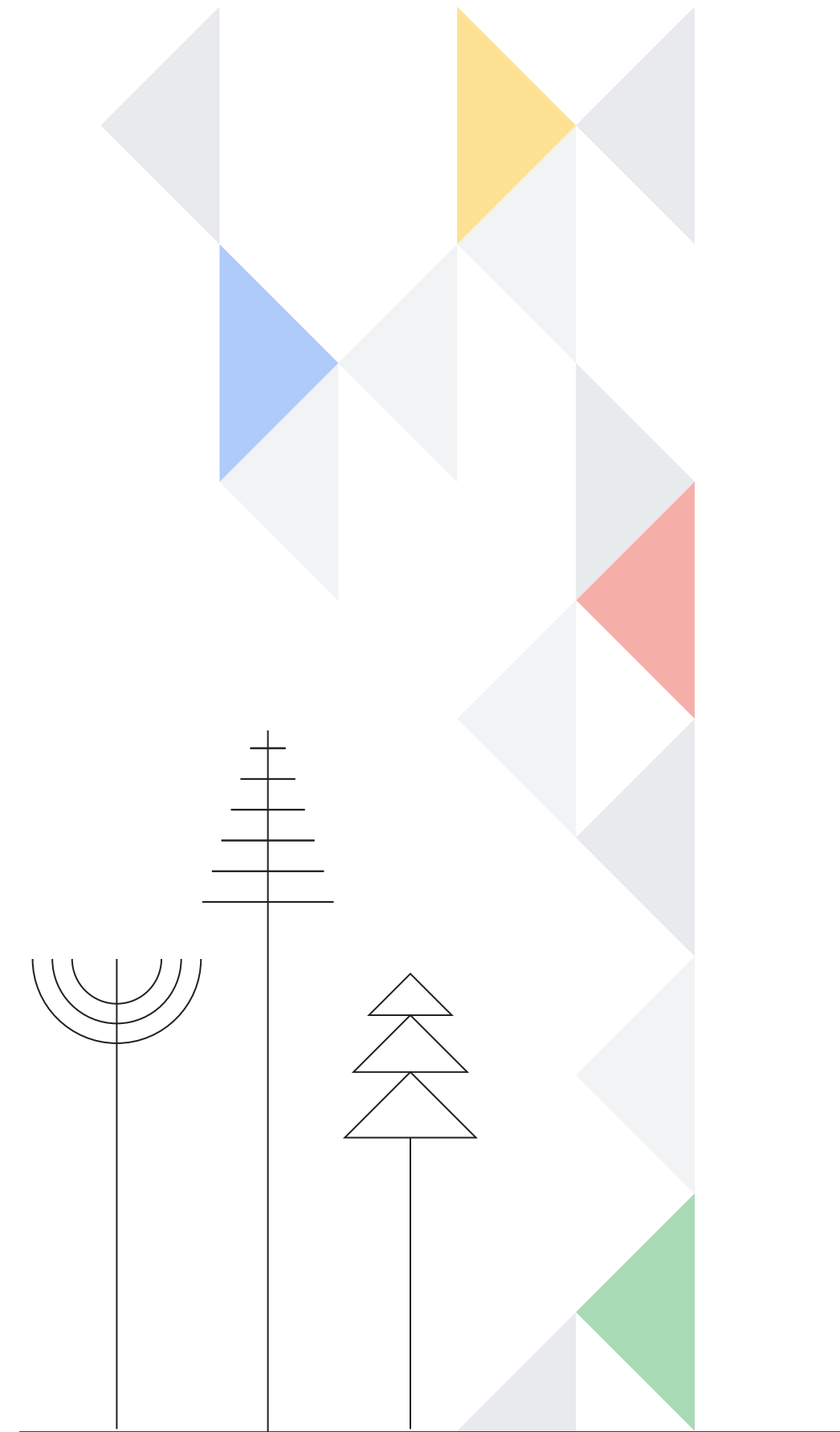


# CIAO Torino

## Secure Your Software Delivery from Dev to Prod

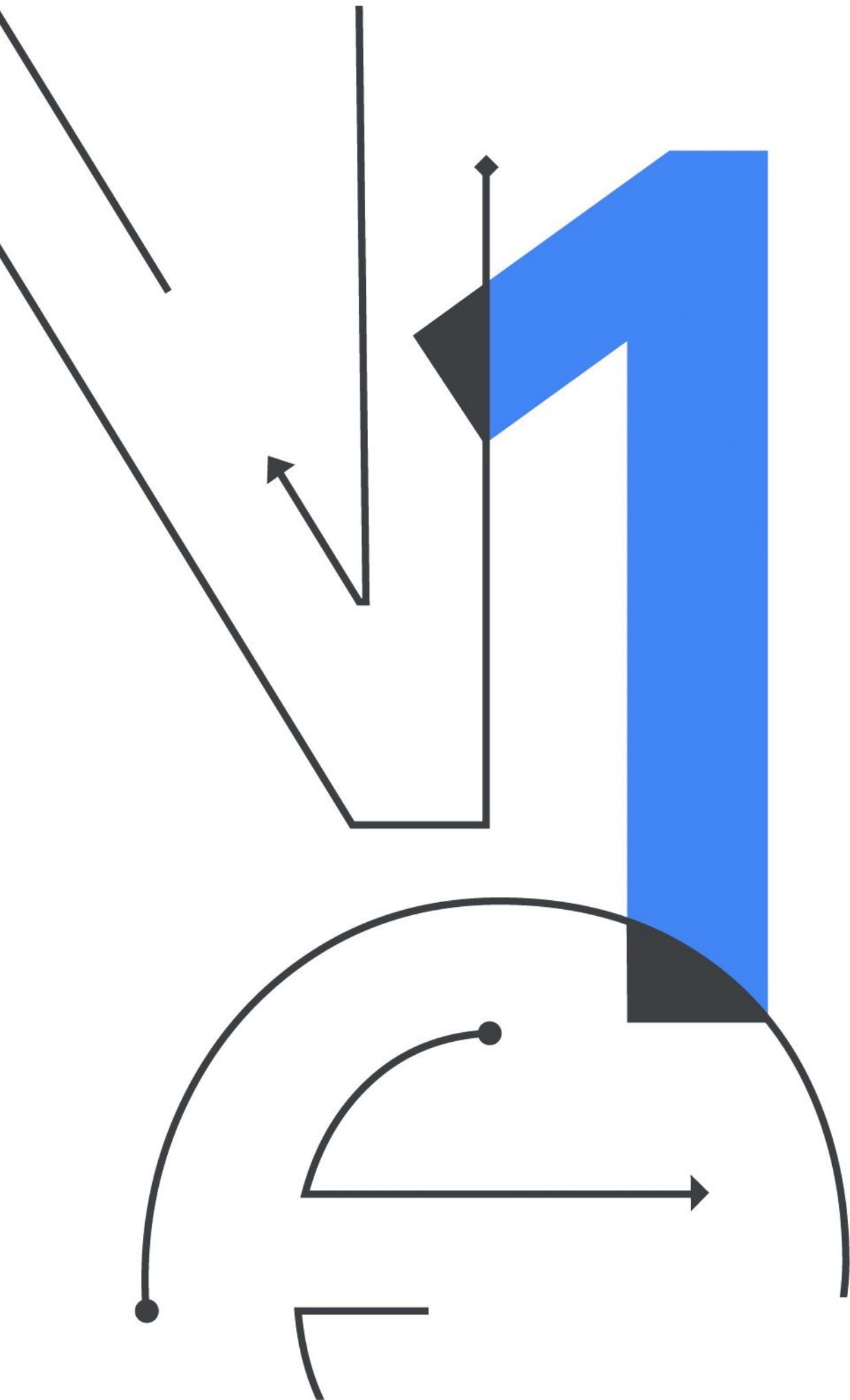


# Abdel Sghiouar

Senior Cloud Developer Advocate @Google  
Kubernetes Podcast co-host

**Twitter: @boredabdel**





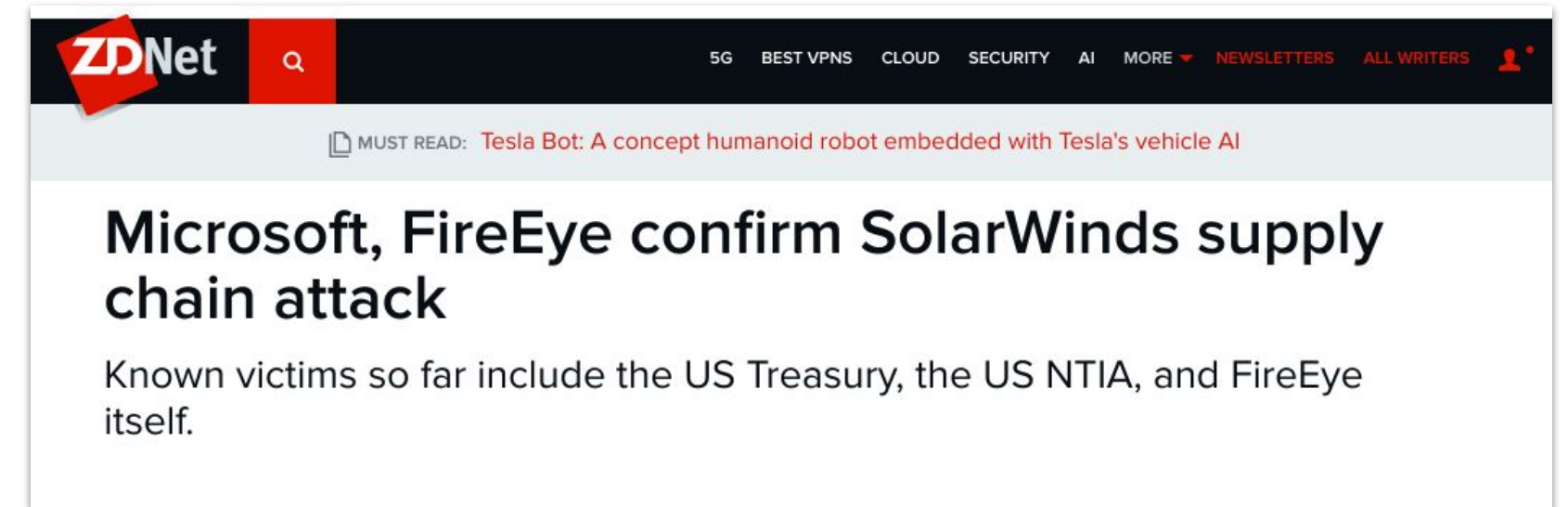
Why it matters

# Software Supply Chain Security

# Growing number of attacks against Software

## Supply Chain

- **SolarWinds** supply chain attack impacted 18,000 customers
- **Log4j** vulnerability affected millions of **Java** applications that “**set the Internet on fire**”.
- **Next-gen** Supply chain attacks surge **430%** according to Sonatype
- **Security vendor** FireEye red team tools stolen in cyber attacks

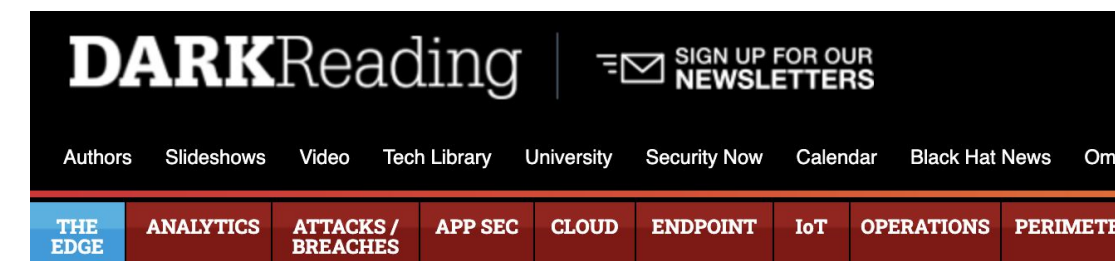


WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS SCIENCE SECURITY

LILY HAY NEWMAN SECURITY 12.18.2021 02:54 PM

## 'The Internet Is on Fire'

A vulnerability in the Log4j logging framework has security teams scrambling to put in a fix.



### APPLICATION SECURITY

8/21/2020  
10:20 AM

## 'Next-Gen' Supply Chain Attacks Surge 430%



Attackers are increasingly seeding open source projects with compromised components.



