

Software Bill of Materials

Grundlage für das Management von CVE

Sebastian Hempel

ITCONSULTINGHEMPEL

- selbständiger Software-Entwickler seit 2003
- Java, Puppet
- Linux und OpenSource

<https://it-hempel.de>

@sepp@chaos.social



**Das Ganze ist nur so sicher wie jedes
seiner Teile.**

Sebastian Hempel (in Anlehnung an Aristoteles)

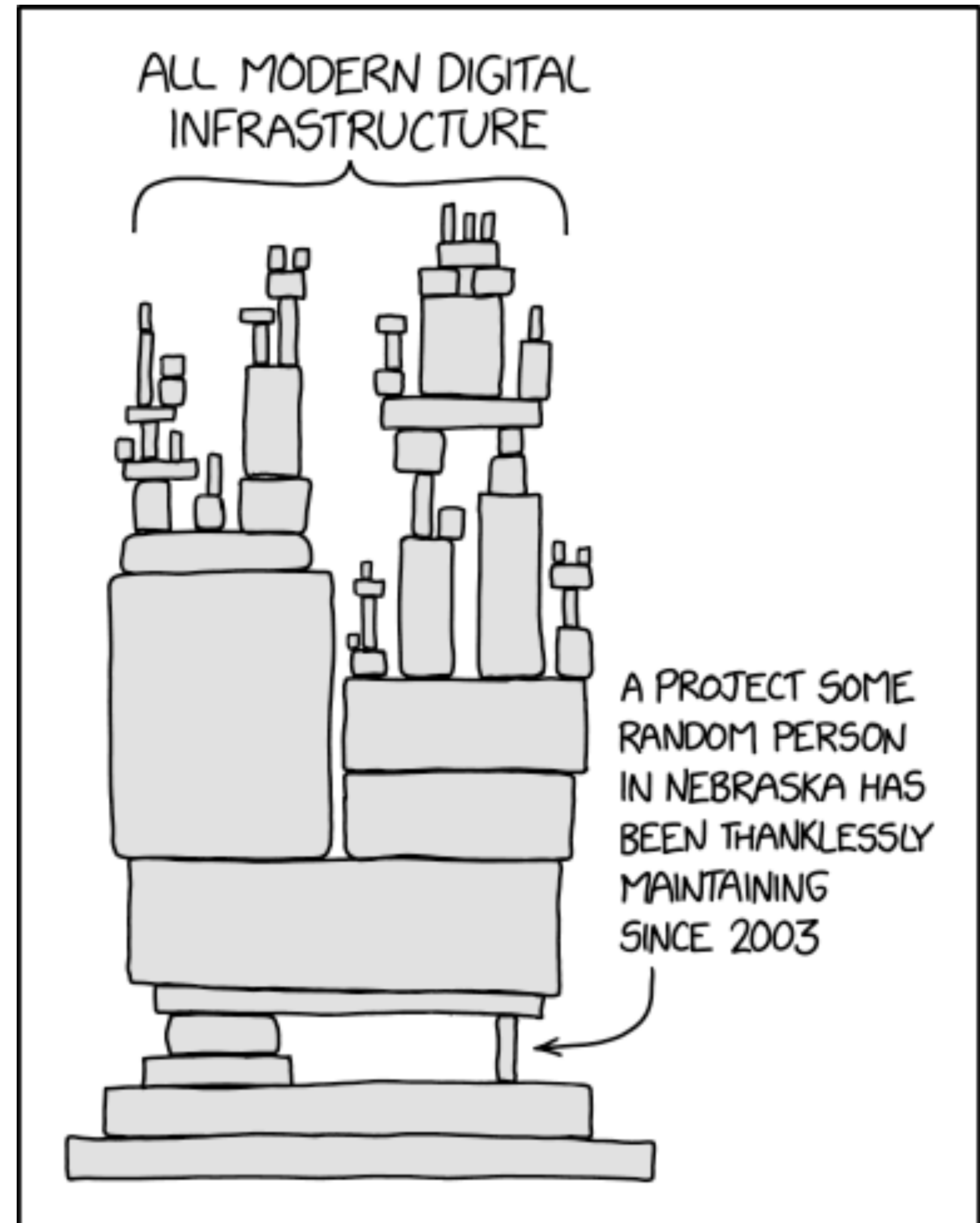
Software besteht aus mehreren Teilen

- JDK
- Framework
- Libraries
- eigene Komponenten

