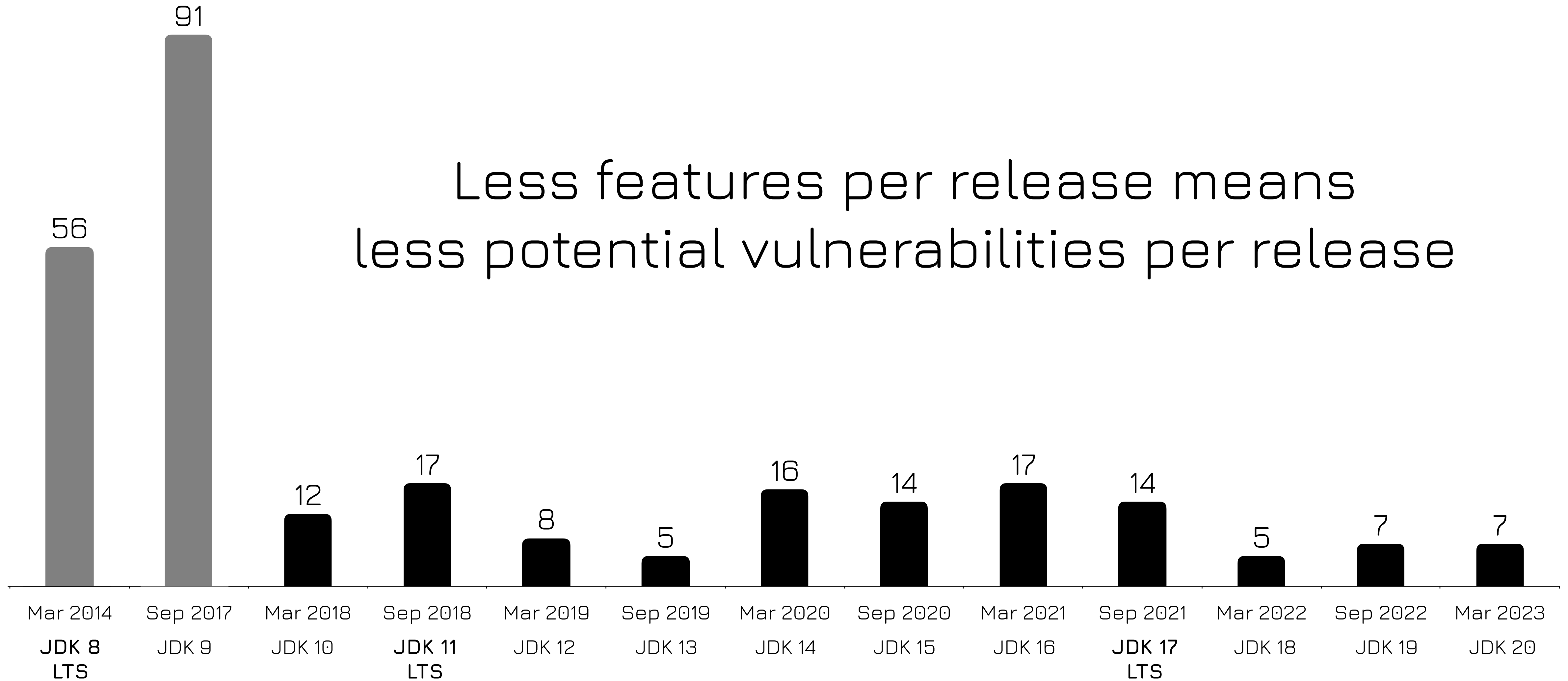
JAVA RELEASE CYCLE

Features per release



JAVA RELEASE CYCLE

Features per release



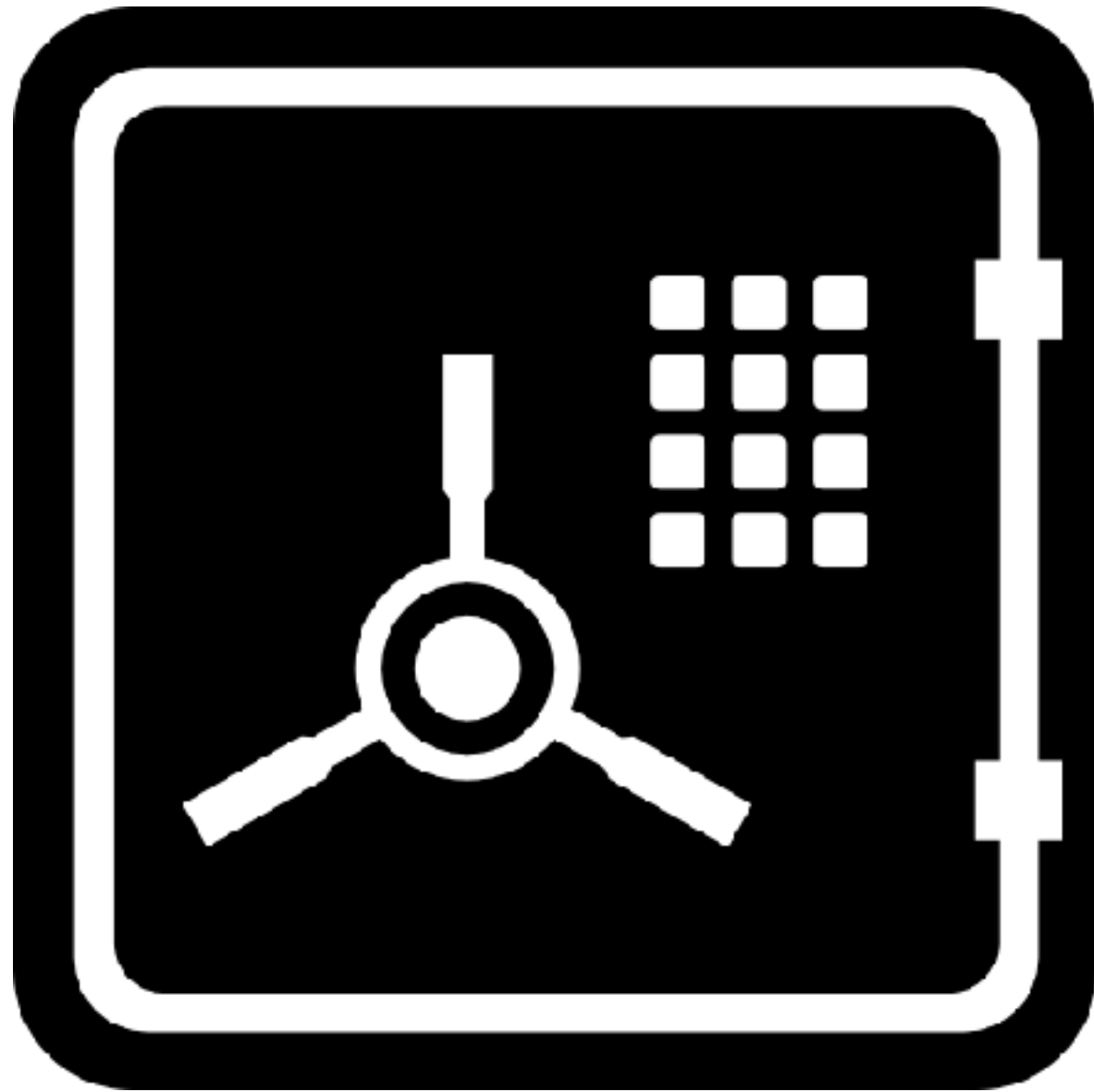
JAVA UPDATES

CPU

Critical Patch Update

CPU

Critical Patch Update



Safe to use in production

Contains

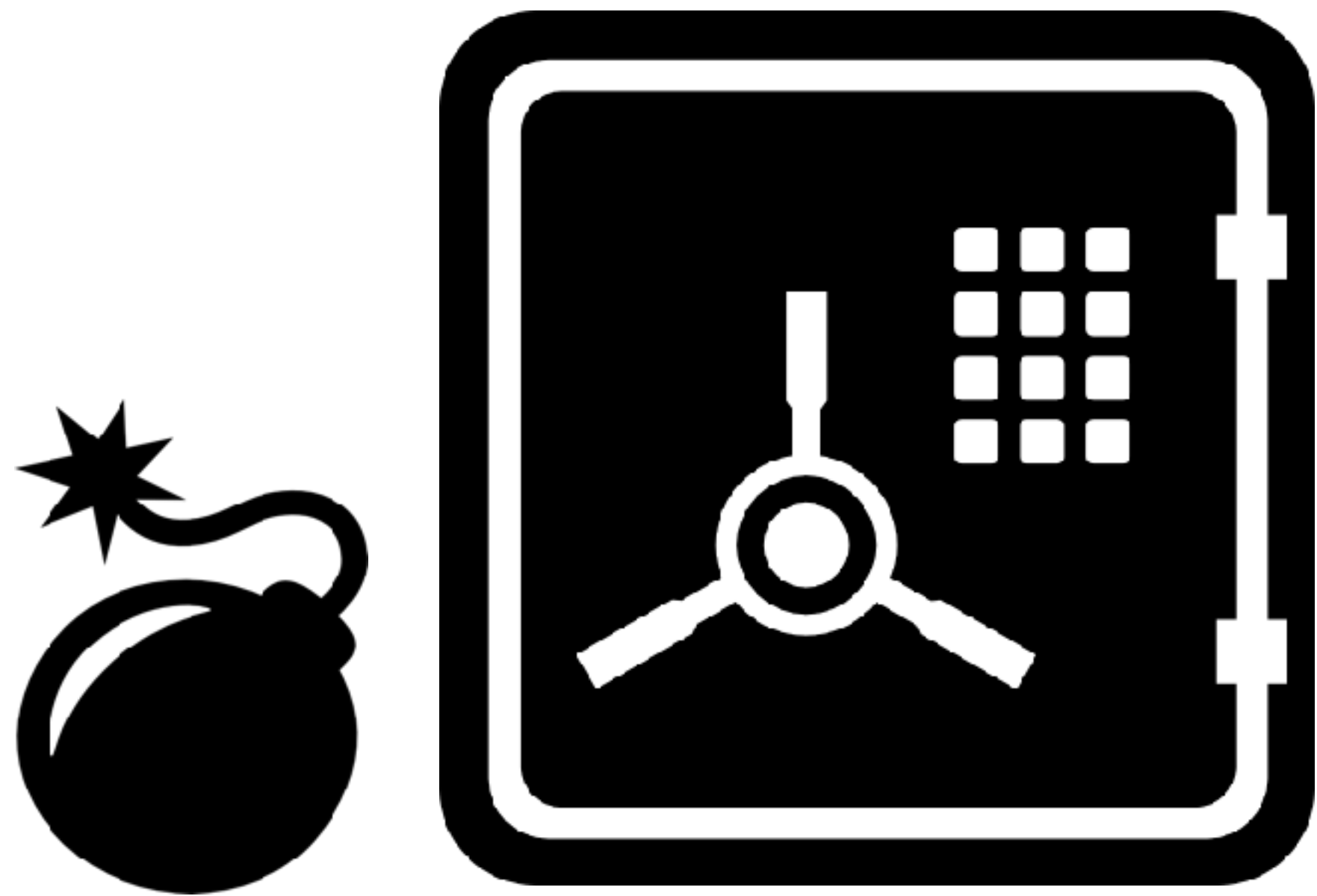
- ✦ Fixes vulnerabilities
- ✦ Fixes critical issues

PSU

Patch Set Update

PSU

Patch Set Update



Could possibly introduce new vulnerabilities !!!

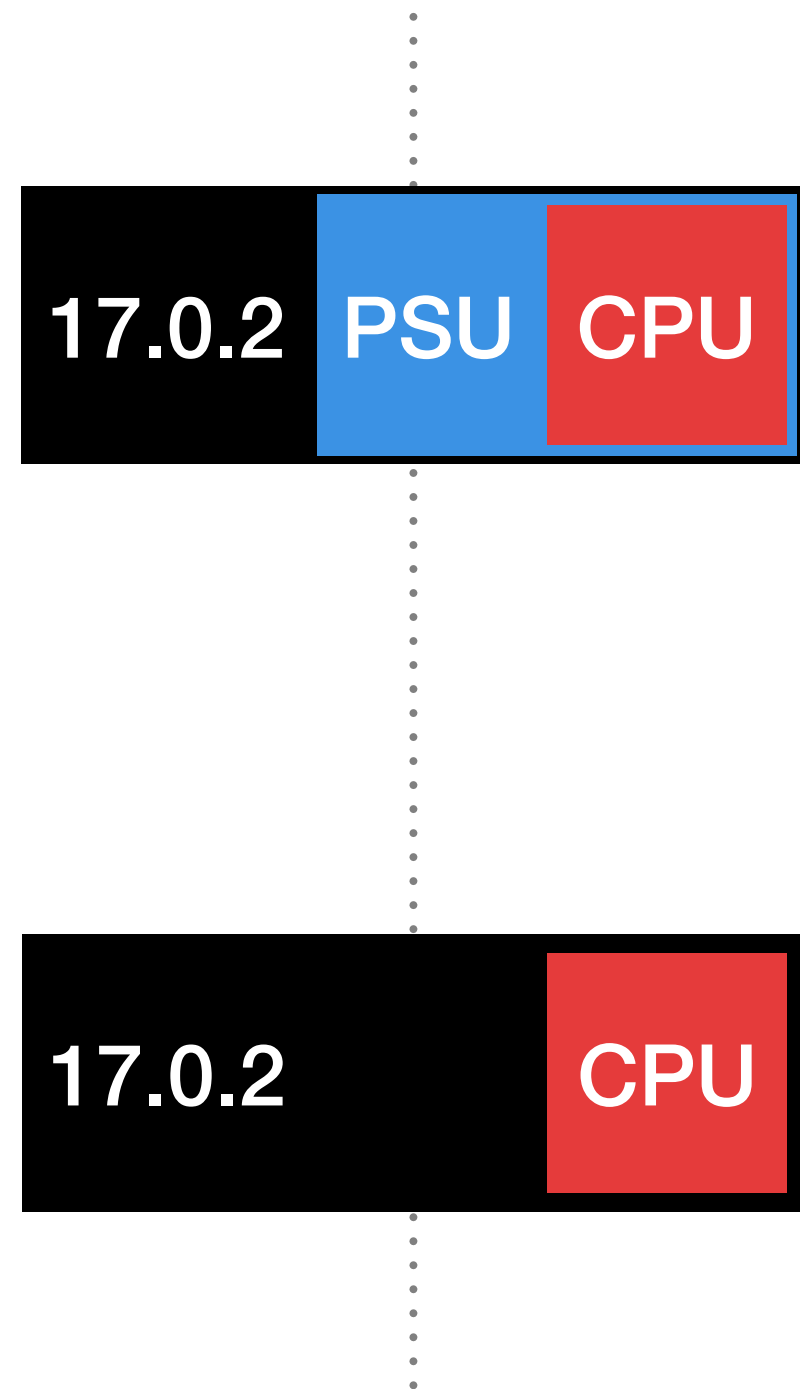
Superset of CPU

Contains

- ✦ Fixes vulnerabilities
- ✦ Fixes critical issues
- ✦ Fixes non critical issues
- ✦ New features

UPDATES

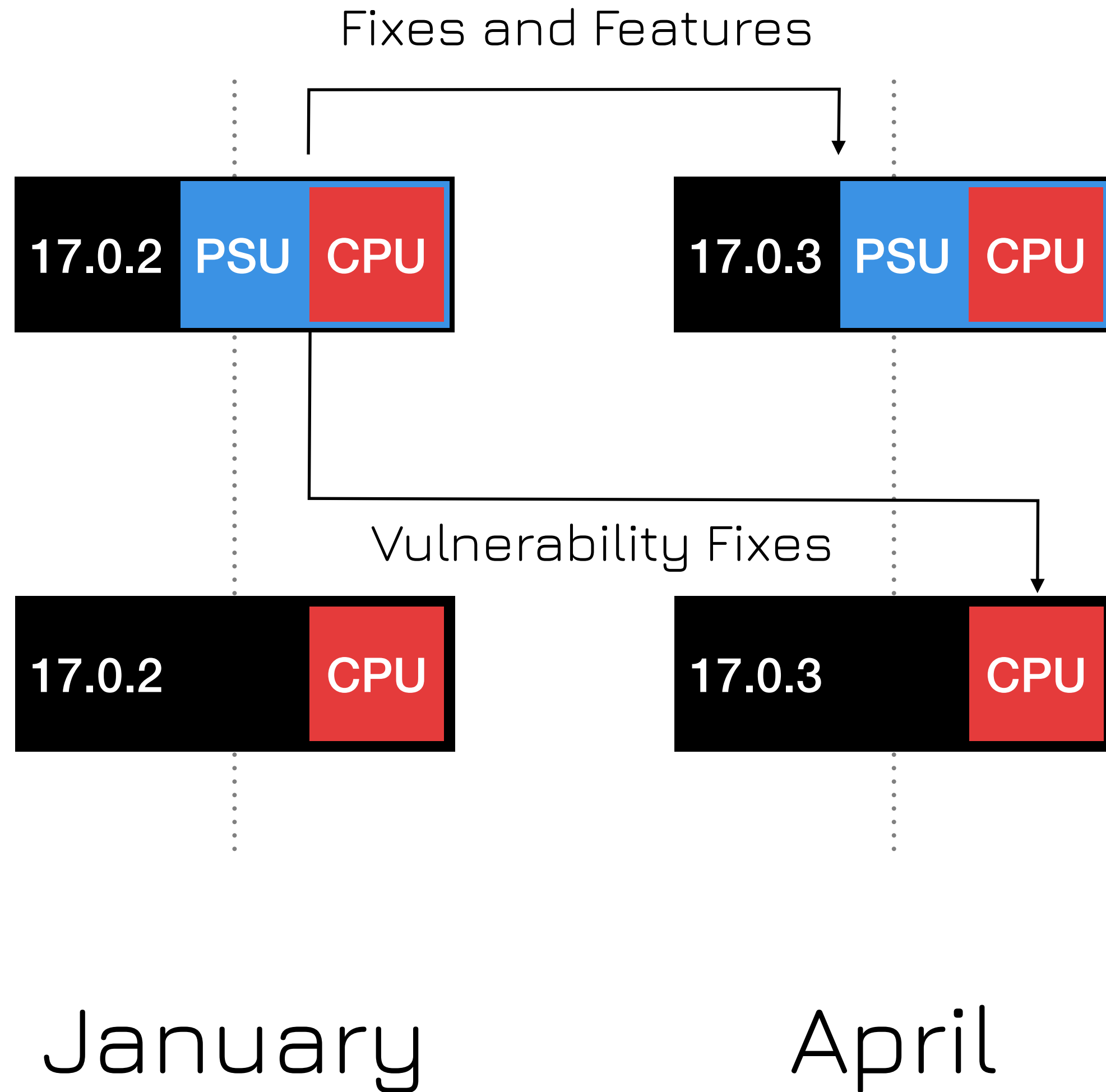
Four times a year



January

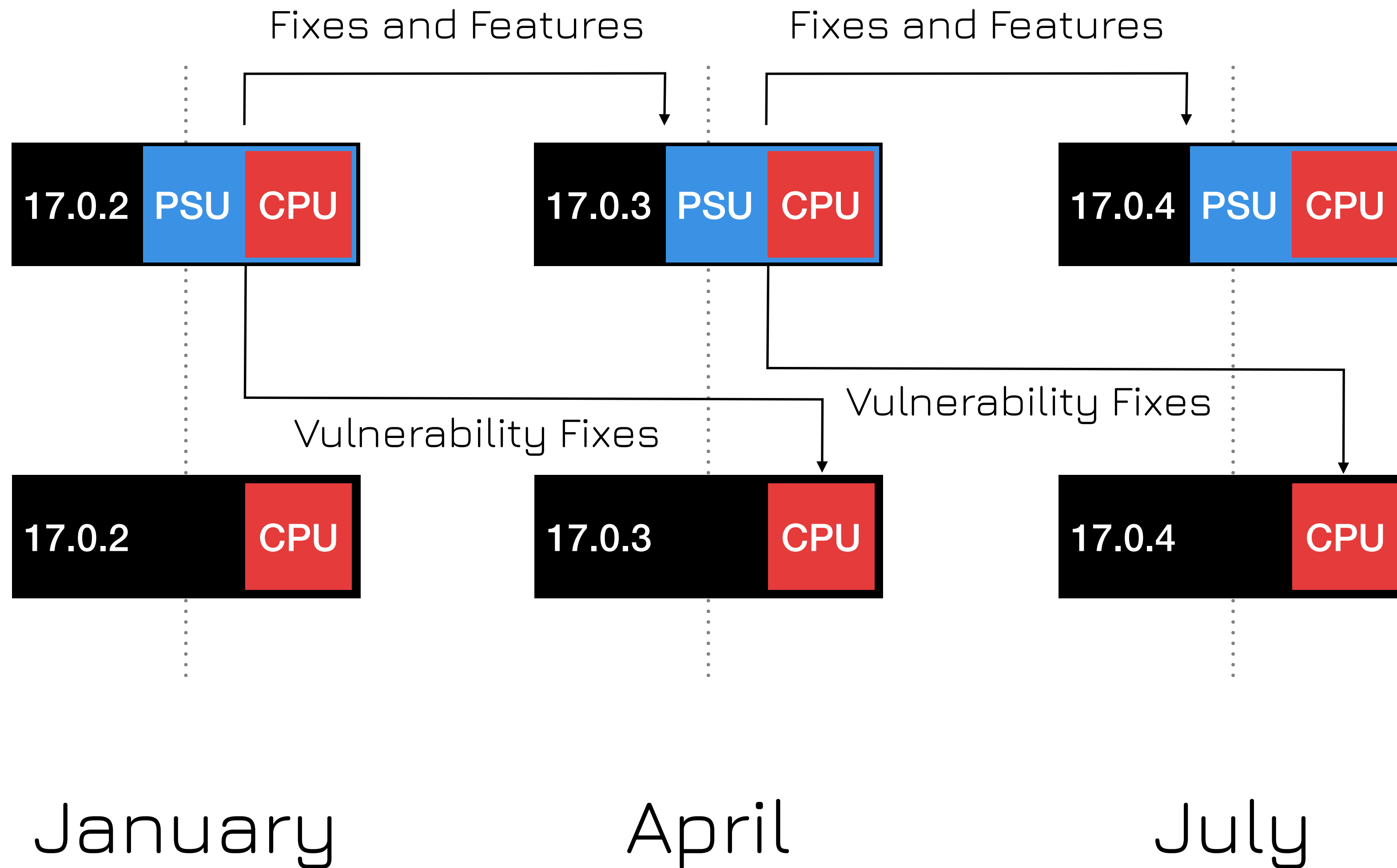
UPDATES

Four times a year



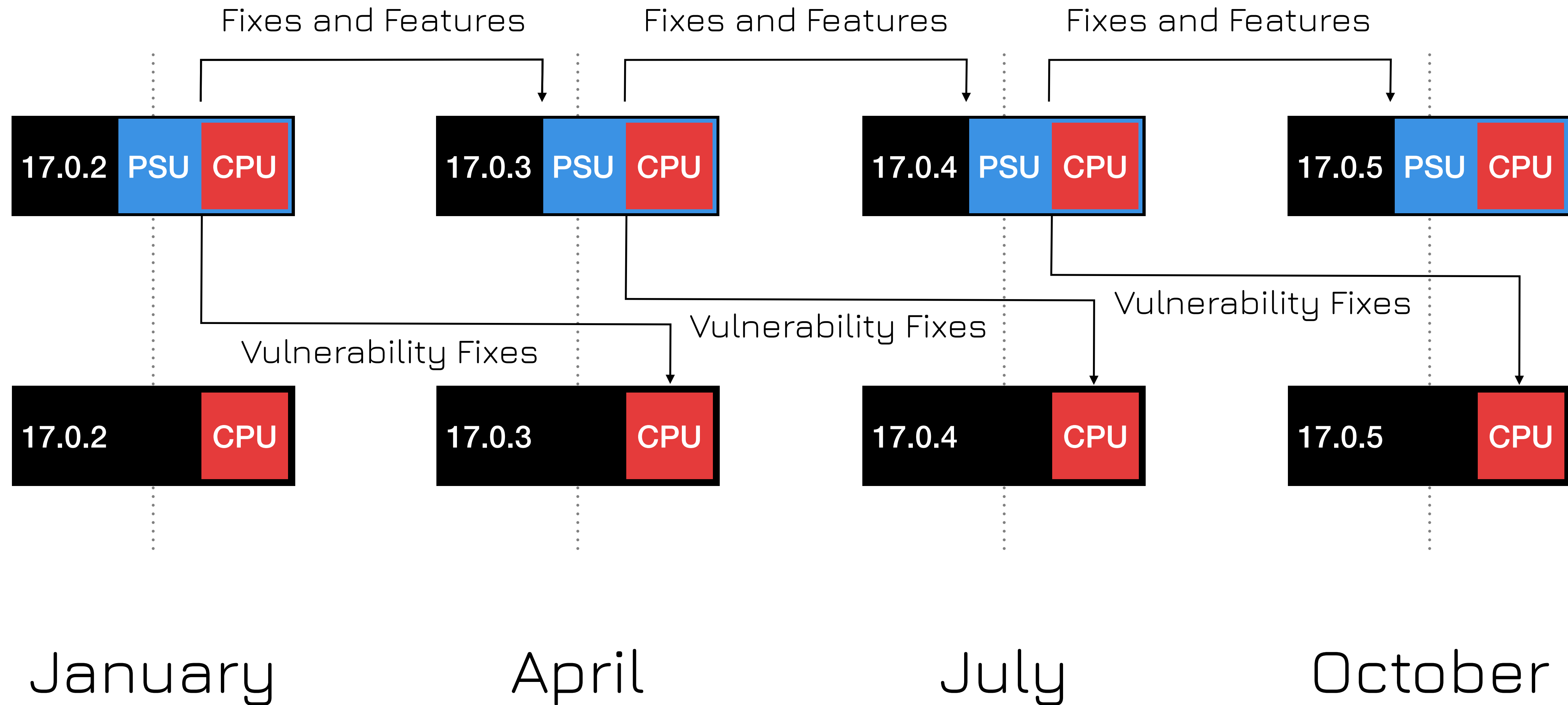
UPDATES

Four times a year



UPDATES

Four times a year



UPDATES

Keep in mind

- ✦ Updates are available 4 times a year (every 3 months starting from January)
- ✦ Patch Set Updates (PSU) contains the CPU plus non-critical fixes and small features
- ✦ Critical Patch Updates (CPU) contain only critical vulnerability fixes and are feature-wise always one step behind the PSU