# Case in Point: Coop Sverige



Coop



IT Company

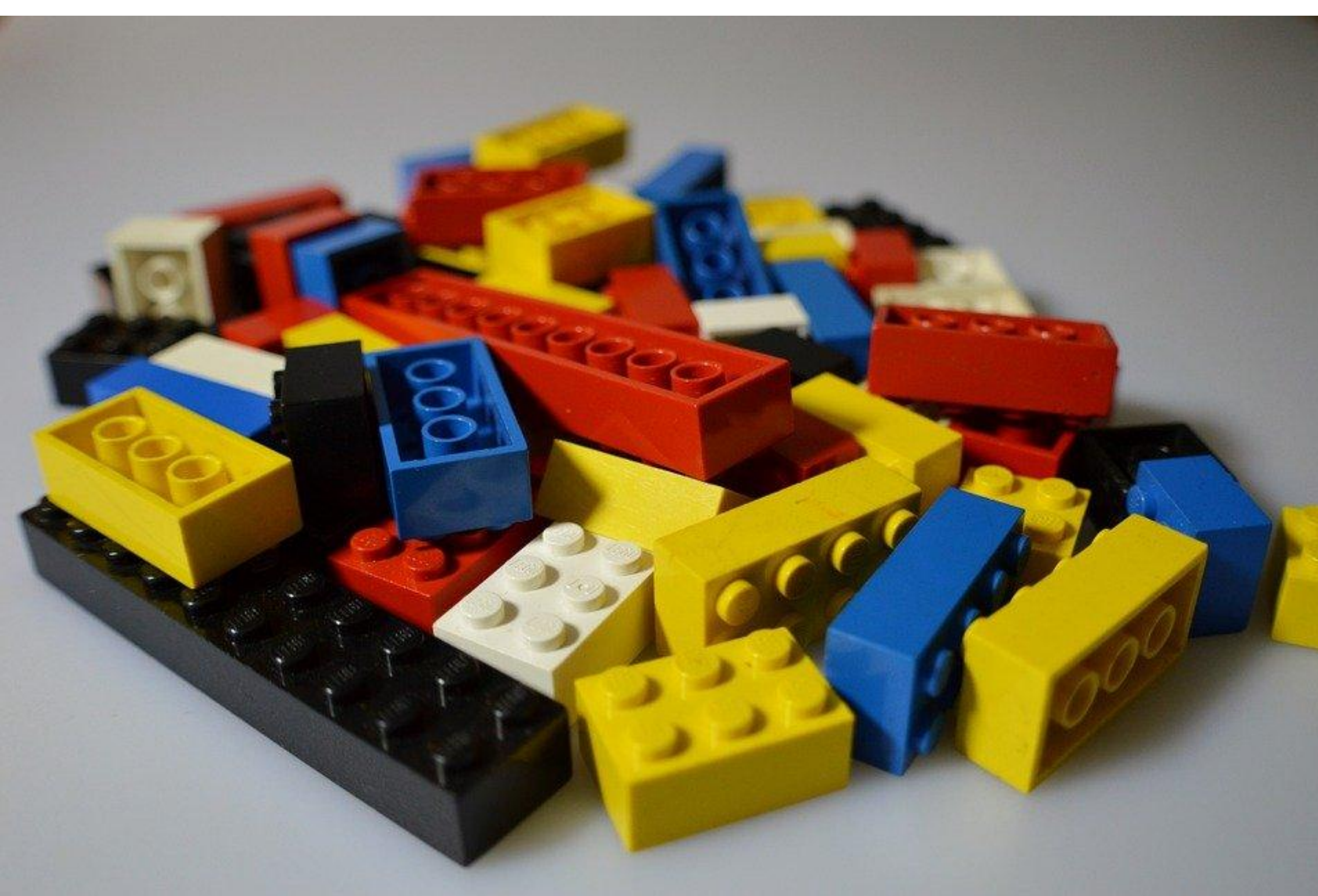


Kaseya

500

Stores closed for >24h

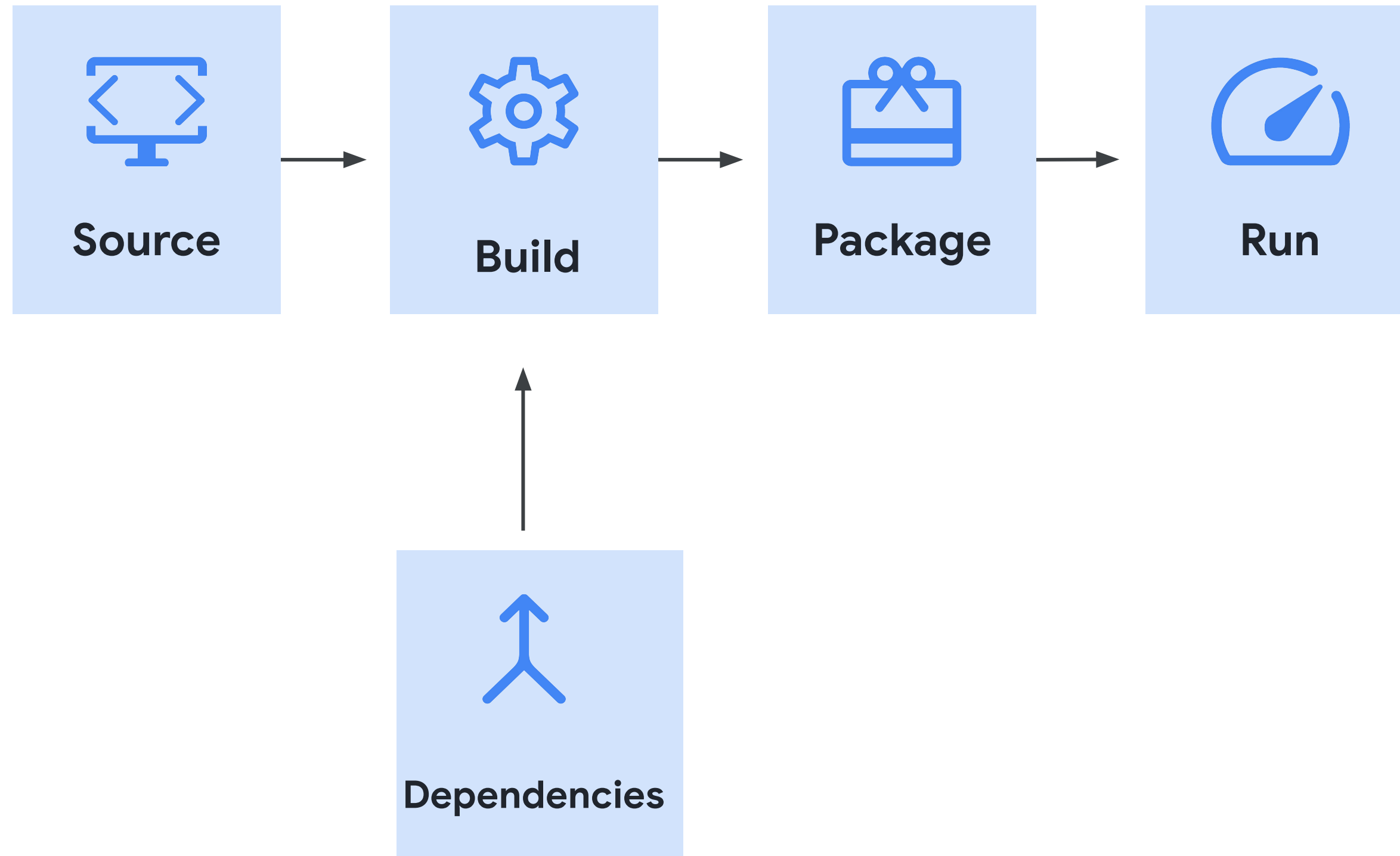




