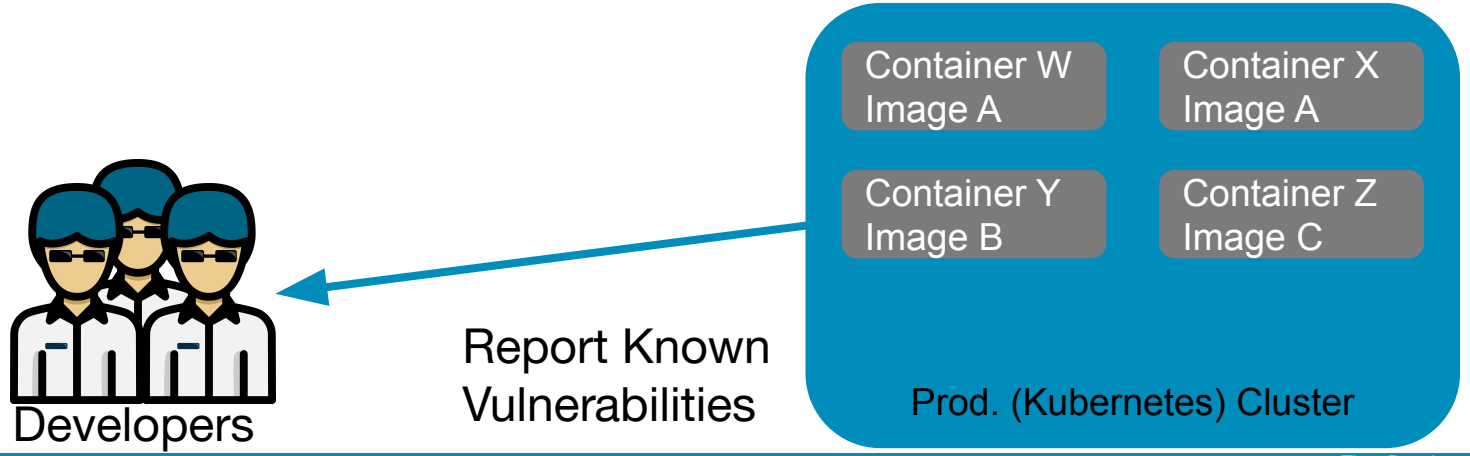
Timo Pagel

ClusterImageScanner

Kieler Linux Tage 2021

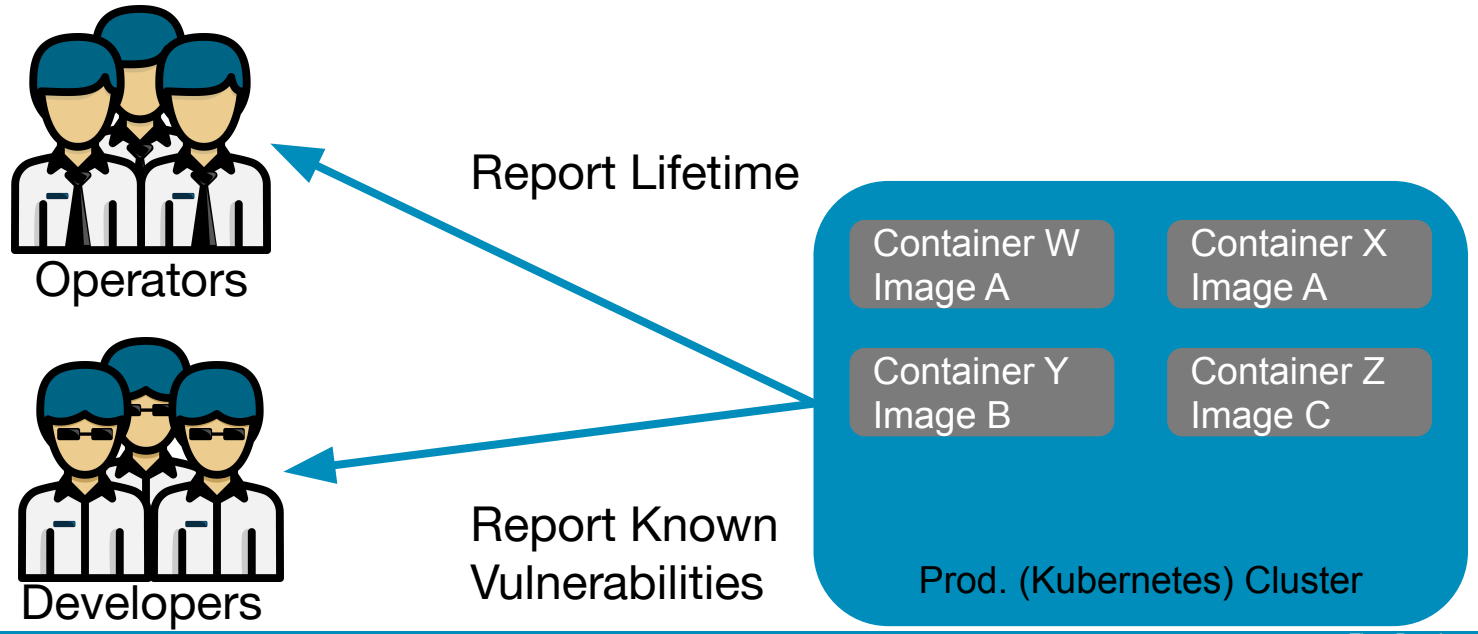
Solution Overview

Handling of security misconfigurations and known vulnerabilities

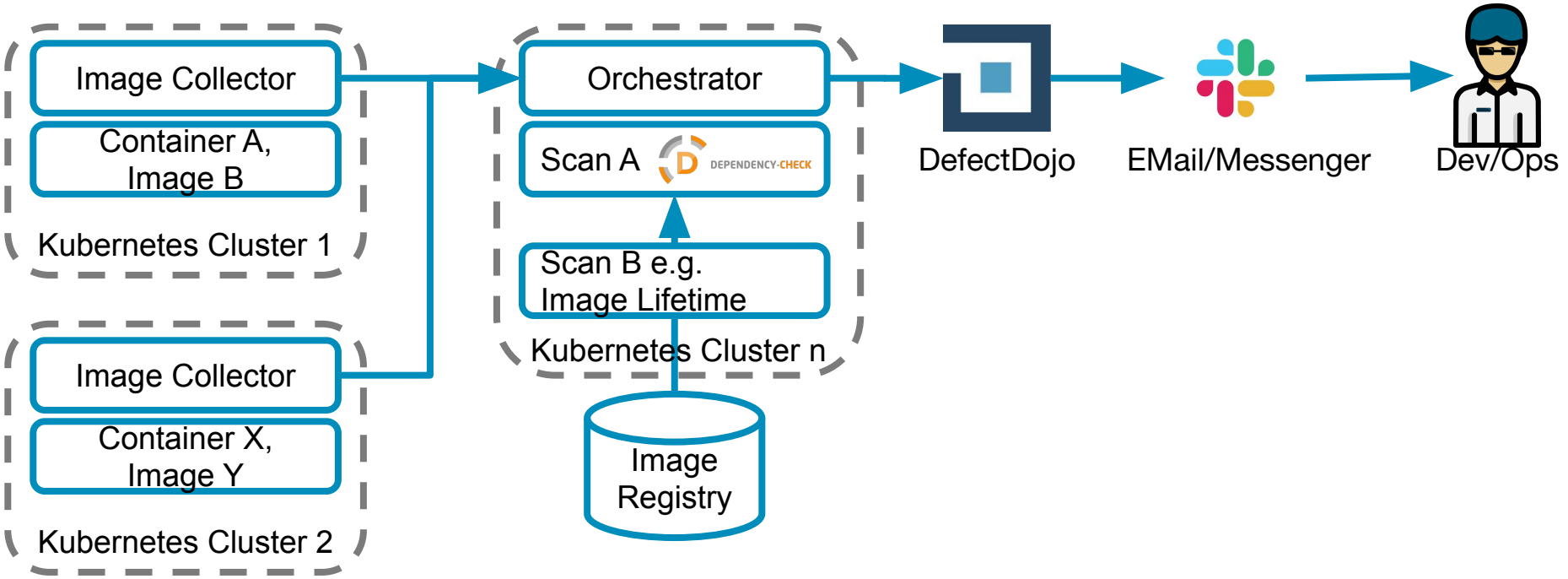


Solution Overview

Handling of security misconfigurations and known vulnerabilities



SDA SE ClusterScanner Overview

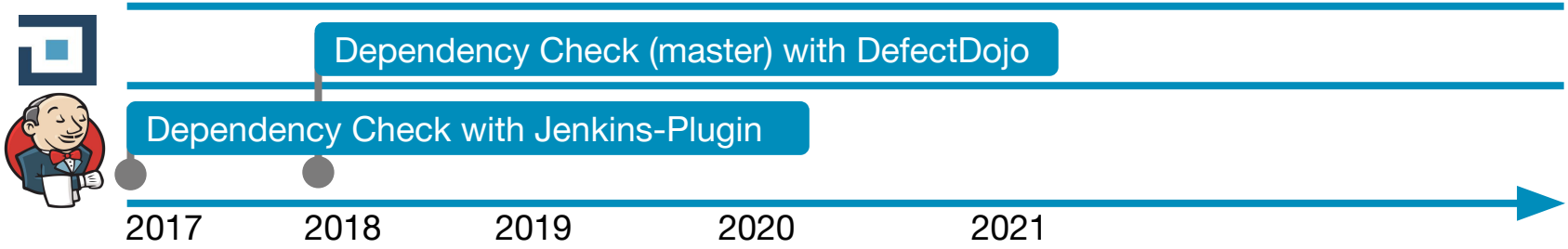


DevSecOps Test Journey

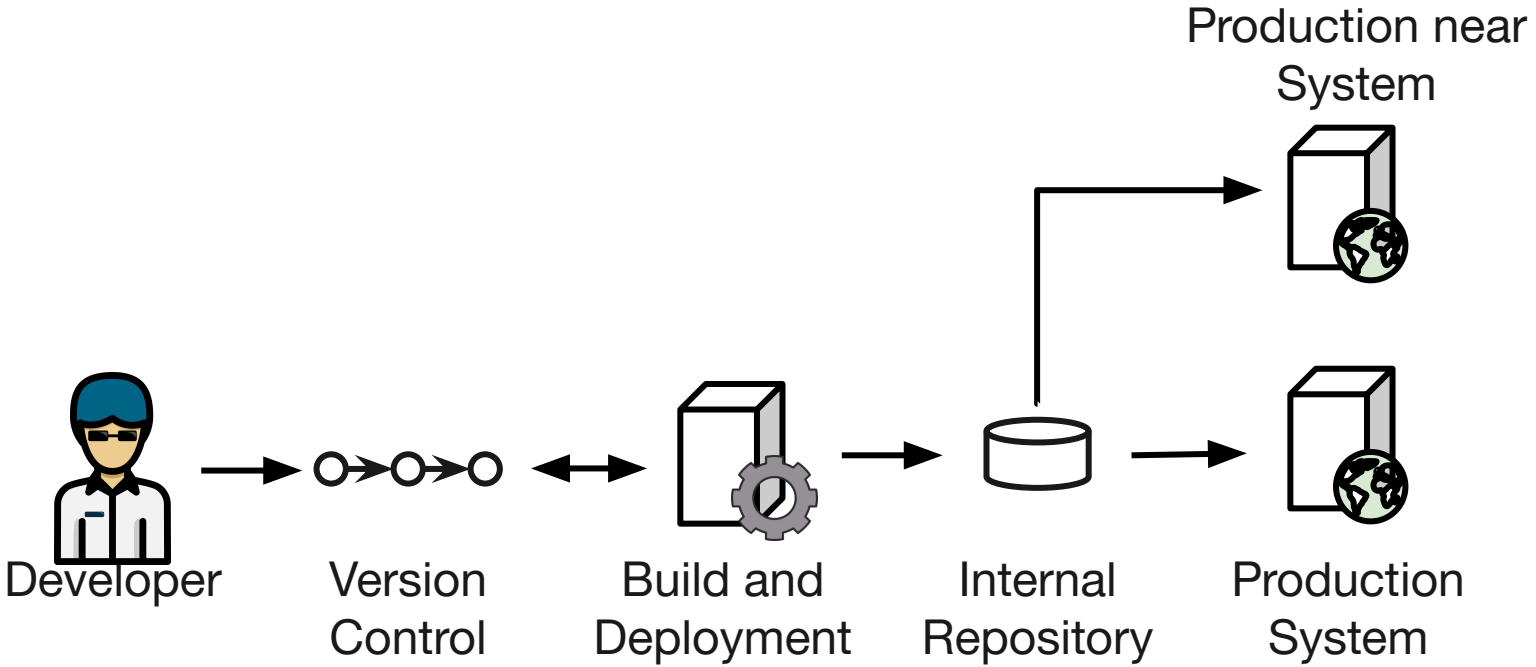
Dependency Check with Jenkins-Plugin

2017 2018 2019 2020 2021

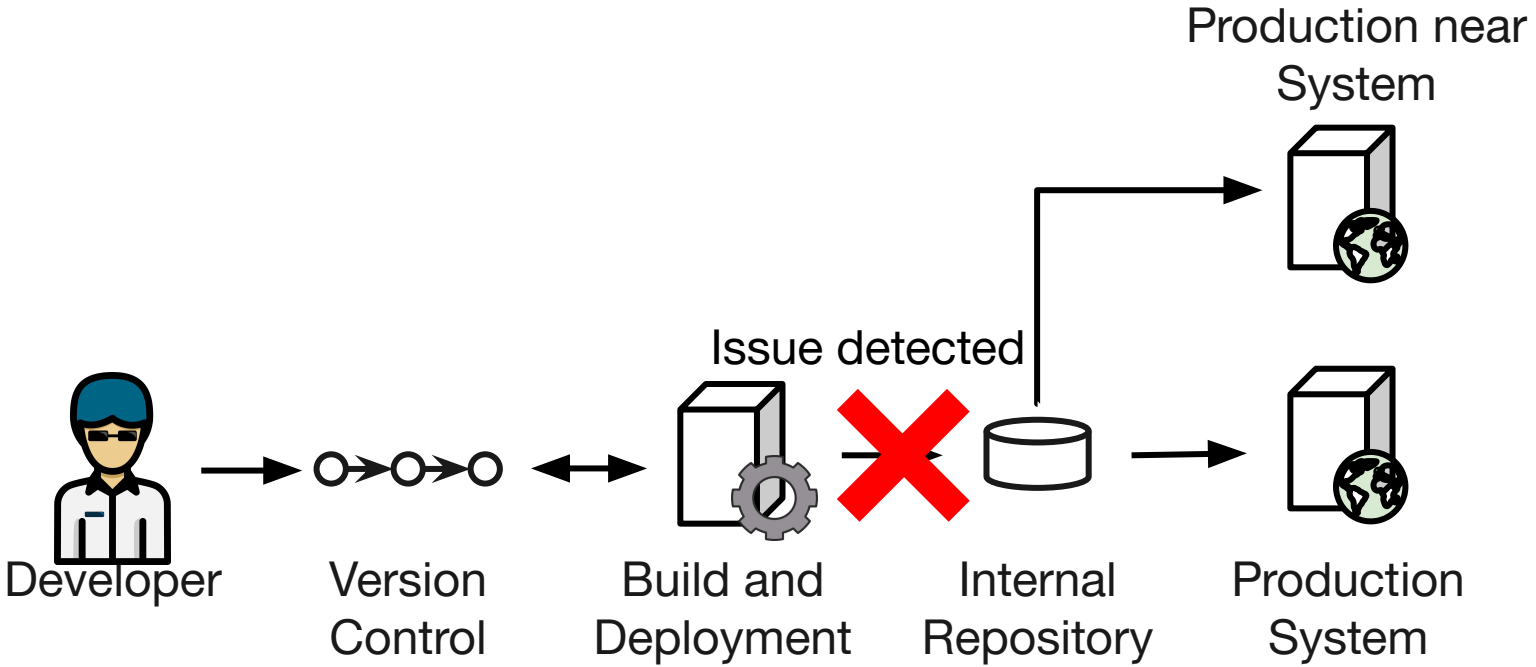
DevSecOps Test Journey



Build and Deployment Process



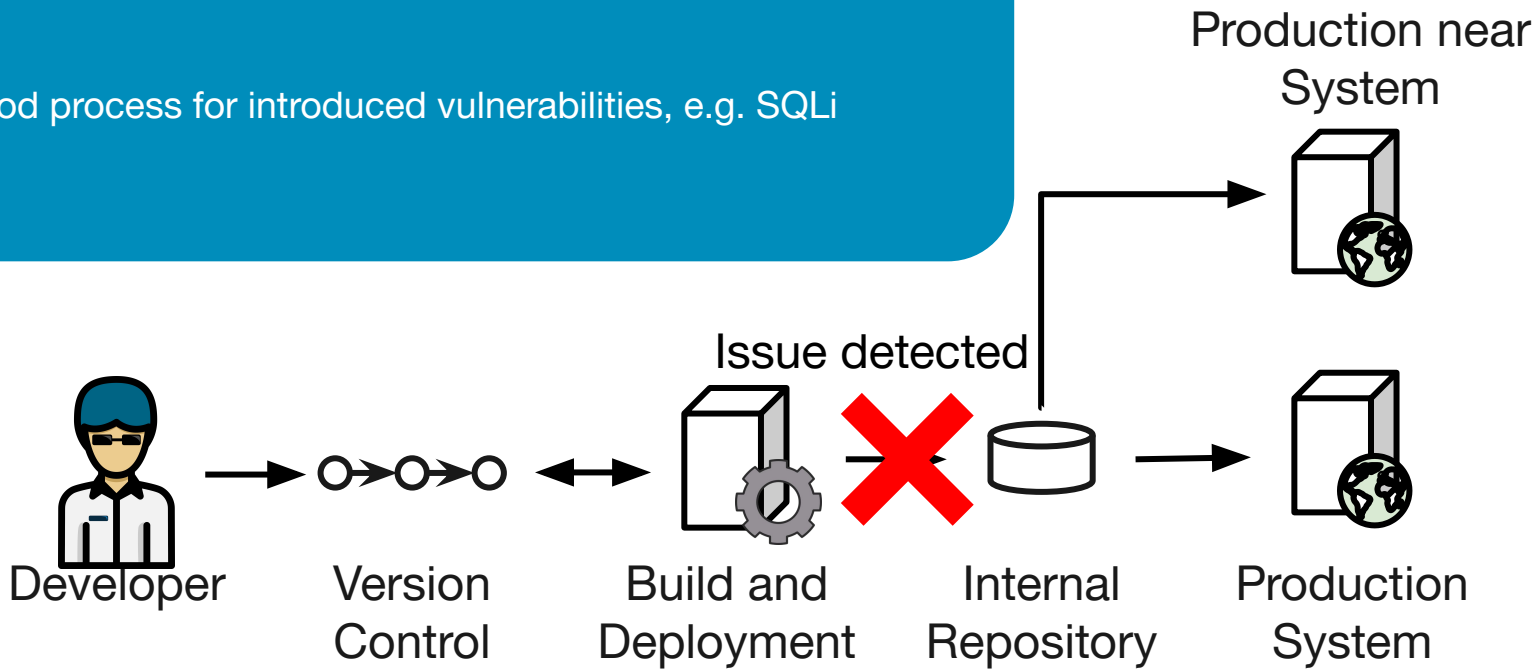
Build and Deployment Process



Build and Deployment Process

Developer is working on something else
Maybe the developer patched a vulnerability but another is raised