Software ist nie frei von Fehlern

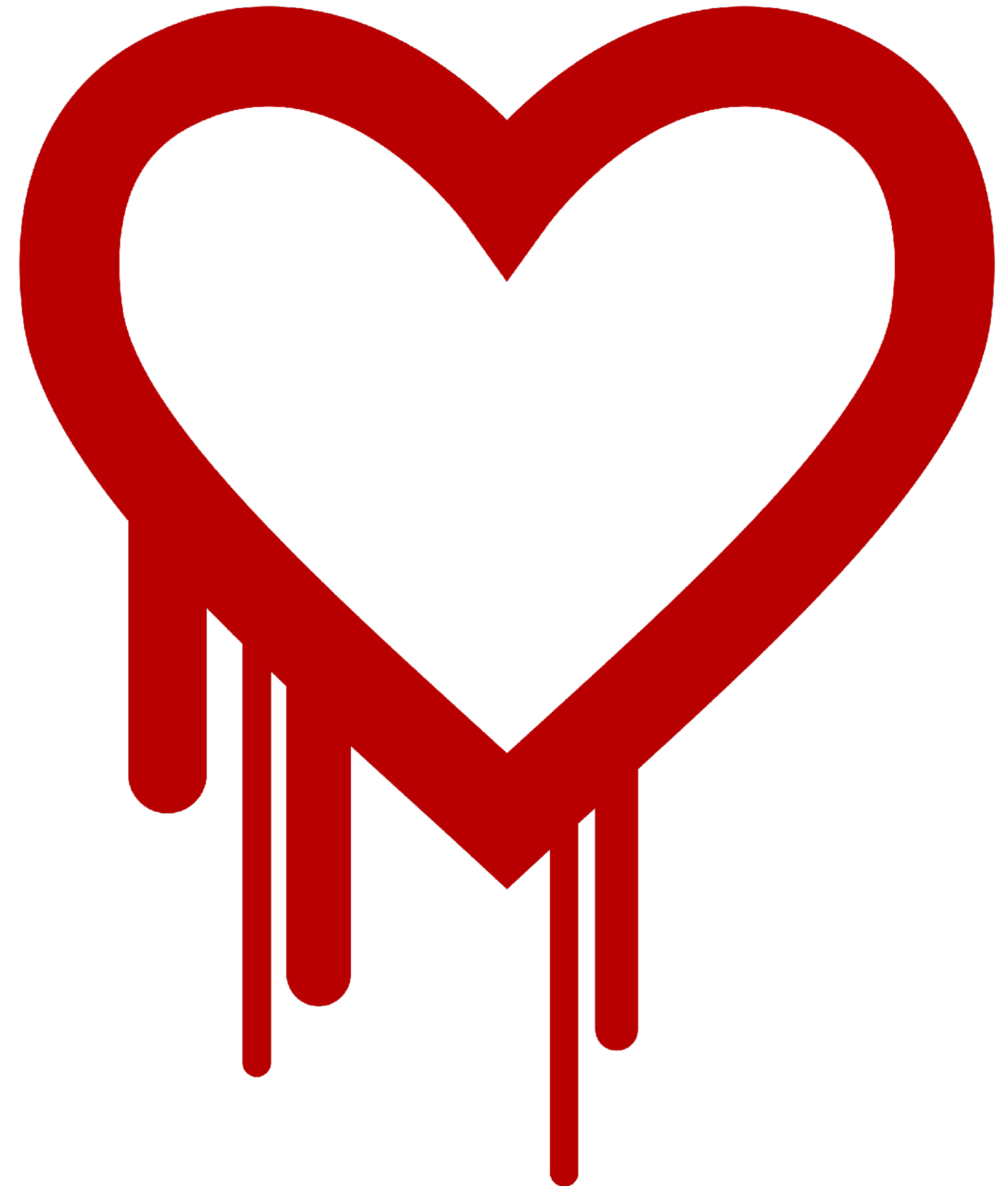
- Bugs
- CVE (common vulnerability and exposures)

Software baut aufeinander auf



Berühmte Katastrophen

- Heartbleed (2014, CVE-2014-0160) - <https://heartbleed.com/>
- Log4Shell (2021, CVE-2021-44228)
- xz Backdoor (2024, CVE-2024-3094)



Es gibt einen neuen CVE

Fragen? - Beispiel Log4j

- Ist meine Software betroffen?
- Welche Software beinhaltet Log4j?
- Welche Version kommt zum Einsatz?