**How software  
is built today**

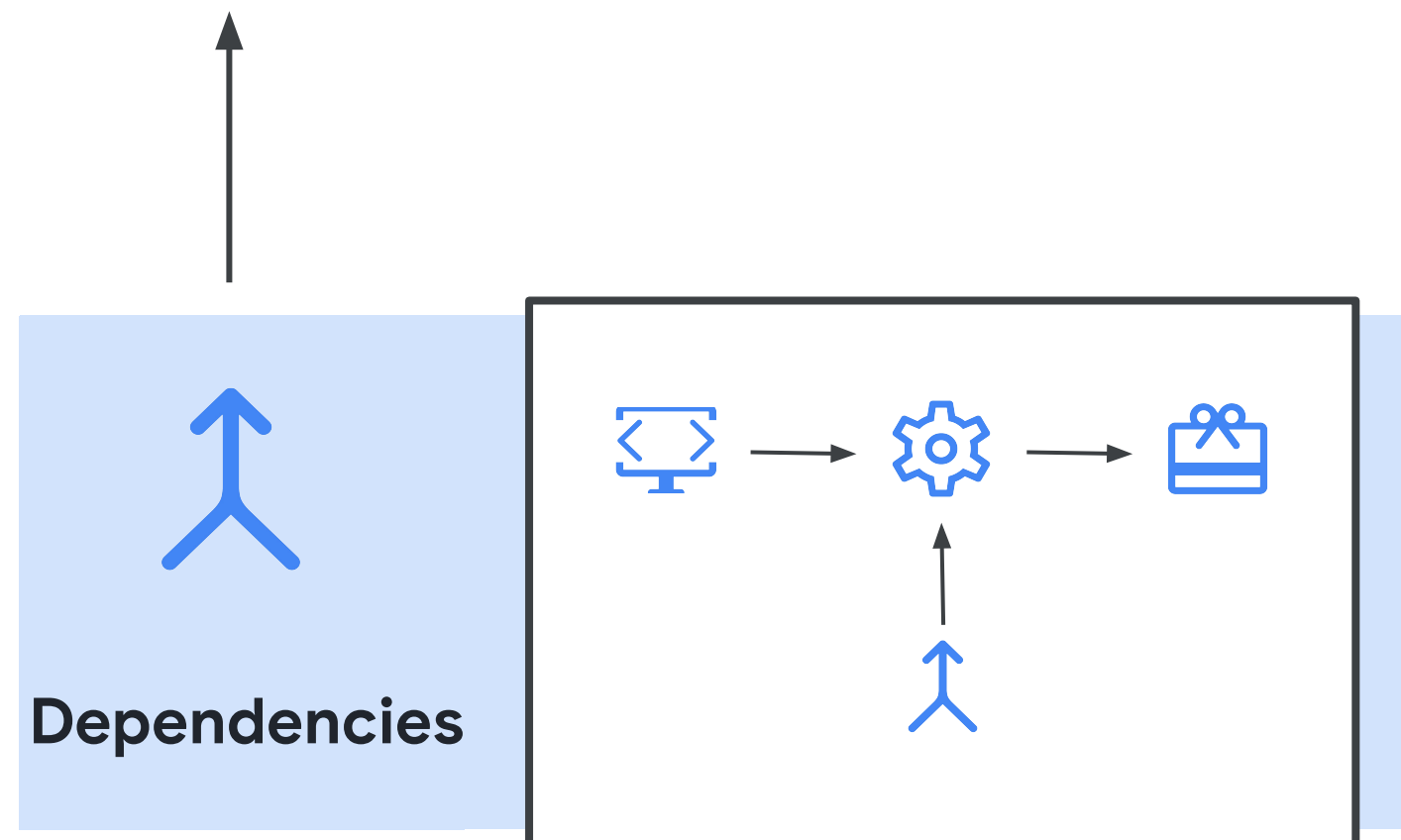
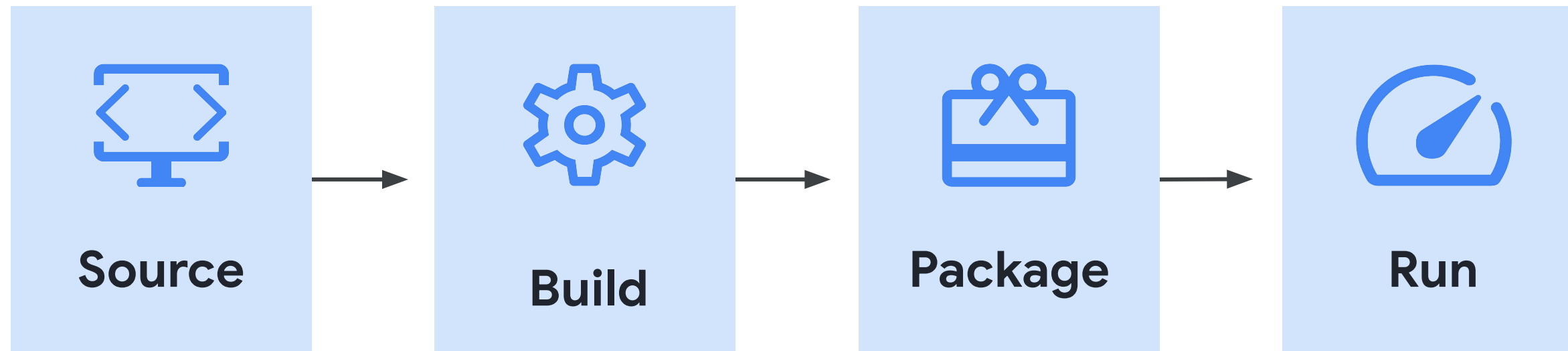
# It looks simple