Is a good process for introduced vulnerabilities, e.g. SQLi

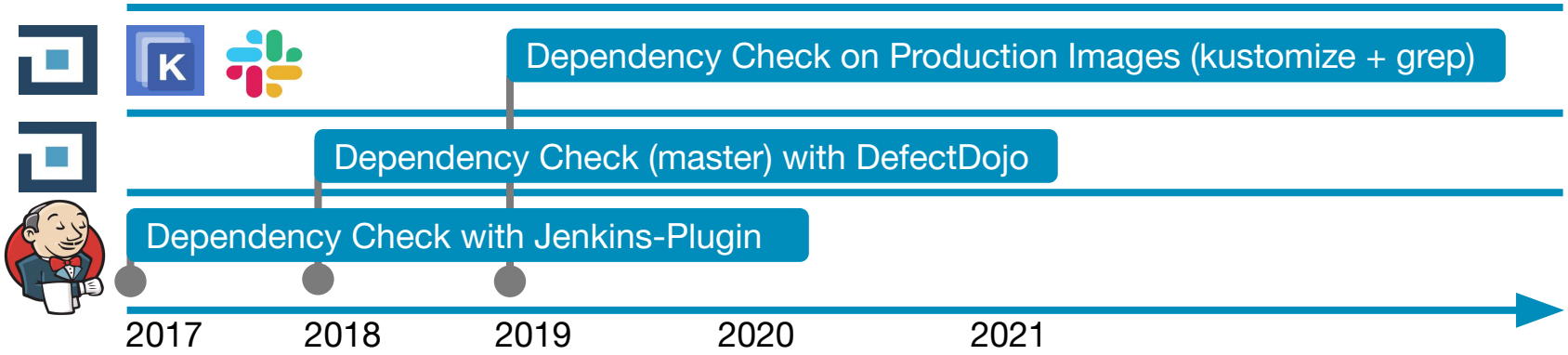


Inform about known vulnerabilities

Developer doesn’t change something

-> Asynchronous information about known vulnerabilities in third party libraries

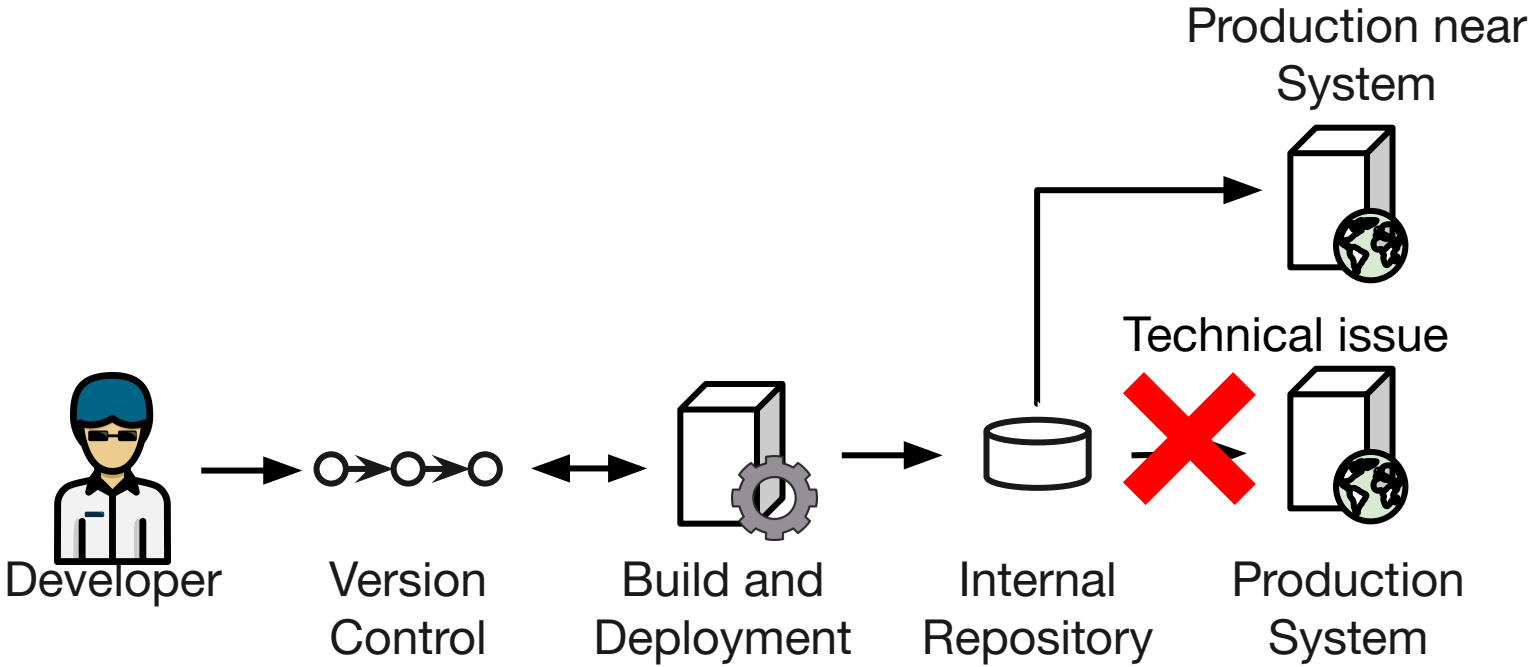
DevSecOps Test Journey



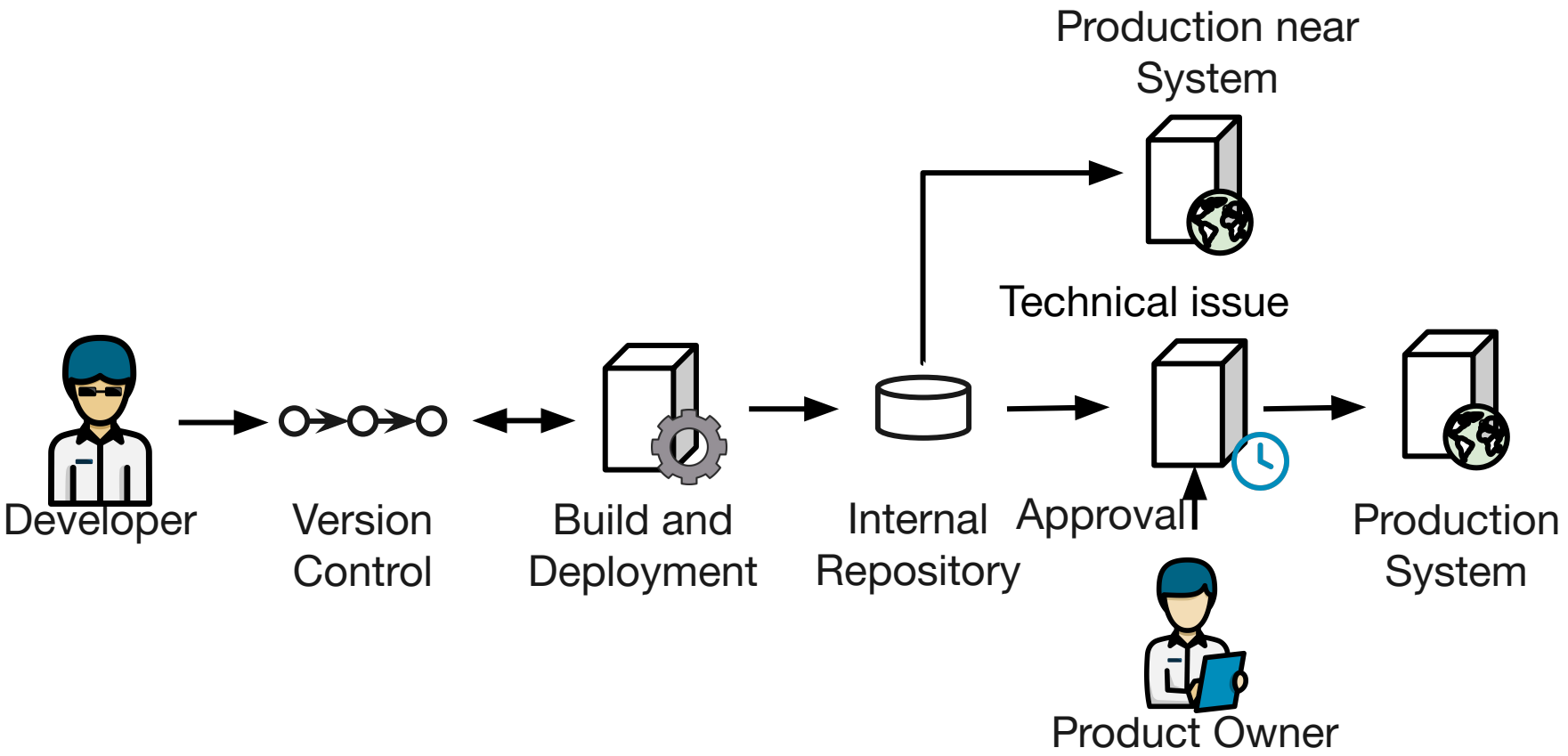
Master != Production

- Process (e.g. approval)
- Technical issues

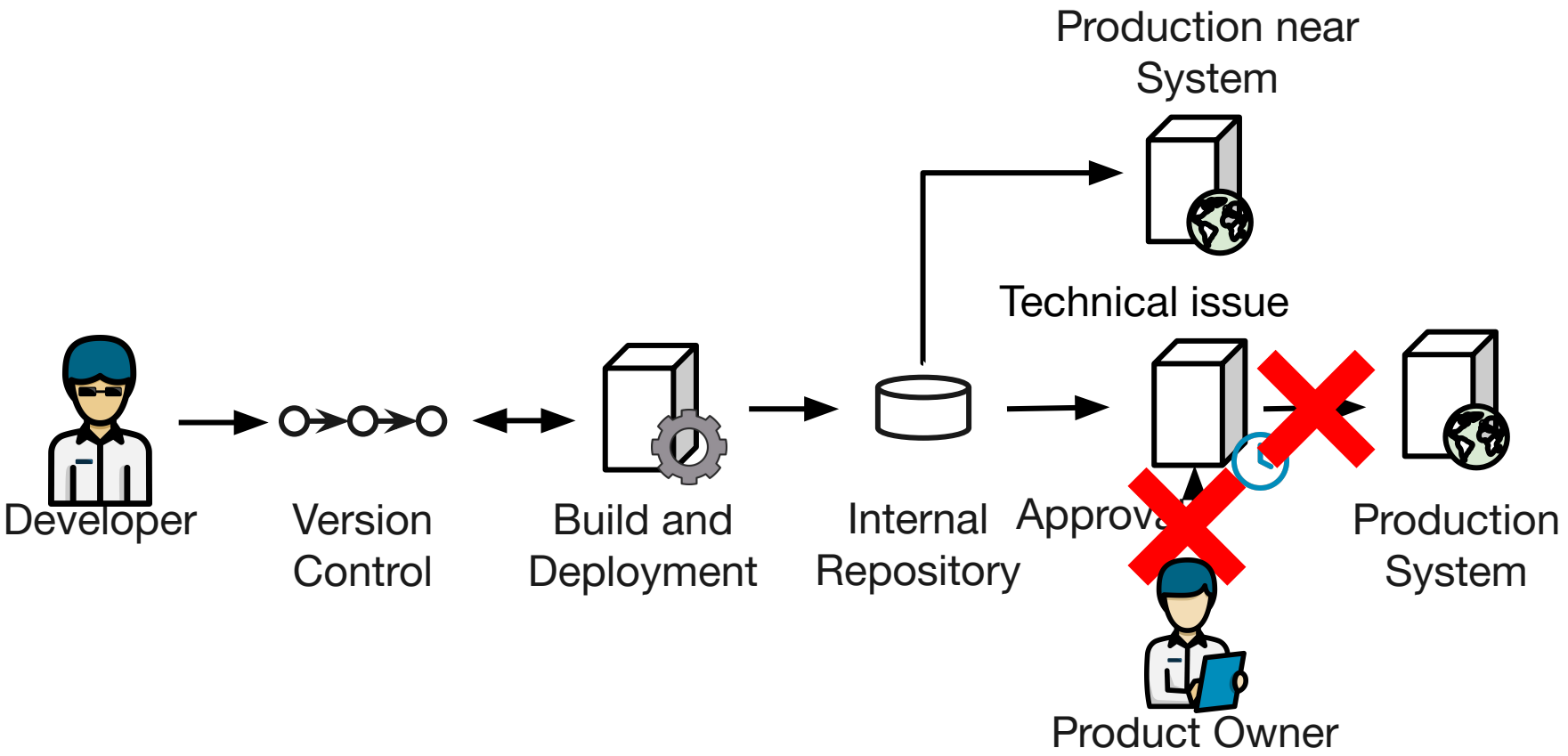
Build and Deployment Process