UPDATES

Why CPUs matter

- ✦ PSU 8u252 introduced a change that prevented Hadoop cluster and Solr from running
- ✦ CPU 8u251 only contained security fixes from PSU 8u242 and did not introduce this change

IMPACT

WITHOUT

UPDATES

JDK 17

14.09.2021

17.0.0

19.10.2021

17.0.1

| | |
|----------------|-----|
| CVE-2021-35567 | 6.8 |
| CVE-2021-35586 | 5.9 |
| CVE-2021-35564 | 5.3 |
| CVE-2021-35561 | 5.3 |
| CVE-2021-35559 | 5.3 |
| CVE-2021-35578 | 5.3 |
| CVE-2021-35556 | 5.3 |
| CVE-2021-35603 | 3.7 |

8 CVE's

18.01.2022

17.0.2

| | |
|----------------|-----|
| CVE-2022-21341 | 5.3 |
| CVE-2022-21365 | 5.3 |
| CVE-2022-21282 | 5.3 |
| CVE-2022-21291 | 5.3 |
| CVE-2022-21277 | 5.3 |
| CVE-2022-21305 | 5.3 |
| CVE-2022-21299 | 5.3 |
| CVE-2022-21296 | 5.3 |
| CVE-2022-21283 | 5.3 |
| CVE-2022-21340 | 5.3 |
| CVE-2022-21293 | 5.3 |
| CVE-2022-21294 | 5.3 |
| CVE-2022-21360 | 5.3 |
| CVE-2022-21366 | 5.3 |
| CVE-2022-21248 | 3.7 |

15 CVE's

19.04.2022

17.0.3

| | |
|----------------|-----|
| CVE-2022-21449 | 7.5 |
| CVE-2022-21496 | 5.3 |
| CVE-2022-21434 | 5.3 |
| CVE-2022-21426 | 5.3 |
| CVE-2022-21443 | 3.7 |

5 CVE's

19.07.2022

17.0.4

| | |
|----------------|-----|
| CVE-2022-34169 | 7.5 |
| CVE-2022-21541 | 5.9 |
| CVE-2022-21540 | 5.3 |

3 CVE's

18.10.2022

17.0.5

| | |
|----------------|-----|
| CVE-2022-21618 | 5.3 |
| CVE-2022-21628 | 5.3 |
| CVE-2022-39399 | 3.7 |
| CVE-2022-21619 | 3.7 |
| CVE-2022-21624 | 3.7 |

5 CVE's

17.01.2023

17.0.6

| | |
|----------------|-----|
| CVE-2023-21835 | 5.3 |
| CVE-2023-21843 | 3.7 |

2 CVE's

18.04.2023

17.0.7

| | |
|----------------|-----|
| CVE-2023-21930 | 7.4 |
| CVE-2023-21954 | 5.9 |
| CVE-2023-21967 | 5.9 |
| CVE-2023-21939 | 5.3 |
| CVE-2023-21938 | 3.7 |
| CVE-2023-21937 | 3.7 |
| CVE-2023-21968 | 3.7 |

7 CVE's

18.07.2023

17.0.8

| | |
|----------------|-----|
| CVE-2023-22041 | 5.1 |
| CVE-2023-25193 | 3.7 |
| CVE-2023-22044 | 3.7 |
| CVE-2023-22045 | 3.7 |
| CVE-2023-22049 | 3.7 |
| CVE-2023-22036 | 3.7 |
| CVE-2023-22006 | 3.1 |

7 CVE's

17.10.2023

17.0.9

| | |
|----------------|-----|
| CVE-2023-22081 | 5.3 |
| CVE-2023-22025 | 3.7 |

2 CVE's

IMPACT WITHOUT UPDATES

JDK 17

54

If you stick to 17.0.0 you are vulnerable to 54 CVE's !!!

**IF IT AIN'T BROKE
DON'T FIX IT ?**

**IF IT AIN'T BROKE
AT LEAST KEEP IT
UP TO DATE !**

MODULAR RUNTIME IMAGES

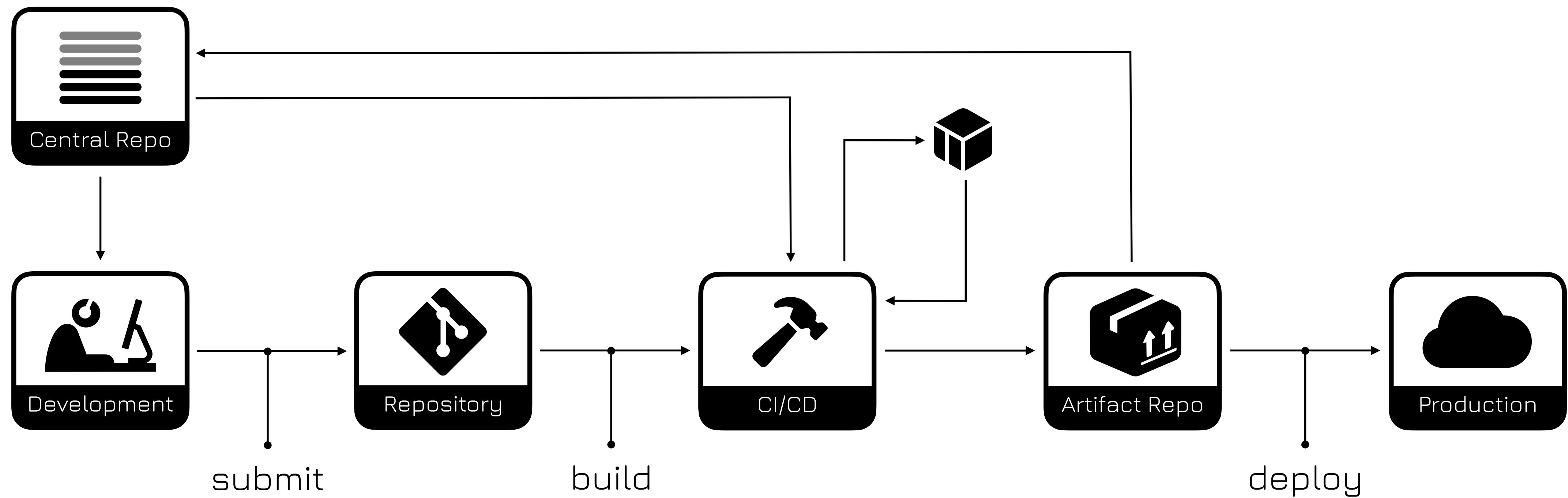
MODULAR RUNTIME IMAGES

JLink

- ✦ Reducing risk by removing modules
- ✦ JLink makes this possible
- ✦ Removing unused modules means reducing risk for vulnerabilities
- ✦ Hackers cannot attack what isn't there

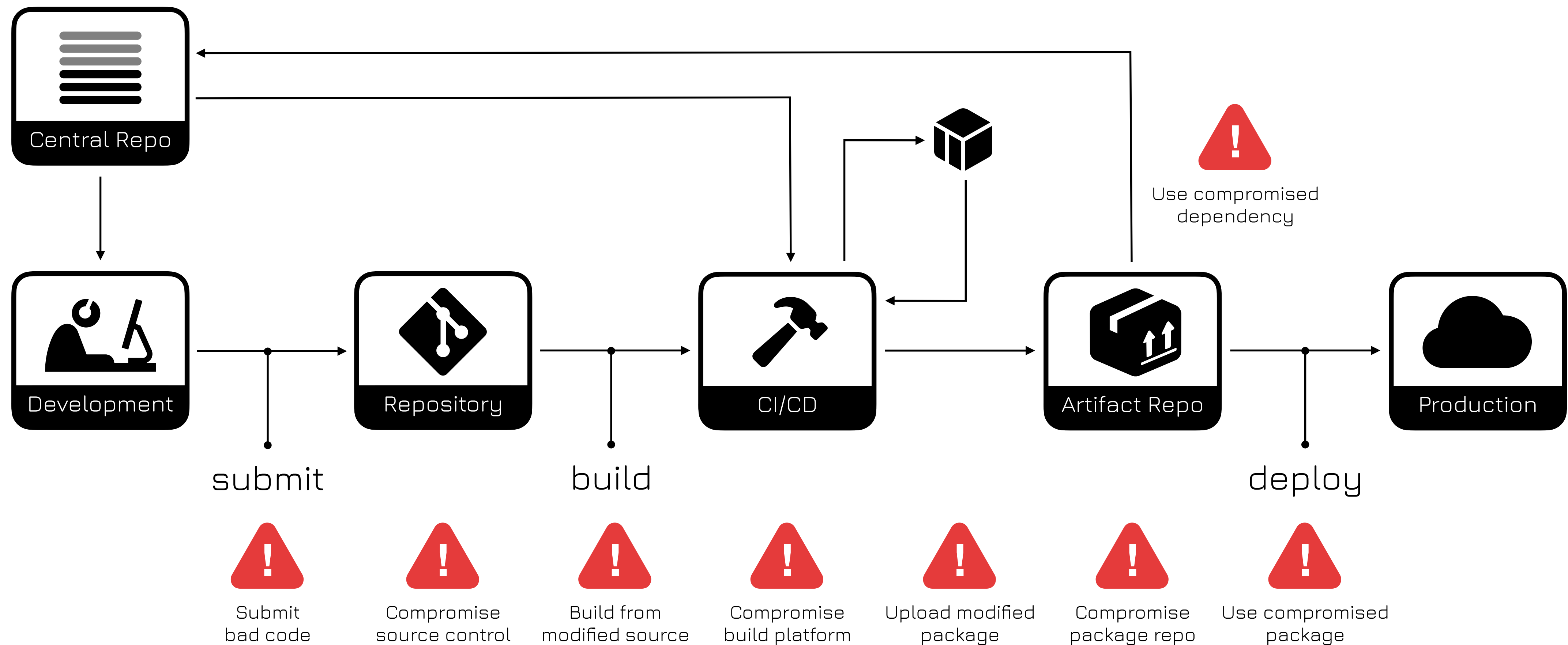
SOFTWARE SUPPLY CHAIN

SOFTWARE SUPPLY CHAIN



SOFTWARE SUPPLY CHAIN

And it's vulnerabilities



SOME FACTS

ATTACKS

Software Supply Chain attacks



742%

Increase over
the past

3

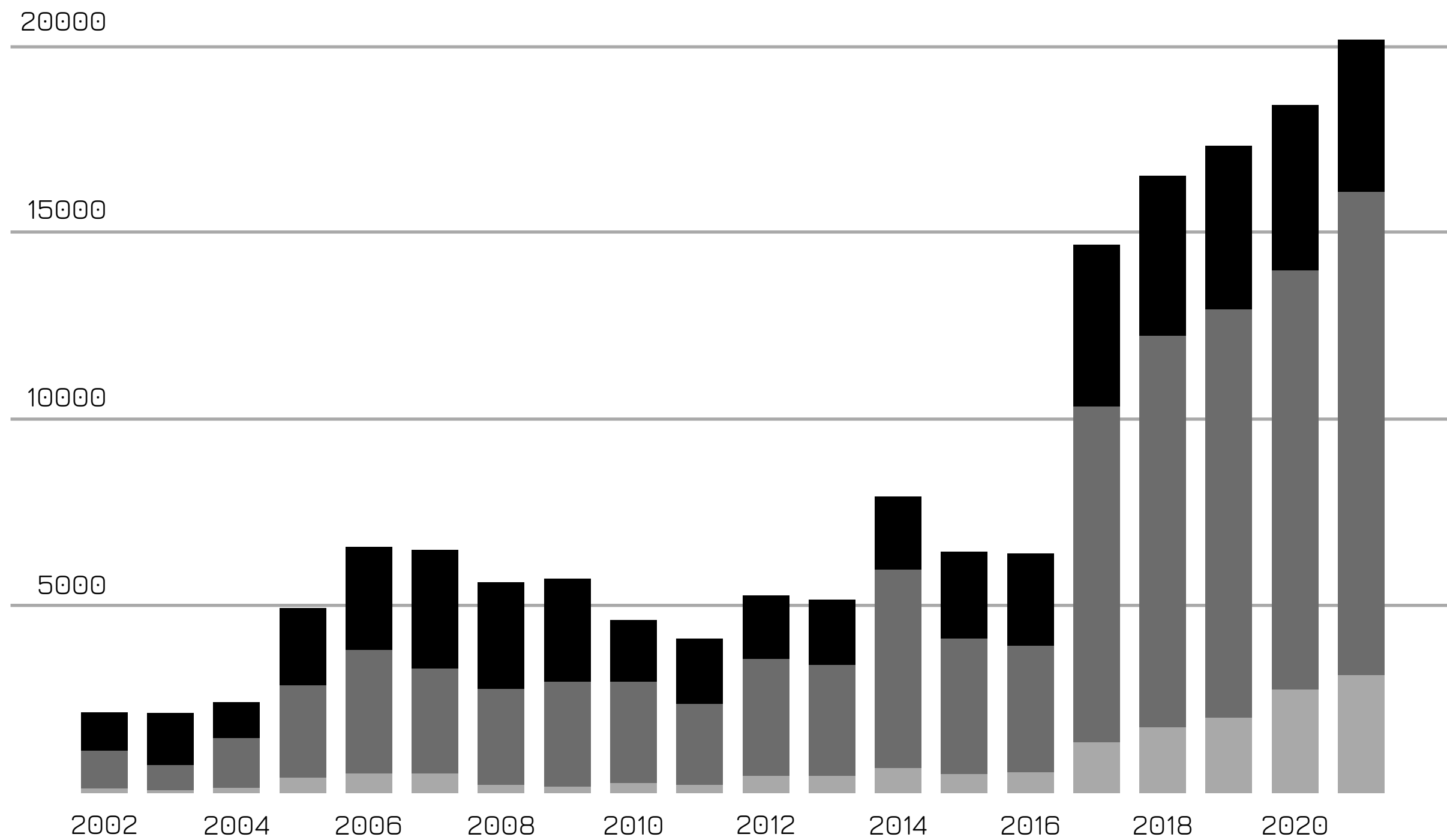
years

(Sonatype State of the Software Chain report)

VULNERABILITIES

Distribution by severity over time

Low Medium High

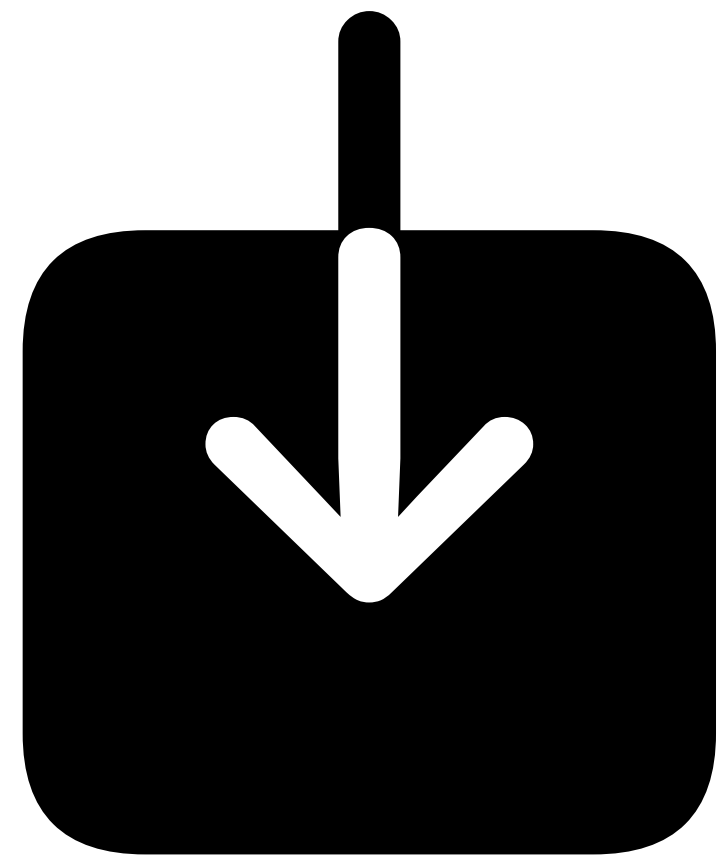


(NIST National Vulnerability Database)

The year 2021 saw
20 142
unique bugs and
security vulnerabilities
recorded

USER LAZINESS

Downloaded versions of Log4j



20%

(Christian Grobmeier, Log4j maintainer)

Of all Log4j downloads

20%

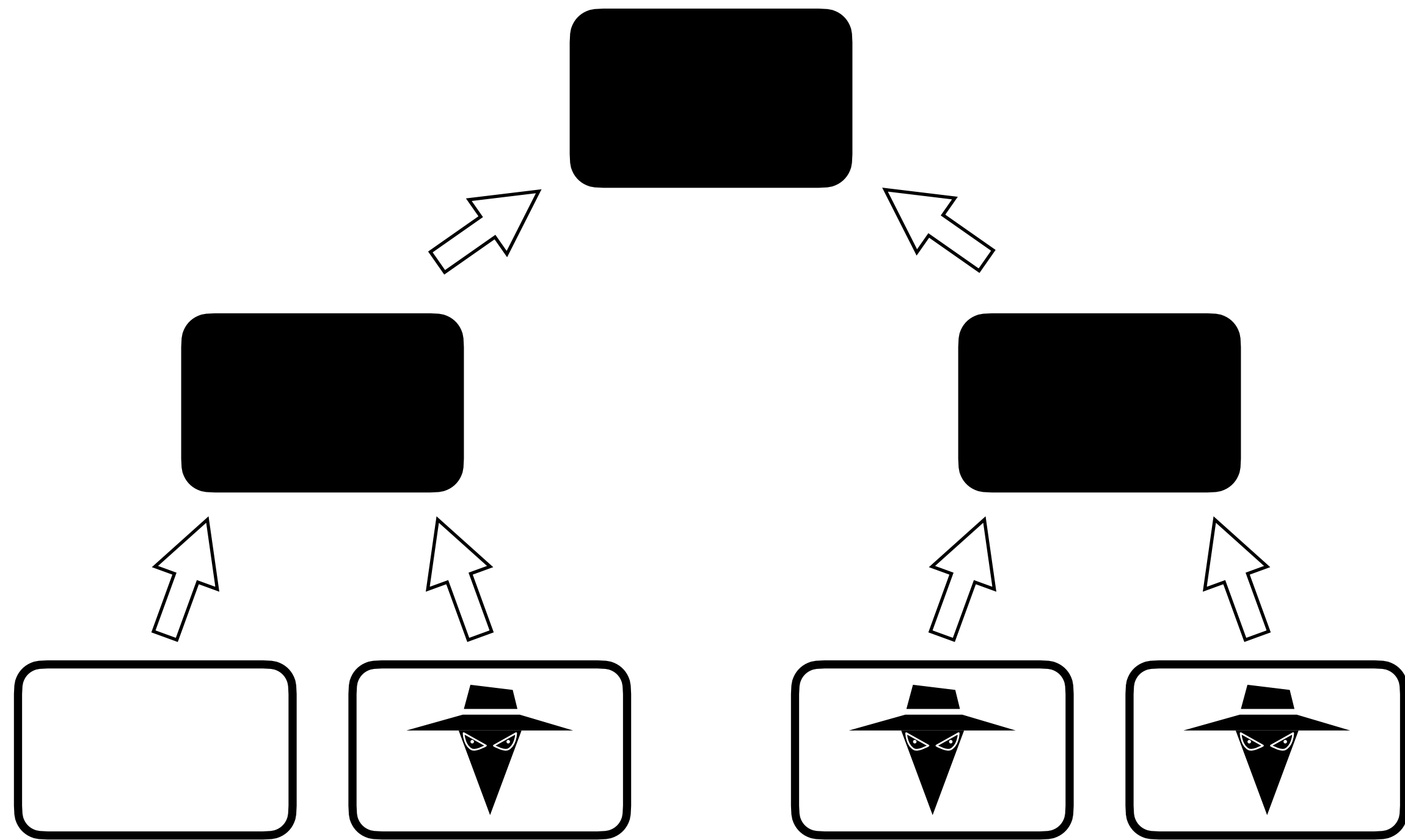
are still vulnerable to
CVE 2021-44228,
even

21 months

after Log4j has
been patched!

VULNERABILITIES

Transitive dependencies

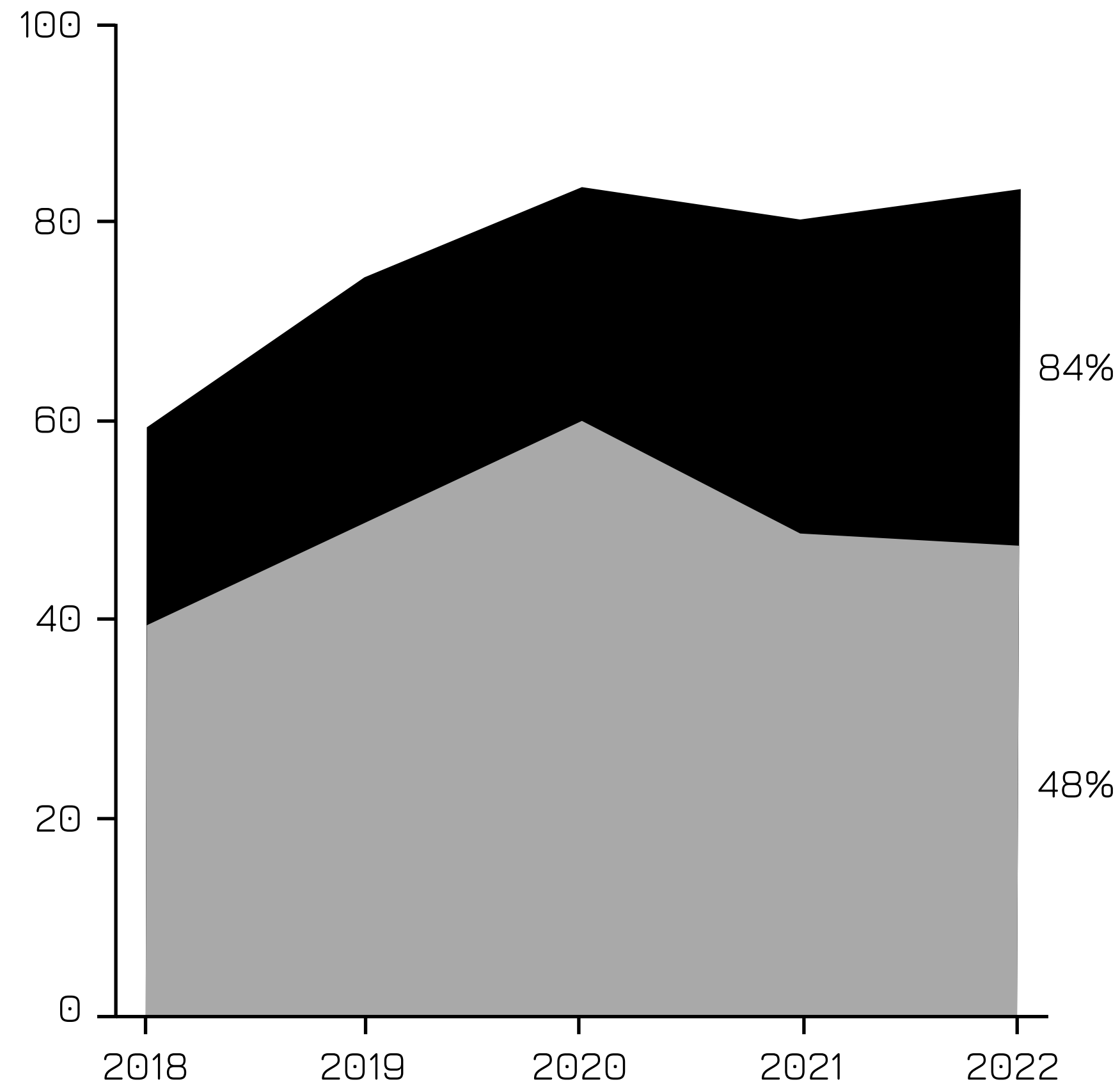


[Sonatype State of the Software Chain report]