Es gibt einen neuen CVE

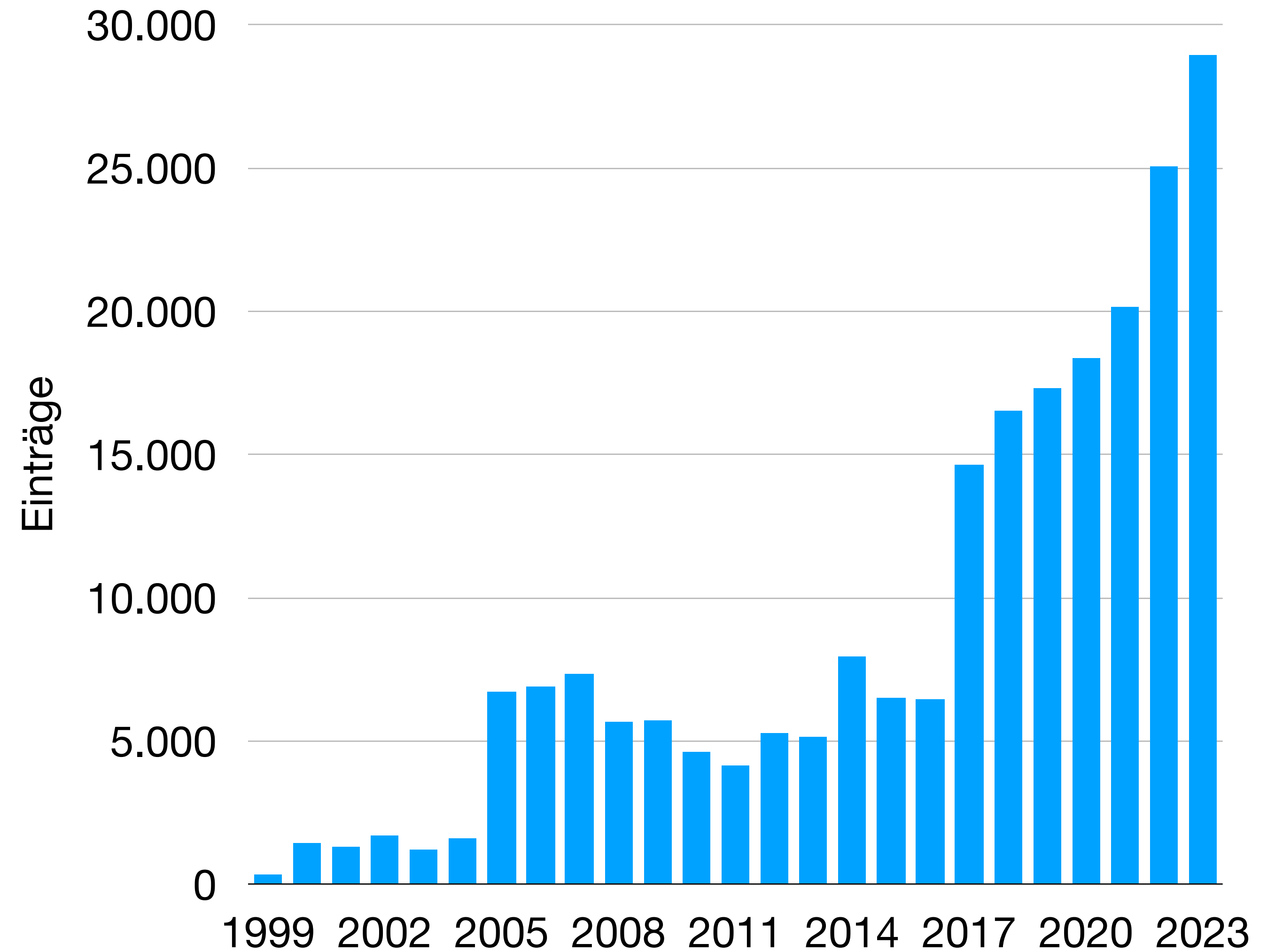
Probleme

- keine schnellen Antworten auf die Fragen
- Es vergeht wertvolle Zeit bis zur Behebung des Problems



CVE Einträge pro Jahr

- komplexere Software
- mehr Angriffe
- mehr Untersuchung
- mehr Beachtung



<https://www.cve.org/About/Metrics>

Supply Chain

- Begriff aus der Logistik
- Woher kommen meine Komponenten?
- Problematisch in Bezug auf OpenSource