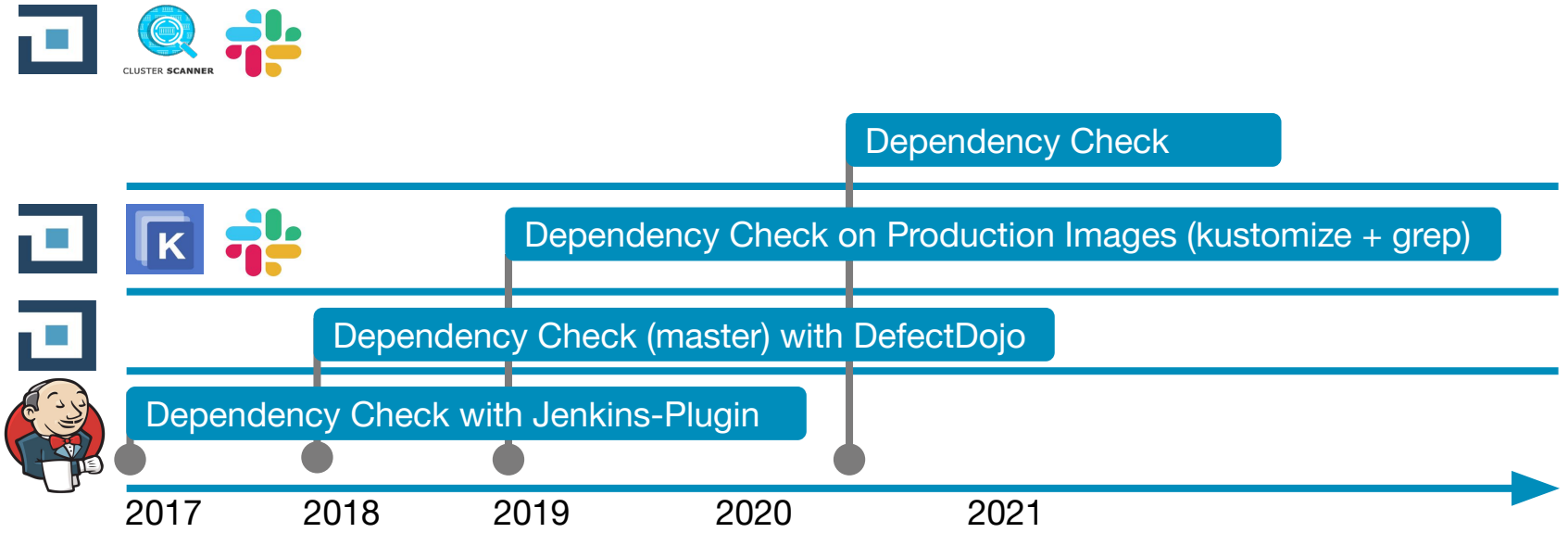
Build and Deployment Process



Build and Deployment Process



DevSecOps Test Journey



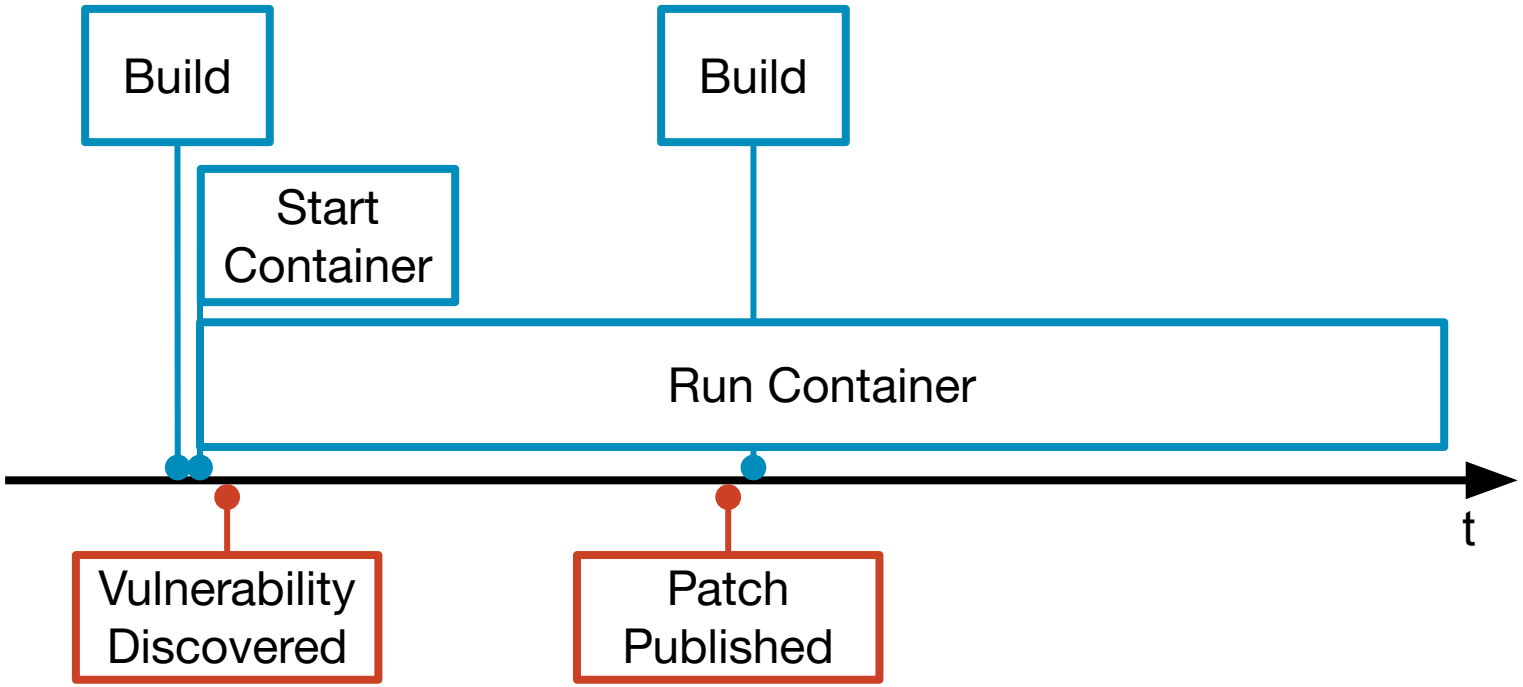
Cluster Components/Layers

Application

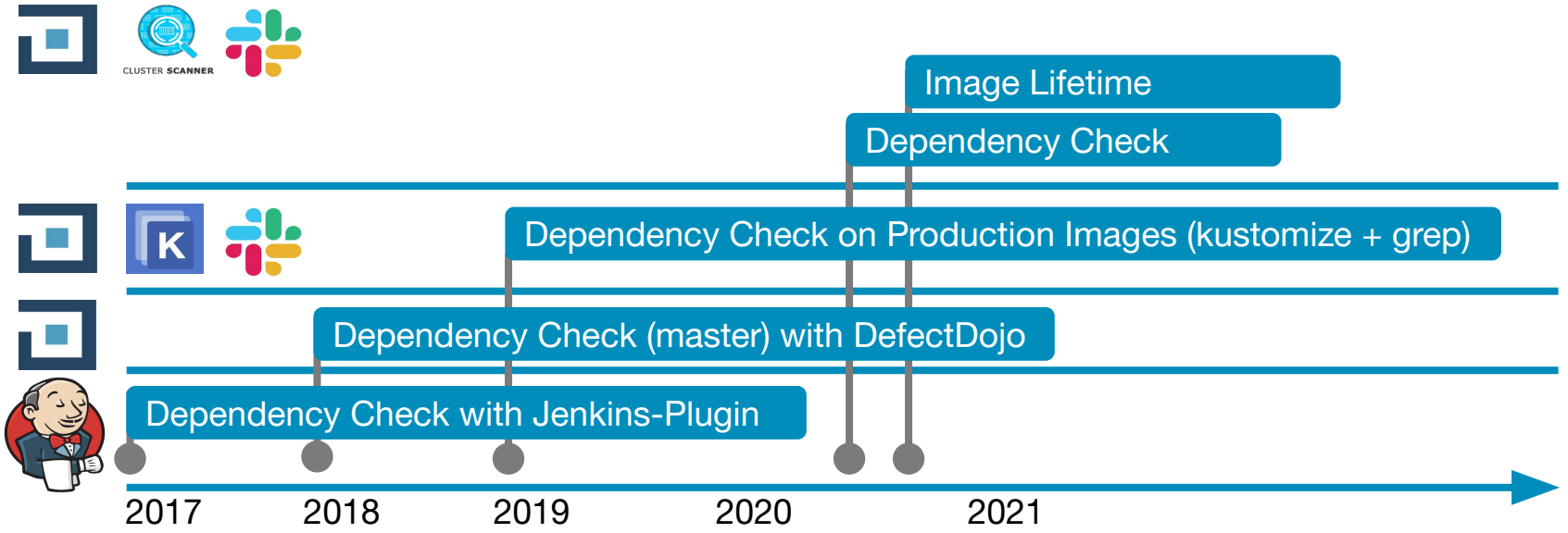
Container Operating System

(Host Operating System)

Patching, a solved issue raises



DevSecOps Test Journey



Images

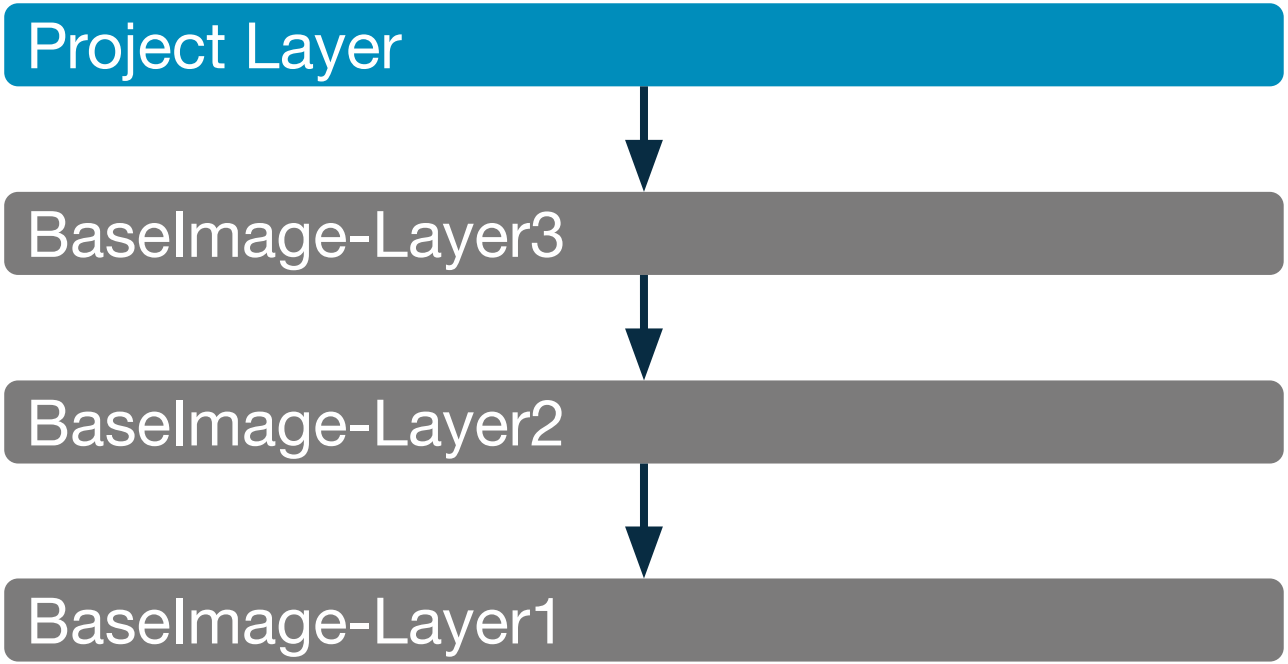
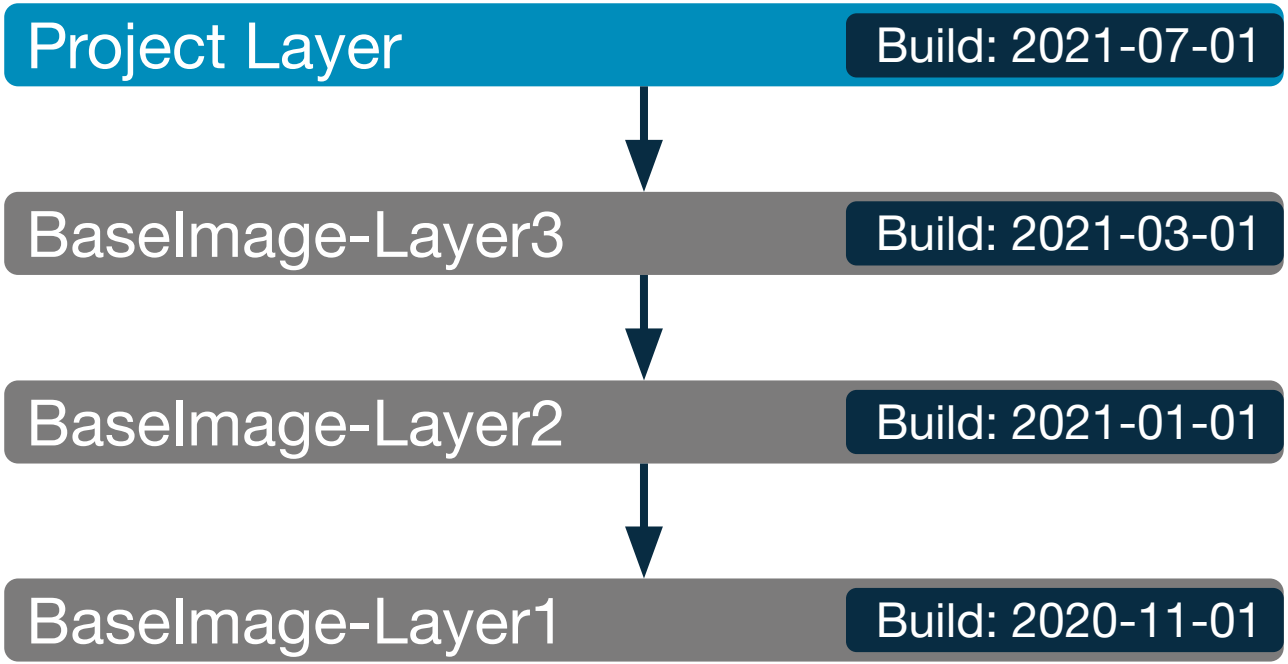
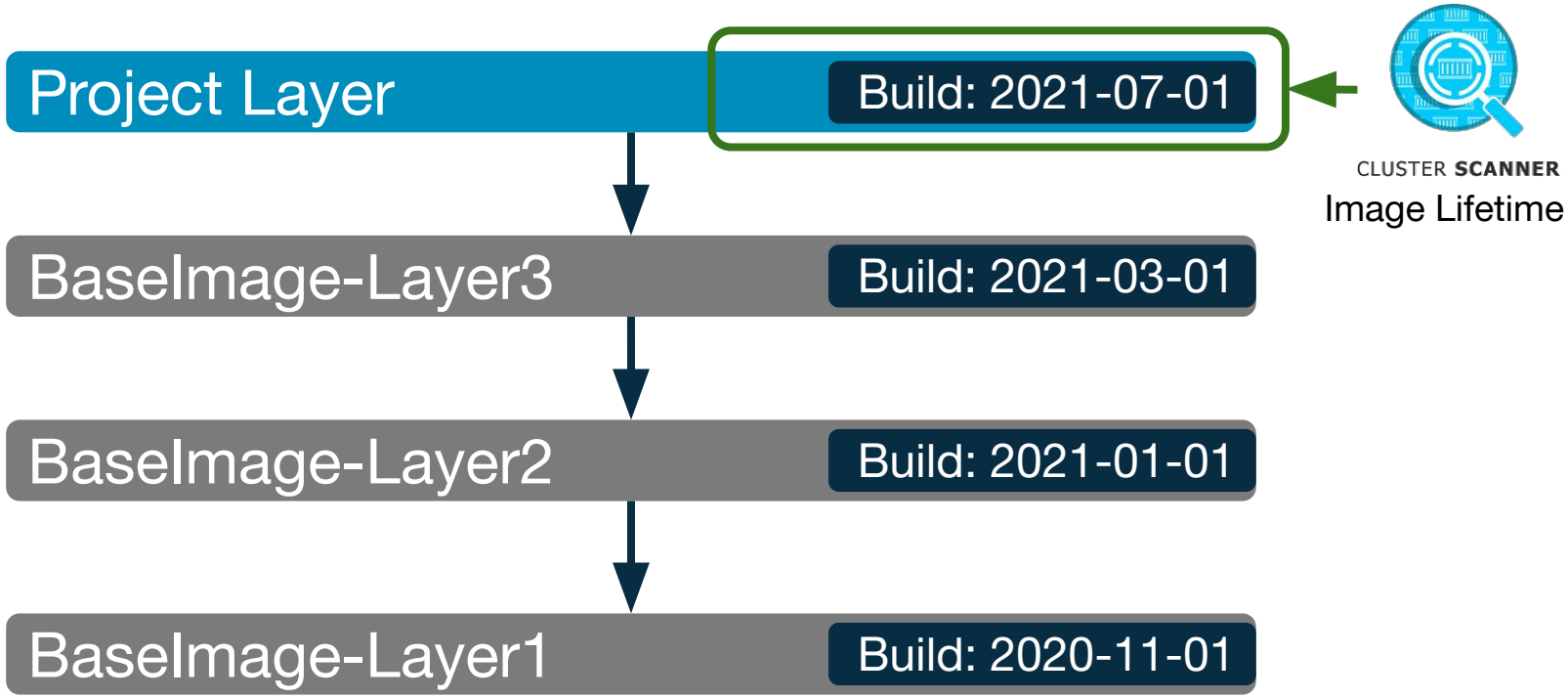