About
6 out of 7
project vulnerabilities
come from transitive
dependencies

SECURITY RISK

Is prevalent



(Synopsys OSS security and risk analysis report)

At least one
vulnerability in

84%

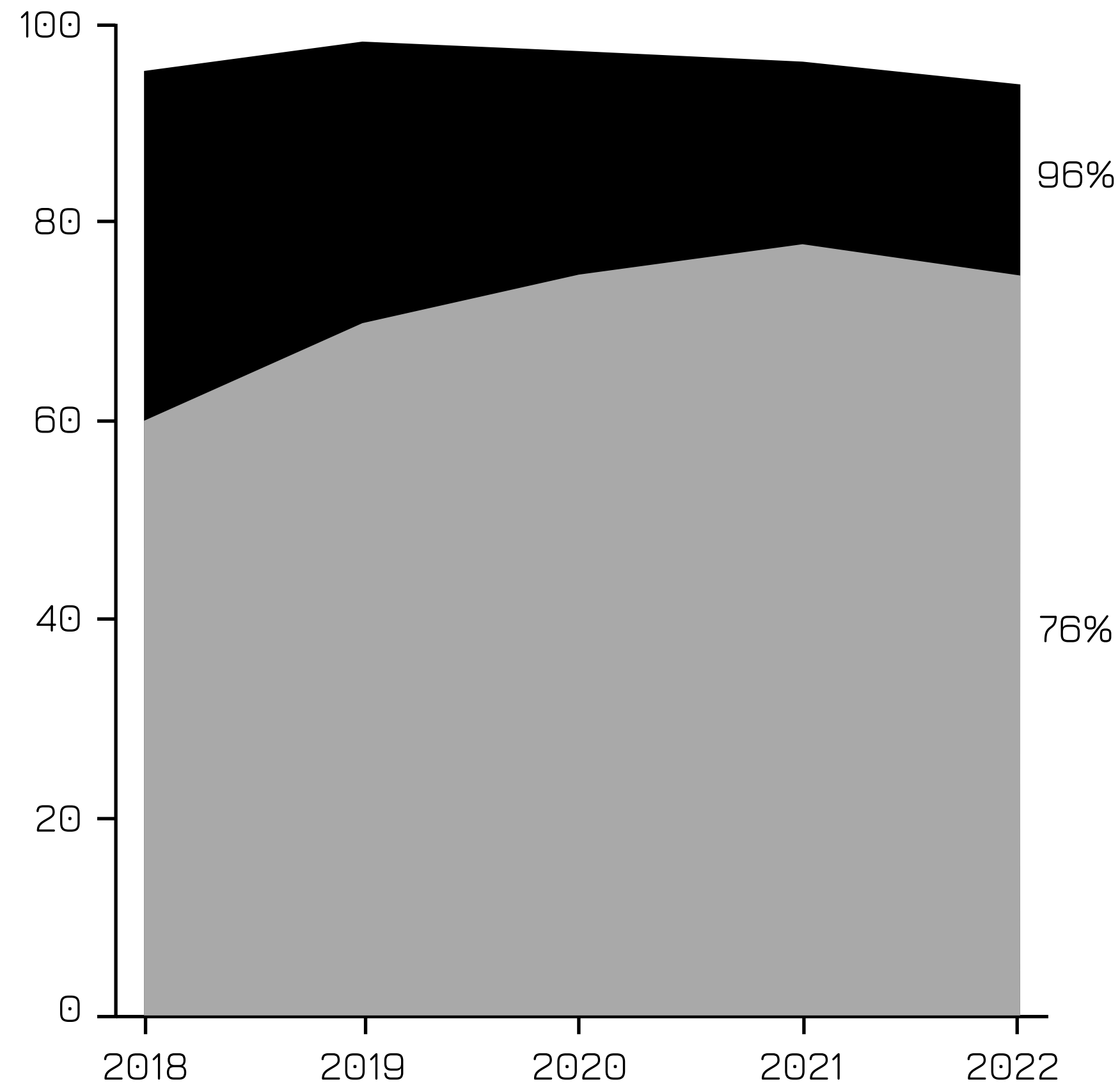
of all scanned
codebases and

48%

contained high-risk
vulnerabilities

OPEN SOURCE

Is everywhere



(Synopsys OSS security and risk analysis report)

Open Source used in

96%

of all scanned
codebases and

76%

of code in codebases
was Open Source

OPEN SOURCE

LICENSE

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LICENSE

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

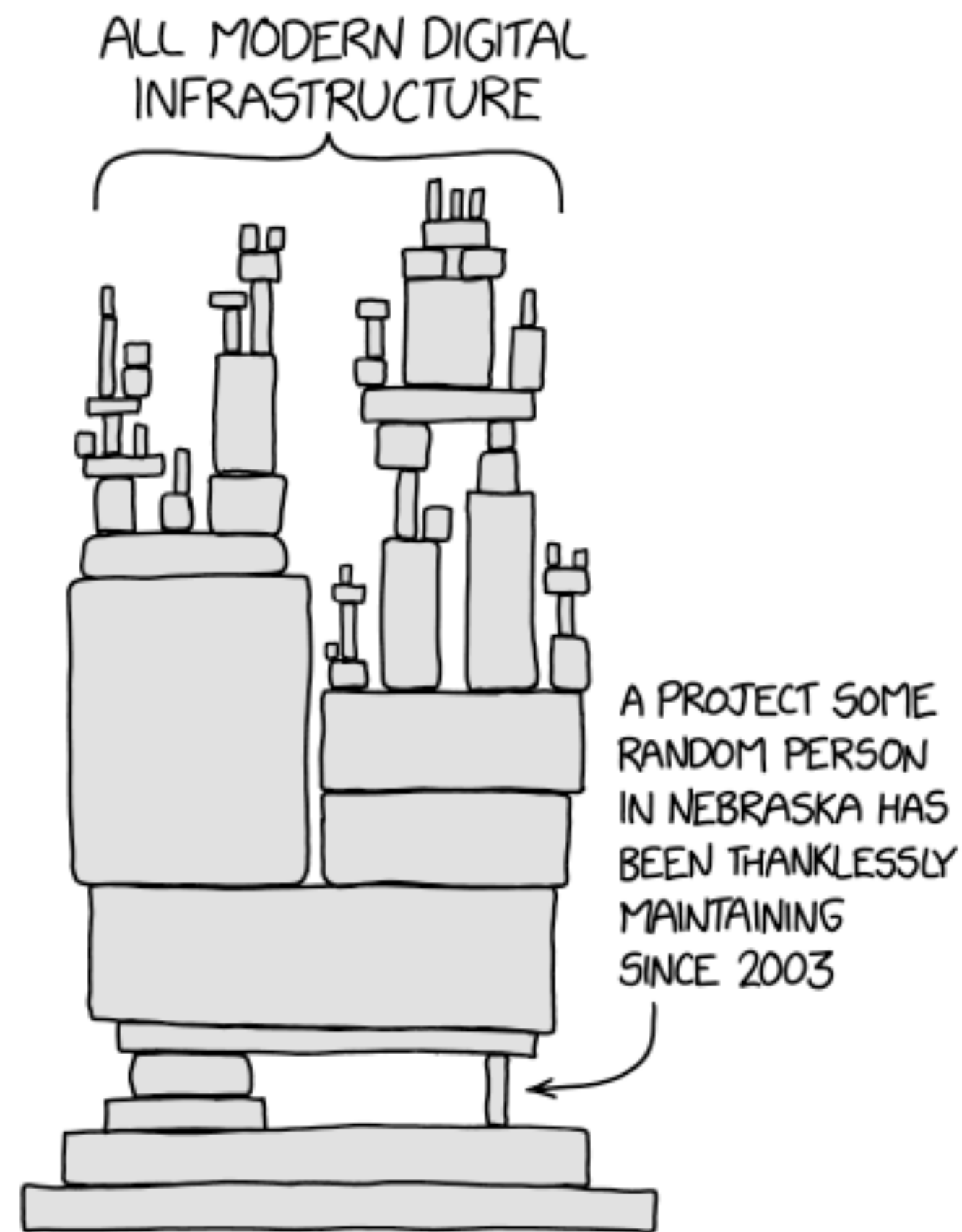
MEANS...

I OWE YOU

NOTHING !

OPEN SOURCE

Providers...not suppliers



<https://xkcd.com/2347/>

Keep in mind that:

OpenSource maintainer
are not suppliers !

You don't have a business
relationship with them !

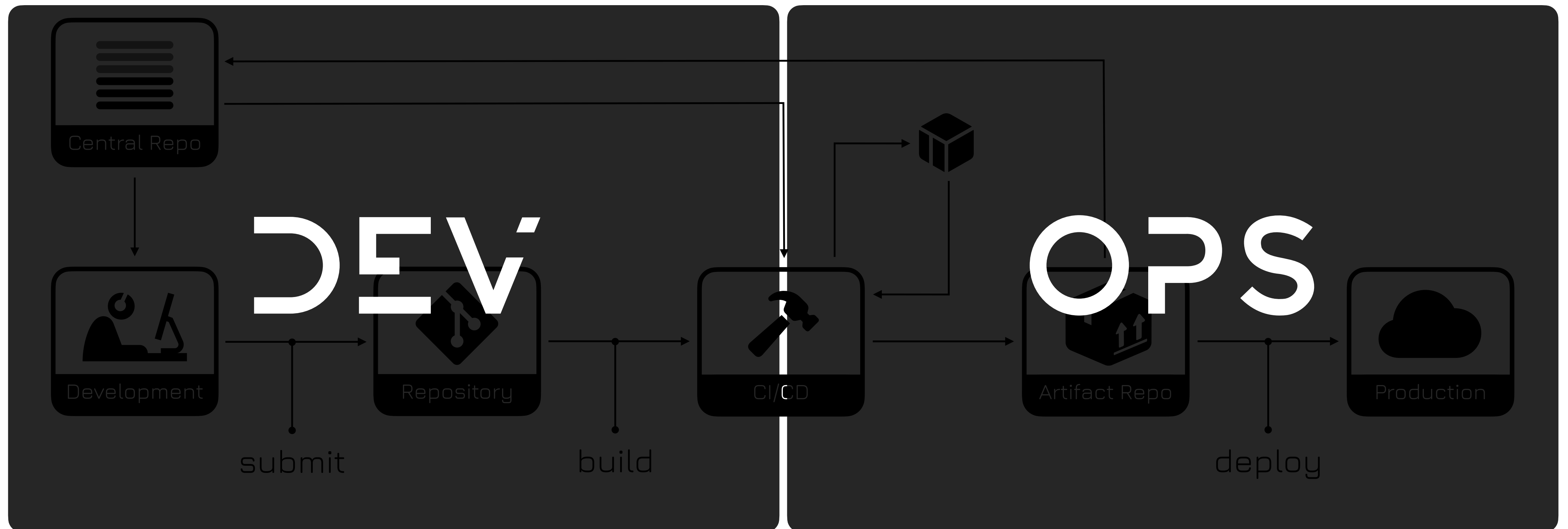
If you use their code, it's
up to you to make sure it's
up to date and secure !

WHAT CAN

WE DO ?