Port Loading Container - makabera - Pixabay Content License

Rechtlicher Hintergrund

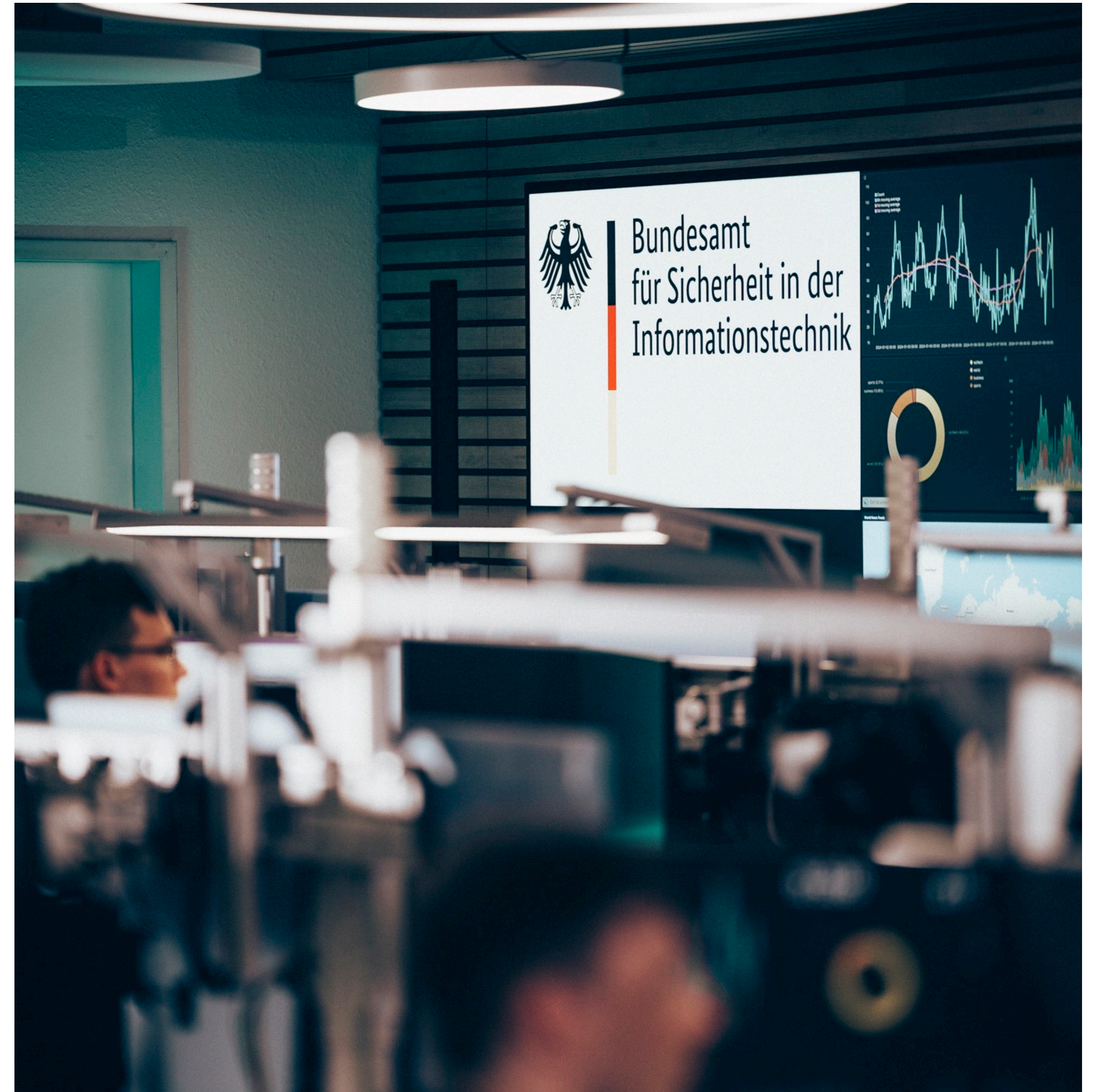


BSI

SBOM Richtlinie

TR-03183-2

04.08.2023



Nationales IT-Lagezentrum - BSI/Bernd Lammel/bundesfoto

BSI SBOM-Richtlinie

- TR-03182 „Cyber-Resilienz-Anforderungen“
- Empfehlung zur Gestaltung von SBOMs
- Verweis auf Cyber Resilience Act
- CISA (cybersecurity & infrastructure security agency)

NIS-2-Richtlinie

EU Richtlinie 2022/2555

Inkrafttreten: 16.01.2023

Anwendung: 18.10.2024



European Union Parliament - Dusan Cvetanovic - Pixabay Content License

NIS-2-Richtlinie

- Network and Information Security (NIS)
- Regelt Cyber- und Informationssicherheit von Unternehmen in 18 Sektoren
 - Cyber-Risikomanagement
 - Sicherheit in der Lieferkette
 - ...
- Richtlinie, muss in nationales Recht umgesetzt werden
- Gesetzentwurf des BMI vom Mai 2024

Cyber Resilience Act (CRA)