Project Layer
BaseImage-Layer3
BaseImage-Layer2
BaseImage-Layer1

Image Build Date

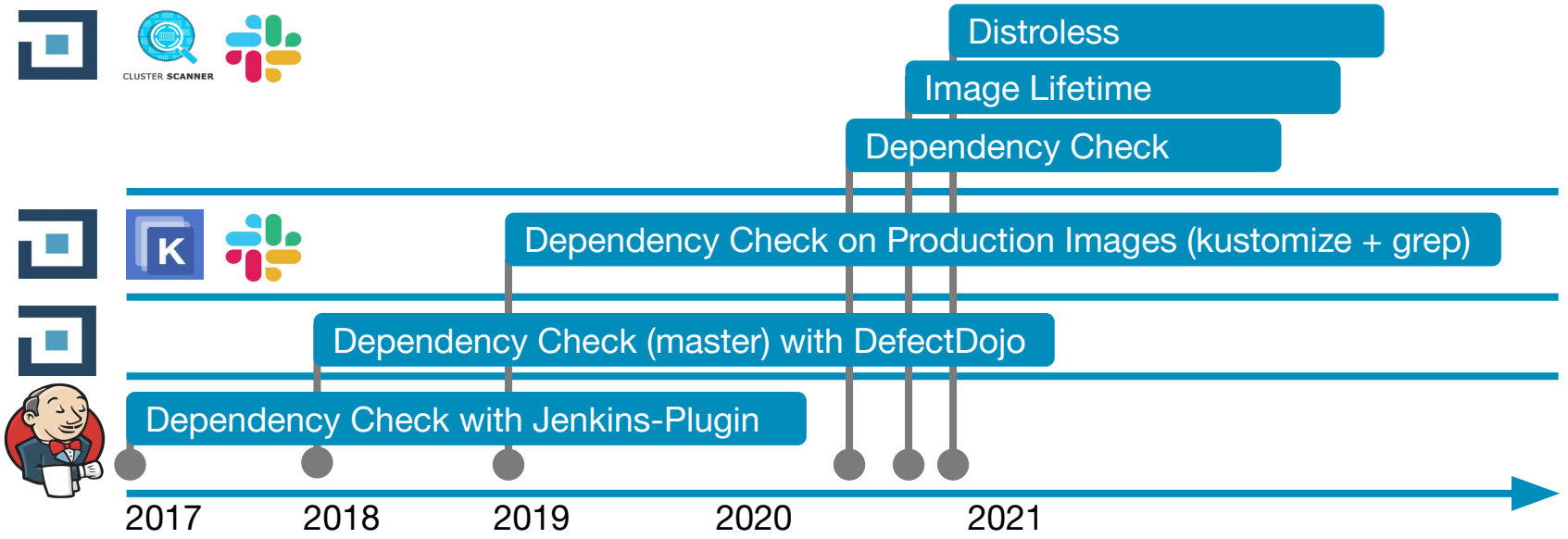


ImageLifetime Scan

SDA SE
OPEN INDUSTRY SOLUTIONS

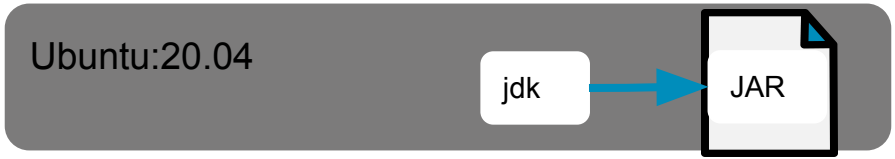


DevSecOps Test Journey



Distroless (Package-Manager Based)

Build App (temporary)



Distroless (Package-Manager Based)

Build App (temporary)

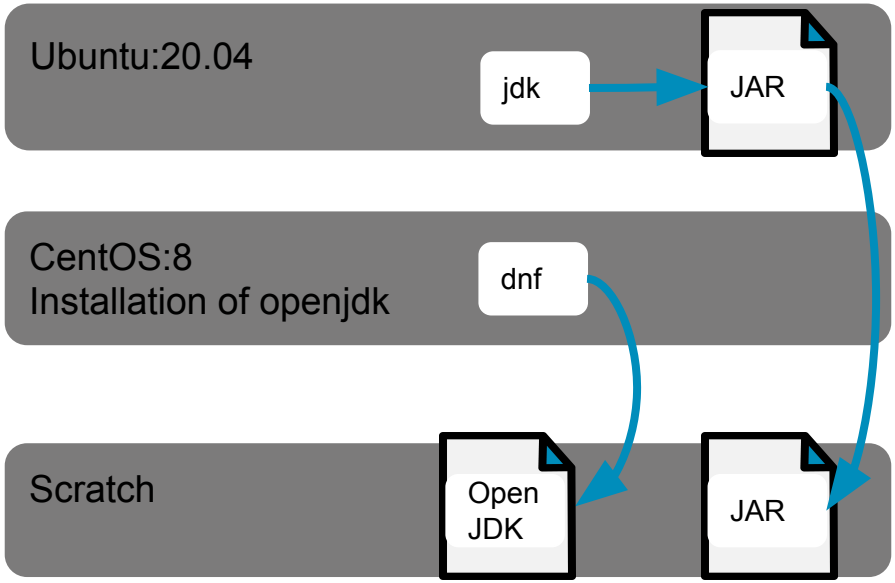
Build Container OS

Distroless (Package-Manager Based)

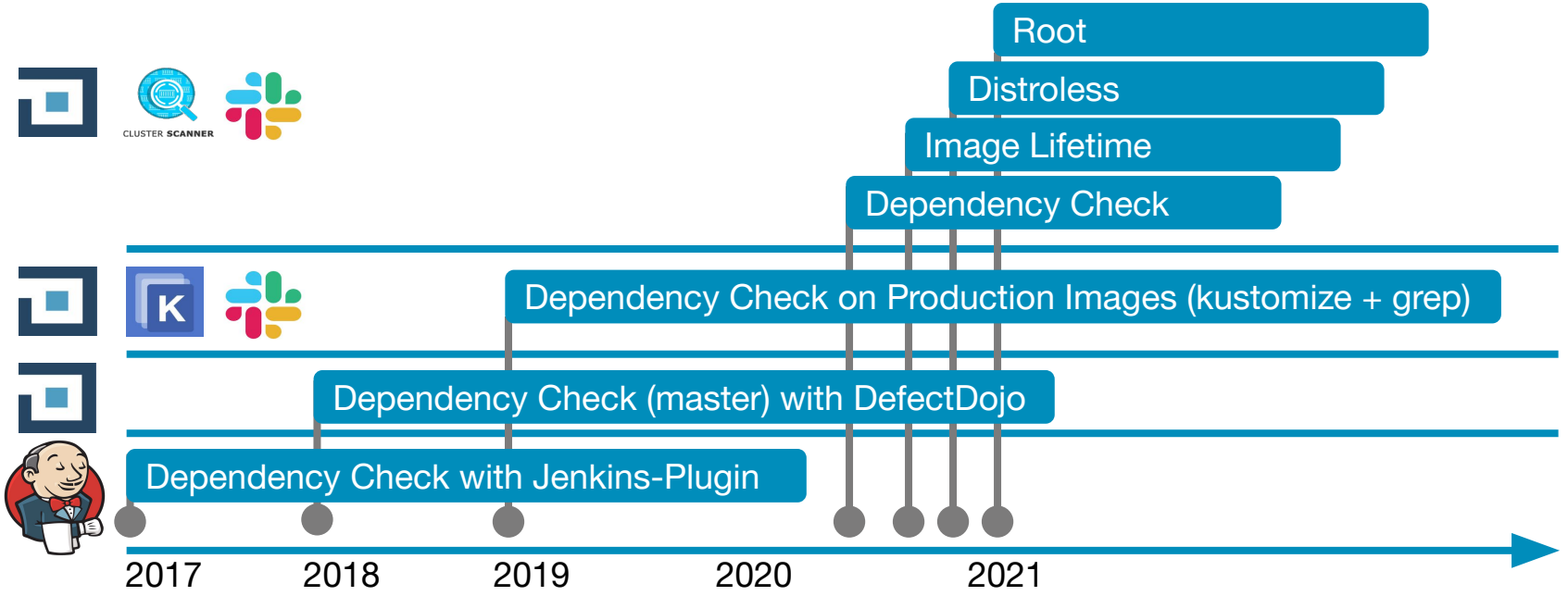
Build App (temporary)

Build Container OS

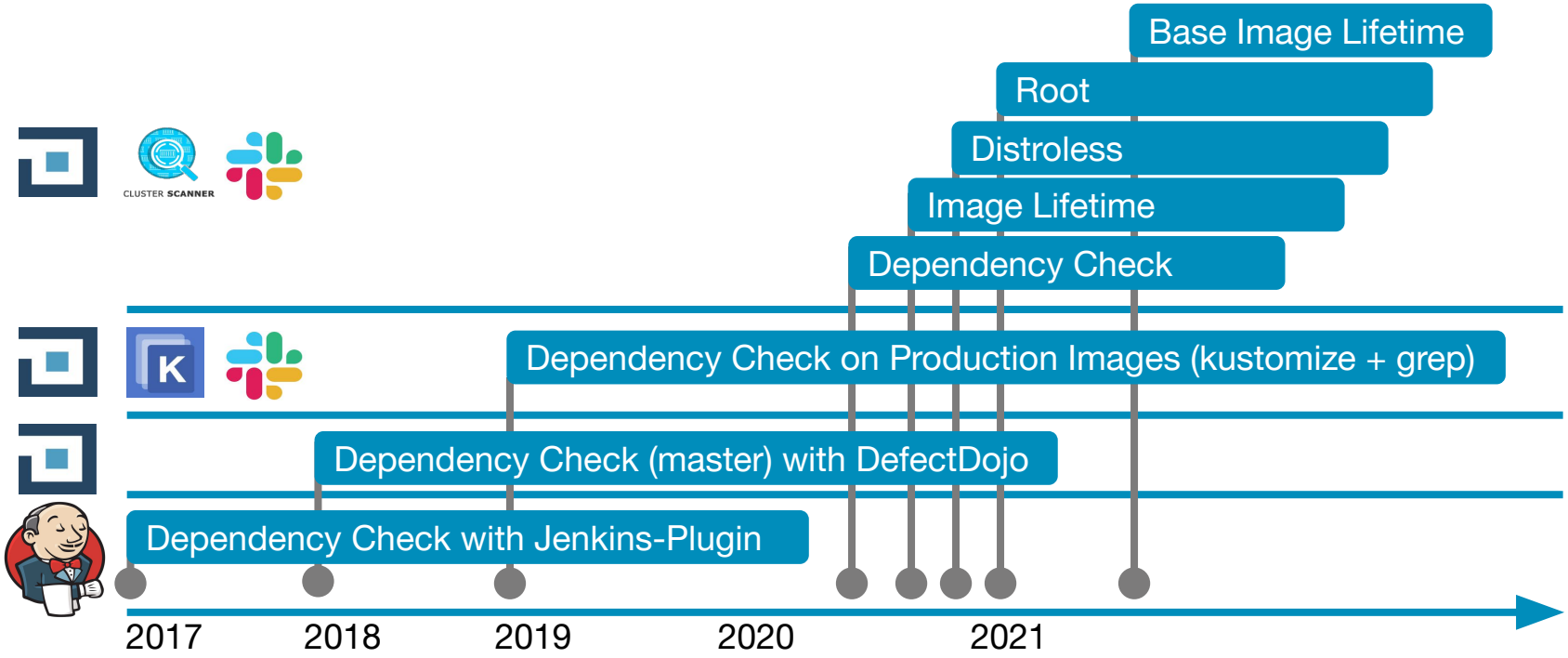
Runtime (versioned)