SHIFT LEFT

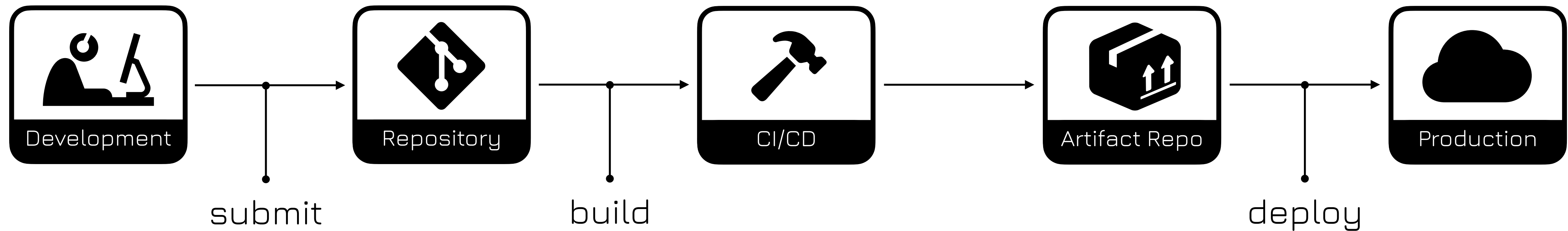
SHIFT LEFT



Software Supply Chain

SHIFT LEFT

Security should start more on the left side of the diagram



Software Supply Chain

DESIGN OPS

DEV OPS LOOP

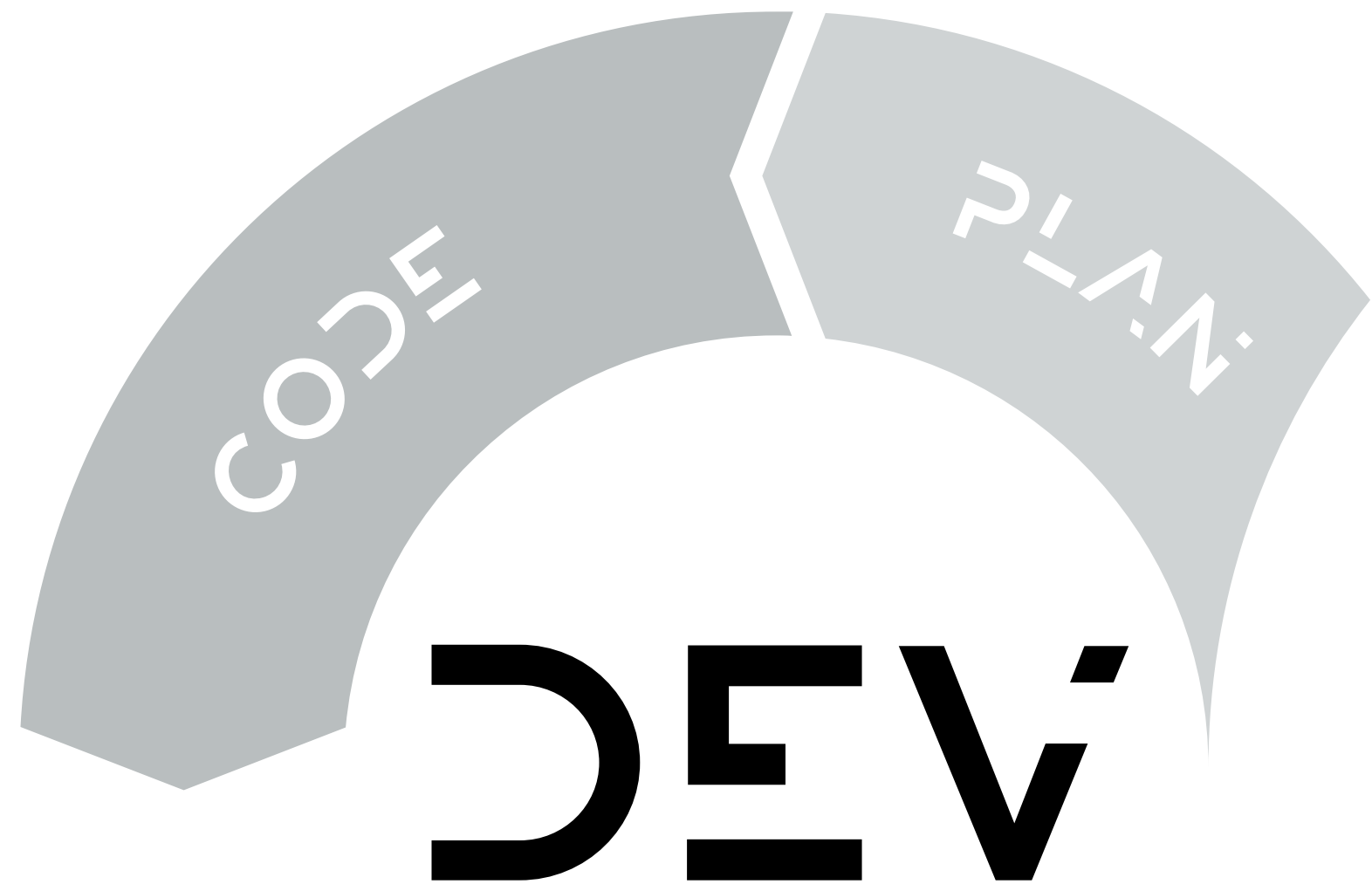
Plan

DEV

PLAN

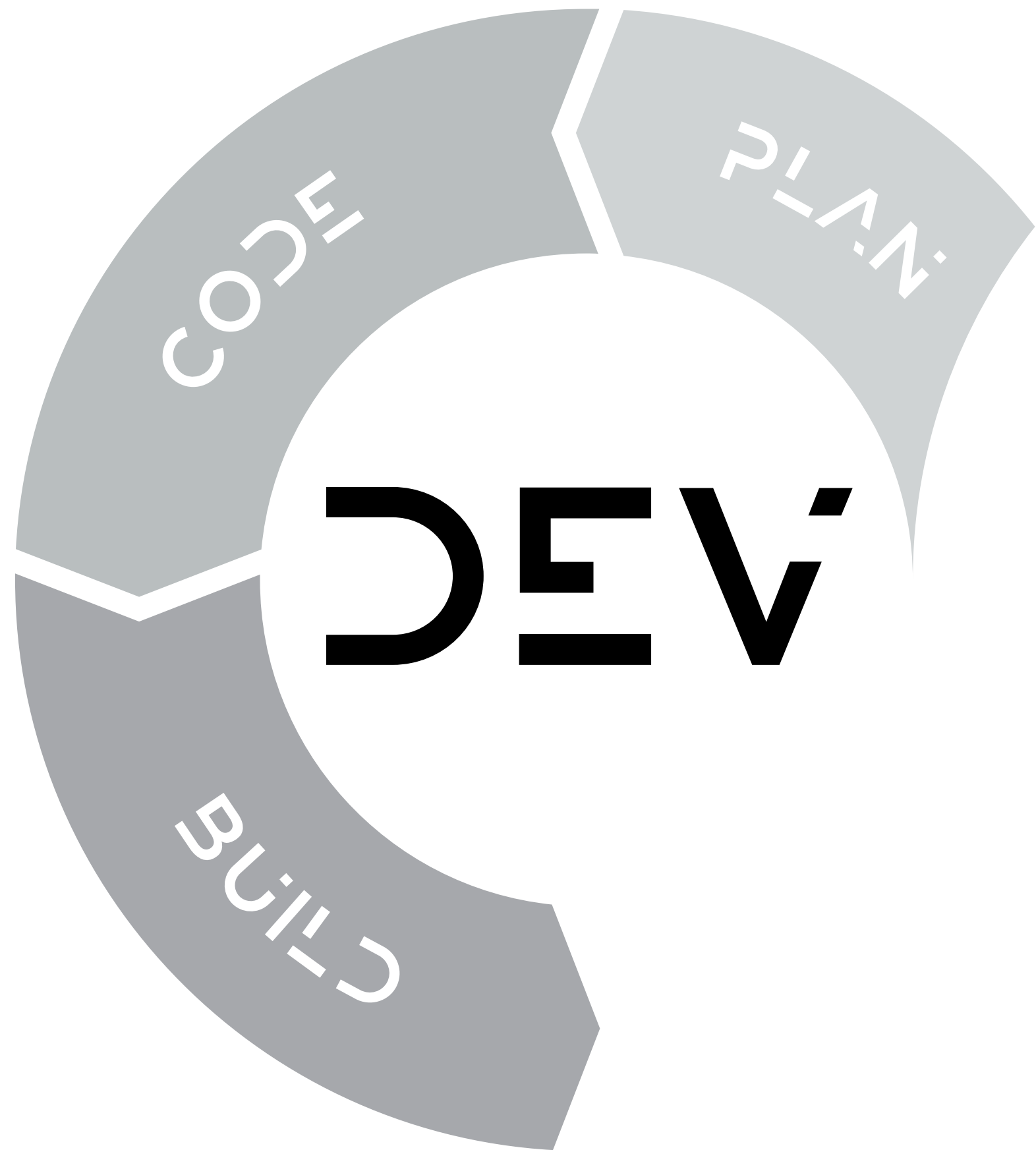
DEV OPS LOOP

Code



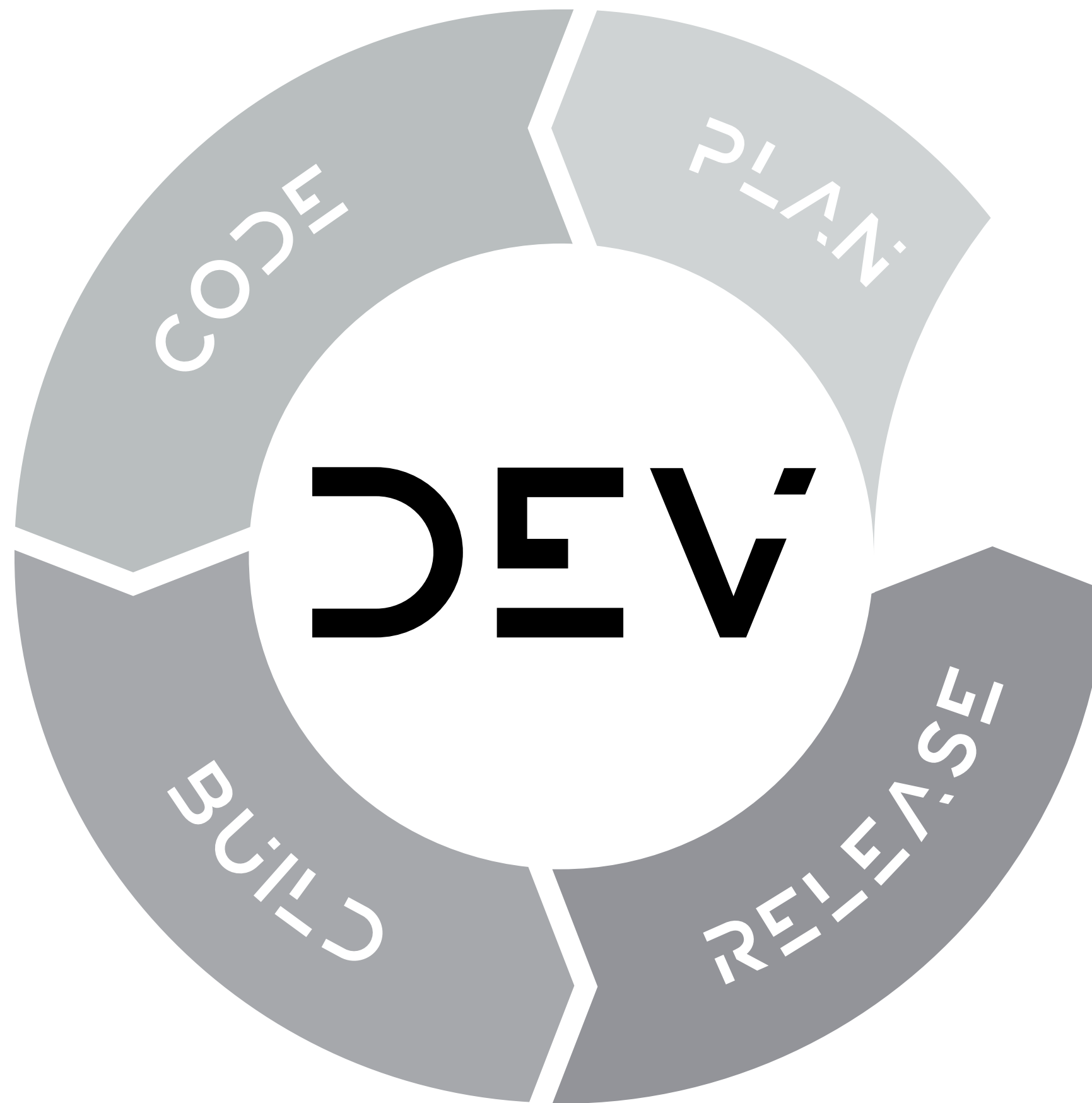
DEV OPS LOOP

Build



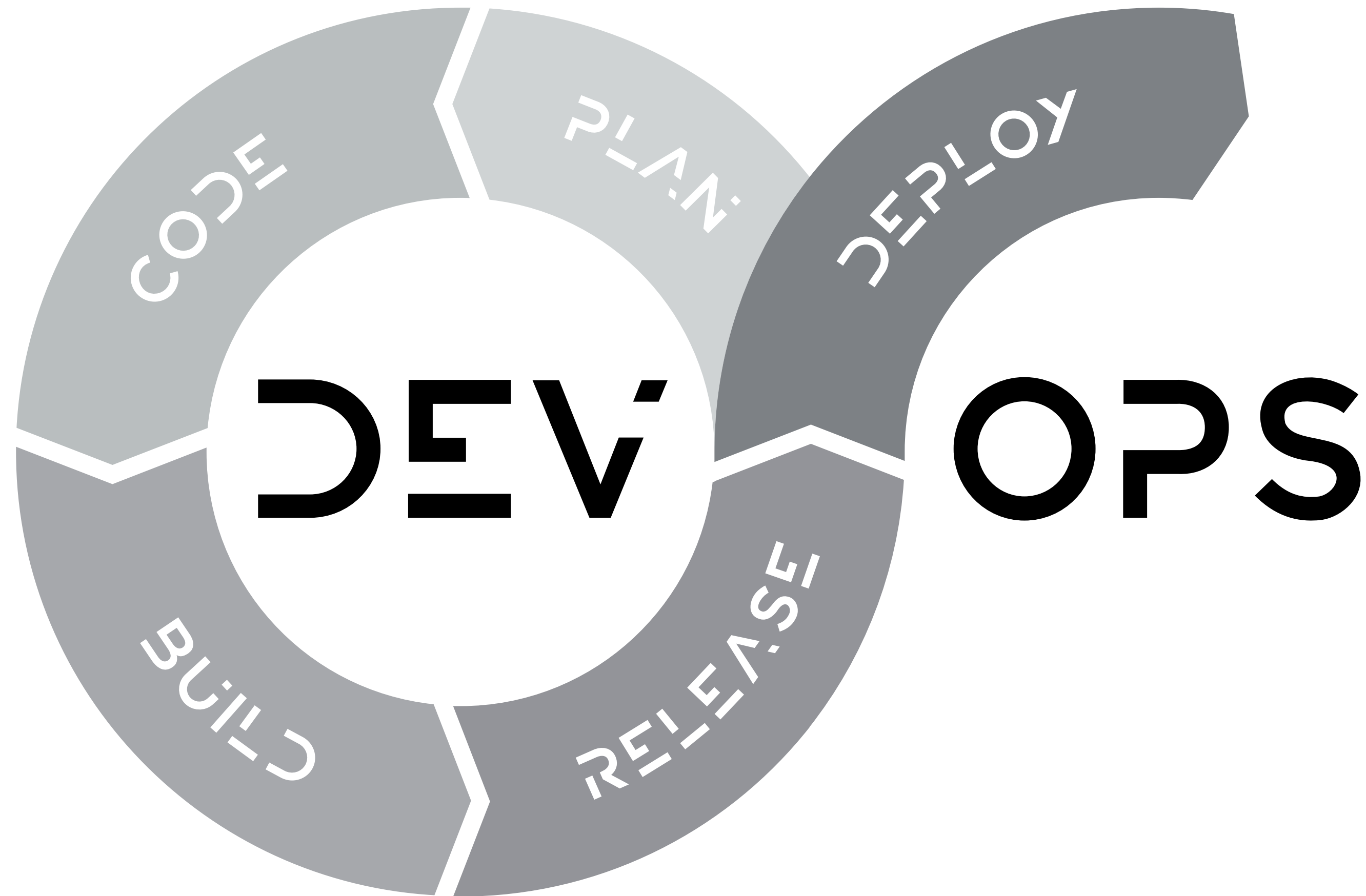
DEV OPS LOOP

Release



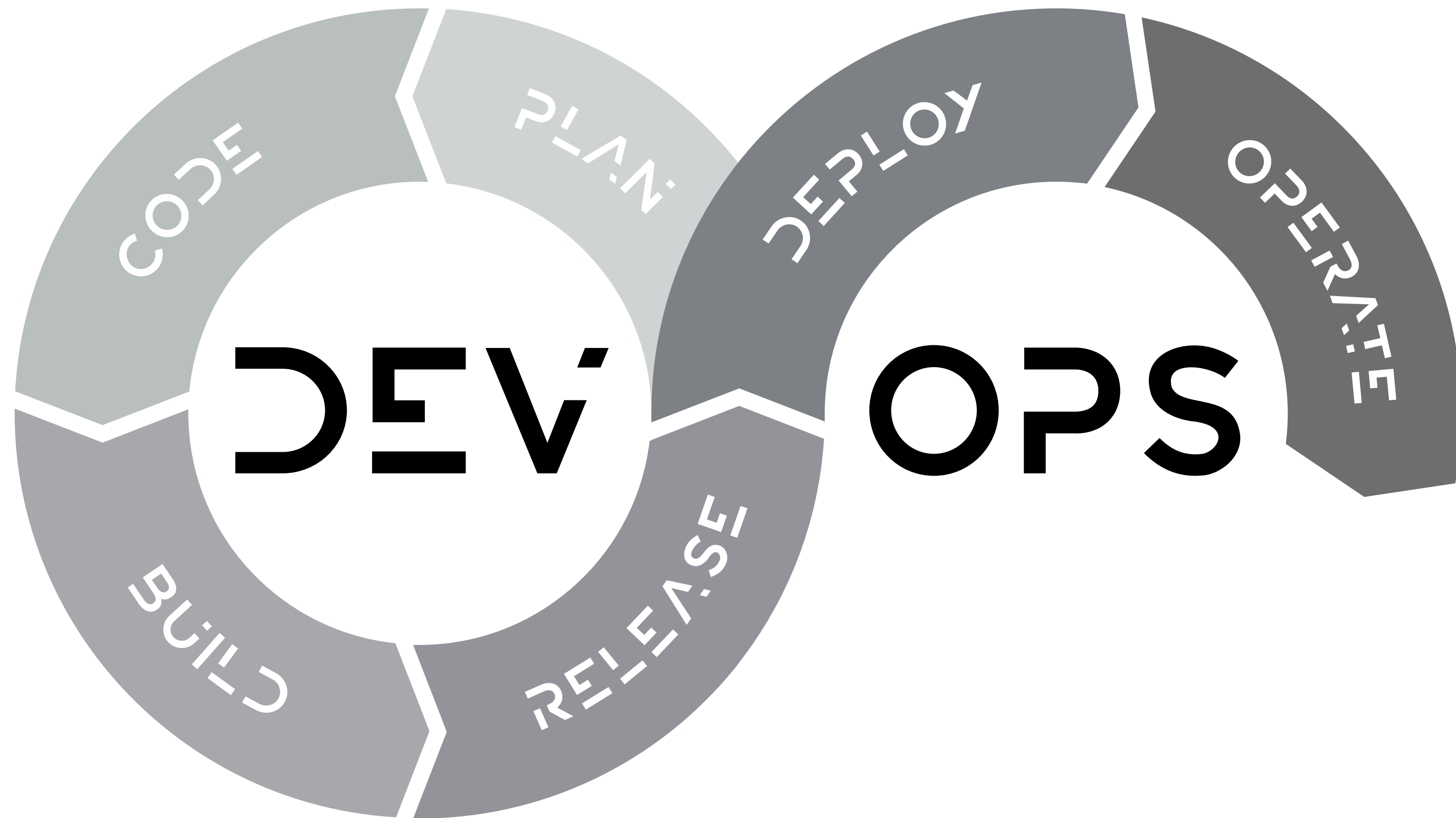
DEV OPS LOOP

Deploy



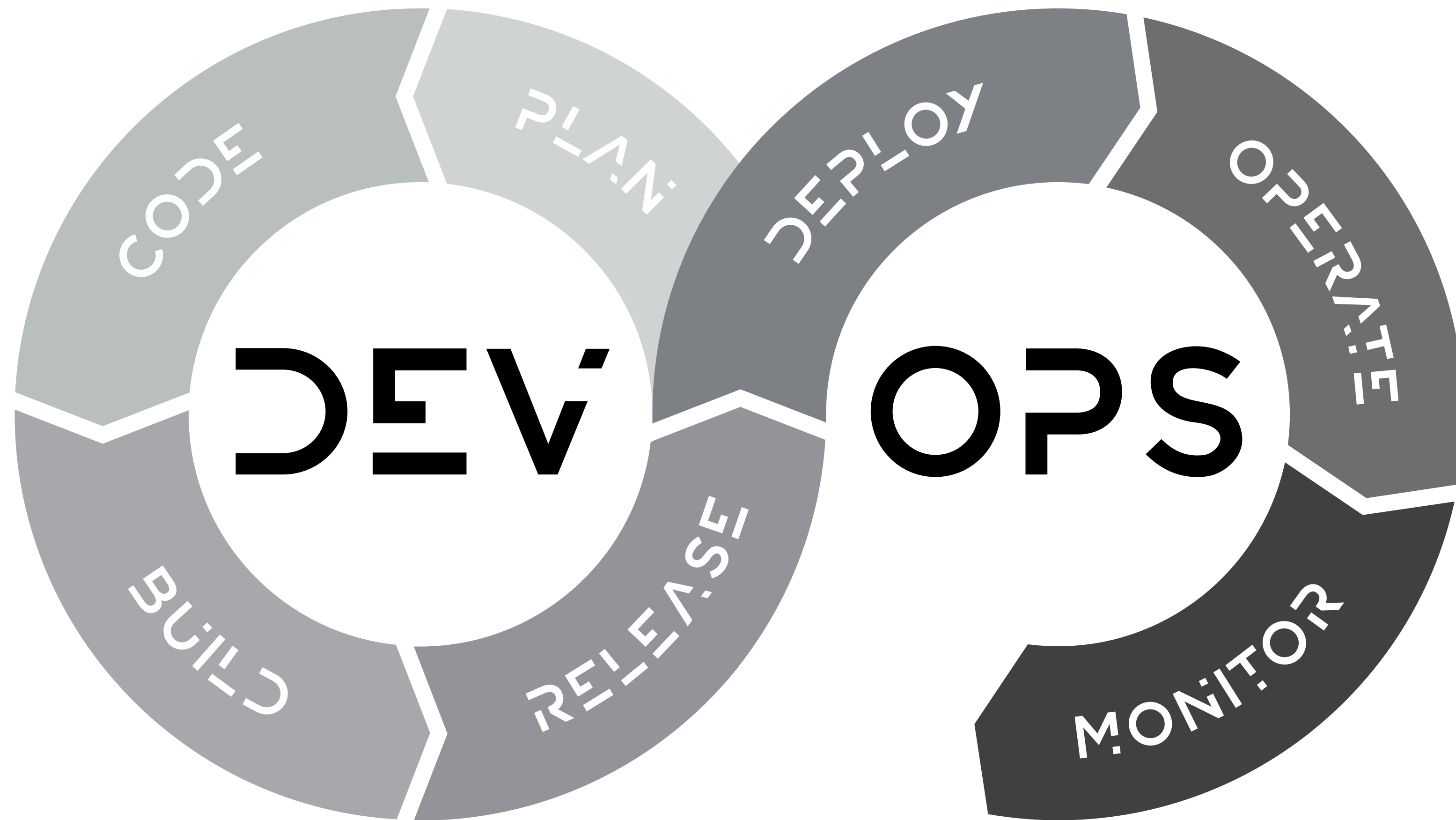
DEV OPS LOOP

Operate



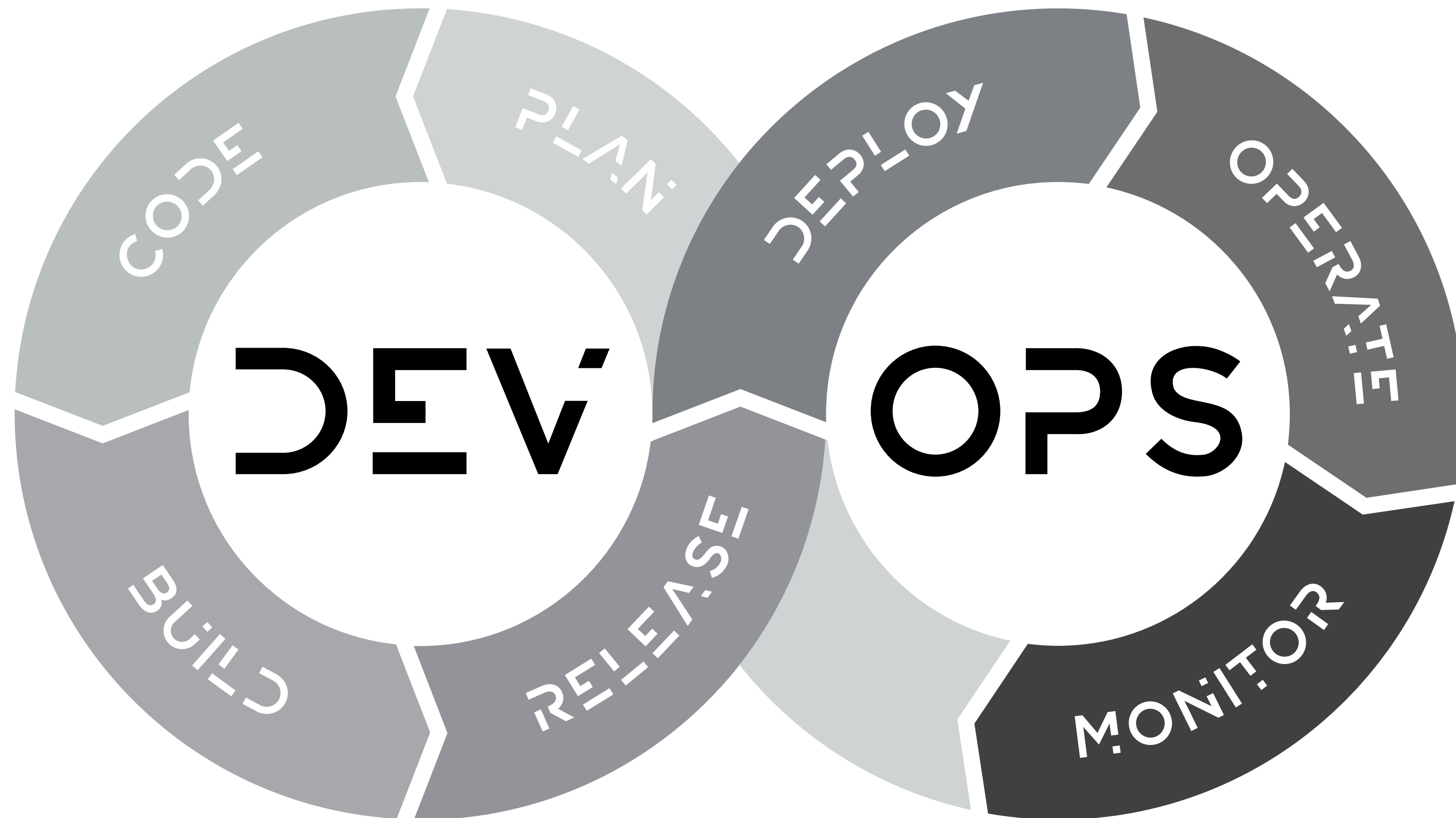
DEV OPS LOOP

Monitor



DEV OPS LOOP

Plan

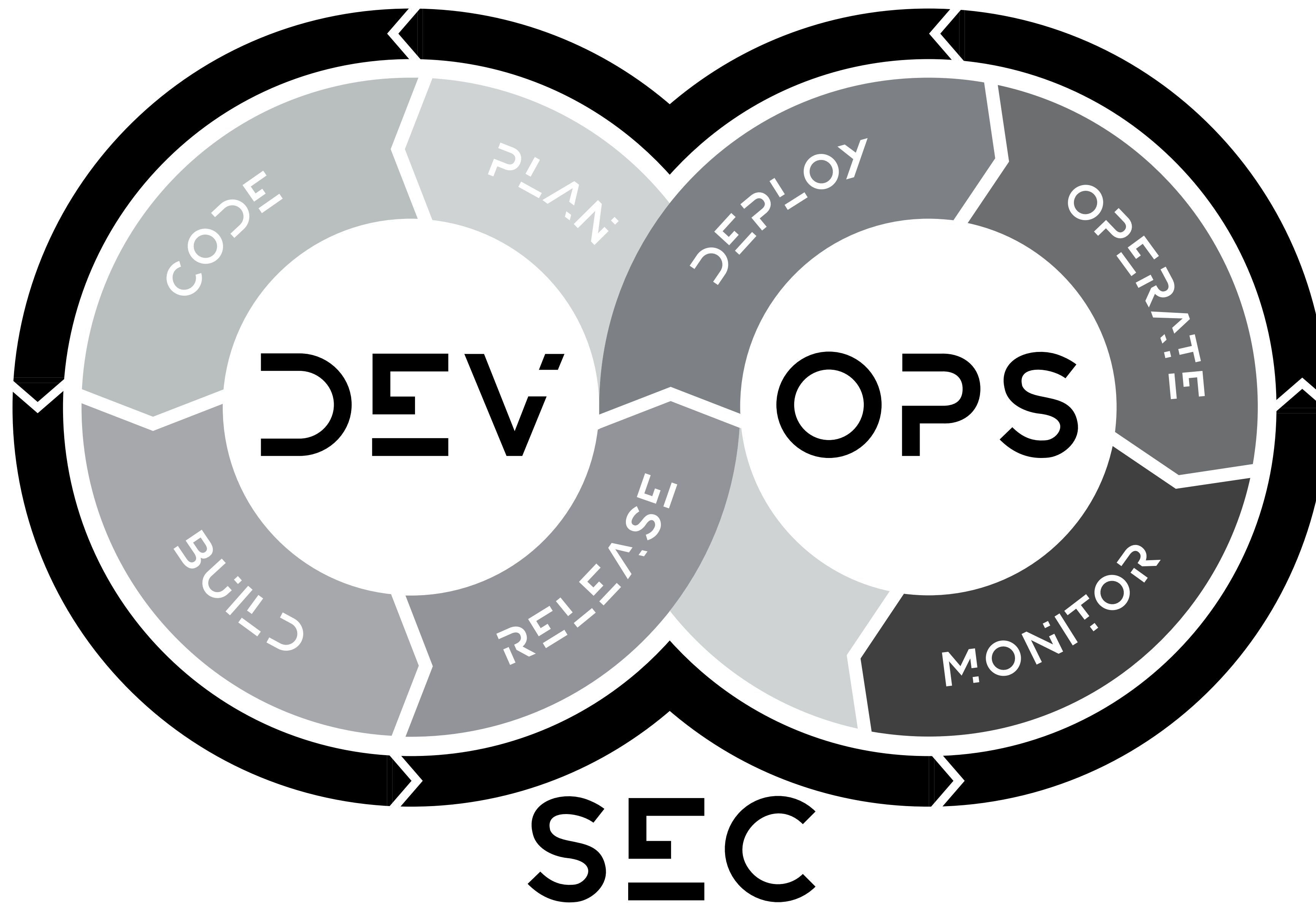


DEV OPS LOOP

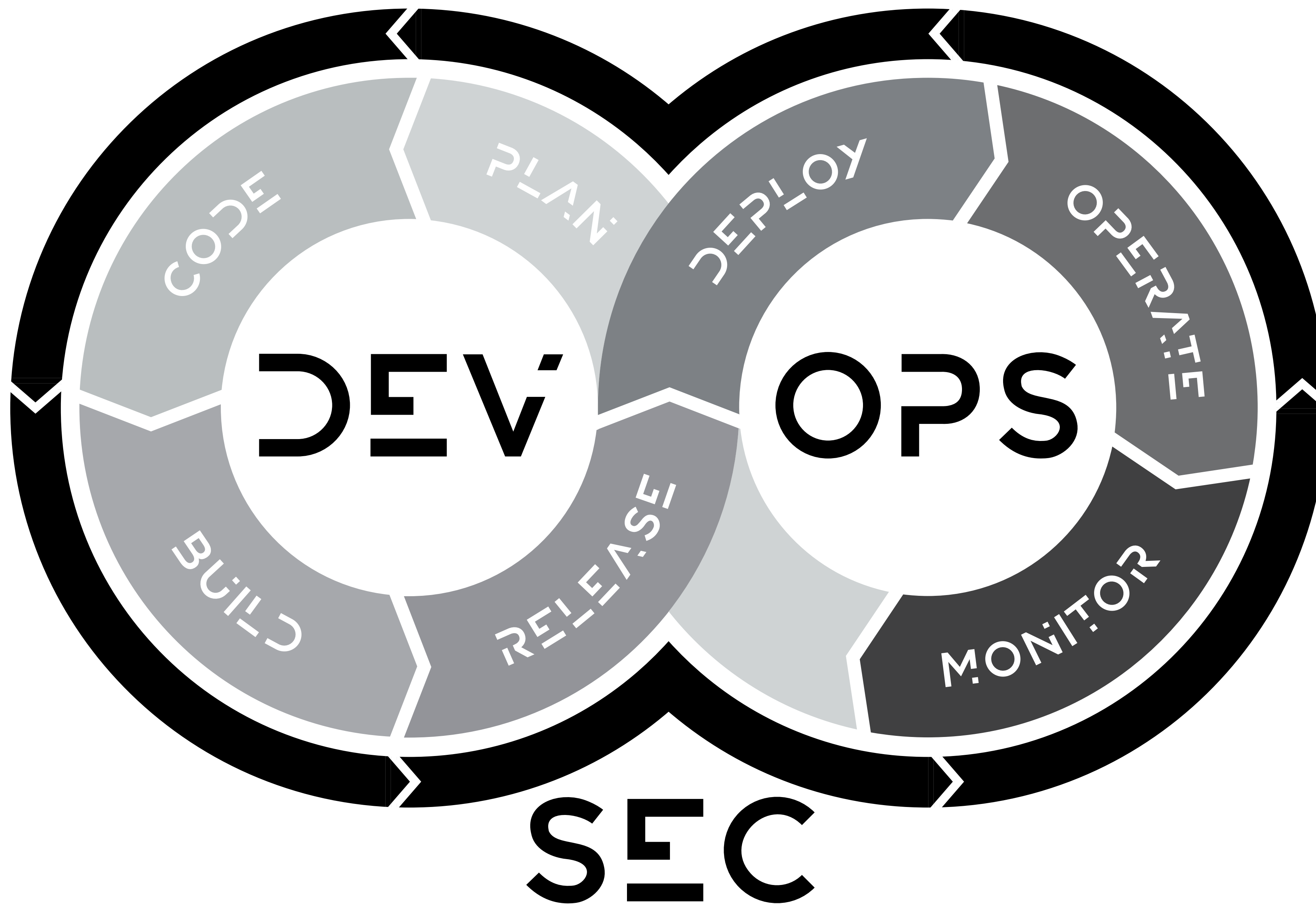
What about security?



DEV SEC OPS



SECURITY APPLIES TO ALL AREAS



SHIFT LEFT ?

YES...BUT

ALSO

VALIDATE

RIGHT?