verabschiedet am 10.10.2024



European Union Parliament - Dusan Cvetanovic - Pixabay Content License

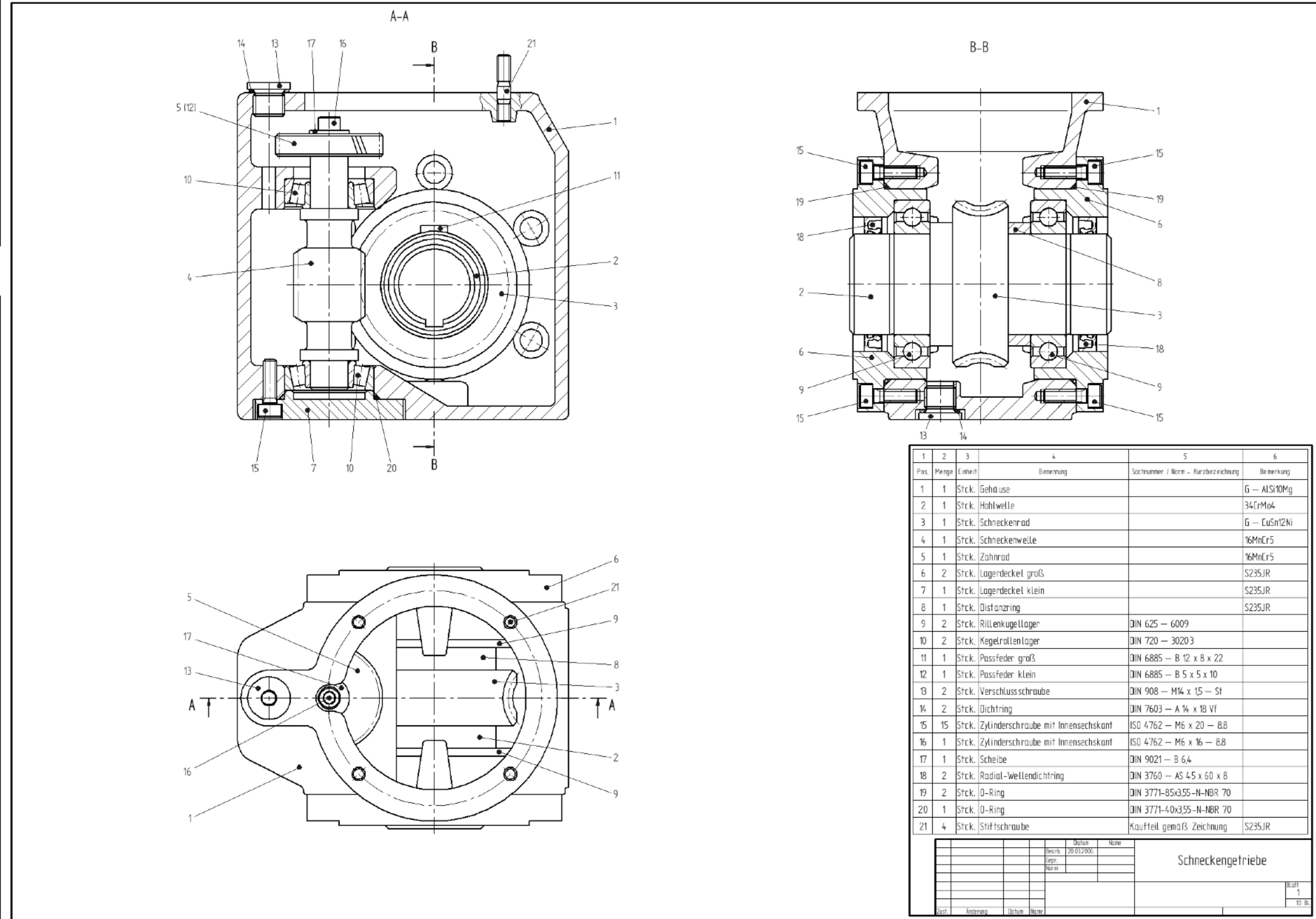
Cyber Resilience Act

„Der Cyber Resilience Act (CRA) zielt darauf ab, Verbraucher und Unternehmen zu schützen, die Produkte oder Software mit einer digitalen Komponente kaufen oder verwenden.“

- Meldung über Schwachstellen
- Lieferung von Security Updates
- Sonderbehandlung von OpenSource Komponenten
- SBOMs sind verpflichtend

Was ist eine SBOM?

Formate und Inhalte



Was ist eine SBOM?

- Stückliste aller Komponenten einer Software
- maschinenlesbar
- Angabe pro Komponente
 - Name
 - Version
 - Lizenz
- enthält direkte und transitive Abhängigkeiten

Was ist eine SBOM?

- weiteres Artefakt meiner Software
- „Beipackzettel“ bei Auslieferung
- SBOM pro Software Version / Build
- SBOM Central

Was ist eine SBOM?

Beispiel

```
<component type="library"  
  bom-ref="pkg:maven/de.ithempe1.sbom/sbom@1.0-SNAPSHOT?type=jar">  
  <group>de.ithempe1.sbom</group>  
  <name>sbom</name>  
  <version>1.0-SNAPSHOT</version>  
  <licenses/>  
  <purl>pkg:maven/de.ithempe1.sbom/sbom@1.0-SNAPSHOT?type=jar</purl>  
</component>
```


Was ist eine SBOM?

Beispiel

```
<component type="library" bom-ref="pkg:maven/org.apache.logging.log4j/log4j-api@2.23.1?type=jar">
  <publisher>The Apache Software Foundation</publisher>
  <group>org.apache.logging.log4j</group>
  <name>log4j-api</name>
  <version>2.23.1</version>
  <description>The Apache Log4j API</description>
  <scope>required</scope>
  <hashes>
    <hash alg="MD5">bee2e2dcbeeb983bdb6b71c9c3476b6a</hash>
    <hash alg="SHA-1">9c15c29c526d9c6783049c0a77722693c66706e1</hash>
    <hash alg="SHA-256">92ec1fd36ab3bc09de6198d2d7c0914685c0f7127ea931acc32fd2ecdd82ea89</hash>
  </hashes>
  <licenses>
    <license>
      <id>Apache-2.0</id>
      <url>https://www.apache.org/licenses/LICENSE-2.0</url>
    </license>
  </licenses>
  <purl>pkg:maven/org.apache.logging.log4j/log4j-api@2.23.1?type=jar</purl>
  <externalReferences><reference type="website"><url>https://logging.apache.org/log4j/2.x/log4j/log4j-api/</url></reference><reference
type="build-system"><url>https://github.com/apache/logging-log4j2/actions</url></reference><reference type="distribution"><url>https://
logging.apache.org/logging-parent/latest/#distribution</url></reference><reference type="distribution-intake"><url>https://
repository.apache.org/service/local/staging/deploy/maven2</url></reference><reference type="issue-tracker"><url>https://github.com/apache/
logging-log4j2/issues</url></reference><reference type="mailing-list"><url>https://lists.apache.org/list.html?log4j-
user@logging.apache.org</url></reference><reference type="vcs"><url>https://github.com/apache/logging-log4j2</url></reference></
externalReferences>
```

SBOM Formate

Cyclone DX

- OWASP
- XML oder JSON
- Option für CVEs
- <https://cyclonedx.org/>
- guter Java Support



SBOM Formate

SPDX (Software Package Data Exchange)

- Linux Foundation
- JSON, YAML, XML, xls
- ISO / IEC 5962:2021
 - <https://spdx.dev/>
- SPDX Export von GitHub



Erstellen einer SBOM

Ansätze in Java Ökosystem



Stückliste von was?