DevSecOps Test Journey

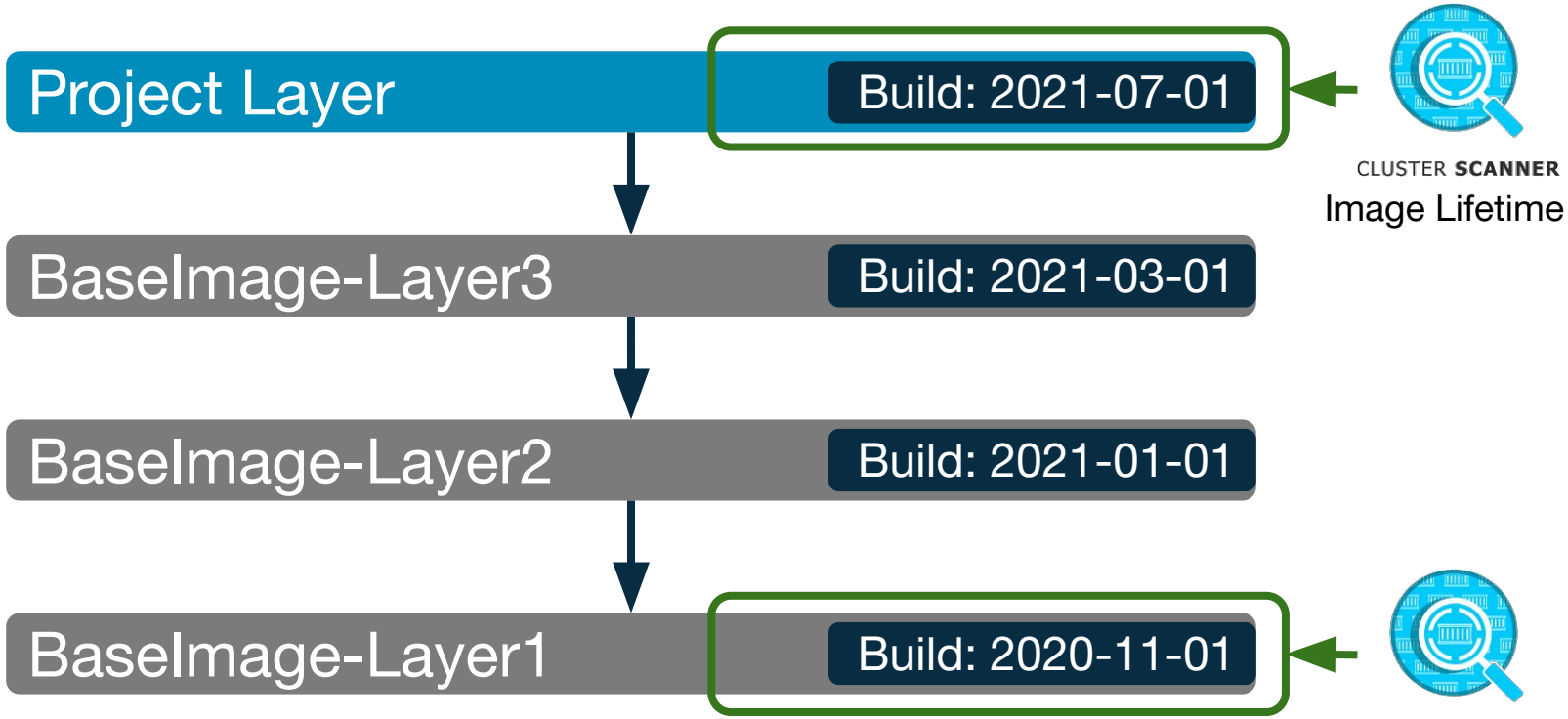


DevSecOps Test Journey



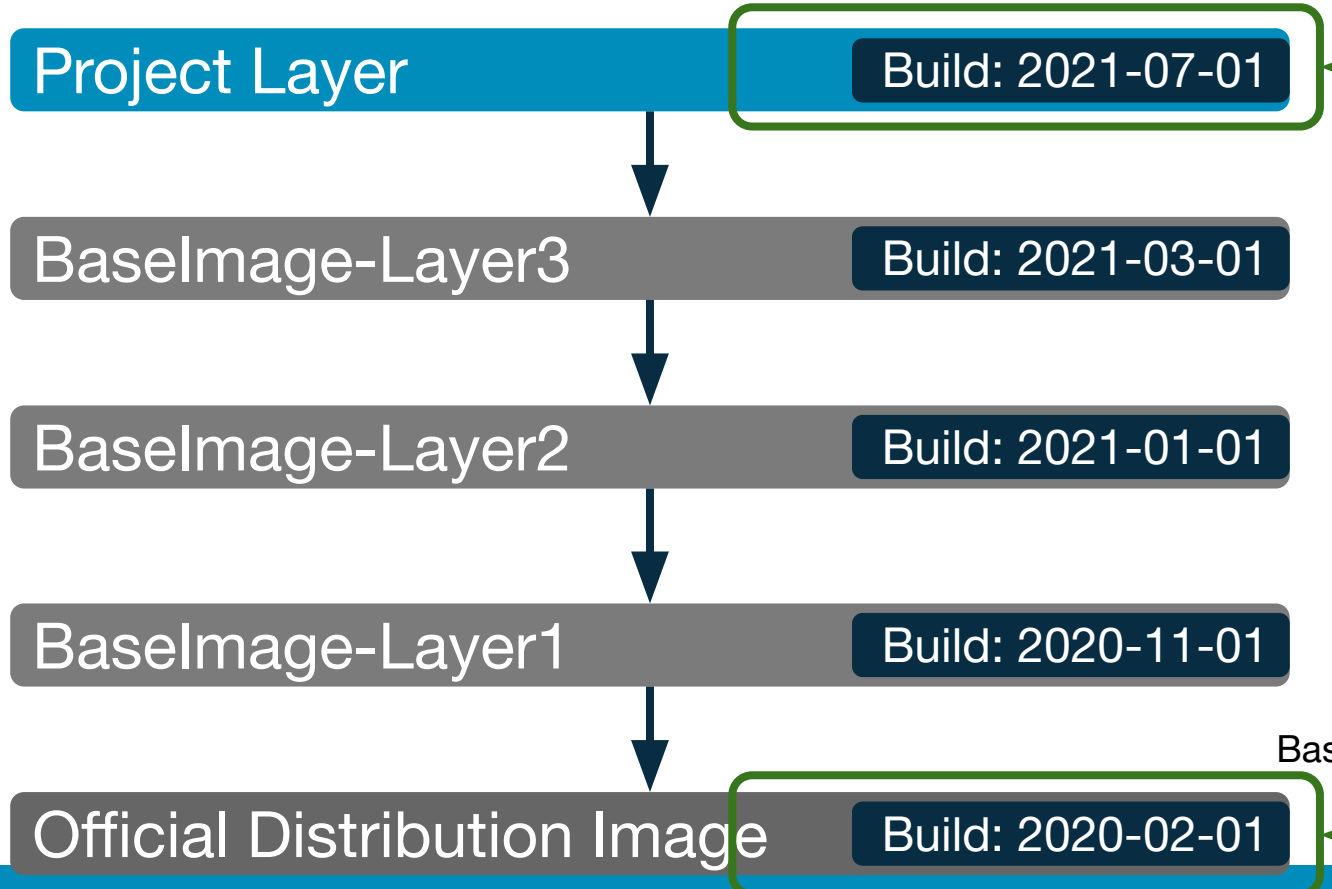
ImageLifetime Scan

SDA SE
OPEN INDUSTRY SOLUTIONS



BaseImageLifetime Scan

SDA SE
OPEN INDUSTRY SOLUTIONS

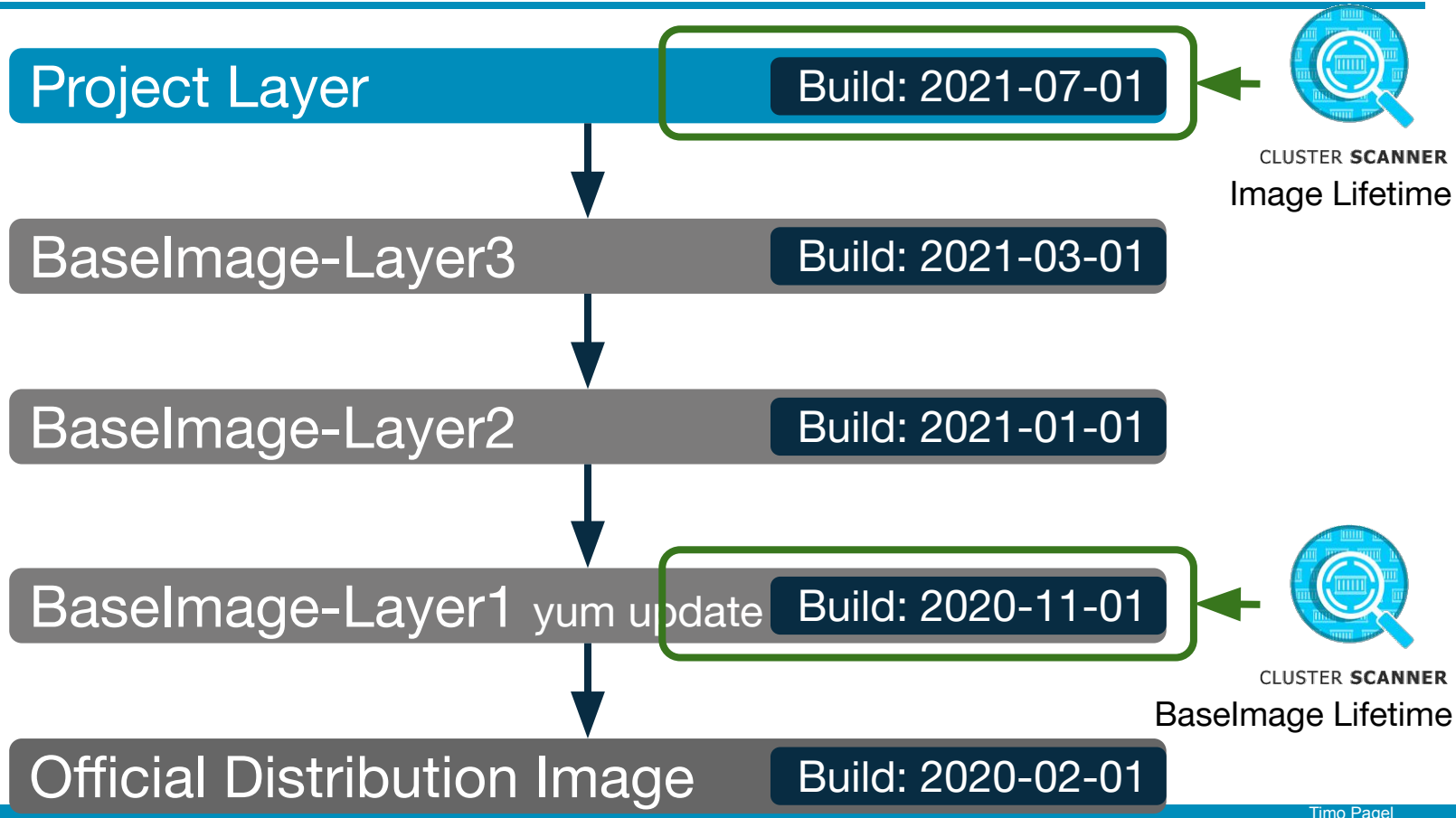


CLUSTER SCANNER
Image Lifetime

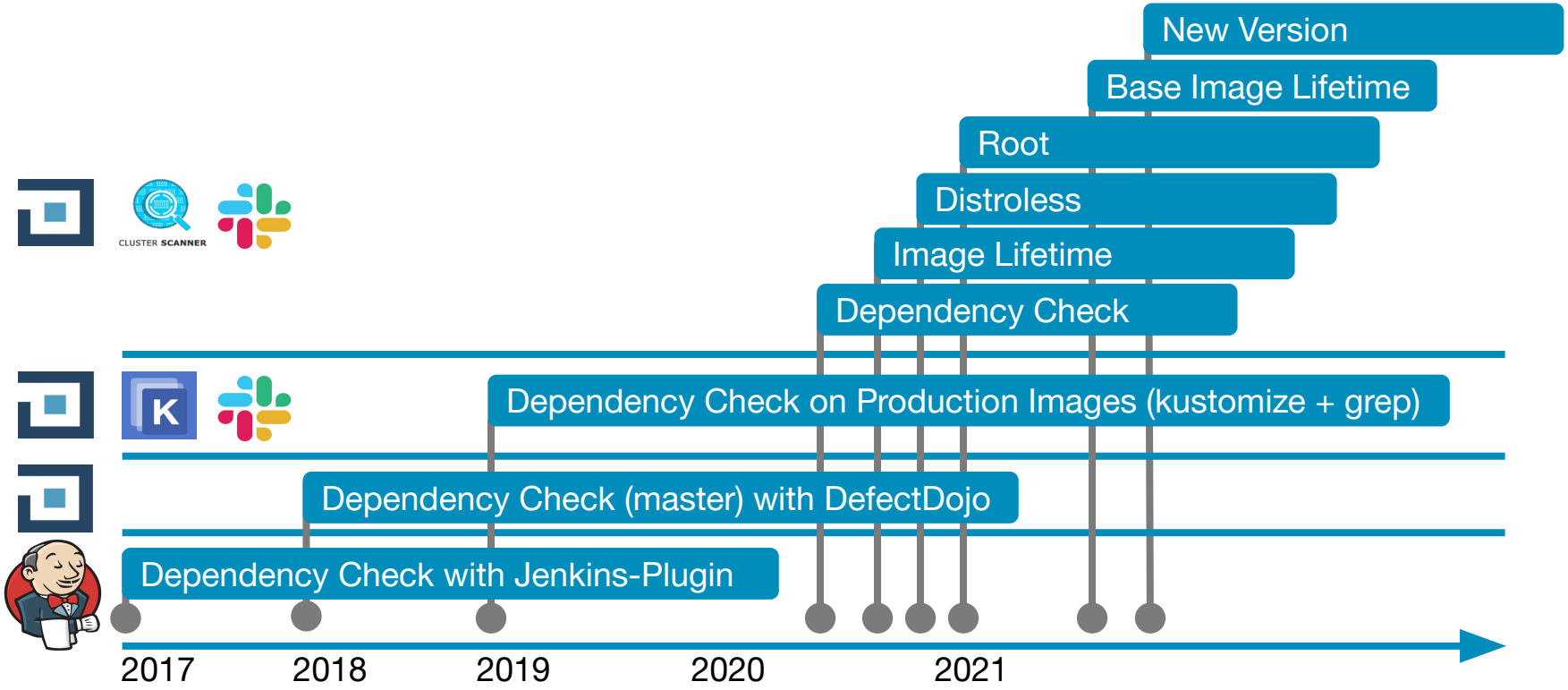
BaseImage Lifetime

BaseImageLifetime Scan

SDA SE
OPEN INDUSTRY SOLUTIONS



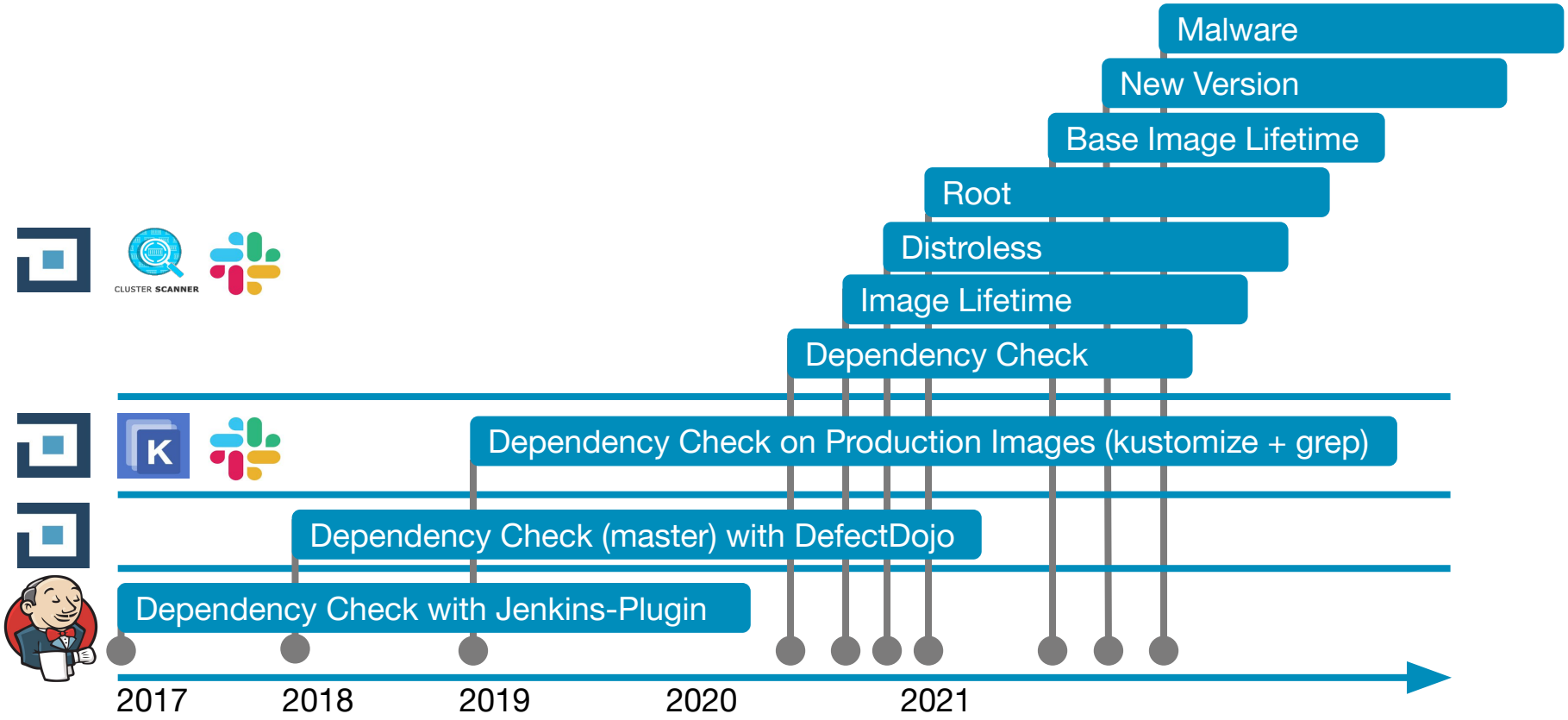
DevSecOps Test Journey



New Version

To reduce pulls
-> filter for quay.io/sdase
-> not checking for newest (e.g. 2.0.1)

DevSecOps Test Journey



Component Overview

- Running Application/Infrastructure Containers

- Image Collector
- Image Scanner
- Vulnerability Management DefectDojo
- Notification: Slack/E-Mail