---

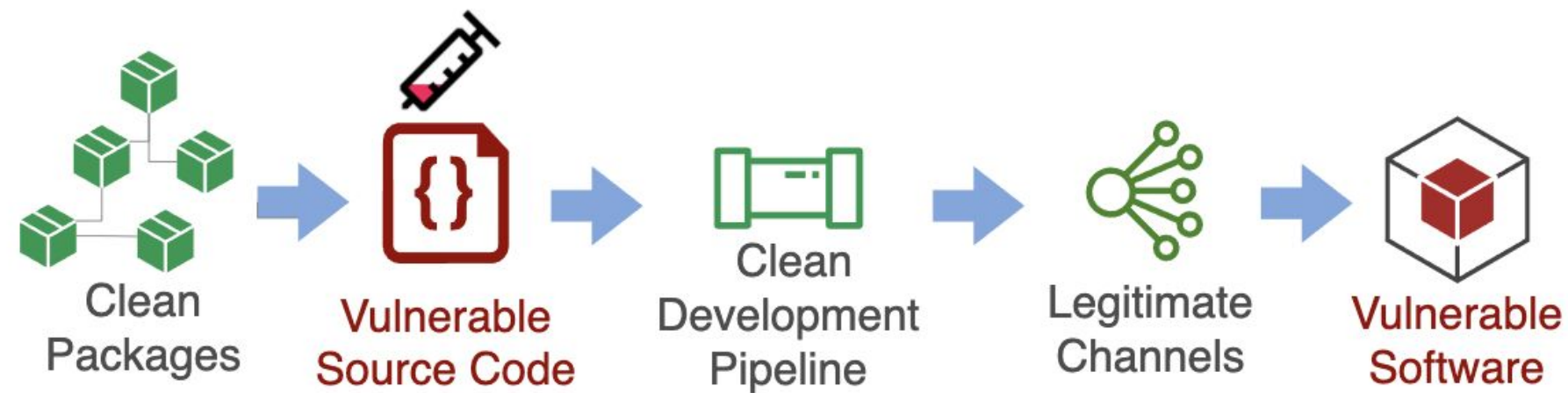
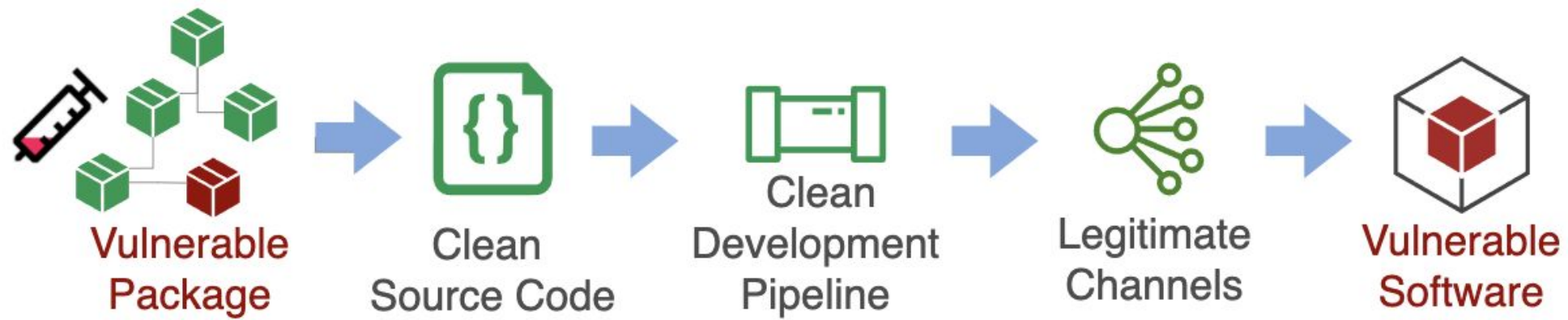


# It's not

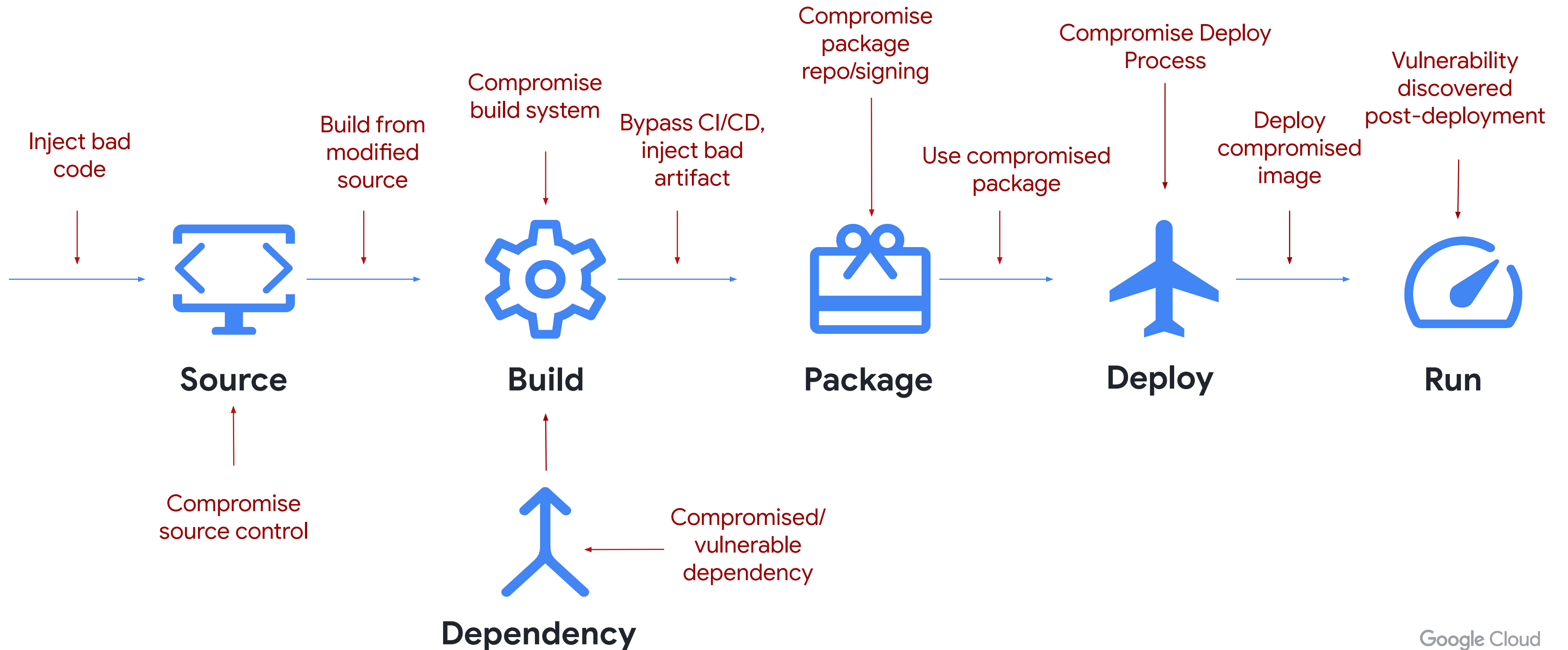
---







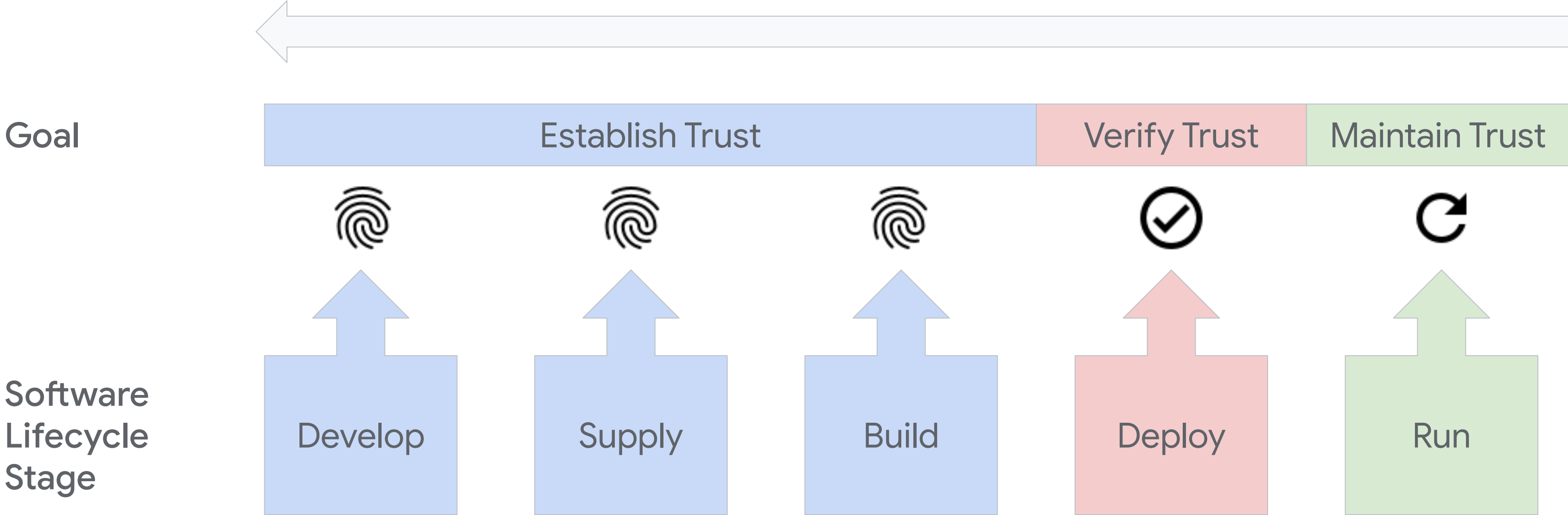
# Attack vectors





—●  
**Now what ?**

# Zero-trust and shift-left





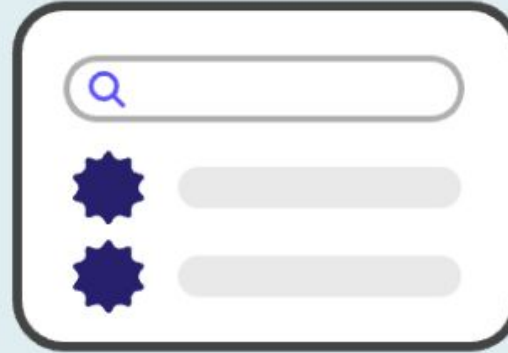
# Sigstore.dev



## Sign


Easy authentication and smart cryptography work in the background. Just push your code, sigstore can handle the rest.