- Design SBOM
- Source SBOM
- **Build SBOM** (pom.xml, build.gradle)
- **Analyzed SBOM** / 3rd party SBOM (JAR)
- **Deployed SBOM** (Docker Image)
- Runtime SBOM / Dynamic SBOM

Build SBOM

Maven / Gradle

- Ermittlung aller Abhängigkeiten des Projekts
- Nutzung vorhandener Informationen
- weiterer Build-Schritt
- weiteres Artefakt



CycloneDX Maven Plugin

OWASP Foundation

- Erstellt SBOM pro pom.xml
- Option zur Aggregation mehrere SBOMs eines multi-module Builds
- CycloneDX Format der OWASP
- Artefakt Upload in Repository

CycloneDX Maven Plugin

Konfiguration

```
<plugins>
  <plugin>
    <groupId>org.cyclonedx</groupId>
    <artifactId>cyclonedx-maven-plugin</artifactId>
    <version>${cyclonedx.version}</version>
    <executions>
      <execution>
        <phase>package</phase>
        <goals>
          <goal>makeAggregateBom</goal>
        </goals>
      </execution>
    </executions>
  </plugin>
```


CycloneDX Maven Pluing

Spring Boot

- Spring Boot ≥ 3.3
- erkennt CycloneDX Plugin
- Quelle: Thomas Vitale via LinkedIn



Thomas Vitale • 2.

Software Engineer | Author of "Cloud Native Spr...

17 Std. • Bearbeitet •

+ Folgen

Spring Boot 3.3 includes out-of-the-box support for SBOMs. If you include the OWASP CycloneDX Maven/Gradle plugin in your project, Spring Boot will detect it and automatically generate an SBOM when you build the application. It will export the SBOM as a JSON file, and serve it via a dedicated Actuator endpoint. You can then upload the JSON file to your favourite solution for managing supply chain security risks, such as the open-source solution from OWASP, DependencyTrack.

[#SpringBoot](#) [#SBOM](#) [#SupplyChainSecurity](#) [OWASP CycloneDX SBOM/xBOM Standard](#)

SBOMs in Spring Boot 3.3

SBOM Plugin

```
build.gradle
```

```
plugins {  
    id 'org.cyclonedx.bom' version '1.8.2'  
}
```

Out-of-the-box SBOM via Actuator Endpoint and JSON File

`http :8080/actuator/sbom` `bom.json`

SPDX Maven Plugin

Linux Foundation

- Erstellt SBOM pro pom.xml
- Lifecycle Phase verify
- SPDX Format der Linux Foundation

SPDX Maven Plugin

Konfiguration

```
<plugin>
  <groupId>org.spdx</groupId>
  <artifactId>spdx-maven-plugin</artifactId>
  <version>${spdx.version}</version>
  <executions>
    <execution>
      <phase>verify</phase>
      <goals>
        <goal>createSPDX</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <excludeFilePatterns>
      <excludeFilePattern>*.spdx</excludeFilePattern>
    </excludeFilePatterns>
  </configuration>
</plugin>
```

Analyzed SBOM JAR

- Ermittlung auf dem Artefakt
- Scanner benötigt Wissen über Format
- Beispiel: Syft



Syft

Artefakt Scanner

- Scanner basierend auf Artefakt
- Umfangreicher Support
- JAR



syft

[syft](#)