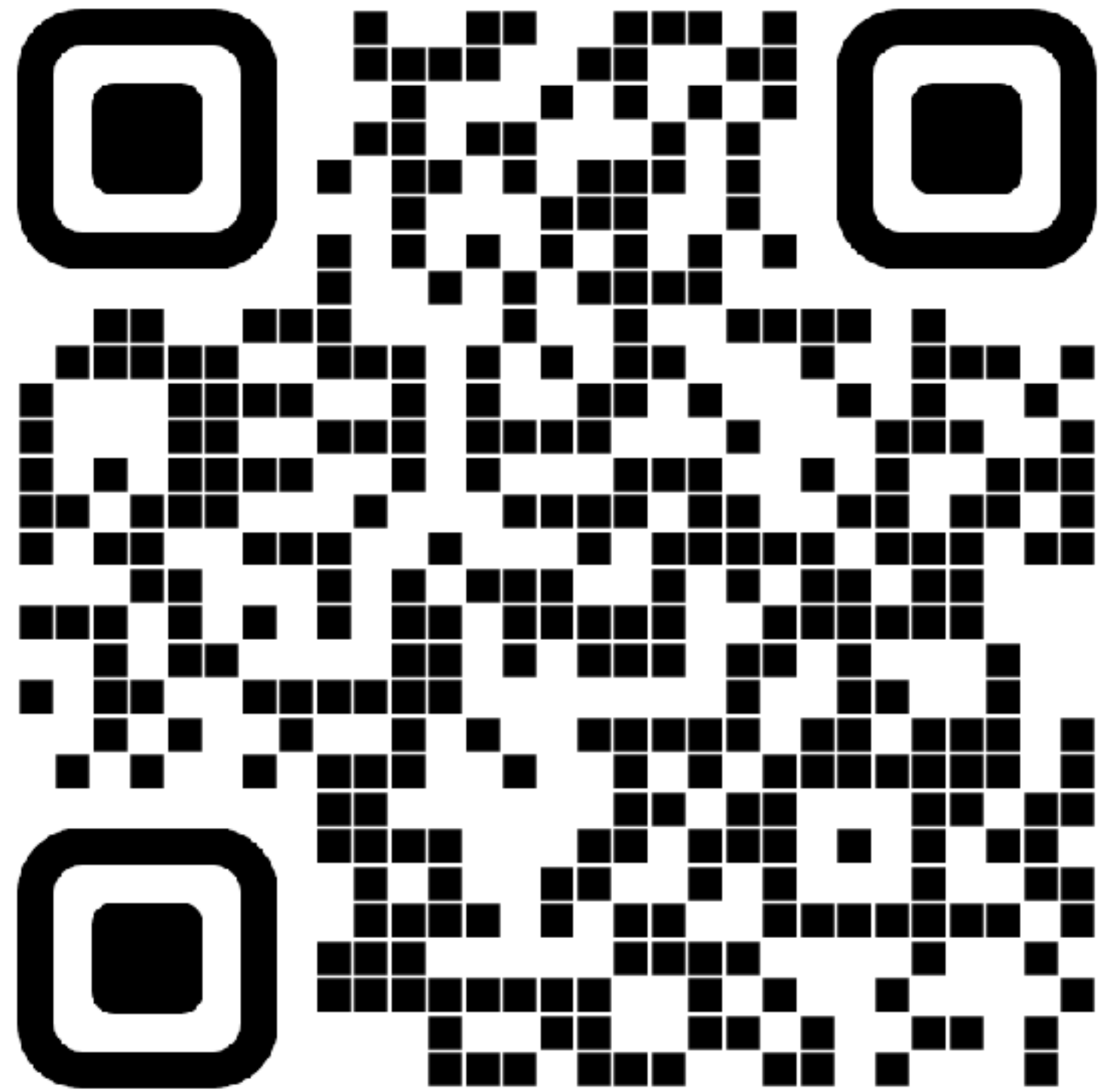
UPDATE YOUR

BOOK

SDKMAN



Command line application



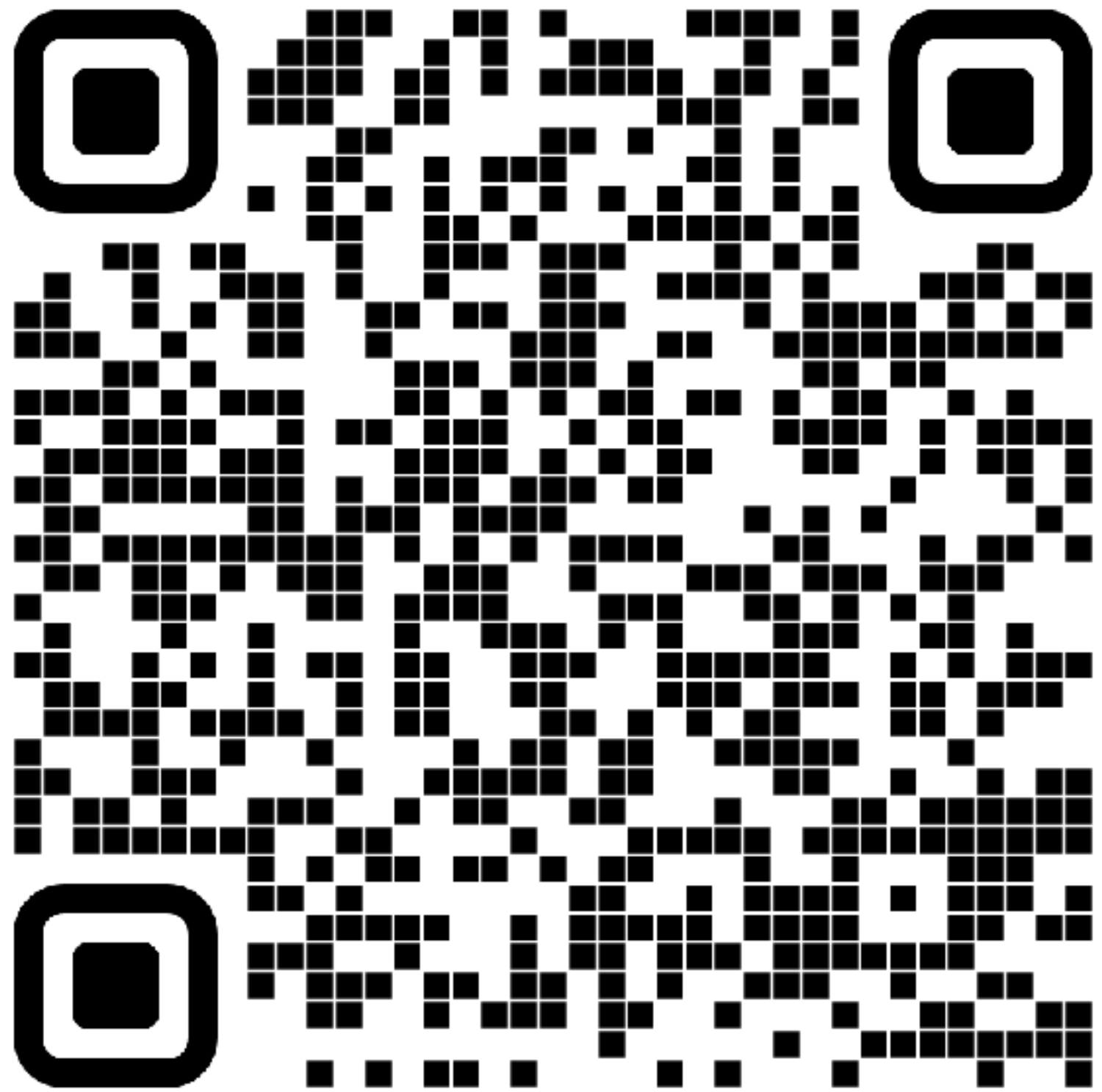
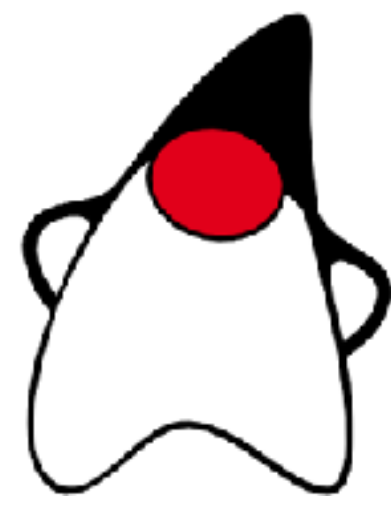
<https://sdkman.io/>

Facts

- ✦ Supports many JDK distributions
- ✦ Commandline only
- ✦ Linux, MacOS
- ✦ Download and install JDK's

JDKMON

Desktop application



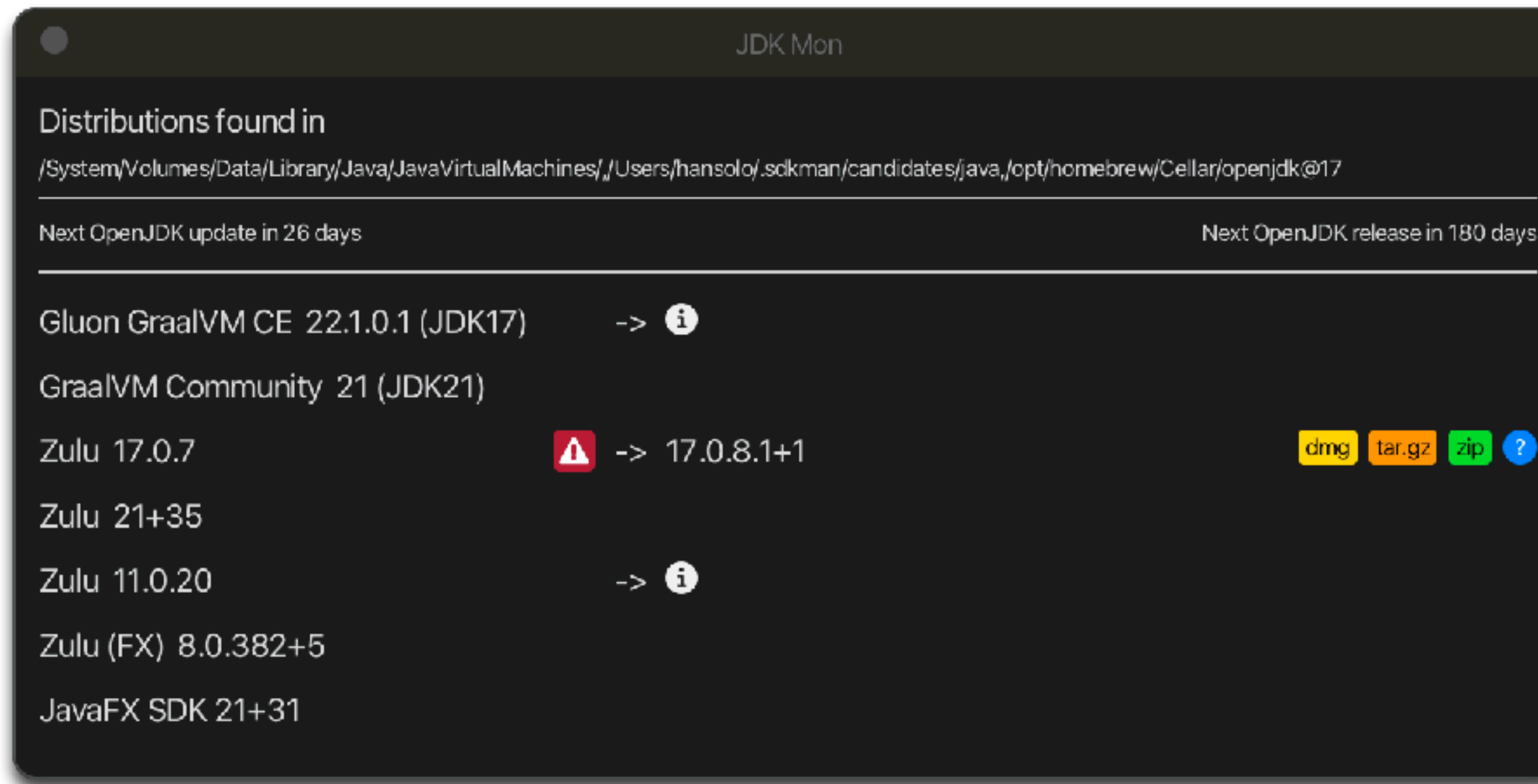
<https://github.com/HanSolo/JDKMon/releases>

Facts

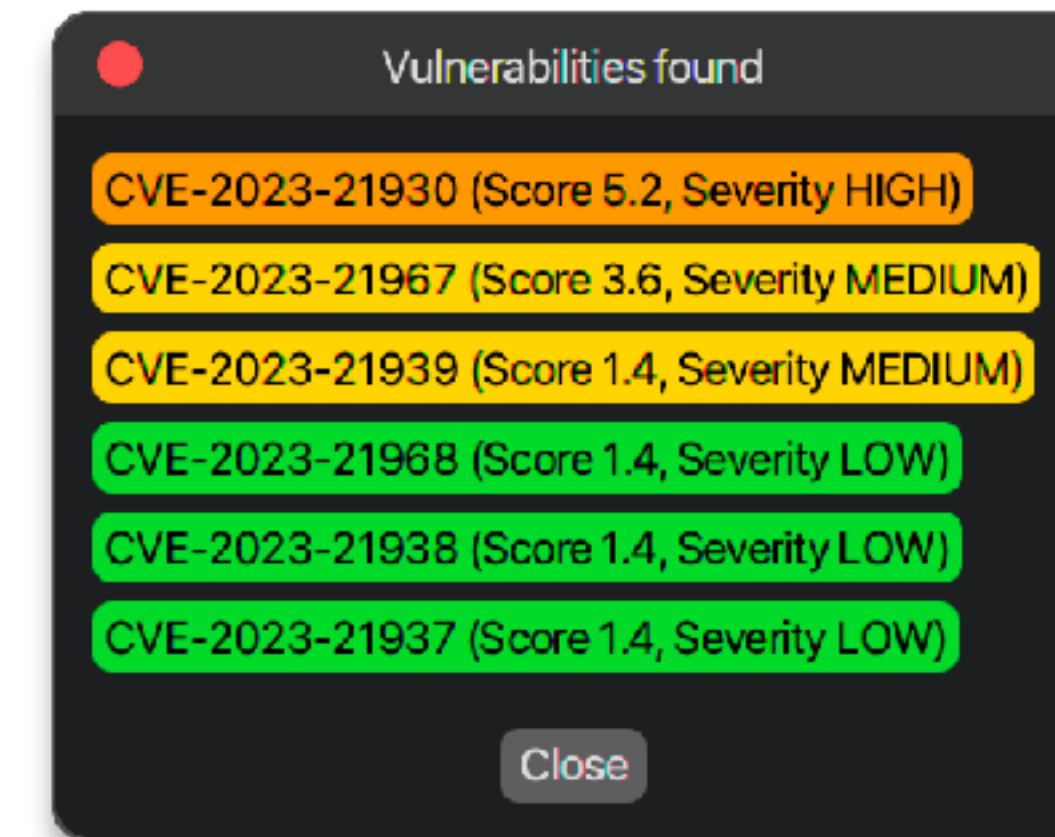
- ✦ Info about JDK updates
- ✦ Supports "all" JDK distributions
- ✦ Taskbar application
- ✦ Windows, Linux, MacOS
- ✦ Info about vulnerabilities in JDK distribution

JDKMON

Desktop application



Installed JDK distributions



Vulnerabilities found for JDK

STATIC CODE ANALYSIS


STATIC CODE ANALYSIS

What is it ?

- ✦ Usually part of a code review (white-box testing)
- ✦ Identifies vulnerabilities in source code
- ✦ At the implementation phase
- ✦ Inexpensive because adjustments can be done easily
- ✦ Standalone tools / IDE plugins

STATIC CODE ANALYSIS

Source Code Security Analyzers

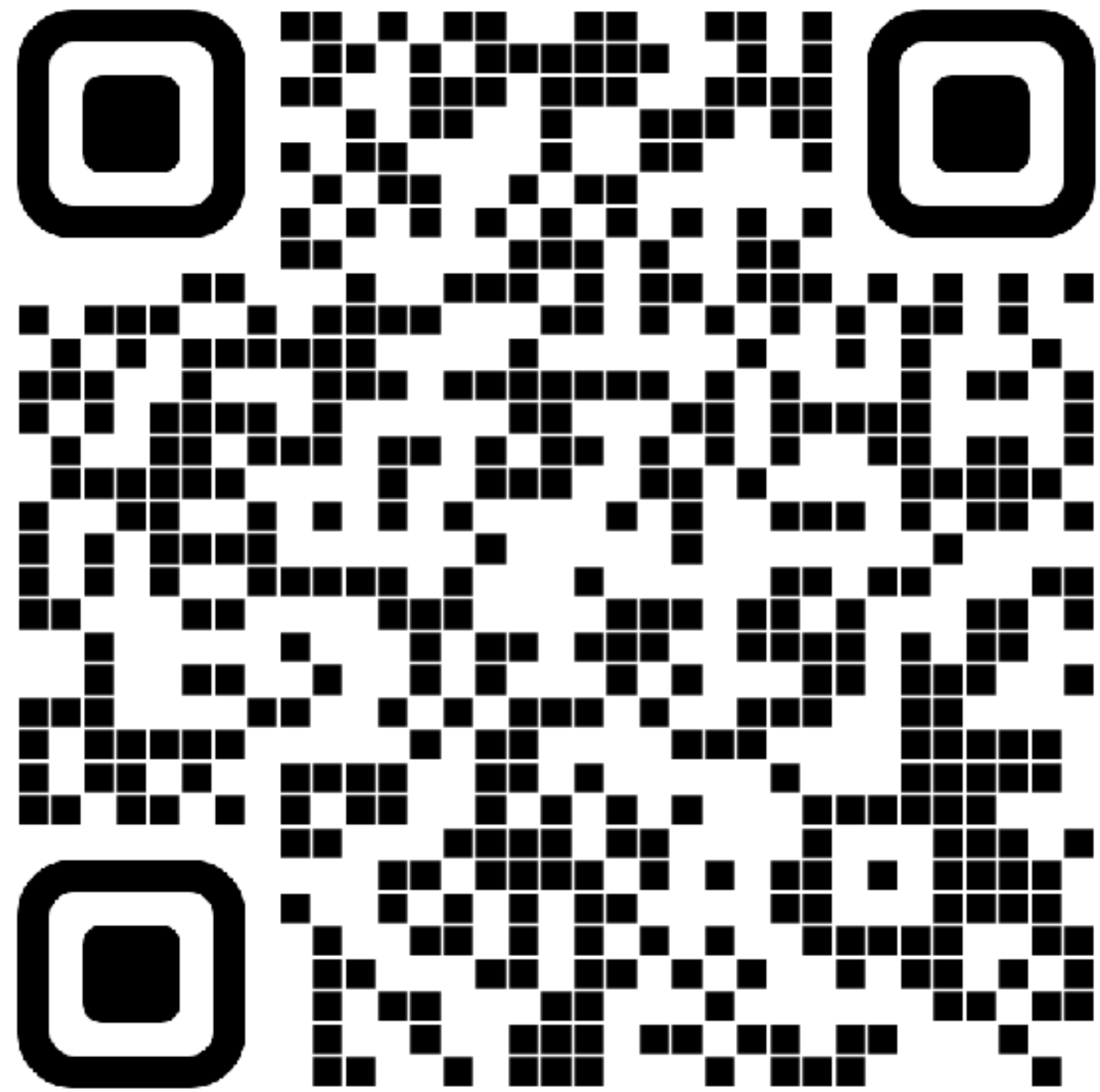
| | | | |
|----------------------|--|---|---|
| ✦ AppSonar/CodeSonar | by CyberTest | | Ⓢ |
| ✦ Codiga | by Codiga | Ⓢ | Ⓢ |
| ✦ DerScanner | by DerSecur Ltd. | | Ⓢ |
| ✦ FindSecurityBugs | free  | Ⓢ | |
| ✦ Snyk Code | by Snyk Limited | Ⓢ | Ⓢ |
| ✦ SonarQube | by SonarSource | Ⓢ | Ⓢ |
| ✦ Static Reviewer | by Security Reviewer | | Ⓢ |

taken from <https://www.nist.gov/itl/ssd/software-quality-group/source-code-security-analyzers>

FIND SECURITY BUGS

FIND SECURITY BUGS

SpotBugs plugin



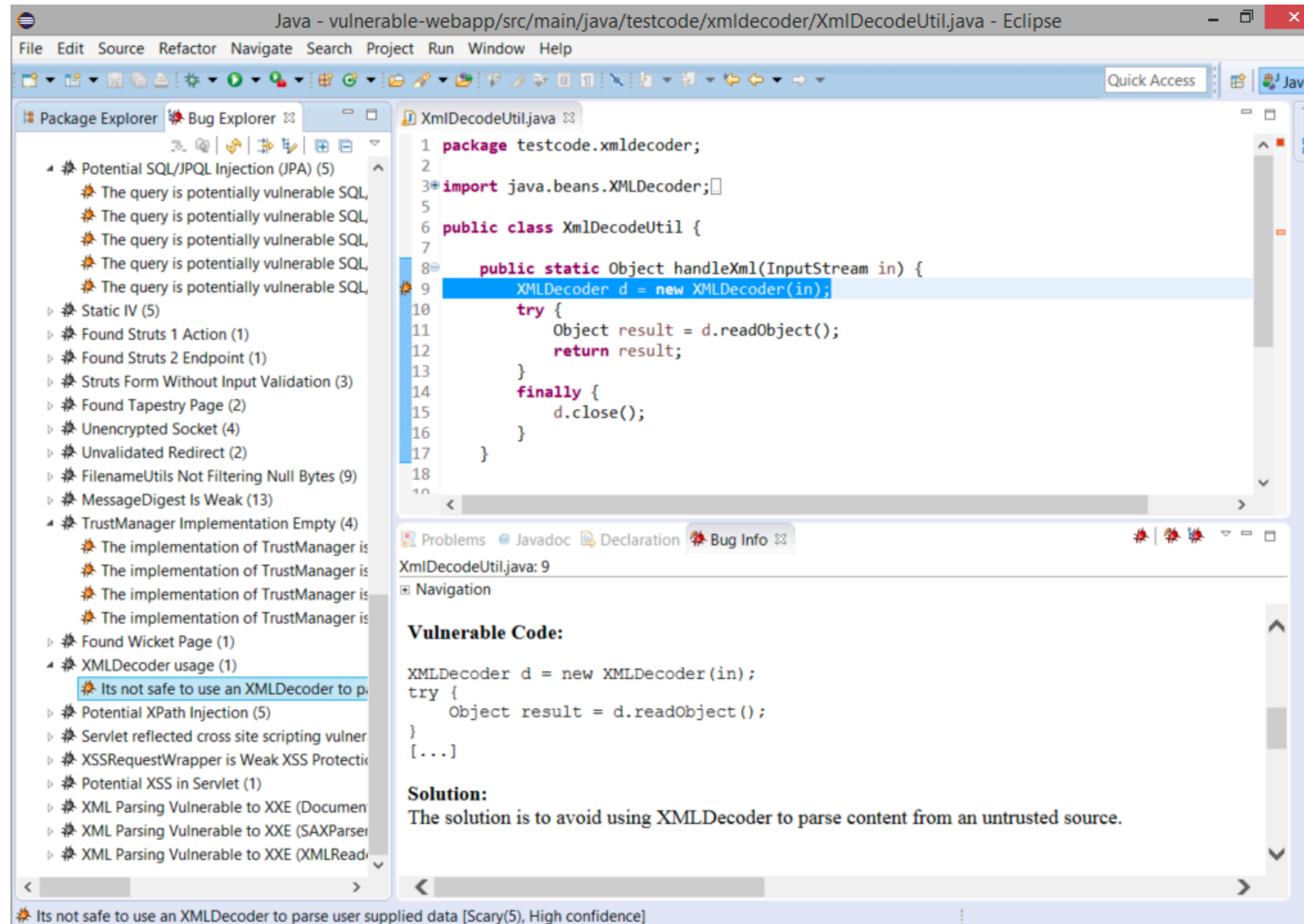
<https://find-sec-bugs.github.io/>

Facts

- ✦ Free of charge
- ✦ Extends SpotBugs
- ✦ 400+ bug patterns
- ✦ Plugin

FIND SECURITY BUGS

Eclipse Plugin



VULNERABILITY SCANNERS

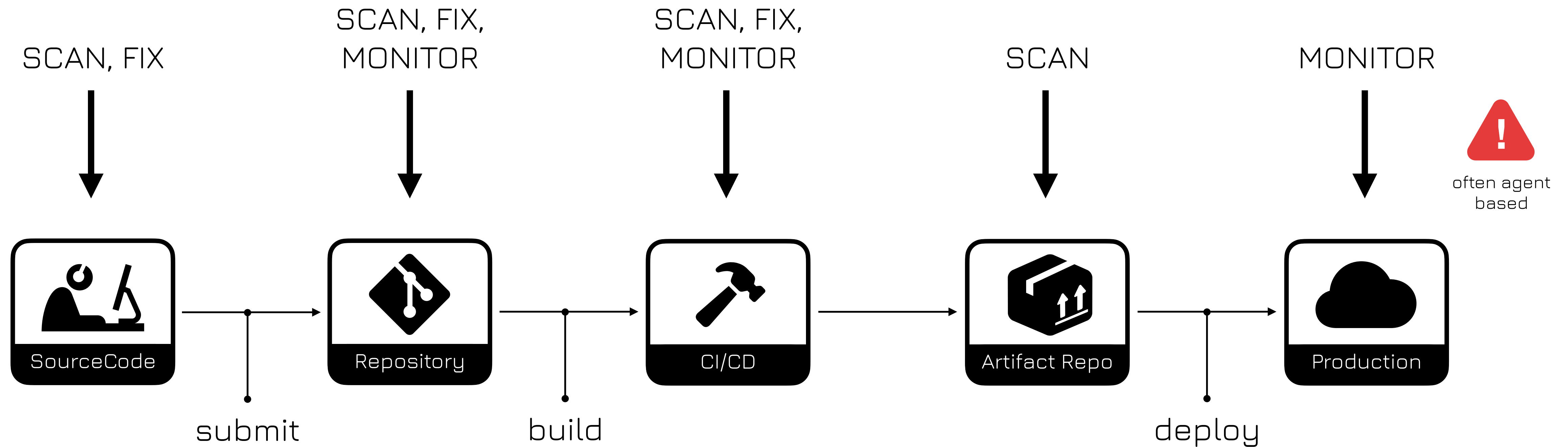
VULNERABILITY SCANNERS

What is it?