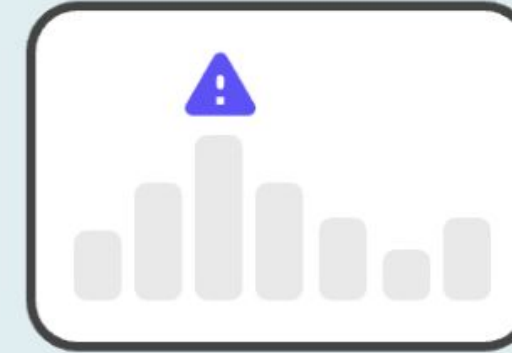
[Learn more](#) 



## Verify


Rekor transparency logs store unique identification like who created it and where it was built, so you know it hasn't been changed.

[Learn more](#) 



## Monitor

Data stored in the logs is readily auditable, a foundation for future monitors and integrations to build into your security workflow.

[Learn more](#) 

# Sigstore.dev

In collaboration with



**2800+**

COMMITTS

**1200+**

MEMBERS

**20+**

ORGS

Now generally available!

[Find out more](#)

# Sigstore components

## Cosign

For container signing, verification and storage in an Open Container Initiative (OCI) registry, making signatures invisible infrastructure.

[View the repo](#) 

## OpenID Connect

An identity layer that checks if you're who you say you are. It lets clients request and receive information about authenticated sessions and users.

[Learn more](#) 

## Certificate Authority

A mechanism that generates certificates, binding cryptographic keys to an identity and an independent check over an artifact's information.

## Rekor

A built in transparency and timestamping service, Rekor records signed metadata to a ledger that can be searched, but can't be tampered with.

[View the repo](#) 

## Fulcio

A free root certification authority, issuing temporary certificates to an authorized identity and publishing them in the Rekor transparency log.

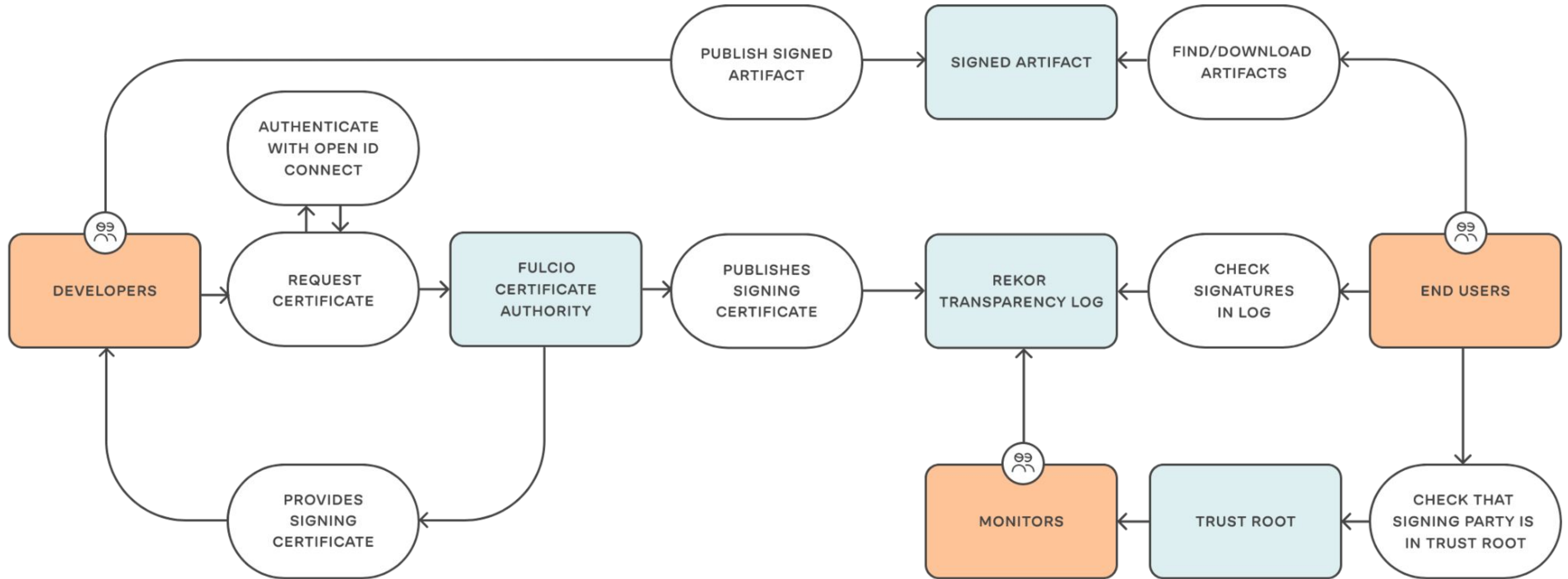
[View the repo](#) 

## Trust root

The foundation for trust behind the whole of sigstore, our keyholders and ways of working to protect the root keys.

[sigstore's trust root](#) 

# Software Supply Chain (Sigstore edition)





# What is SLSA?

It's a security [framework](#), a [check-list](#) of standards and controls to [prevent tampering](#), [improve integrity](#), and [secure packages](#) and infrastructure in your projects, businesses or enterprises. It's how you get from safe enough to being as resilient as possible, at any link in the chain.



## Level 1

Easy to adopt, giving you supply chain visibility and being able to generate provenance



## Level 2

Starts to protect against software tampering and adds minimal build integrity guarantees



## Level 3

Hardens the infrastructure against attacks, more trust integrated into complex systems



## Level 4

The highest assurances of build integrity and measures for dependency management in place

# How does SLSA help?

		Required at			
Requirement		SLSA 1	SLSA 2	SLSA 3	SLSA 4
Source	Version Controlled		✓	✓	✓
	Verified History			✓	✓
	Retained Indefinitely			18 mo.	✓
	Two-Person Reviewed				✓
Build	Scripted	✓	✓	✓	✓
	Build Service		✓	✓	✓
	Ephemeral Environment			✓	✓
	Isolated			✓	✓
	Parameterless				✓
	Hermetic				✓
	Reproducible				○
Provenance	Available	✓	✓	✓	✓
	Authenticated		✓	✓	✓
	Service Generated		✓	✓	✓
	Non-Falsifiable			✓	✓
	Dependencies Complete				✓
Common	Security				✓
	Access				✓
	Superusers				✓

○ = required unless there is a justification

- Define what does **good** look like
- Provide a **framework** for **assessing** existing **software development lifecycle**
- Provide a **framework** for **continuous improvement**
- **Shift** security **to the left**
- Enforce **provenance** of the build
- Enable **end-to-end supply chain trust**
- Enable software development **observability**

# You fav OSS projects are dancing SLSA





# SBOM: Software Bill of Materials

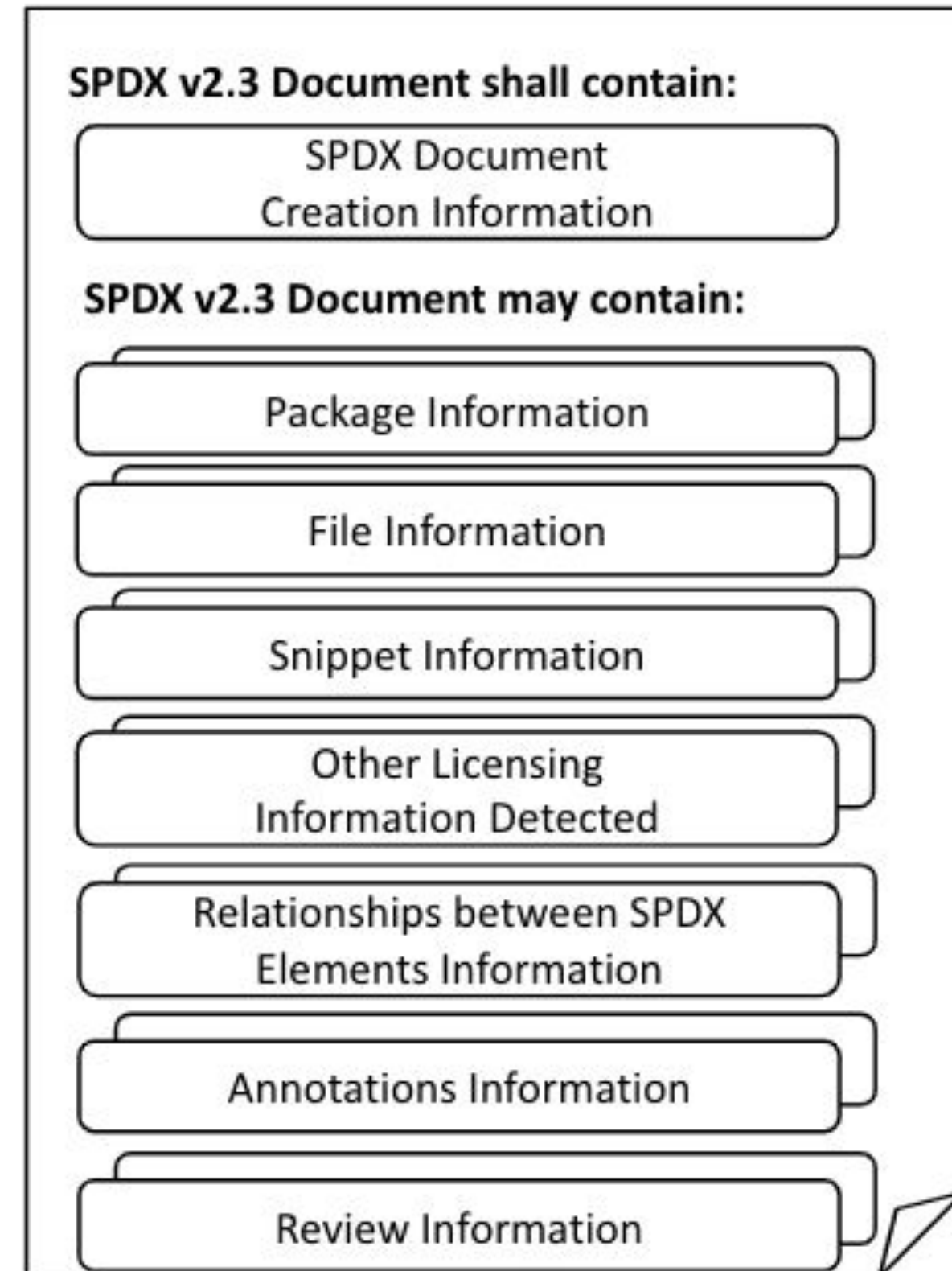
A software bill of materials often captures:

- Supplier name
- Component name and version
- Other unique identifiers
- Dependency relationships
- Author of the SBOM data
- Timestamp

SBOMs come in two formats:

- CycloneDX
- **SPDX**

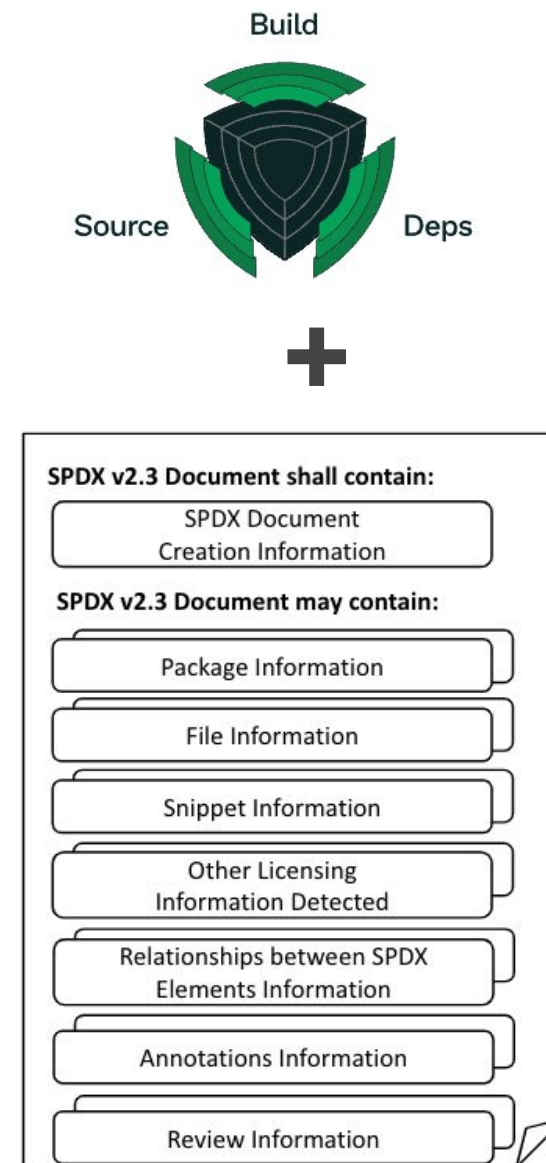
[cisa.gov/sbom](https://cisa.gov/sbom)





# SLSA and Software Bill of Materials (SBOM)

- SLSA and SBOM are **complementary**
- SLSA can make it easier to generate SBOMs
- Major **SLSA** principle - generate **tamper-proof provenance data**
  - Who performed the release process for an artifact
  - Materials used in production
  - Whether the artifact was protected from tampering
- SBOMs hinge on accuracy, completeness, and trust
  - Having **SLSA provenance** for an artifact **improves** the **quality** and **integrity** of its **SBOM**.



Demo





# Software Delivery Shield - Goals



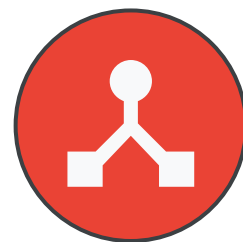
Holistic

Provide holistic product solution encompassing software development lifecycle, dependencies and runtimes