Context

Kubernetes

Cluster Scanner Image Collector

Cluster Scanner Orchestration

Scans

DefectDojo Client (SecureCodeBox) / OWASP DefectDojo

Notifier

Architecture Overview

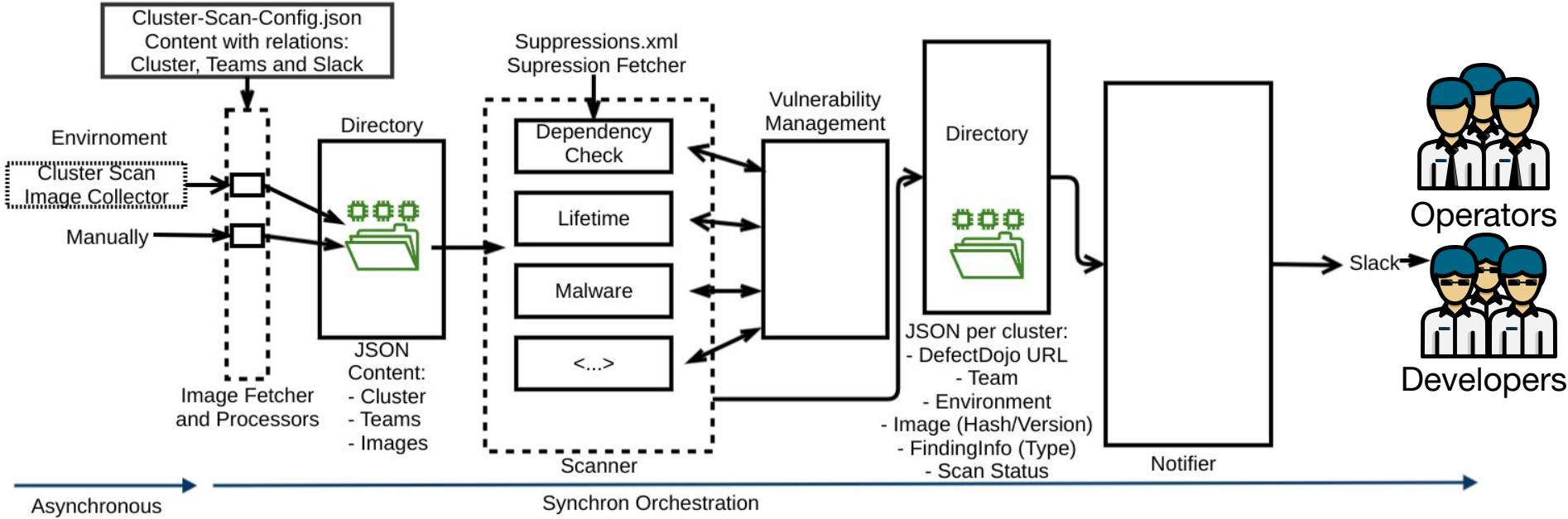
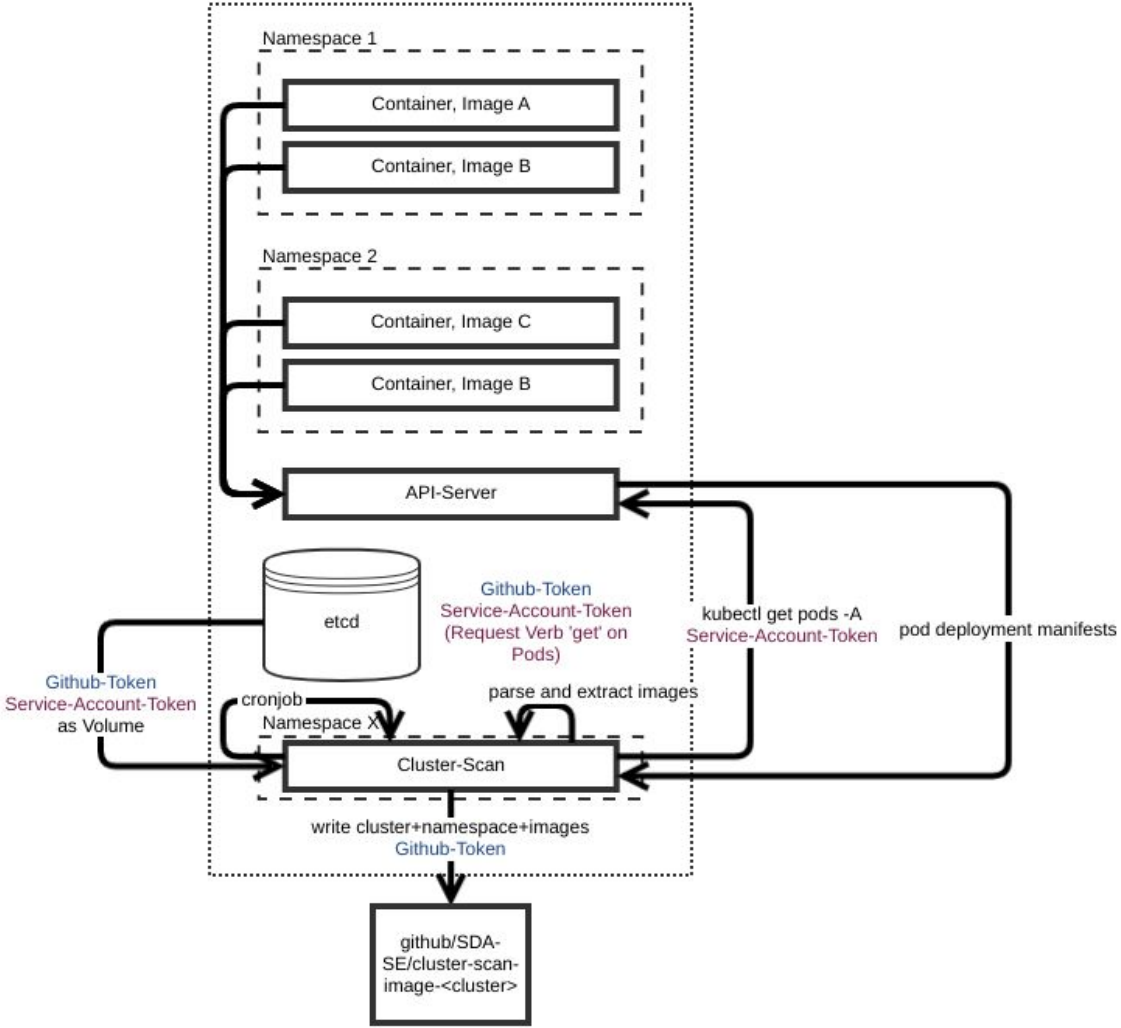
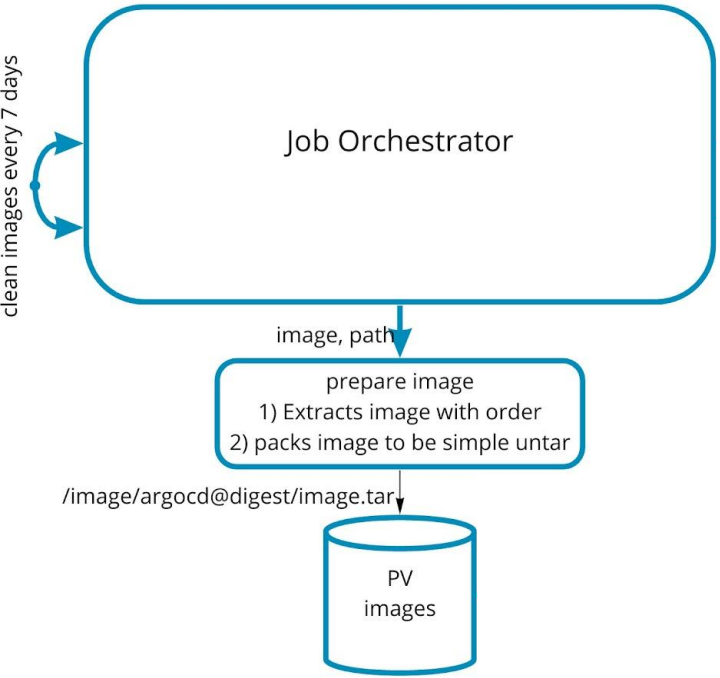


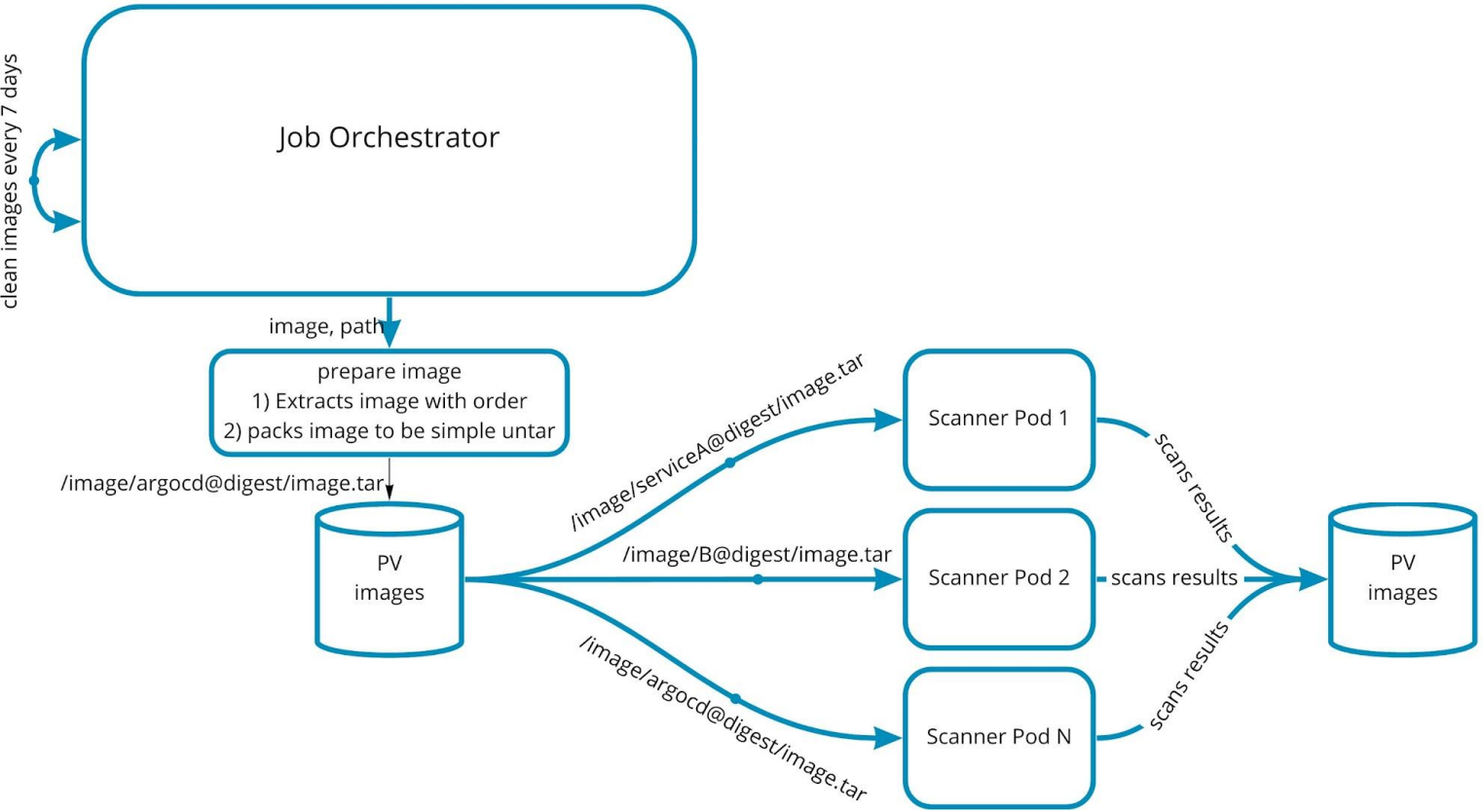
Image Collector



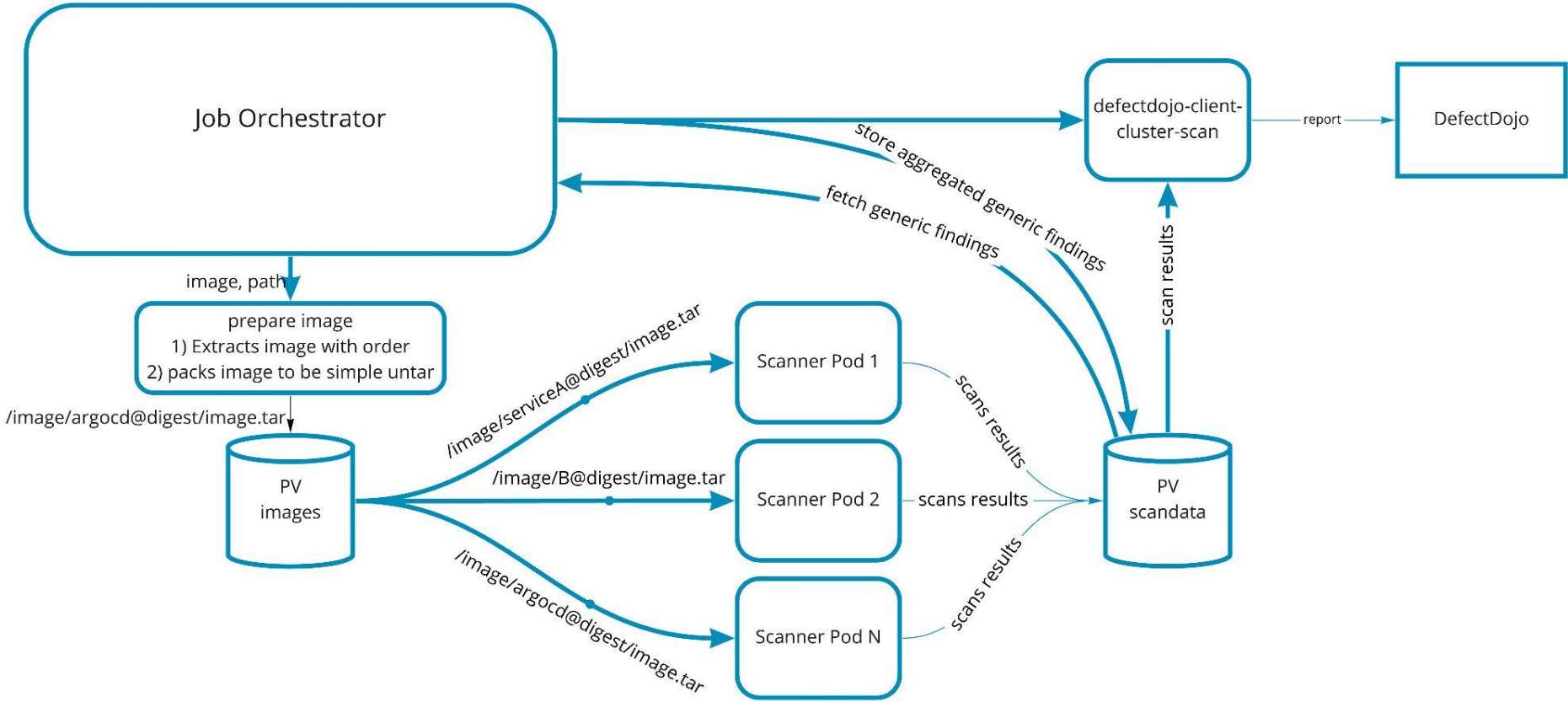
Storage


clean images every 7 days
Job Orchestrator
image, path
prepare image
1) Extracts image with order
2) packs image to be simple untar
/image/argocd@digest/image.tar
PV
images

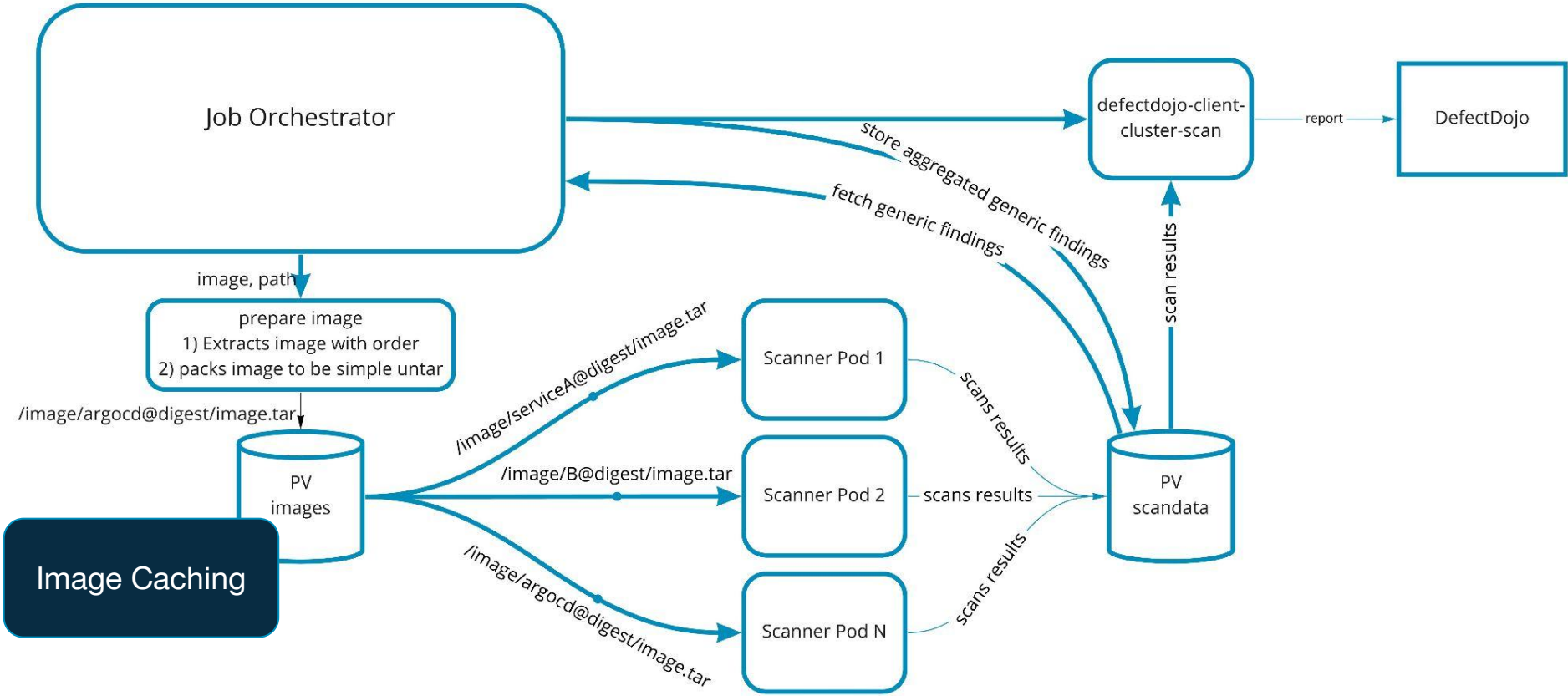
Storage


clean images every 7 days
Job Orchestrator
image, path
prepare image
1) Extracts image with order
2) packs image to be simple untar
/image/argocd@digest/image.tar
PV
images
/image/serviceA@digest/image.tar
/image/B@digest/image.tar
/image/argocd@digest/image.tar
Scanner Pod 1
Scanner Pod 2
Scanner Pod N
scans results
scans results
scans results
PV
images