- ✦ Detect vulnerabilities (using a database / probing for common flaws)
- ✦ Monitor misconfigurations and coding flaws
- ✦ Help using only artifacts from reliable sources
- ✦ Help using only latest secure version (without known vulnerabilities)
- ✦ Monitor appearance of new packages with fixed vulnerabilities
- ✦ Update dependencies (as soon as new versions are available)

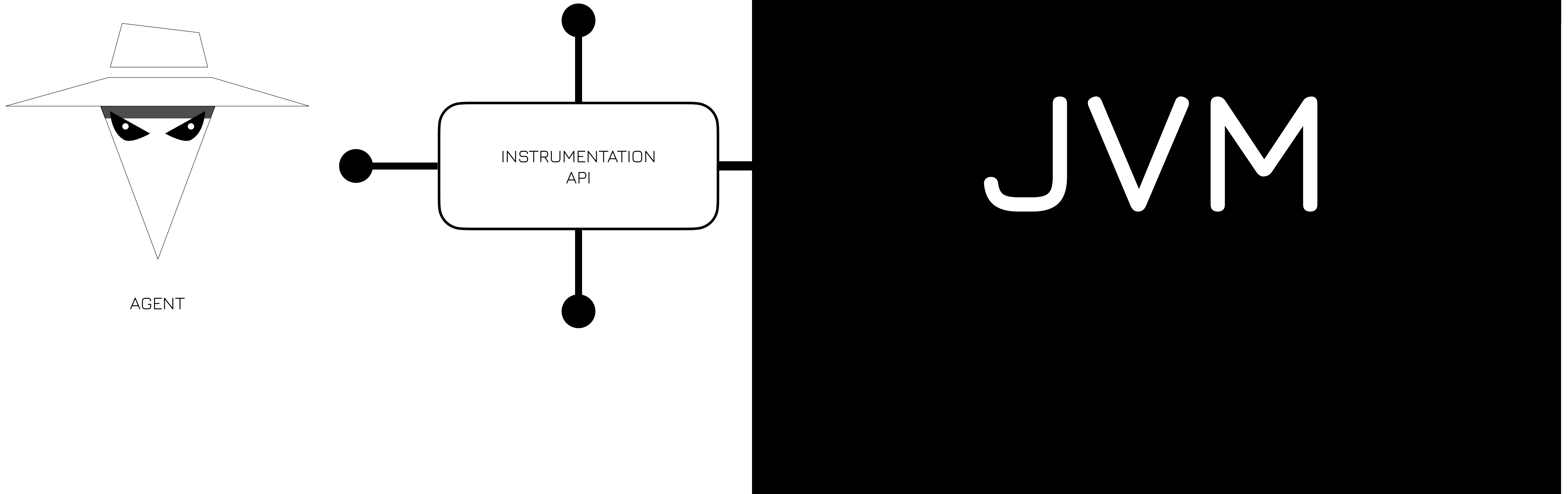
VULNERABILITY SCANNERS

How they work



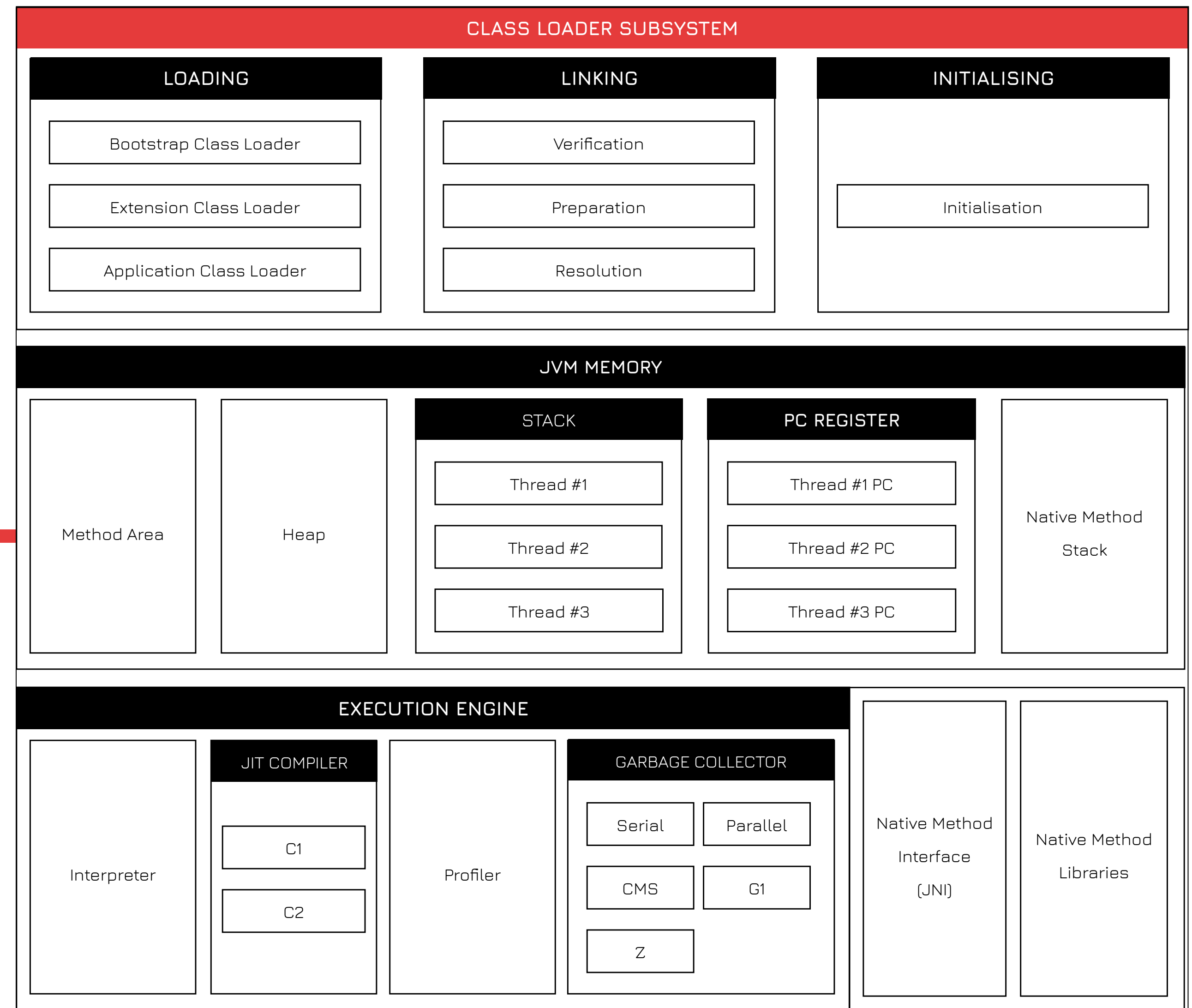
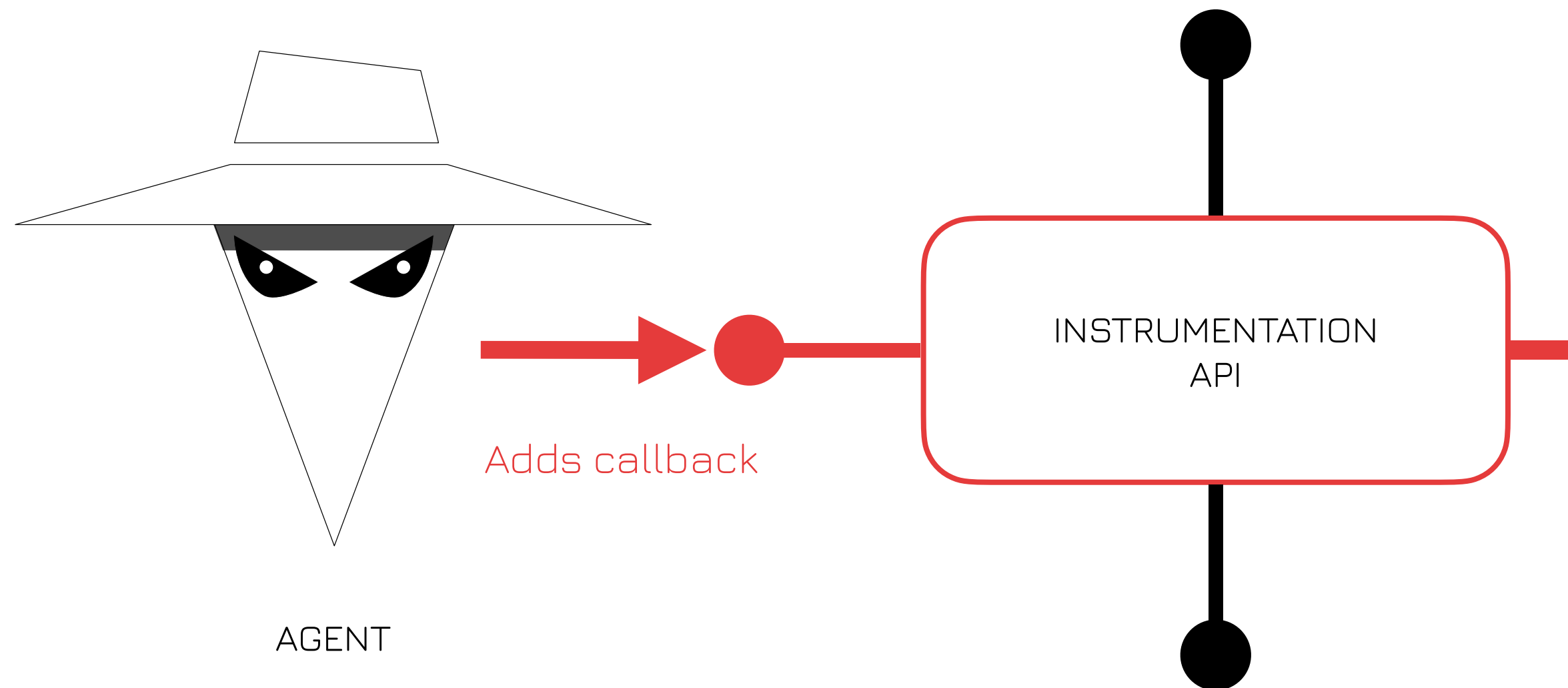
VULNERABILITY SCANNERS

Agent based monitoring



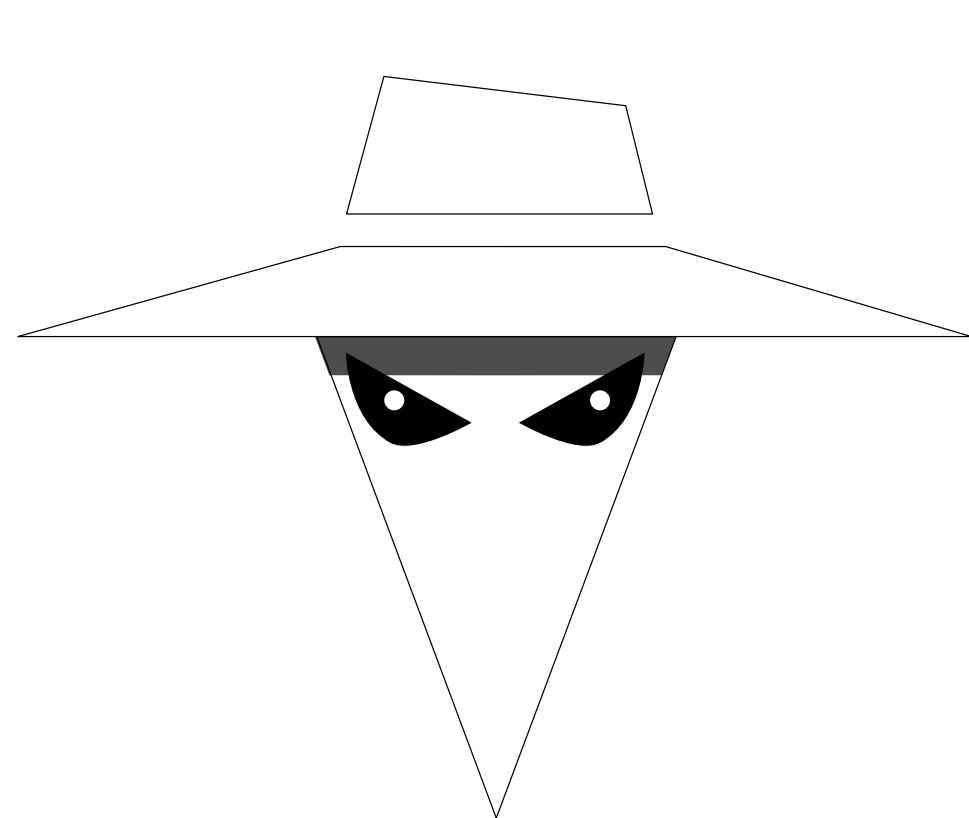
VULNERABILITY SCANNERS

Agent based monitoring

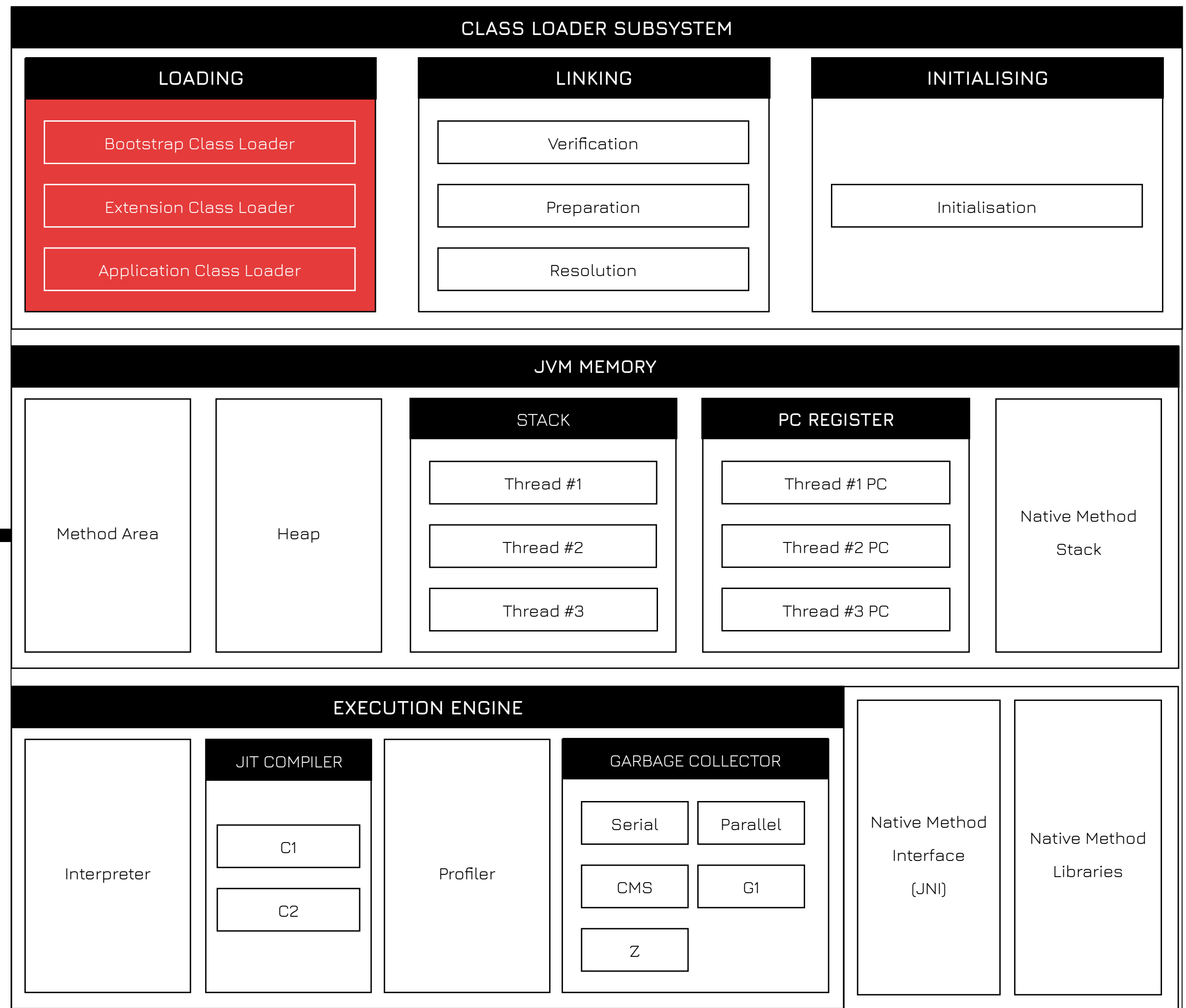
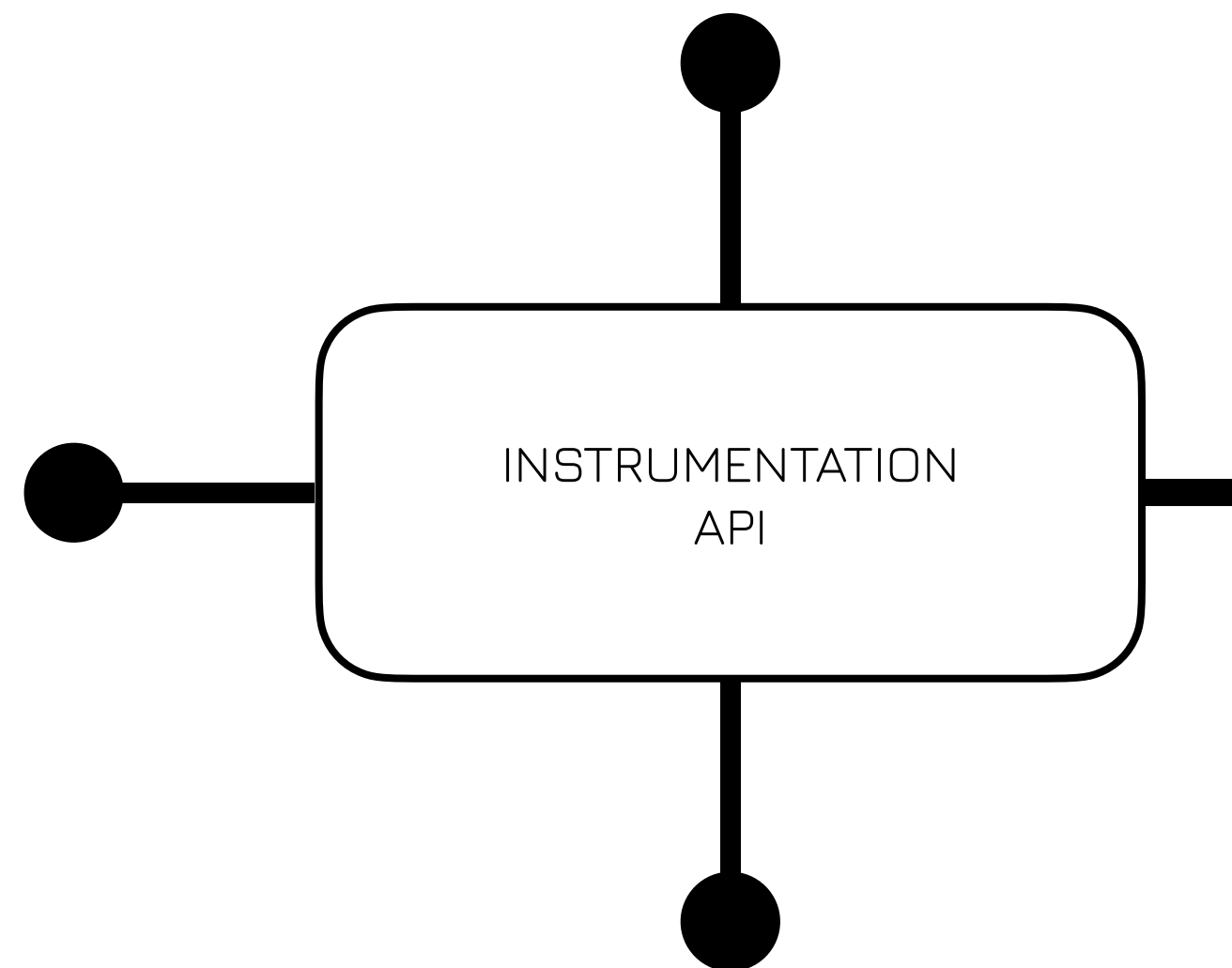


VULNERABILITY SCANNERS

Agent based monitoring

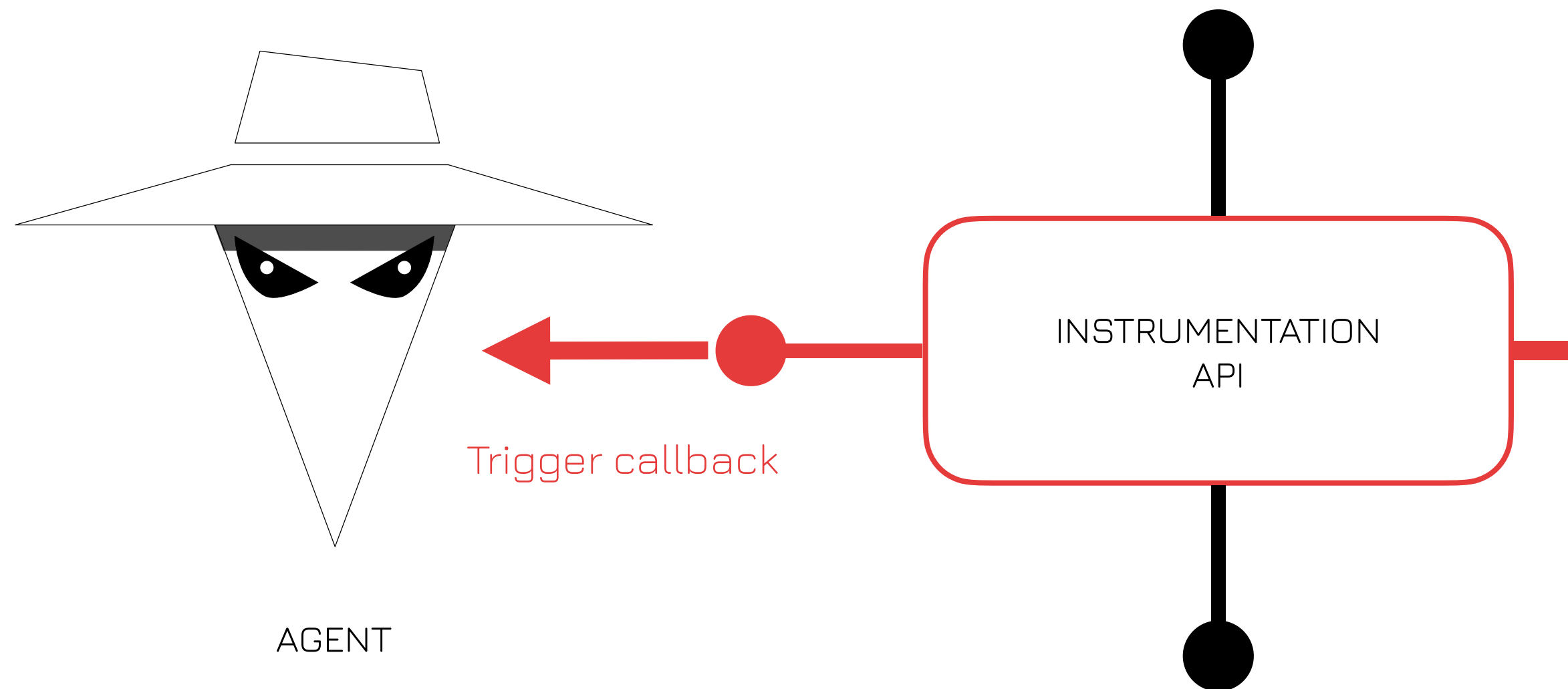


AGENT

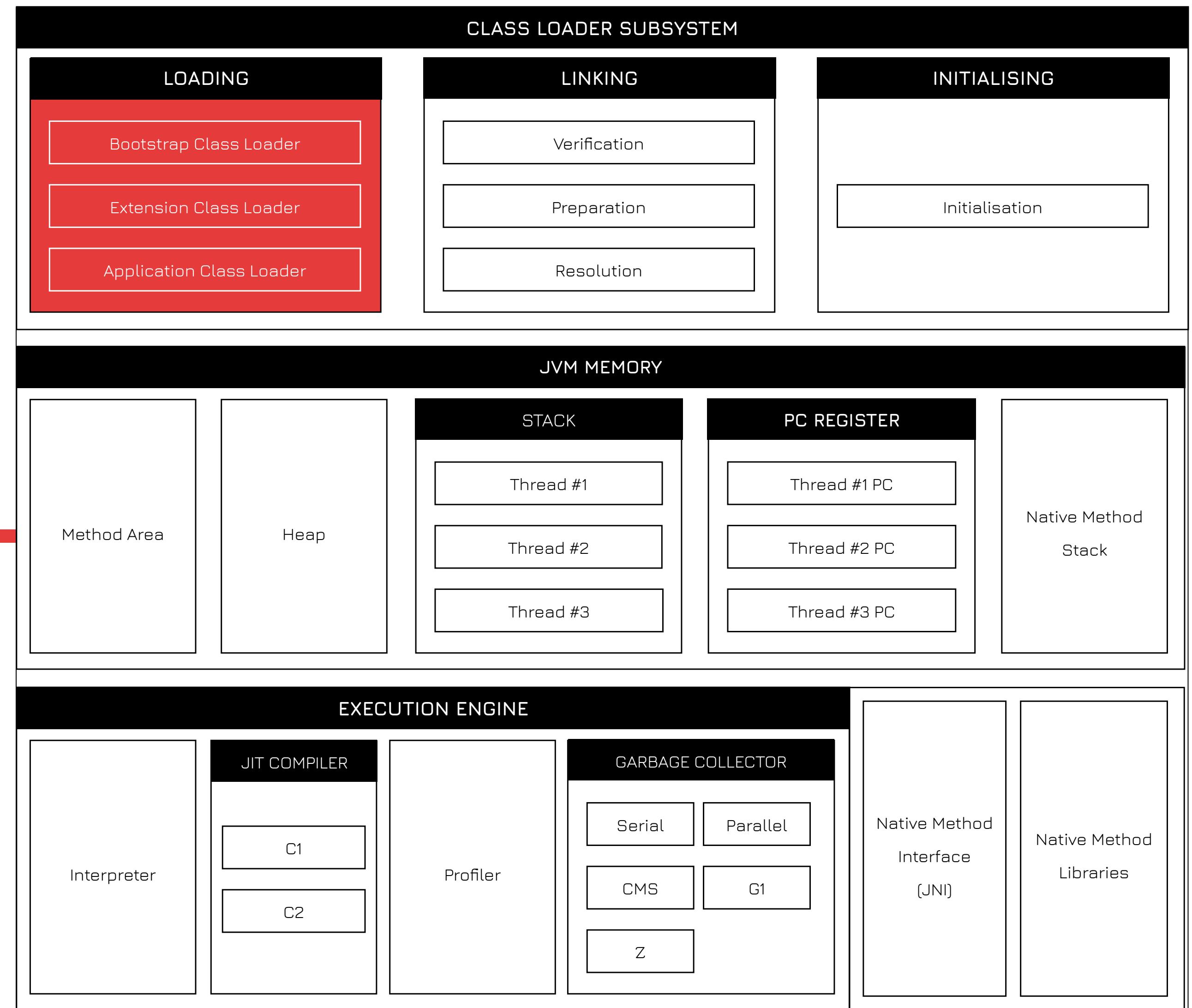


VULNERABILITY SCANNERS

Agent based monitoring



Performance hit of 10% or more !