Best practices

Time tested Google best practices inside



Modular

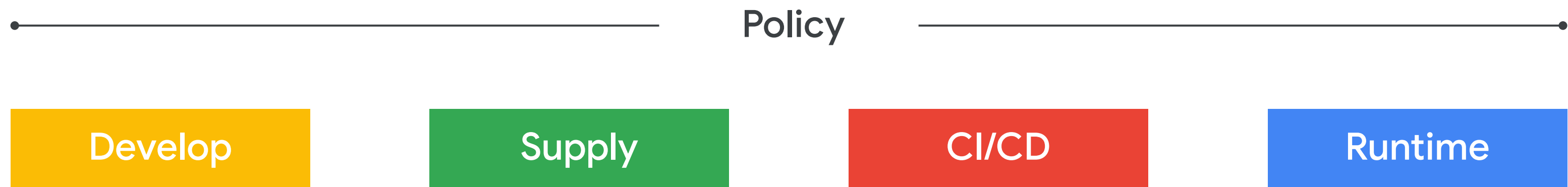
Incremental adoption pathway





# Software Delivery Shield

Fully managed, end-to-end software supply chain security solution

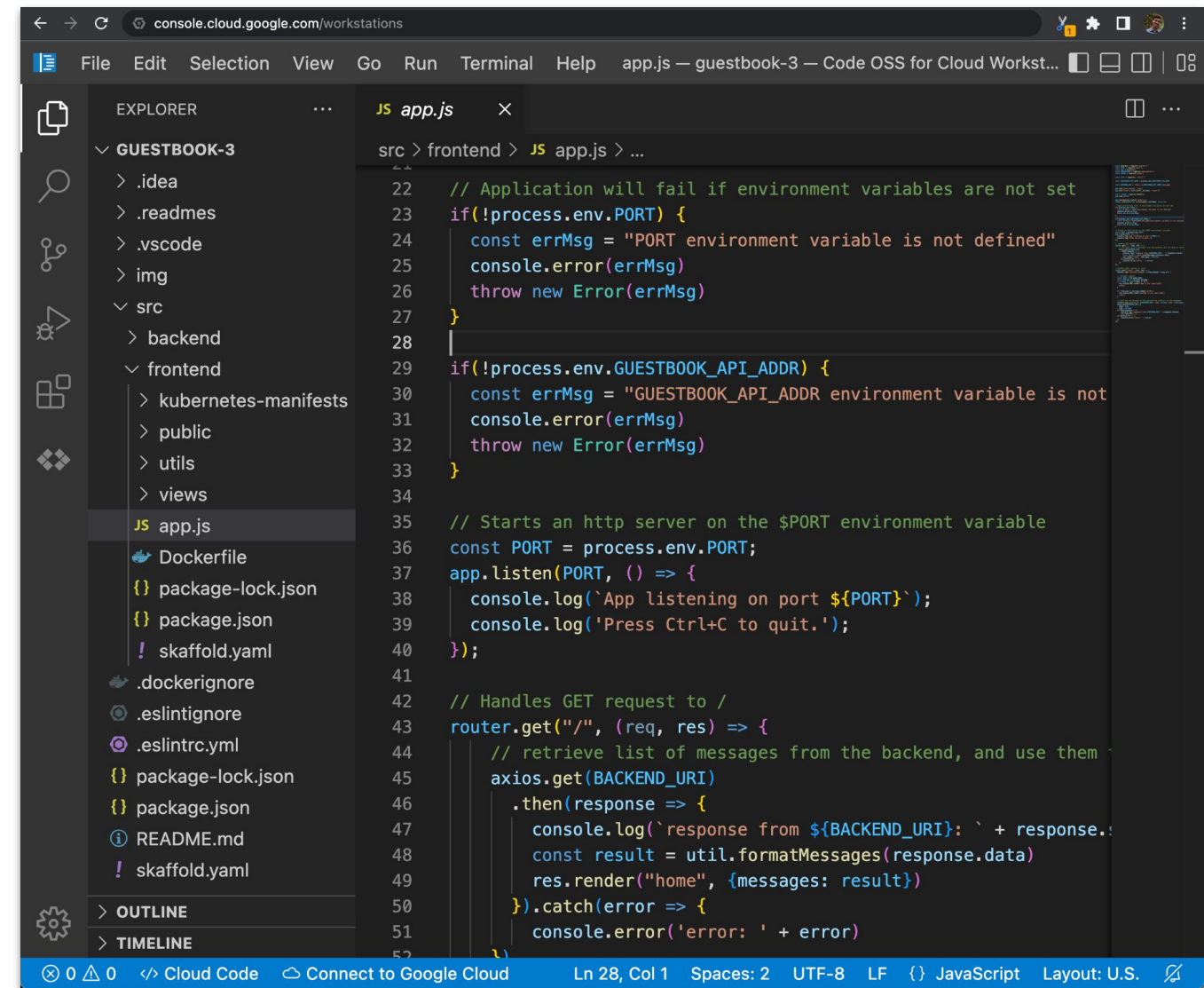


# Fully managed development environments



## Cloud Workstations

- On-demand environments accessible anywhere
- Security policies
- Managed base images
- VPC and VPC-Service controls



```

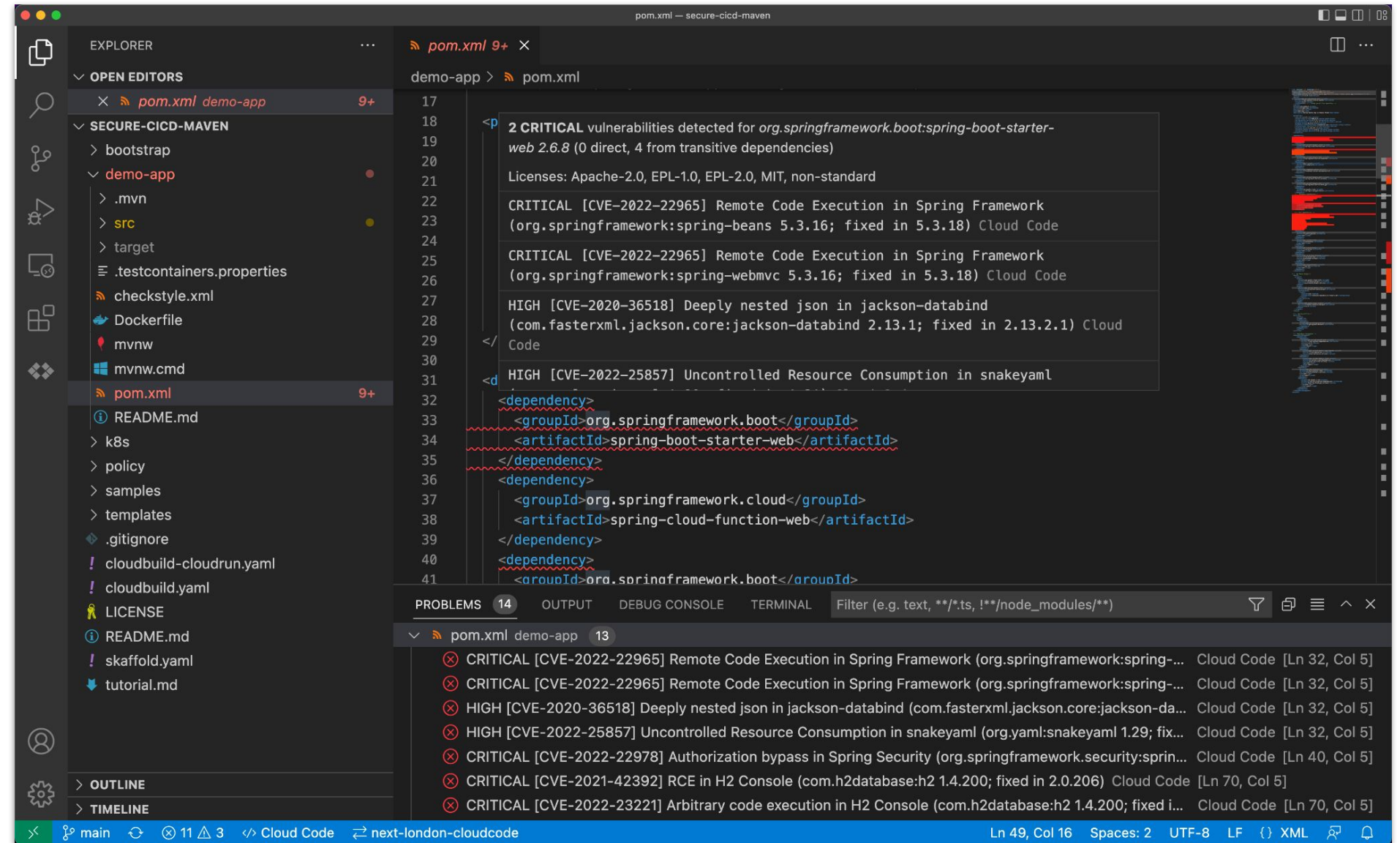
22 // Application will fail if environment variables are not set
23 if(!process.env.PORT) {
24   const errMsg = "PORT environment variable is not defined"
25   console.error(errMsg)
26   throw new Error(errMsg)
27 }
28
29 if(!process.env.GUESTBOOK_API_ADDR) {
30   const errMsg = "GUESTBOOK_API_ADDR environment variable is not
31   console.error(errMsg)
32   throw new Error(errMsg)
33 }
34
35 // Starts an http server on the $PORT environment variable
36 const PORT = process.env.PORT;
37 app.listen(PORT, () => {
38   console.log(`App listening on port ${PORT}`);
39   console.log('Press Ctrl+C to quit.');
```

Preview

# Security assistance in the IDE

## Cloud Code source protect

- Vulnerability detection as you code
- Support for scanning transitive dependencies
- Dependency license reporting



Preview

# Improving security of artifacts and dependencies

## Artifact Registry & Container Analysis

## Assured Open Source Software

- Artifact Registry - Maven virtual and remote repos
- Container Analysis - On-push Maven and Go container scanning and standalone Maven package scanning
- Container Analysis - On-push SBOM dependency list generation for containers
- Assured Open Source Software - 250+ Java and Python packages

**Scan results**

**PREVIEW** Maven and Go scanning are now included. [LEARN MORE](#)

Based on factors such as exploitability, scope, impact, and maturity of the vulnerability.

Scans	Total	Fixes	Critical	High	Medium
3	29	12	4	5	7

**Filter** Filter vulnerabilities

Name	Effective severity	CVSS V2	Fix available	Package	Package type	
CVE-2022-22978	Critical	7.5	Yes	org.springframework.security:spring-security-core	Maven	<a href="#">VIEW FIX</a>
<a href="#">CVE-2022-23221</a>	Critical	10	Yes	com.h2database:h2	Maven	<a href="#">VIEW FIX</a>
<a href="#">CVE-2021-42392</a>	Critical	10	Yes	com.h2database:h2	Maven	<a href="#">VIEW FIX</a>
CVE-2022-22965	Critical	7.5	Yes	org.springframework:spring-beans	Maven	<a href="#">VIEW FIX</a>
CVE-2022-22970	High	3.5	Yes	org.springframework:spring-core	Maven	<a href="#">VIEW FIX</a>
CVE-2022-31197	High	0	Yes	org.postgresql:postgresql	Maven	<a href="#">VIEW FIX</a>
CVE-2021-23463	High	6.4	Yes	com.h2database:h2	Maven	<a href="#">VIEW FIX</a>
CVE-2022-22968	High	5	Yes	org.springframework:spring-core	Maven	<a href="#">VIEW FIX</a>
CVE-2020-36518	High	5	Yes	com.fasterxml.jackson.core:jackson-databind	Maven	<a href="#">VIEW FIX</a>
<a href="#">CVE-2020-16156</a>	Medium	6.8	-	perl	OS	<a href="#">VIEW</a>
CVE-2022-22971	Medium	4	Yes	org.springframework:spring-core	Maven	<a href="#">VIEW FIX</a>
<a href="#">CVE-2022-2509</a>	Medium	0	Yes	gnutls28	OS	<a href="#">VIEW FIX</a>
<a href="#">CVE-2021-31879</a>	Medium	5.8	-	wget	OS	<a href="#">VIEW</a>

Preview





## Software Delivery Shield

# Enhance the security of your CI pipelines



- SLSA Level 3 build support ([slsa.dev](https://slsa.dev))
- Build provenance for non-container Java (Maven) and Python packages
- Security insights panel

Security insights for demo-app

Software Delivery Shield is a new service to safeguard artifact integrity across your entire software delivery lifecycle. [Learn more](#) about how it can prevent tampering, improve integrity, and secure packages and infrastructure.

Achieved **SLSA Build Level 3** [What's this?](#)

### Supply Chain

Supply chain information appears for artifacts that you store in Artifact Registry and Container Registry. If parts of your supply chain are outside of Google Cloud, some information might be unavailable.