Scannen eines JAR

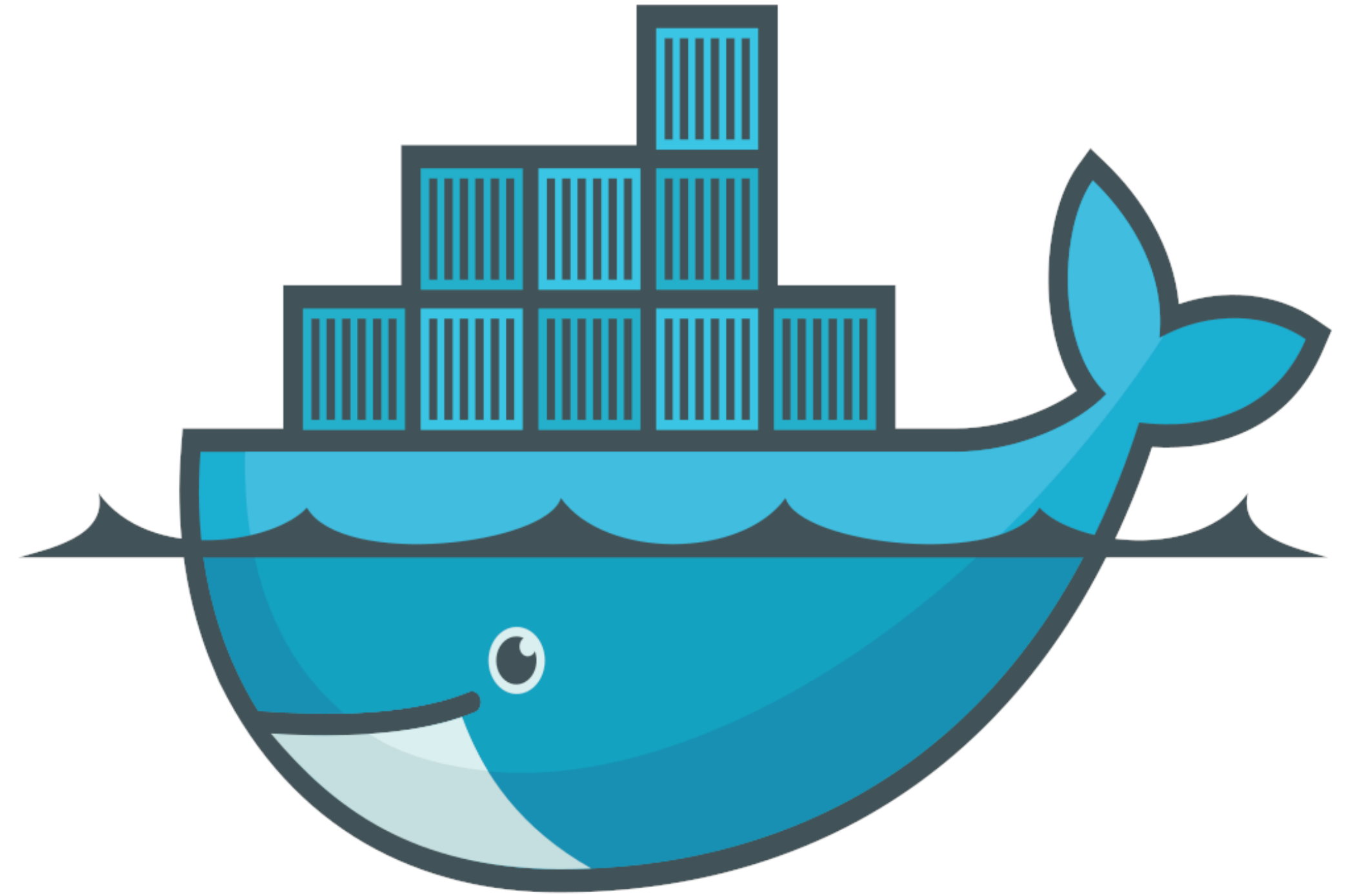
```
syft artifact.jar -o cyclonedx-xml --file bom.syft.cdx.xml
```

- Scannt das Artefakt artifact.jar
- Erstellt ein SBOM im CycloneDX Format (XML)
- erkennt „eingebettete“ JARs

Deployed SBOM

Docker Image

- Ermittlung basierend auf dem Docker Image
- Scanner liest Layer
- Beispiel: Syft



docker



Scannen eines Docker Images

```
syft scan appimage -o cyclonedx-xml --file docker.image.xml
```

- Scann das Image appimage
- Erstellt SBOM im CycloneDX Format (XML)
- erkennt Layer und deren Properties

CVE Scan

Software Composition Analysis (SCA)



CVE Scan

Prüfung

- Prüfung auf CVEs in der SBOM
- Abgleich mit Datenbanken
 - National Vulnerability Database (NVD) des NIST
 - OpenSource Vulnerability Database (OSV) von Google
 - Datenbanken von SCA Herstellern

CVE Scan

Klassifizierung

- Ist der CVE relevant für meine Software?
 - Wird betroffene Funktionalität genutzt?
 - Gibt es Gegenmaßnahmen?
- Vulnerability Exploitability Exchange (VEX)
 - vom Hersteller als eigenes Dokument
 - Referenz auf SBOM
 - eingebettet bei SBOM-Erstellung

CVE Scan

- SBOM ist nach Erstellung statisch
- CVE Scan sollte regelmäßig durchgeführt werden

trivy

**OpenSource Vulnerability
Scanner**





Funktionsumfang

- Command-Line Tool
- Quellen: Dateisystem, Docker Images, Code Repositories und SBOMs
- unterstützt VEX
- erkennt vulnerabilities, misconfiguration, secrets und Lizenzen



Beispiel

```
trivy sbom target/bom.json
```

Prüfung der SBOM auf vulnerabilities

```
trivy sbom --scanners=vuln,license
```

zusätzlich Prüfung der genutzten Lizenzen

Lizenzprüfung

Kompatibilität der FLOSS
Lizenzen



**open source
initiative[®]**

open source initiative - Colin Viebrock - CC BY

Lizenzprüfung

- Auflistung der Lizenzen der Komponenten
- Passen die Komponenten zusammen?
- Nutzung von verbotenen Lizenzen (im Enterprise Umfeld)

Identifizierung

Wie erkenne ich eine Komponente?

Dear Mr Turing,
Following
Please take this
Park.
Please attend
letter with you.

NATIONAL REGISTRATION IDENTITY
NUMBER AND NAME OF HOLDER

NO P Q / 1028 / 709.

ALAN TURING.

Issuing Office H.M.N_1

Name of Issuing Officer (surname in Block Capitals)
James Plister

Signature of Chief Officer

Signature of Bearer

Visible distinguishing marks
— none —

(1) Surname TURING

(2) Surname at Birth — 11 —

(3) Other Names Mattison
(and rank or designation)

(4) Unit or occupation UNIT 8 F

(5) Nationality British

(6) Nationality at Birth — 11 —

(7) Second Nationality, if any — NA —

(8) Place of Birth London, England

(9) Year of Birth 1912

E.D. Letter Code
26.01 / FCN

Schedule Number
00759.