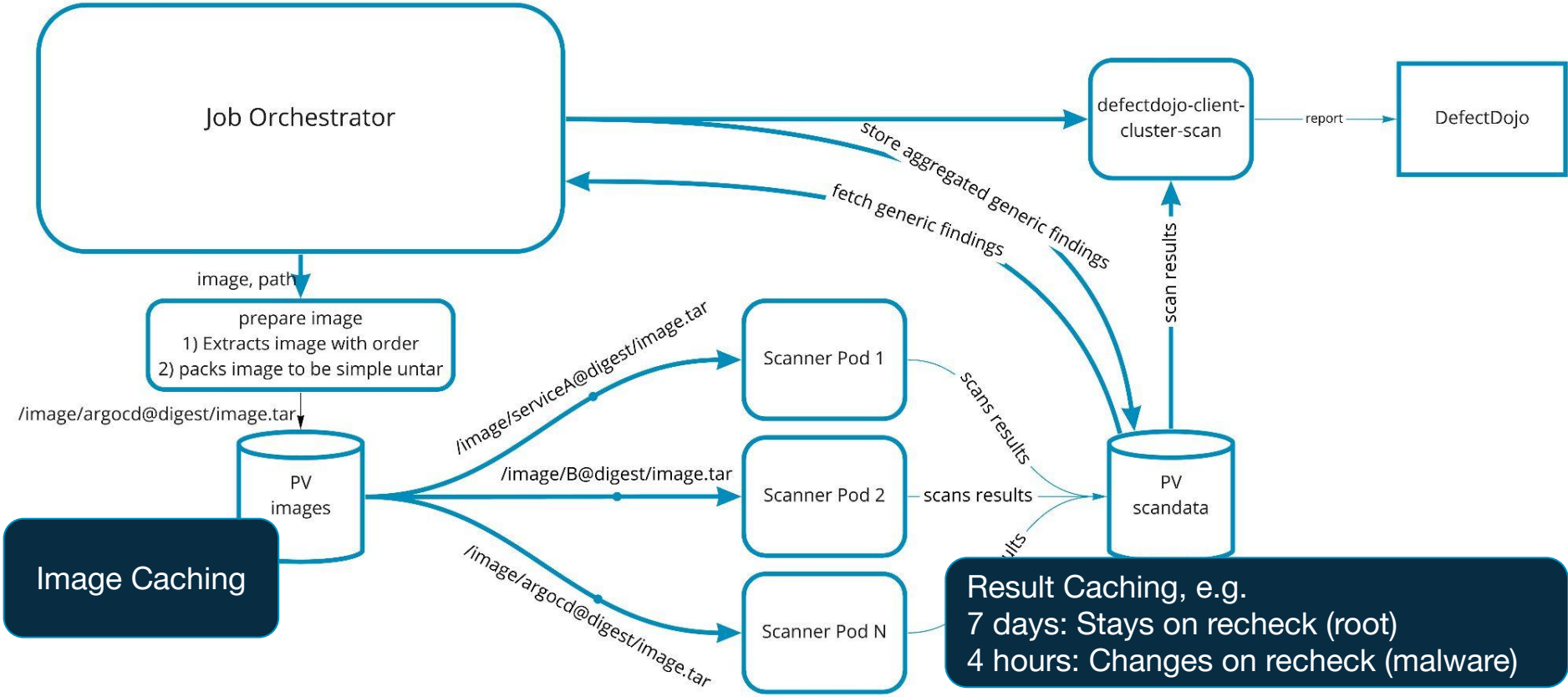
Storage


Job Orchestrator
defectdojo-client-cluster-scan
report
DefectDojo
store aggregated generic findings
fetch generic findings
scan results
image, path
prepare image
1) Extracts image with order
2) packs image to be simple untar
/image/argocd@digest/image.tar
PV images
/image/serviceA@digest/image.tar
/image/B@digest/image.tar
/image/argocd@digest/image.tar
Scanner Pod 1
Scanner Pod 2
Scanner Pod N
scans results
scans results
scans results
PV scandata

Storage


Job Orchestrator
defectdojo-client-cluster-scan
report
DefectDojo
store aggregated generic findings
fetch generic findings
scan results
image, path
prepare image
1) Extracts image with order
2) packs image to be simple untar
/image/argocd@digest/image.tar
PV images
/image/serviceA@digest/image.tar
/image/B@digest/image.tar
/image/argocd@digest/image.tar
Scanner Pod 1
Scanner Pod 2
Scanner Pod N
scans results
scans results
scans results
PV scandata
Image Caching

Storage


Job Orchestrator
defectdojo-client-cluster-scan
report
DefectDojo
store aggregated generic findings
fetch generic findings
scan results
image, path
prepare image
1) Extracts image with order
2) packs image to be simple untar
/image/argocd@digest/image.tar
PV images
/image/serviceA@digest/image.tar
/image/B@digest/image.tar
/image/argocd@digest/image.tar
Scanner Pod 1
Scanner Pod 2
Scanner Pod N
scans results
scans results
PV scandata
Image Caching
Result Caching, e.g.
7 days: Stays on recheck (root)
4 hours: Changes on recheck (malware)

Scan Status

cat ./pvc-[...]/20201230-01/quay.io/sdase/defectdojo-client@sha256:945[...]/module_scan-dependency-check.json


```
{
  "scan-dependency-check": {
        "finishedAt": "2021-02-02 17:04:52.271404683+00:00",
        "startedAt": "2021-02-02 17:03:16.075462195+00:00",
        "status": "completed"
  }
}
```

Scans

- Distroless
- Image Lifetime
- Known Vulnerabilities
- Malware
- Root-User

Job Scheduling Options

- Kubernetes
  - Limit jobs and fire up (fast)
  - Might result in complicated job status review
- argo-workflows 
  - Introduction needed (slow)
  - Might give more visibility about the job status

Scheduling Notifications

- Periodically (Nightly)

Scheduling Notifications

- Nightly
  - Wish from Dev.: Reminder once per day (not multiple notifications during the day)
  - Wish from Ops.: Once every *X* days
  - On Premise: CPU resources available

Scheduling Notifications

- Nightly
  - Wish from Dev.: Reminder once per day (not multiple notifications during the day)
  - Wish from Ops.: Once every *X* days
  - On Premise: CPU resources available
- On Deployment (Kubernetes Operator)
  - Wish from Sec.: Fast information in case of a misconfiguration

cluster-scan-test APP 15:31 Uhr

BaseImage Age > 60 Days in huggingface/mongoku@sha256:c59caa0a5de48906361b7011d98a6cc57577bc9bf3d421454f96db02eebcf2b4

ClusterScanner

Potential unhandled vulnerabilities or misconfigurations found in image

BaseImage is 905 days too old

BaseImage huggingface/mongoku@sha256:c59caa0a5de48906361b7011d98a6cc57577bc9bf3d421454f96db02eebcf2b4

Cluster minikube

Namespace abc

Scan Job Status: completed

image: huggingface/mongoku@sha256:c59caa0a5de48906361b7011d98a6cc57577bc9bf3d421454f96db02eebcf2b4, app: kiali, namespace: abc, envirnoment: minikube

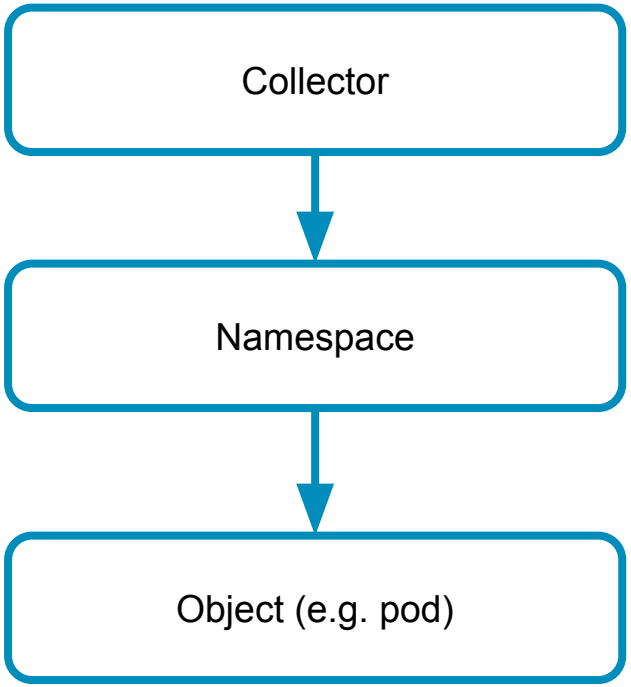
Handle potential vulnerabilities

Contact Information

contact.sdase.org/email='k.panier@sda.se'

contact.sdase.org/slack='#fellowship-security'