### Vulnerabilities

Critical	High	Medium	Low
0	0	0	0

Artifacts scanned [demo-app](#)

### Build

Details

Logs	<a href="#">4b25f15e</a>
Builder	Cloud Build
Completed	4 days ago

Provenance [View](#)

```
["_type": "https://in-toto.io/Statement/v0.1",
```

Preview

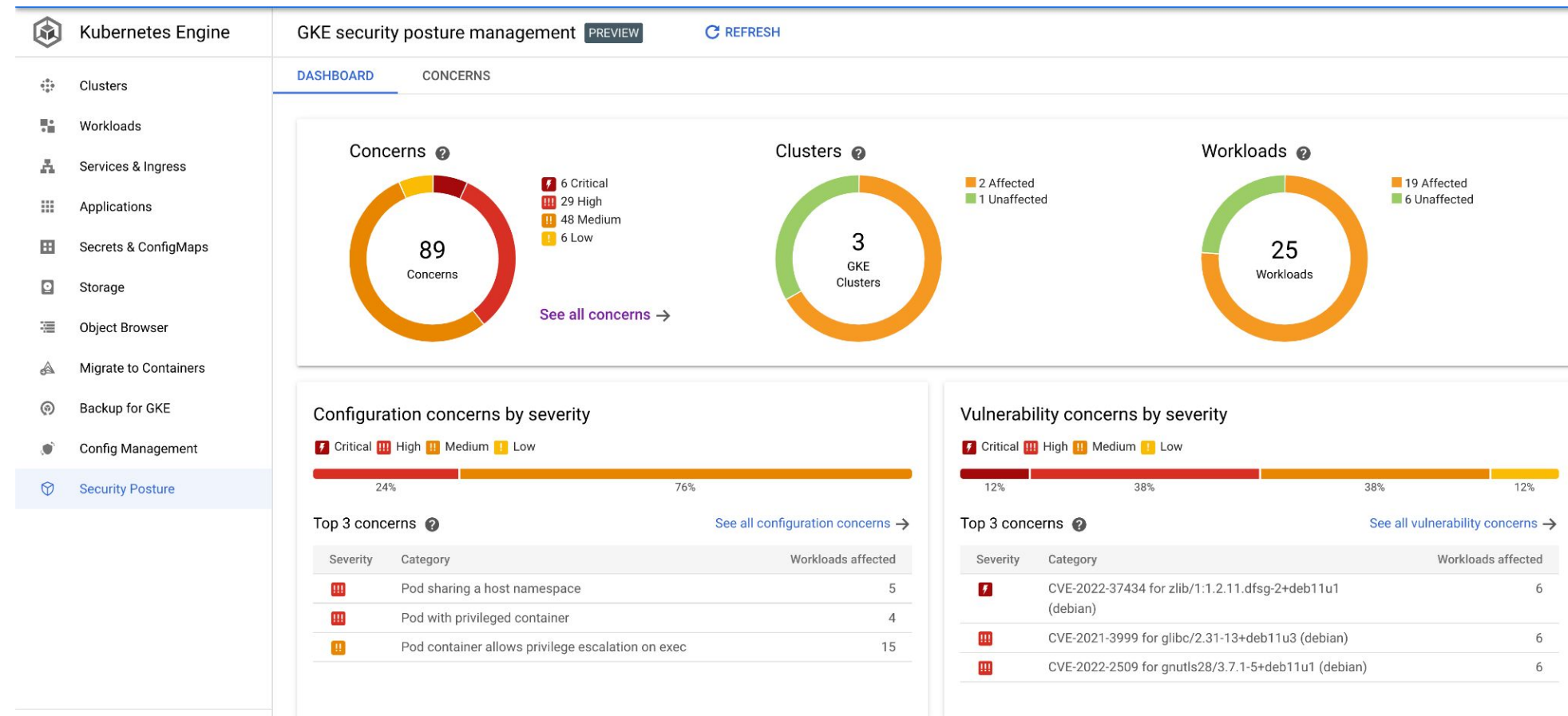
## Software Delivery Shield

# Security insights at the runtime

 GKE security posture

 Cloud Run security insights

- GKE continuous runtime vulnerability and workload configuration scanning
- Cloud Run insights into security target levels, service vulnerabilities, and build provenance



Preview

# Trust based policy




## Binary Authorization

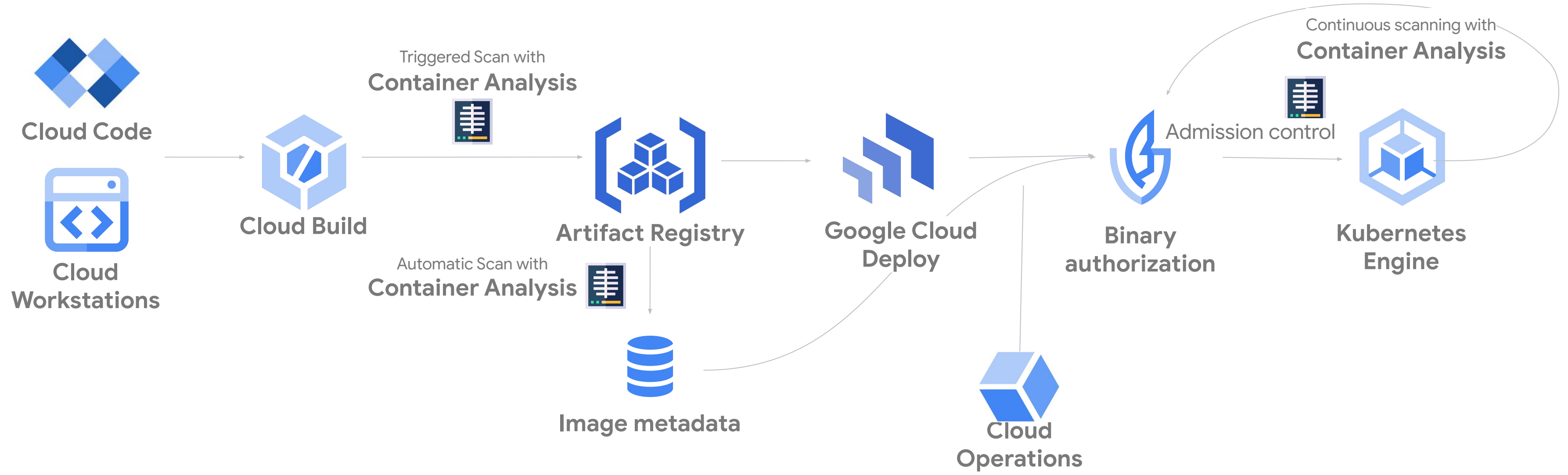
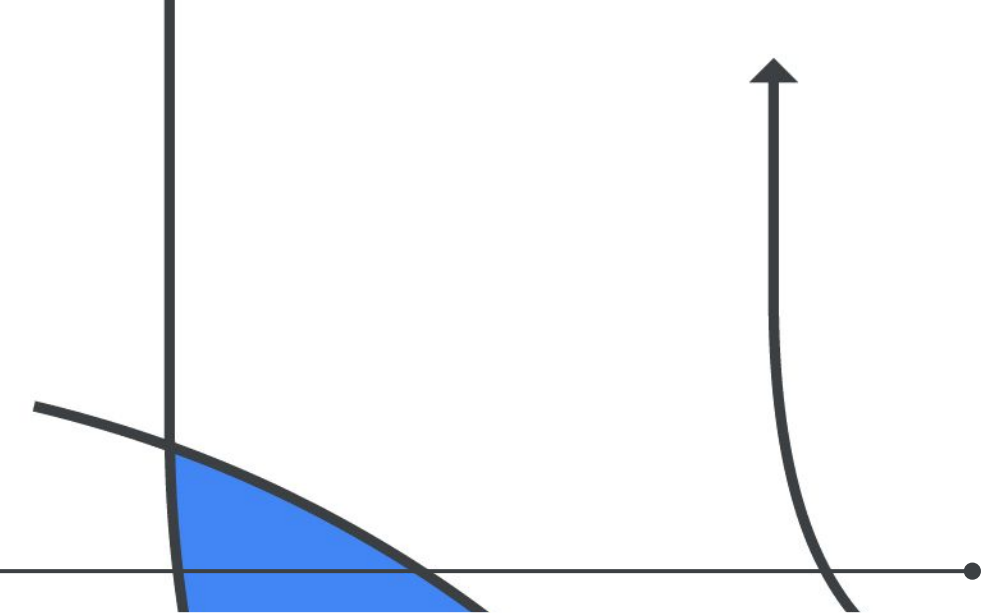
- Trust-based software development lifecycle policy
- Runtime policy enforcement

### Policy deployment rules for "vsz-demo"

[EDIT POLICY](#)

Project default rule	Allow only images that have been approved by all of the following attestors: <ul style="list-style-type: none"><li>• projects/vsz-demo/attestors/built-by-cloud-build</li><li>• projects/vsz-demo/attestors/build-vuln-check</li></ul>
Specific rules	-
Dry-run mode 	Not enabled

# Demo Overview





# Thank you

**Abdel Sghiouar**

Senior Cloud Developer Advocate @Google

Kubernetes Podcast co-host

**Twitter: @boredabdel**