Portrait of Alan Turing

Serial No. BP 120380

Identifizierung

- Standard zur eindeutigen Identifizierung einer Komponente
- Vereinheitlichung der Komponenten in SBOMs
- Wichtig zur Zuordnung von CVEs

SWID Tagging

- Software Identification (SWID) Tagging
- ISO 19770-2:2015
- Ursprünglich für Software Asset Management
- keine Verbreitung
- <https://csrc.nist.gov/Projects/Software-Identification-SWID>

CPE

- Common Platform Enumeration
- zentraler Ansatz / zentrales Verzeichnis
- NIST
- <https://csrc.nist.gov/projects/security-content-automation-protocol/specifications/cpe>

PURL

- Package URL
- Orientierung an Paketmodellen (Maven, NPM, ...)
- dezentraler Ansatz
- OpenSource
- Github: package-url / purl-spec (<https://github.com/package-url/purl-spec>)

PURL

Beispiel

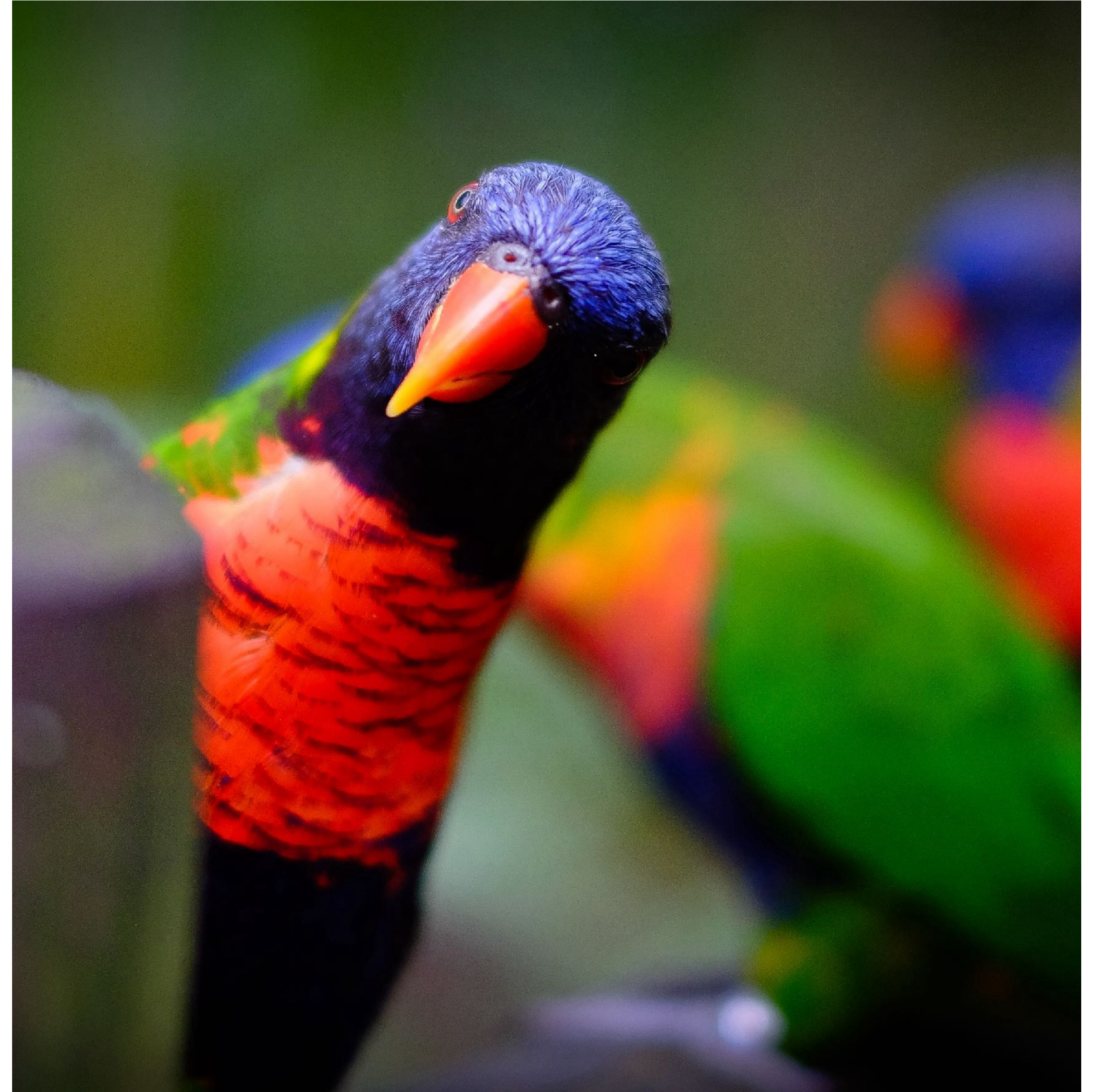
`scheme:type/namespace/name@version?qualifiers#subpath`

`pkg:maven/org.apache.logging.log4j/log4j-api@2.23.1?
type=jar`

`pkg:gem/jruby-launcher@1.1.2?platform=java`

`pkg:npm/%40angular/animation@12.3.1`

Fragen?



any questions? - (Matthias Ripp) - CC BY