VULNERABILITY SCANNERS

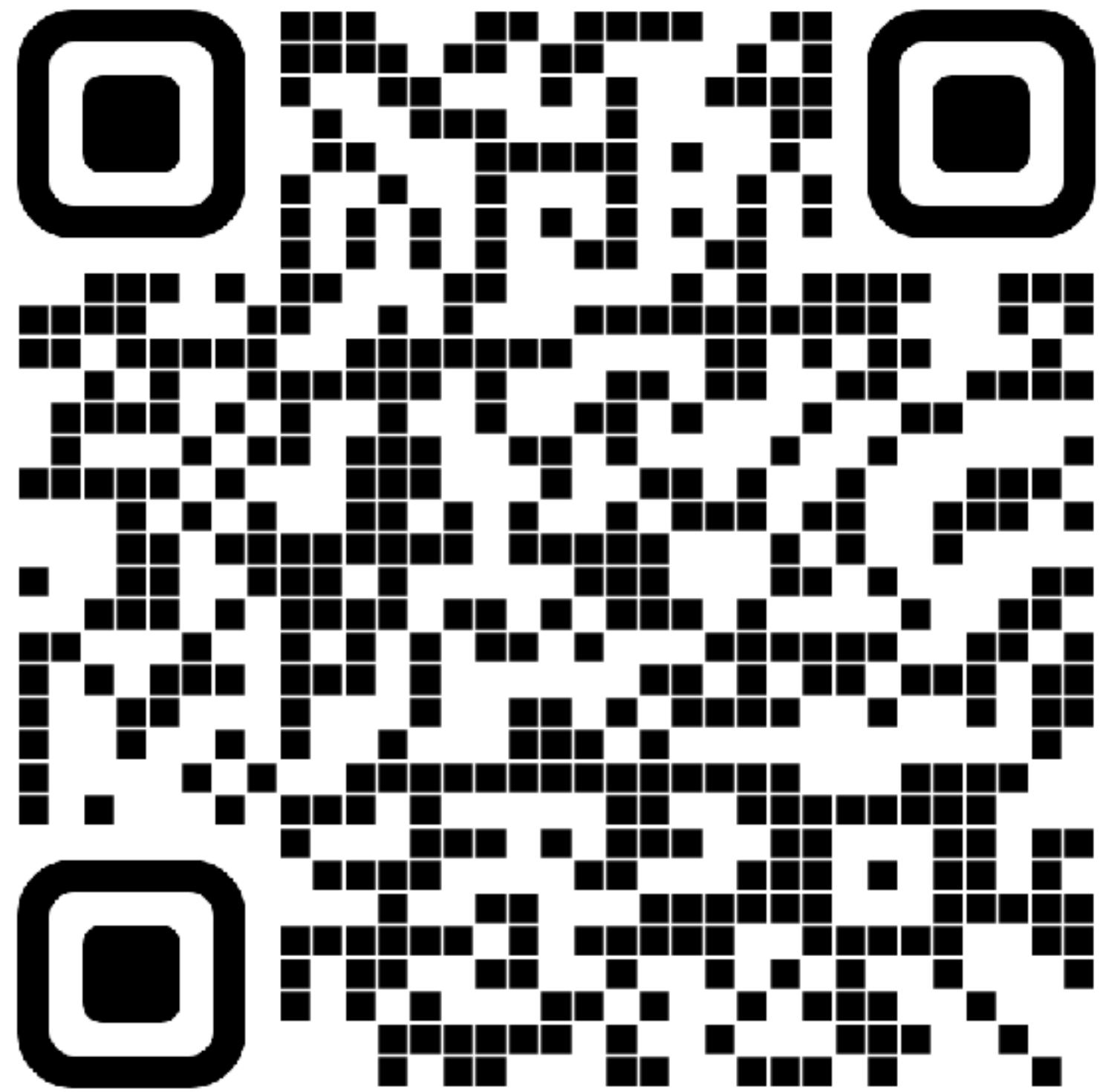
For Java development

| | | | |
|--------------------------------|-----------------|---|---|
| ✦ Azul Vulnerability Detection | by Azul | | Ⓢ |
| ✦ Black Duck | by Synopsys | | Ⓢ |
| ✦ Xray | by JFrog | Ⓢ | Ⓢ |
| ✦ Snyk | by Snyk Limited | Ⓢ | Ⓢ |
| ✦ SonarQube | by SonarSource | Ⓢ | Ⓢ |
| ✦ Trivy | by Aqua | Ⓢ | |

SNYK CODE

SNYK CODE

Static application Security Testing



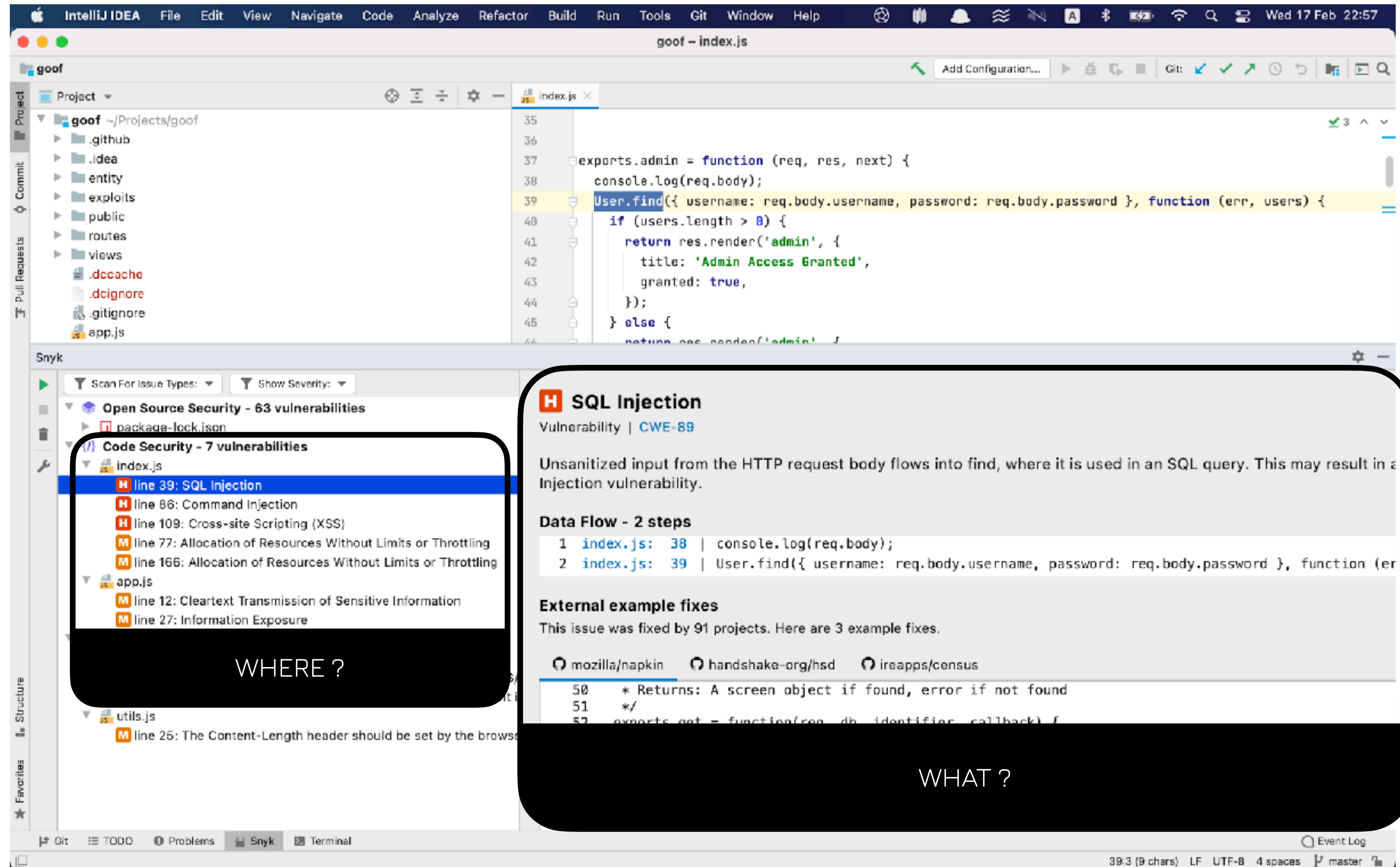
<https://snyk.io/product/snyk-code/>

Facts

- ✦ Free and paid version
- ✦ 9+ languages supported
- ✦ Developer first
- ✦ Standalone
- ✦ IDE Plugin available
- ✦ CI/CD integration

SNYK CODE

IntelliJ Plugin



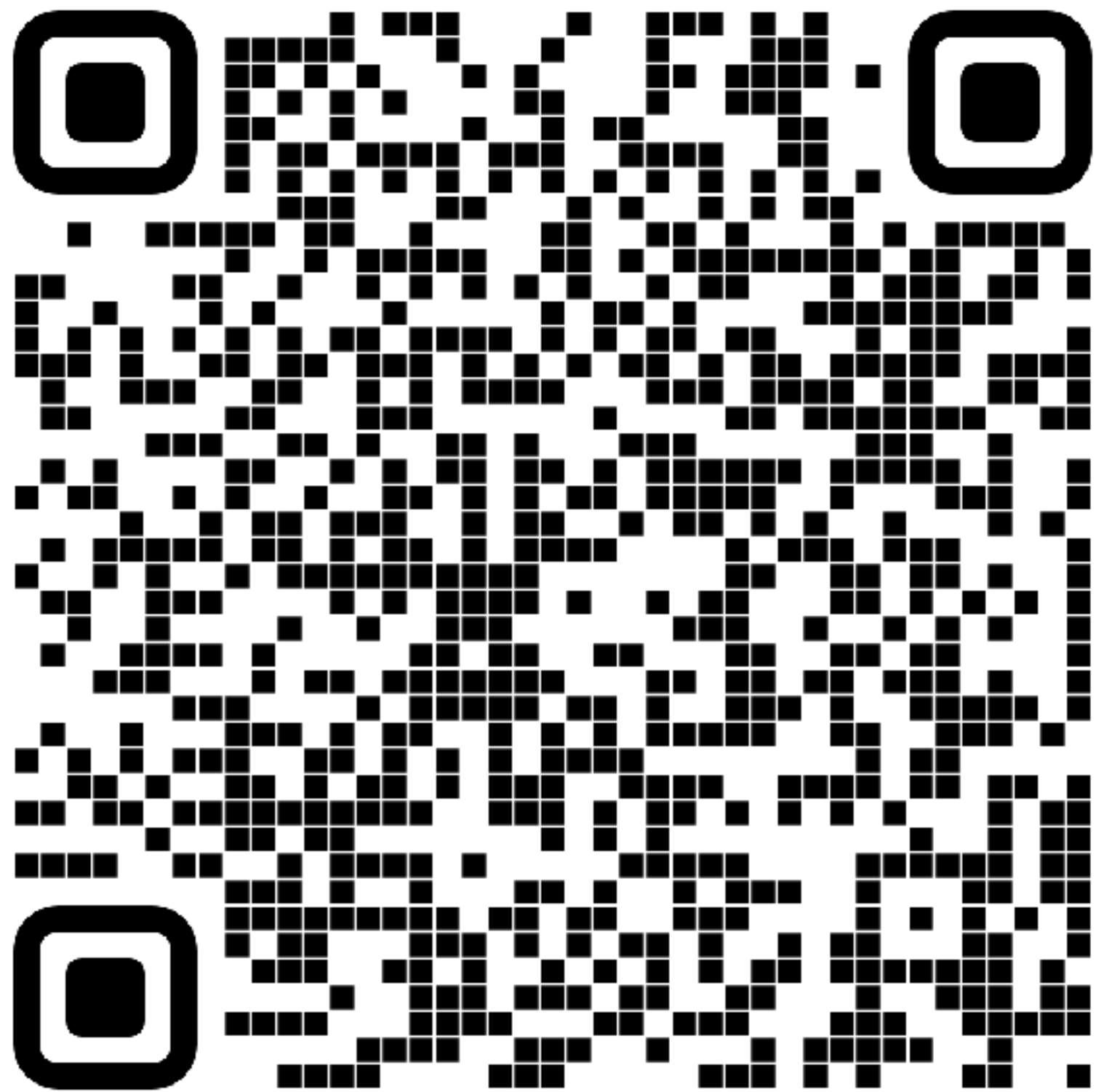
AZUL

VULNERABILITY

DETECTION



Azul Vulnerability Detection



Facts

- ✦ Runs in production
- ✦ Java only
- ✦ Fewer false positives
- ✦ Does code inventory
- ✦ No Java agent -> no performance overhead

<https://www.azul.com/products/vulnerability-detection/>

AZUL VULNERABILITY DETECTION

Web UI

azul

Vulnerability Detection

1m

→

now

CVE Analysis

Show Unaffected ☒

| Component | Version | CVE | CVE Score | CVE Status | Timestamp | Hostname | Instance ID |
|---|--------------|--------------------------------------|-----------|------------|-----------------------|---------------------------|--------------|
| netty | 3.8.0.Final | <div><div></div>CVE-2021-37137</div> | 7.5 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| netty-codec-haproxy | 4.0.29.Final | <div><div></div>CVE-2021-37137</div> | 7.5 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| jackson-databind | 2.6.5 | <div><div></div>CVE-2019-14892</div> | 9.8 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| cp-chill_2.11-0.8.0.jar11254612295605694756.jar | UNKNOWN | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| | UNKNOWN | | | | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| | 2.2.0 | | | | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| cp-jackson-databind-2.6.5.jar6660364613748728752... | UNKNOWN | <div><div></div>CVE-2019-14540</div> | 9.8 | USED | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| javax.inject | 2.1.95 | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| eclipse-collections-9.2.0.jar | UNKNOWN | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| jackson-core-asl | 1.9.13 | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| jackson-databind | 2.6.5 | <div><div></div>CVE-2018-7489</div> | 9.8 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| neo4j-common | 3.5.12 | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| netty-transport | 4.0.29.Final | <div><div></div>CVE-2021-37137</div> | 7.5 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| javassist | 3.18.1-GA | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| jackson-databind | 2.6.5 | <div><div></div>CVE-2020-36182</div> | 8.1 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| netty | 3.8.0.Final | <div><div></div>CVE-2021-43797</div> | 6.5 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| neo4j | 3.5.12 | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| javax.inject | 2.1.86 | None CVE impact | N/a | - | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |
| hadoop-annotations | 2.2.0 | <div><div></div>CVE-2016-6811</div> | 8.8 | PRESENT | 6/24/2022, 9:32:50 AM | skylake02.azulsystems.com | 5ed19d86e1ed |

WHERE ?

CVE

CVE Score

CVE-2021-37137

CVE-2021-37137

CVE-2019-14892

None CVE impact

7.5

7.5

9.8

N/a

VULNERABLE ?

CVE Status

PRESENT

PRESENT

PRESENT

-

USED ?

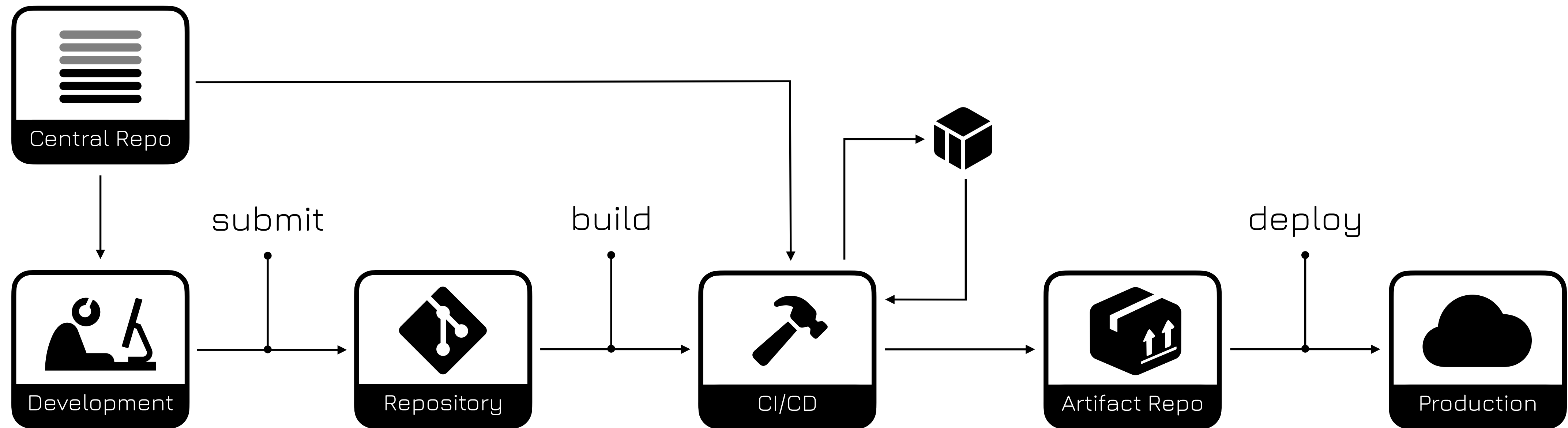
WHERE ?

VULNERABLE ?

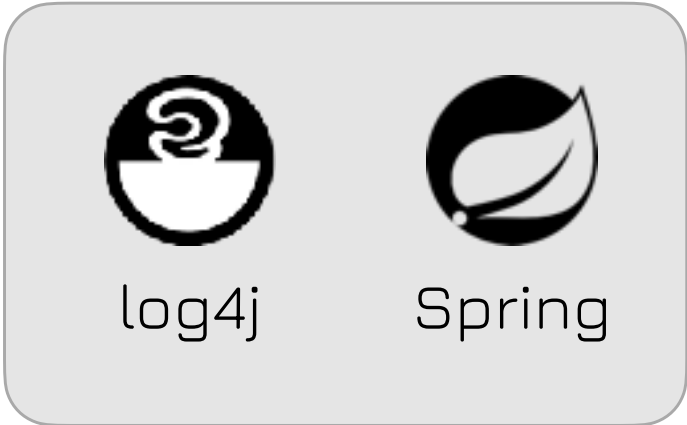
USED ?

A SECURE SOFTWARE SUPPLY CHAIN

EXAMPLE

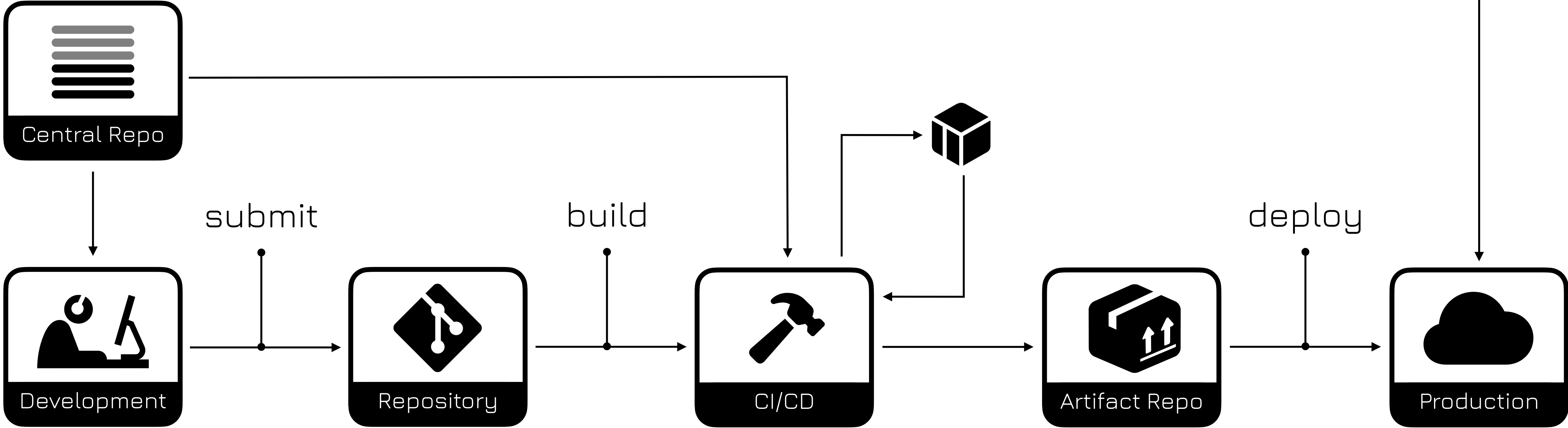
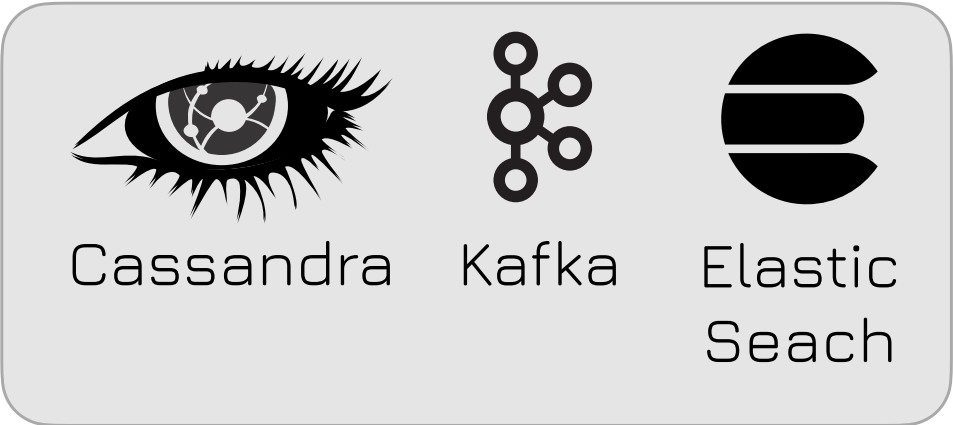


3rd party libraries

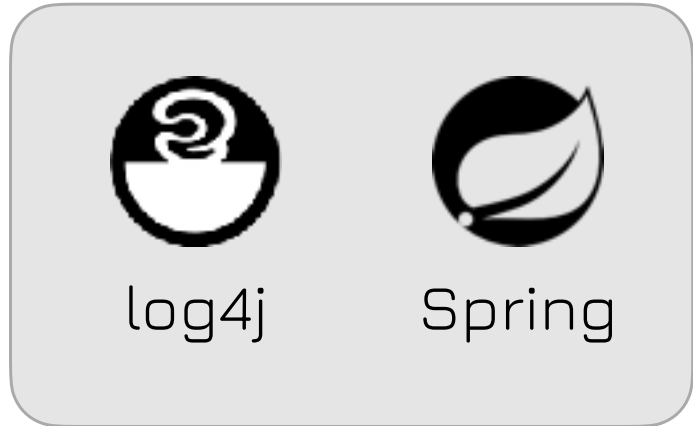


EXAMPLE

3rd party software

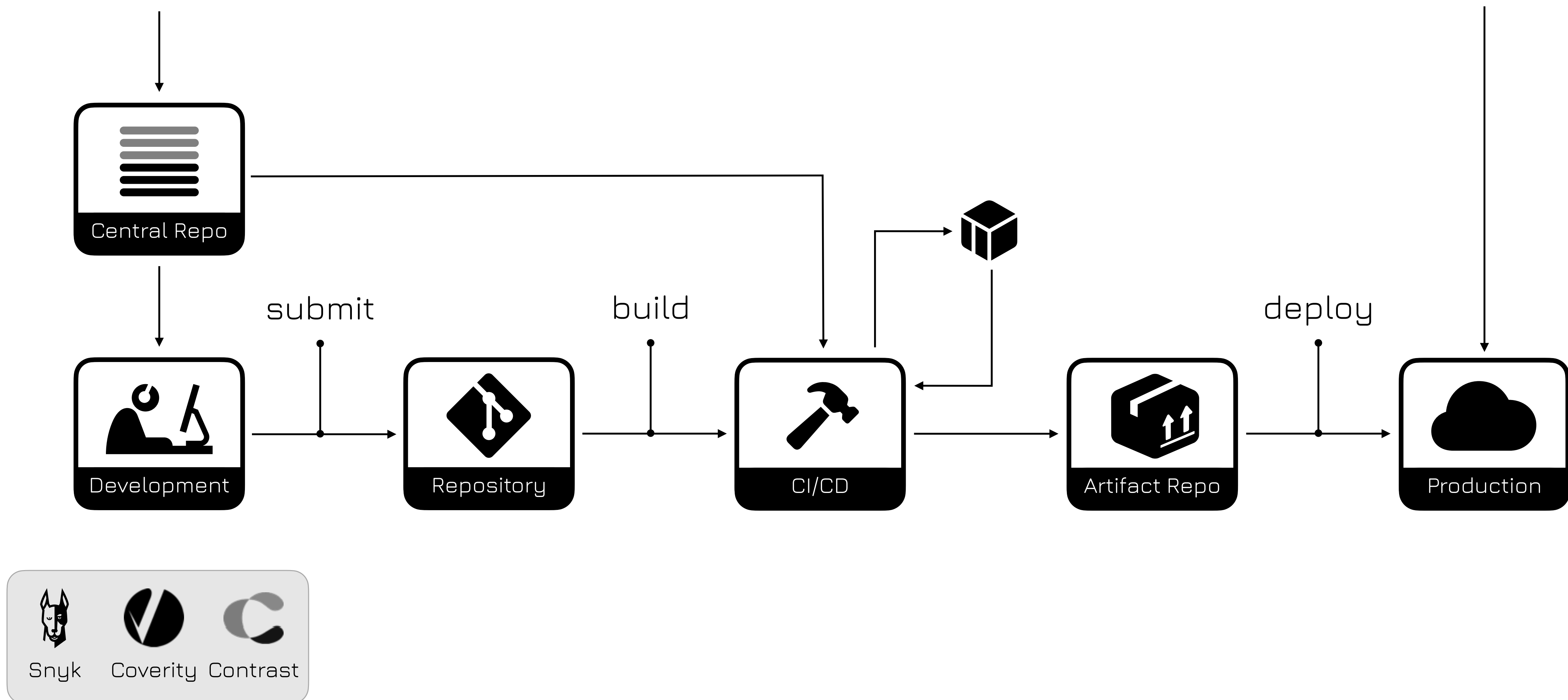
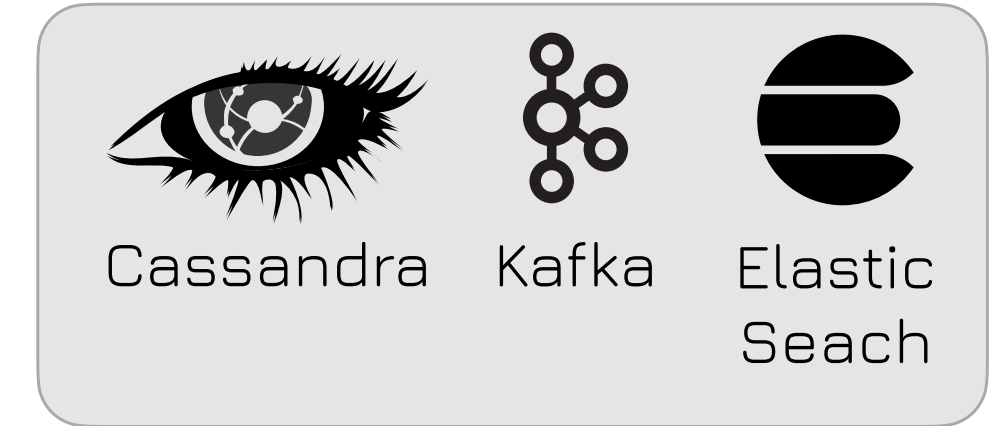


3rd party libraries



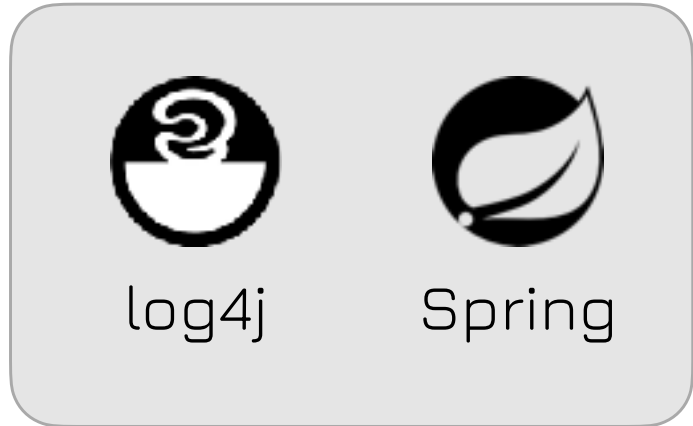
EXAMPLE

3rd party software



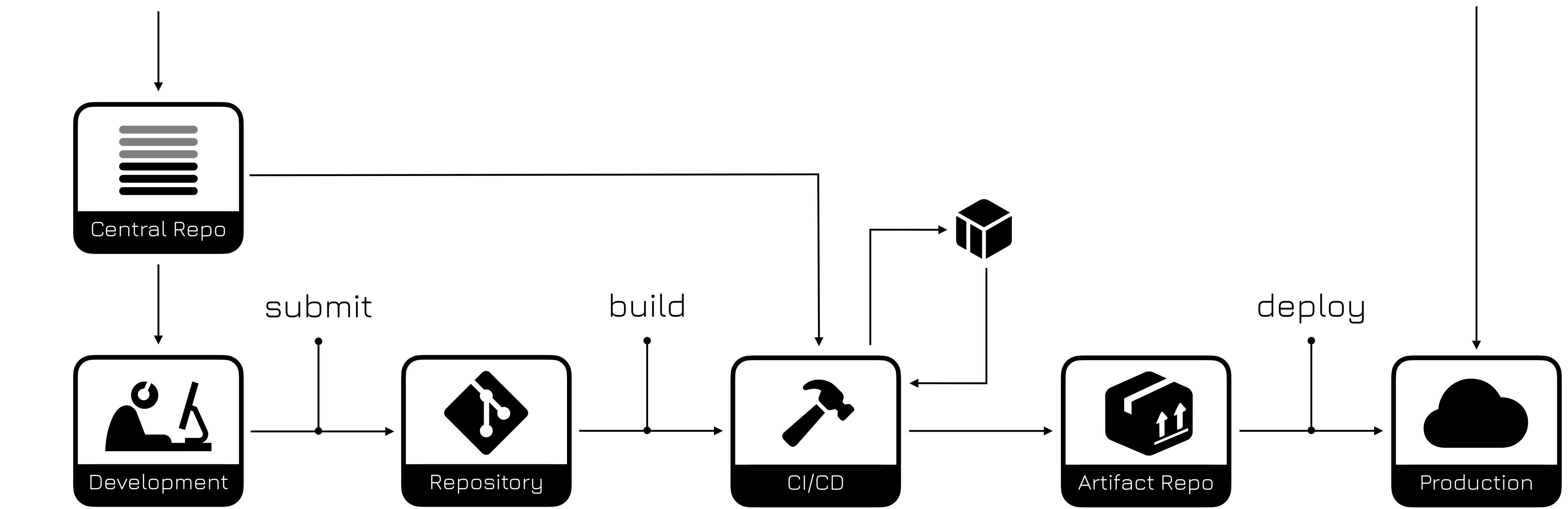
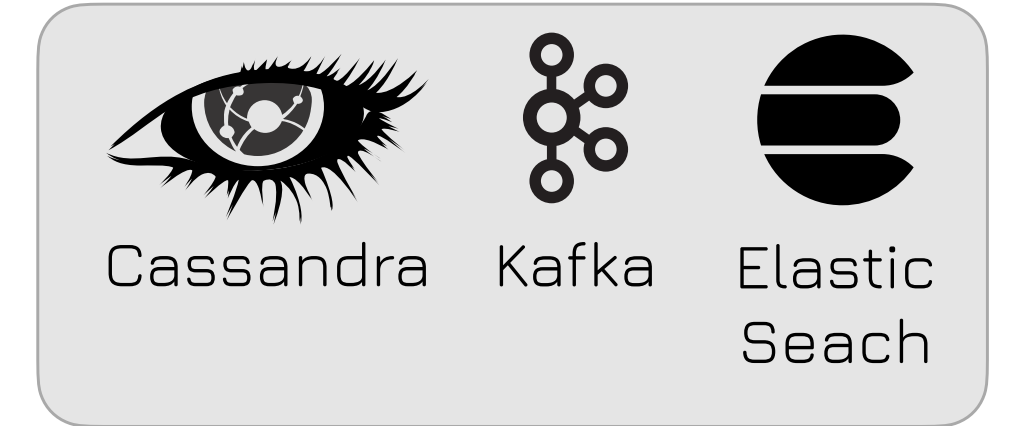
Code scanners

3rd party libraries

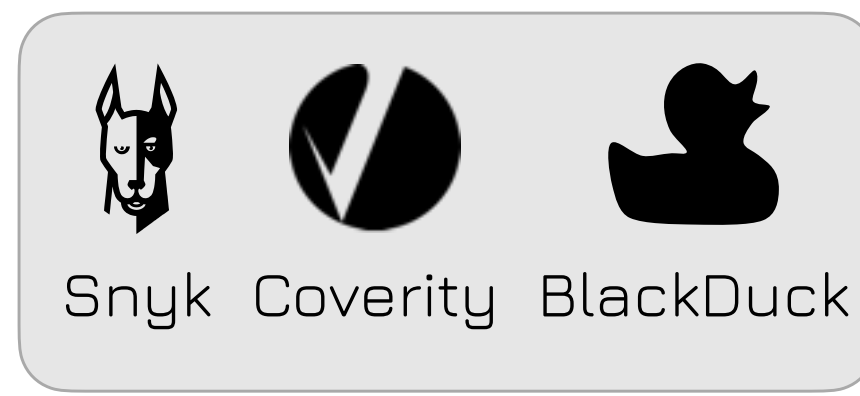


EXAMPLE

3rd party software

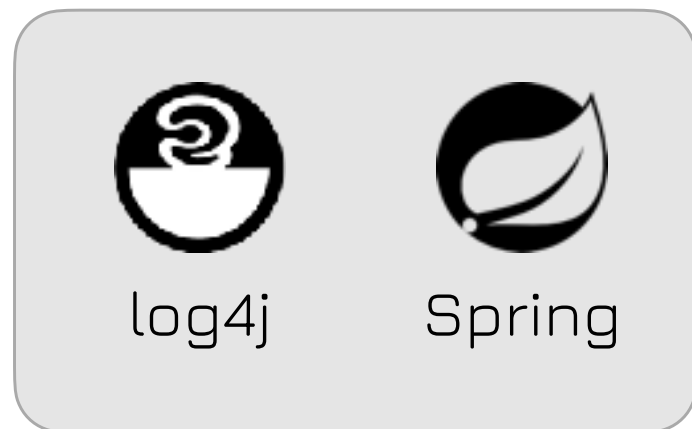


Code scanners



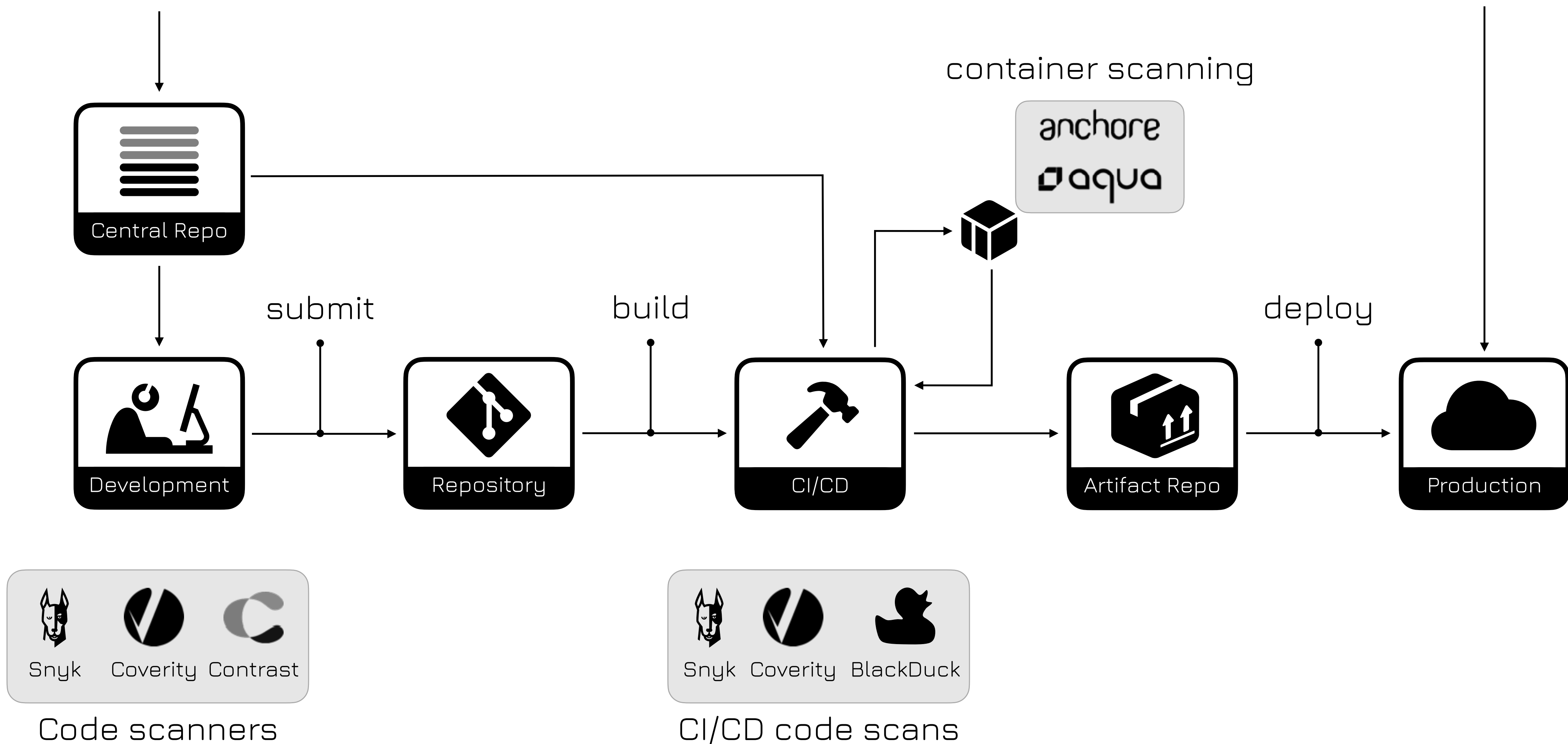
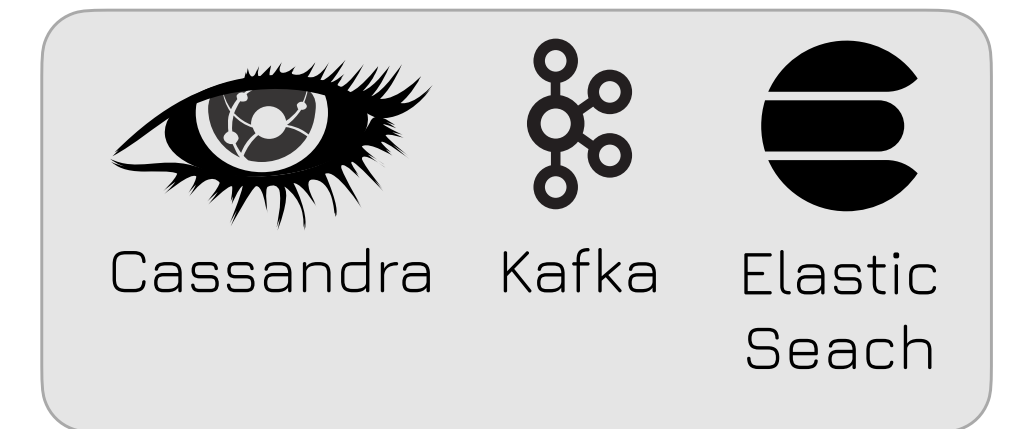
CI/CD code scans

3rd party libraries

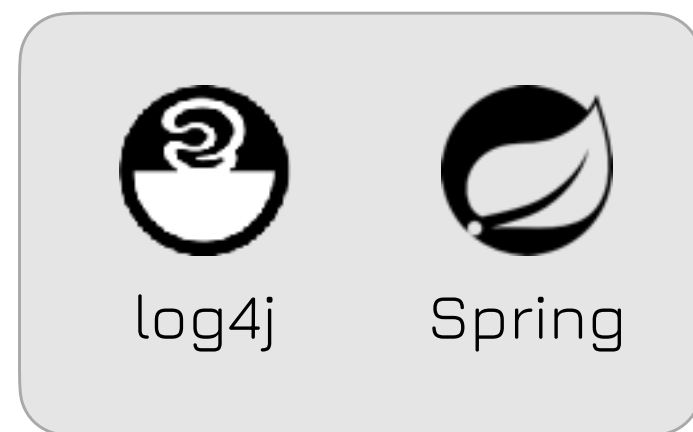


EXAMPLE

3rd party software

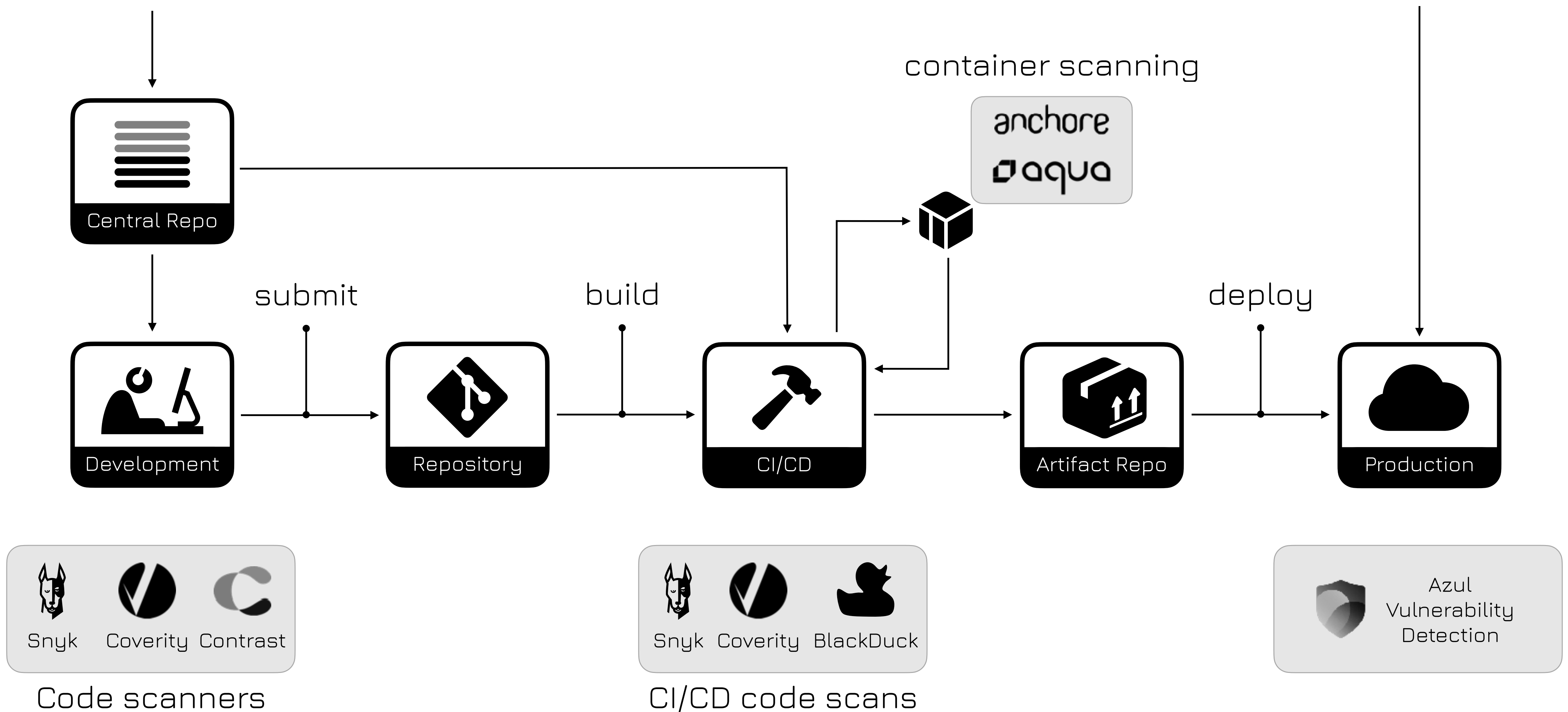
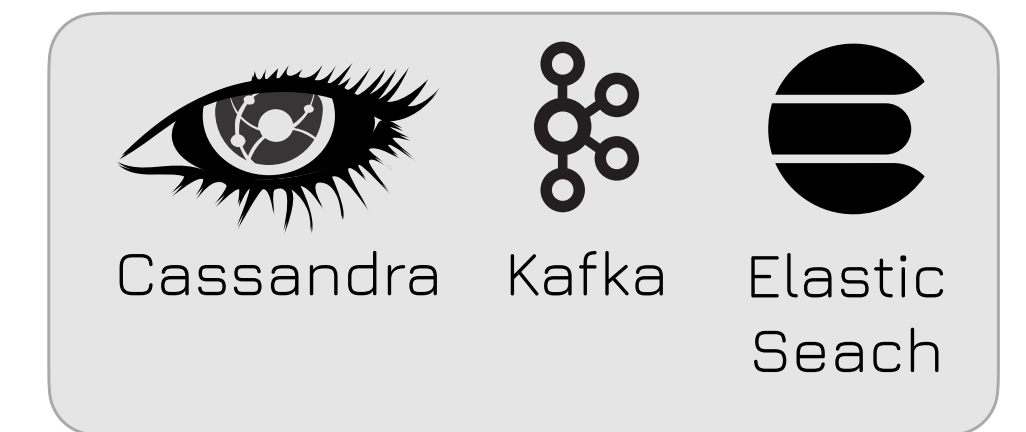


3rd party libraries





EXAMPLE

3rd party software




3rd party libraries



log4j



Spring

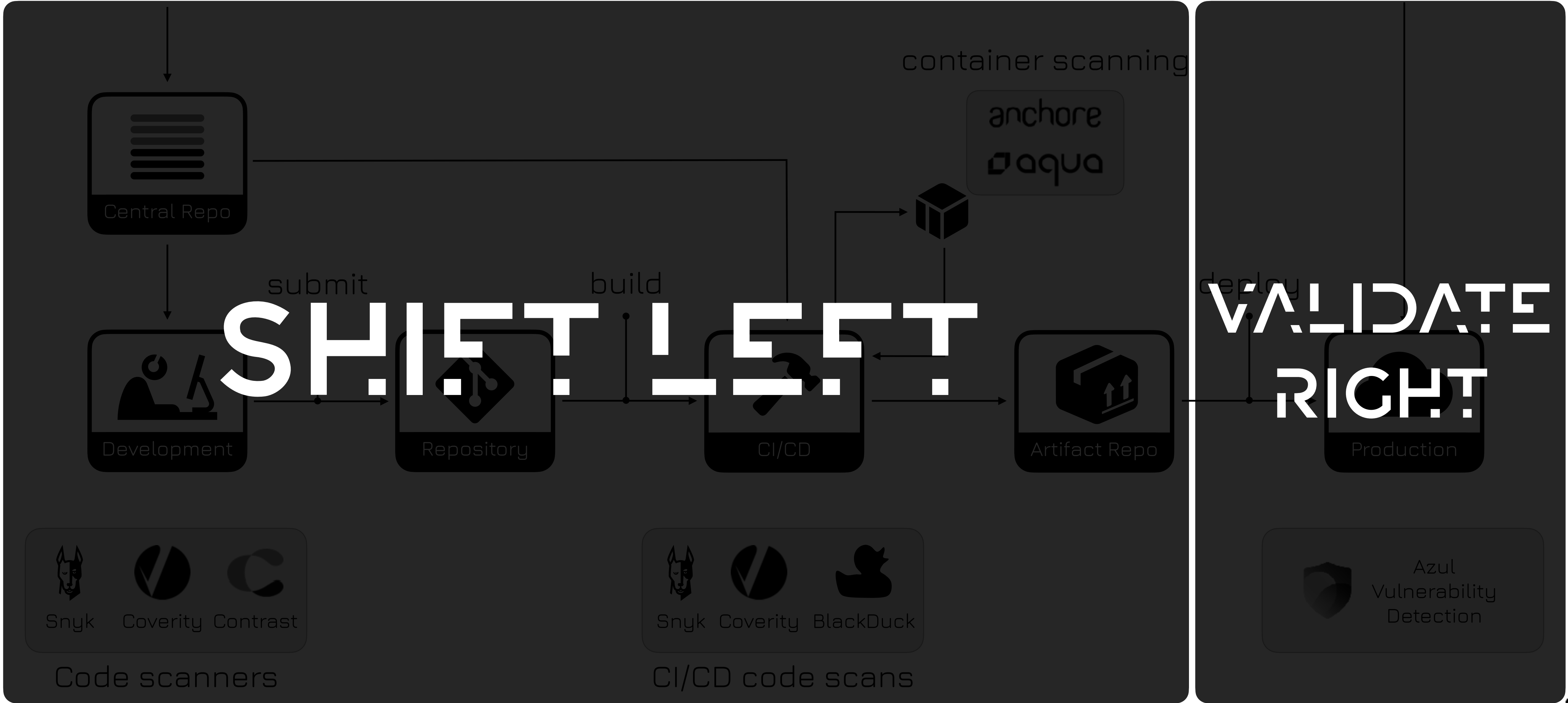
EXAMPLE

3rd party software


Cassandra


Kafka


Elastic Search



TAKEAWAY

TAKEAWAY

- ✦ Follow an automated patch schedule
(in line with your OpenJDK vendors quarterly patch cycle)
- ✦ Automate application packaging with jlink
(removing modules that are not used by your application)
- ✦ Watch for CVE's in libraries
(automate their updates in the line with the OpenJDK quarterly patch schedule)
- ✦ Use vulnerability scanners
(not only in development and CI/CD but also in production)

NEED TO BE A

SECURITY

EXPERT?

NiO...

BUT...

**YOU NEED TO
BE AWARE**



**STAY
SECURE**