Annotation Inheritance

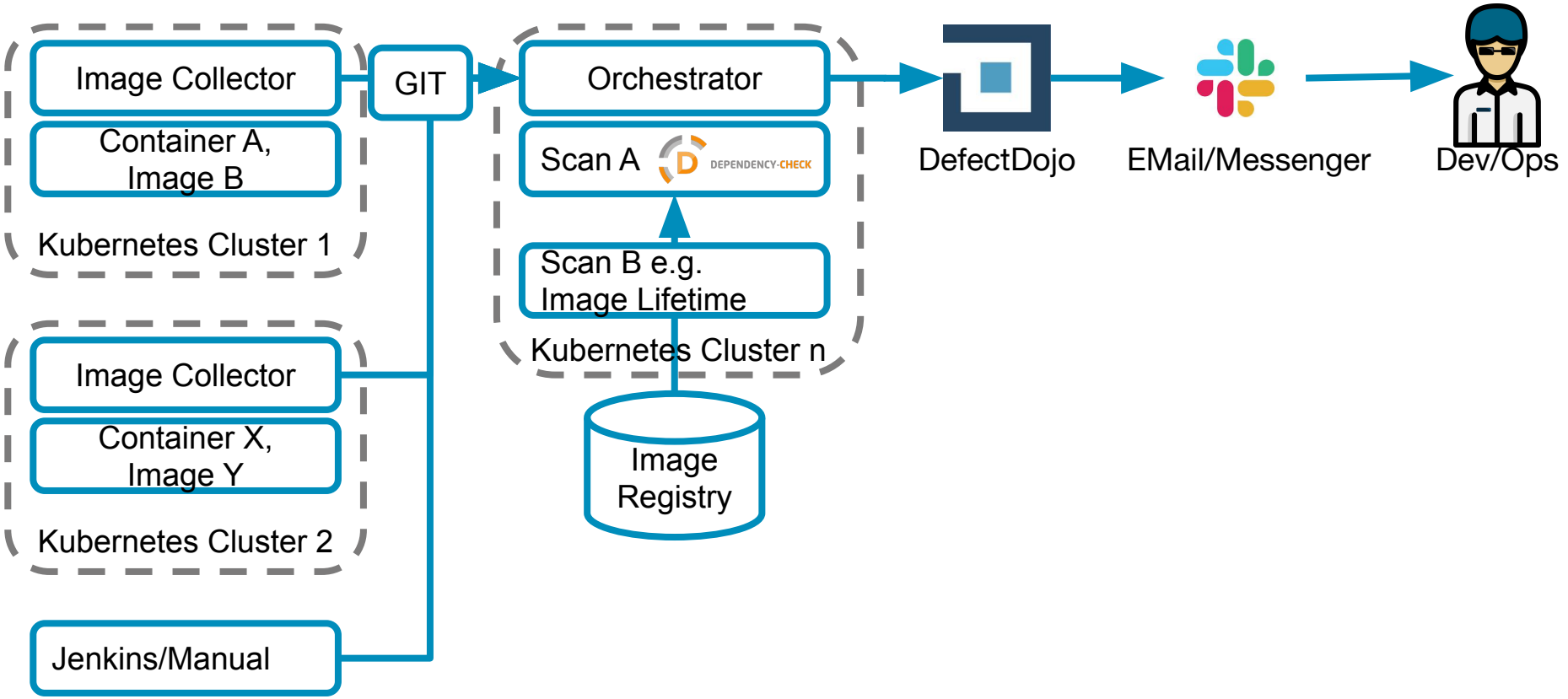


Environment Variable: DEFAULT_TEAM_NAME=team1

Annotation: contact.sdase.org/team=team2

Annotation: contact.sdase.org/team=team3

SDA SE ClusterScanner Overview



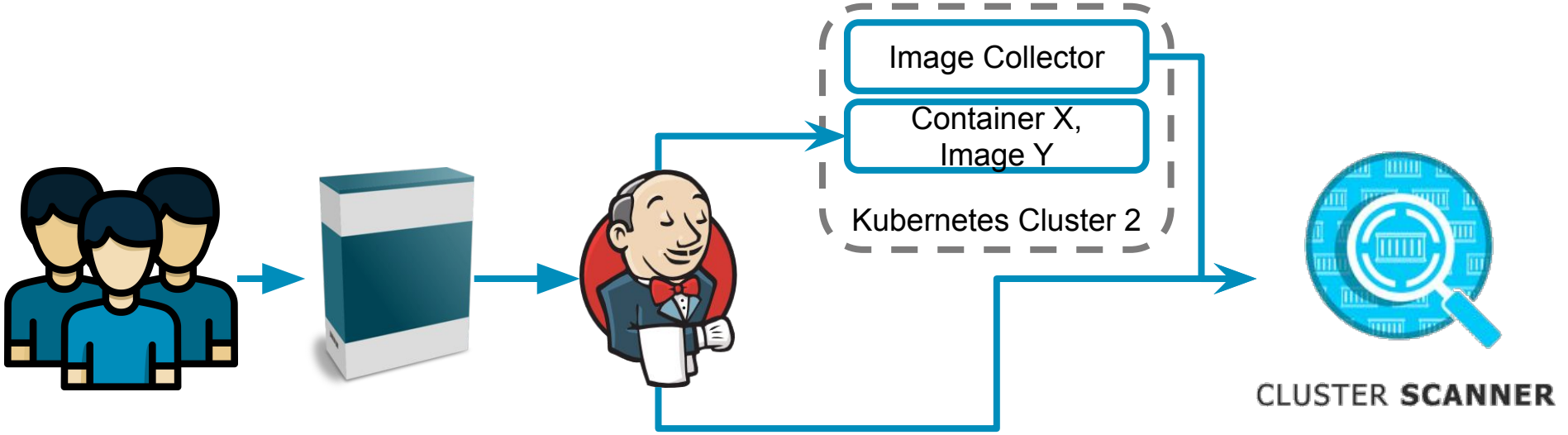
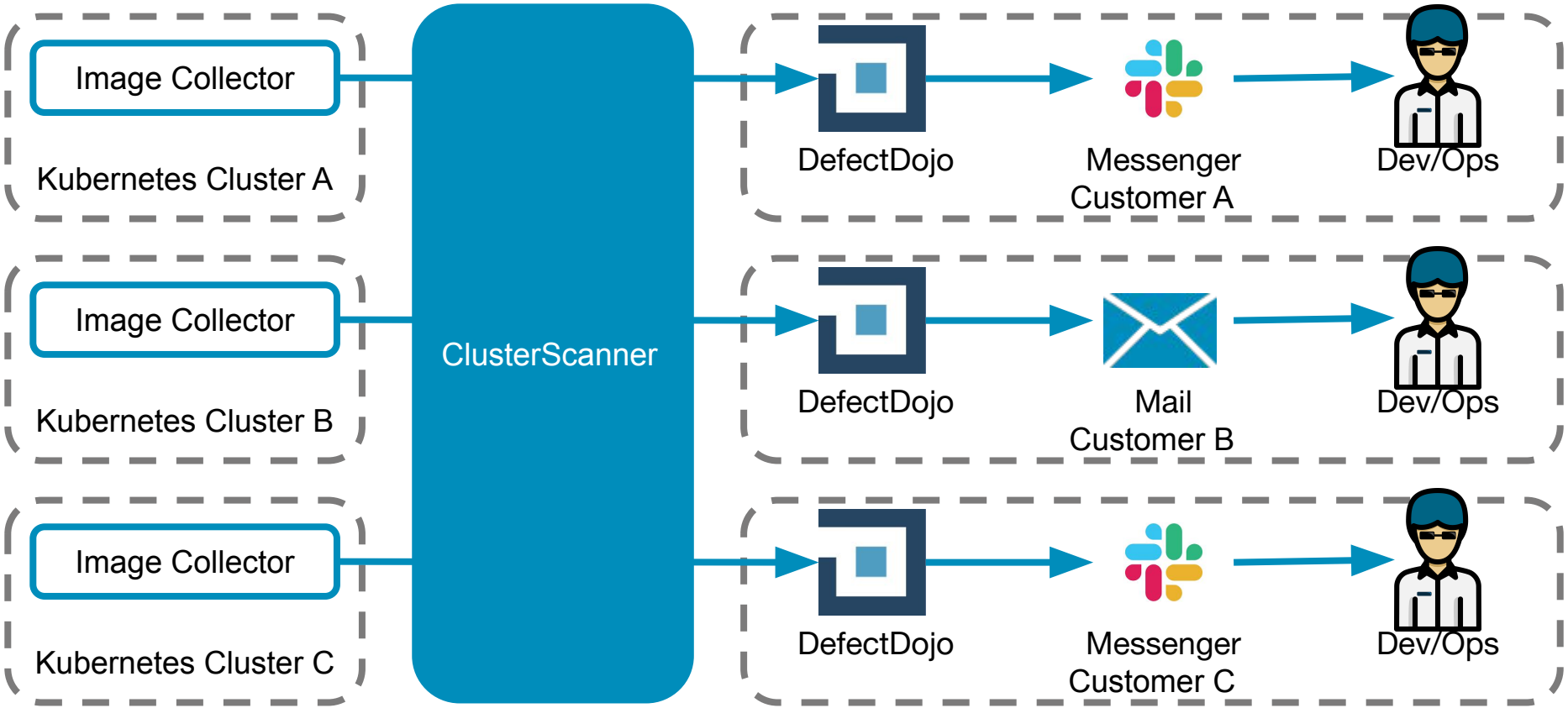
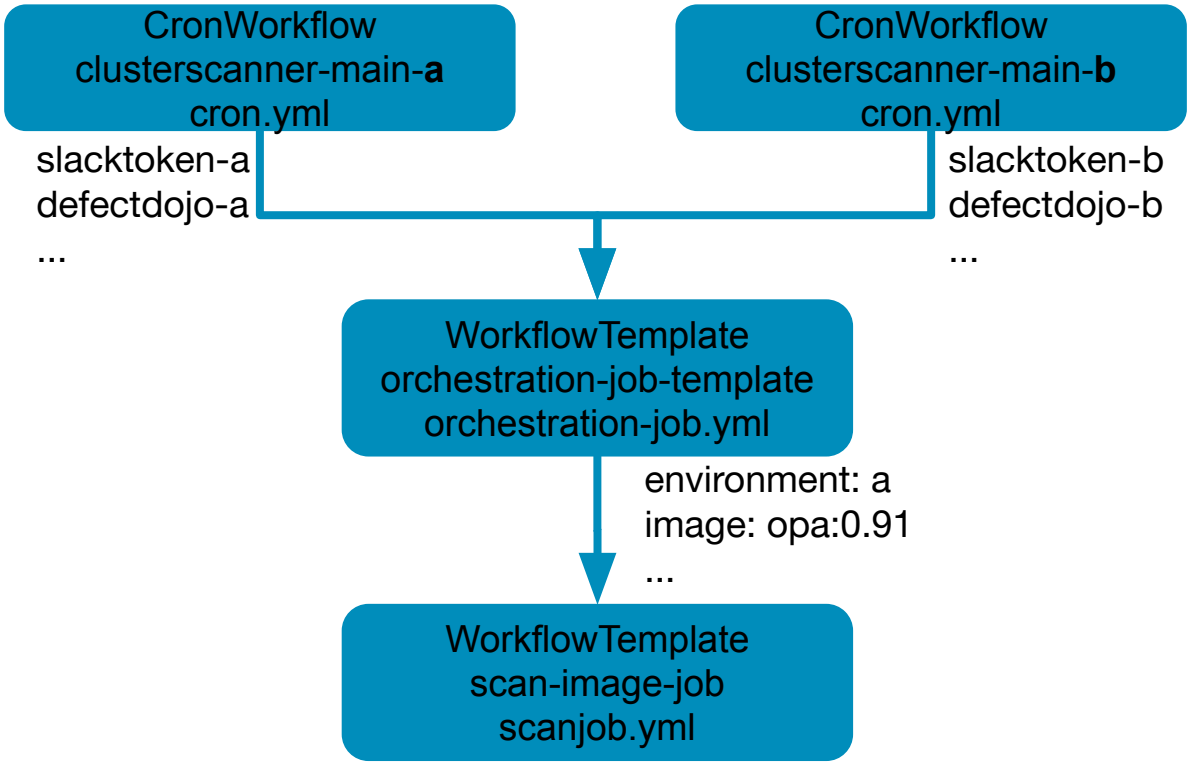


Image Collector
Container X,
Image Y
Kubernetes Cluster 2
CLUSTER SCANNER

Multi Tenant Deployment



Multi Tenant Deployment



Label Inheritance

Inheritance:

- Cluster
- Namespace
- Object

Asset Catalog

- ISMS requires an asset catalog with asset and a description
- Where are my applications deployed?
  Teams -> Repos -> Deployments -> Potential Findings

Asset Catalog

- ISMS requires an asset catalog with asset and a description
- Where are my applications deployed?
  Teams -> Repos -> Deployments -> Potential Findings

Annotation: contact.sdase.org/team="ring-fellowship"

Annotation:
scm.sdase.org/source_url="https://github.com/cluster-image-scanner"

scm.sdase.org/source_branch="master"

scm.sdase.org/release="2.0.0"

Label: app.kubernetes.io/name="two-towers"

Label: app.kubernetes.io/version="v2.0.0-20210915"

Annotation: sdase.org/description="The Two Towers provide [...]"

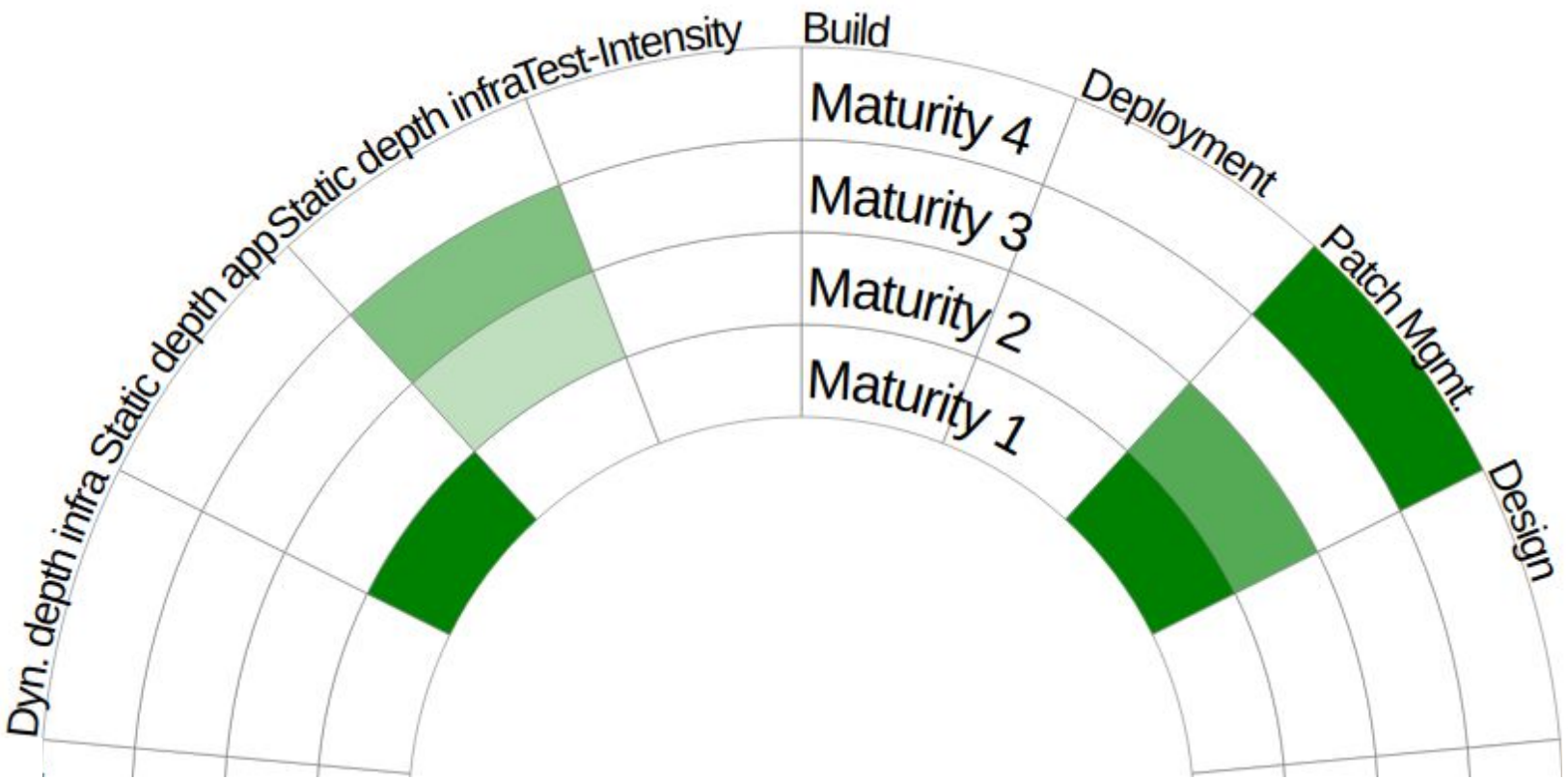
Severity and Notifications

Problem: Some customers do not care about (Base)Image Lifetime and ignore not (but do not want to turn it of

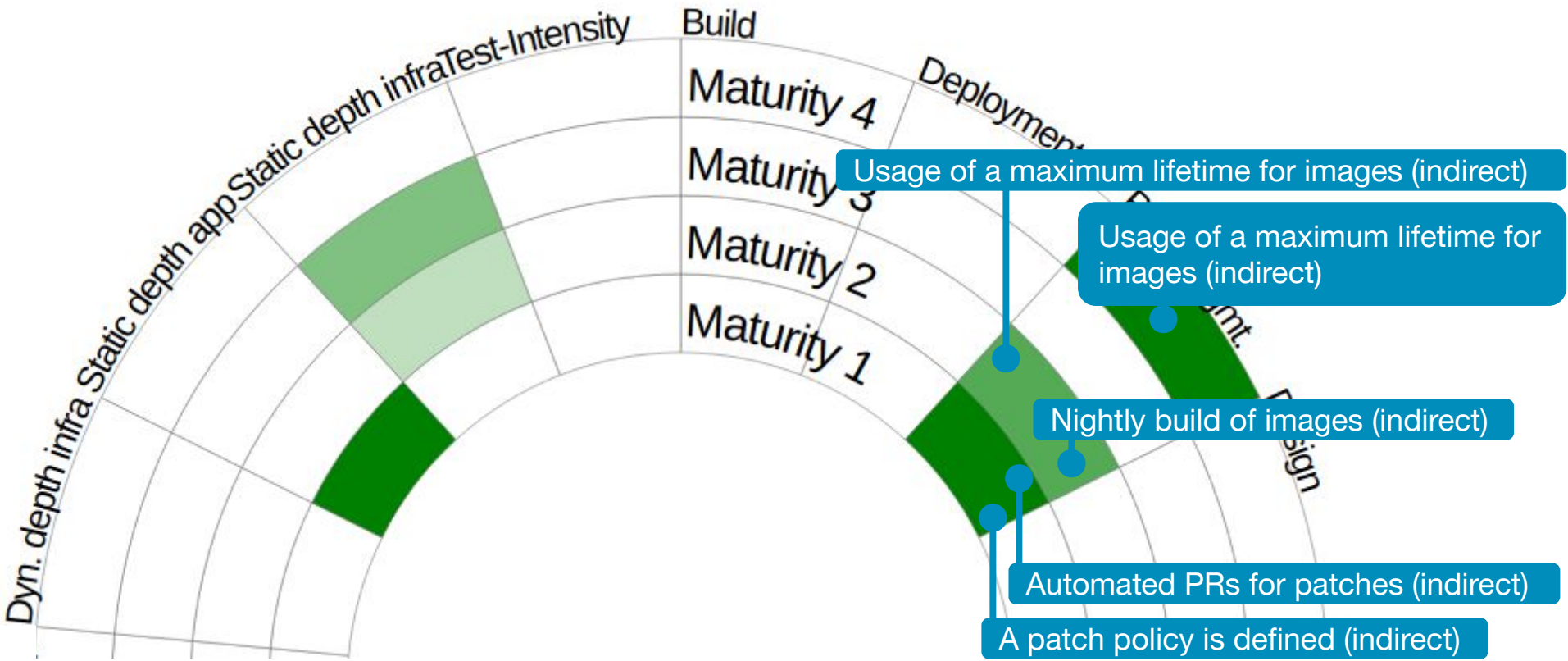
Discussion!

- Take into account interval of notification and DefectDojo
- Severity of finding

Conclusion

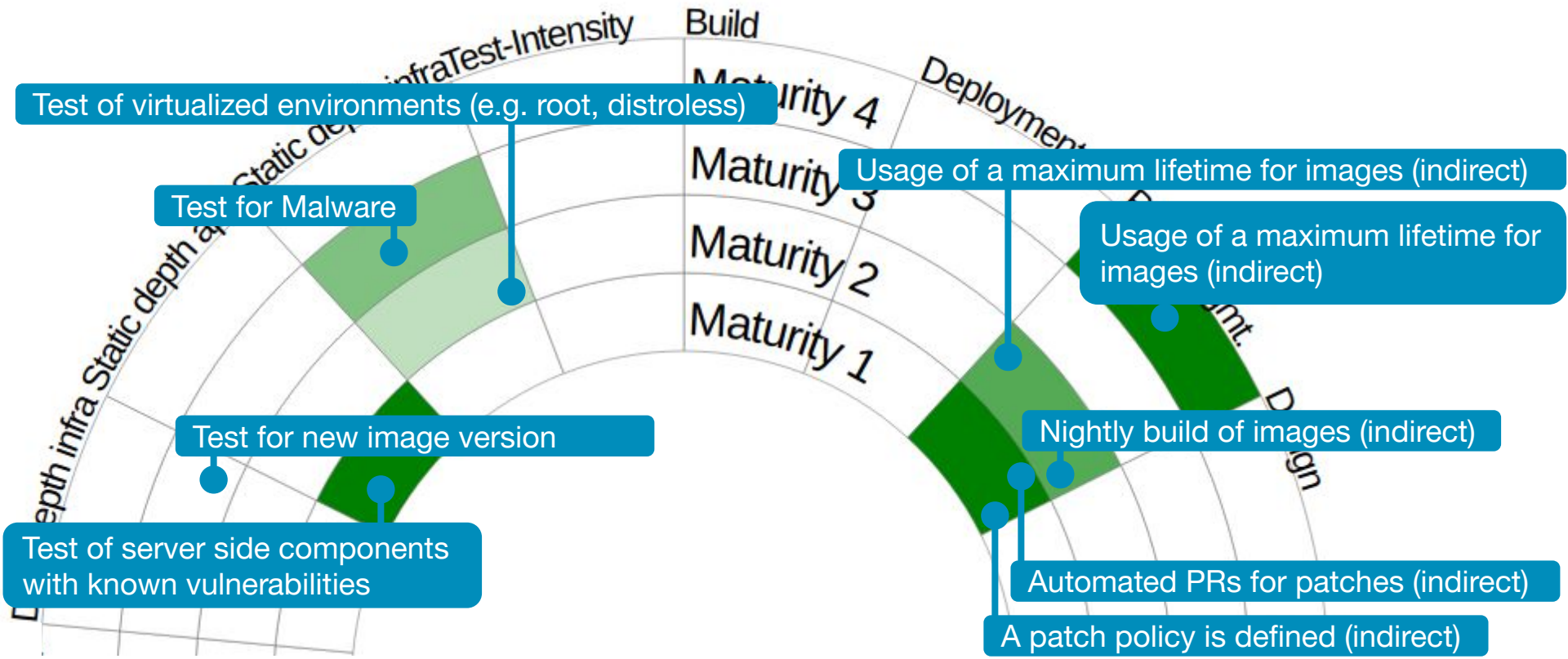

Build
Deployment
Test-Intensity
Static depth infra
Static depth app
Dyn. depth infra
Patch Mgmt.
Design
Maturity 4
Maturity 3
Maturity 2
Maturity 1

Conclusion


Dyn. depth infra
Static depth app
Static depth infra
Test-Intensity
Build
Maturity 4
Maturity 3
Maturity 2
Maturity 1
Usage of a maximum lifetime for images (indirect)
Usage of a maximum lifetime for images (indirect)
Nightly build of images (indirect)
Automated PRs for patches (indirect)
A patch policy is defined (indirect)

Check for new image version

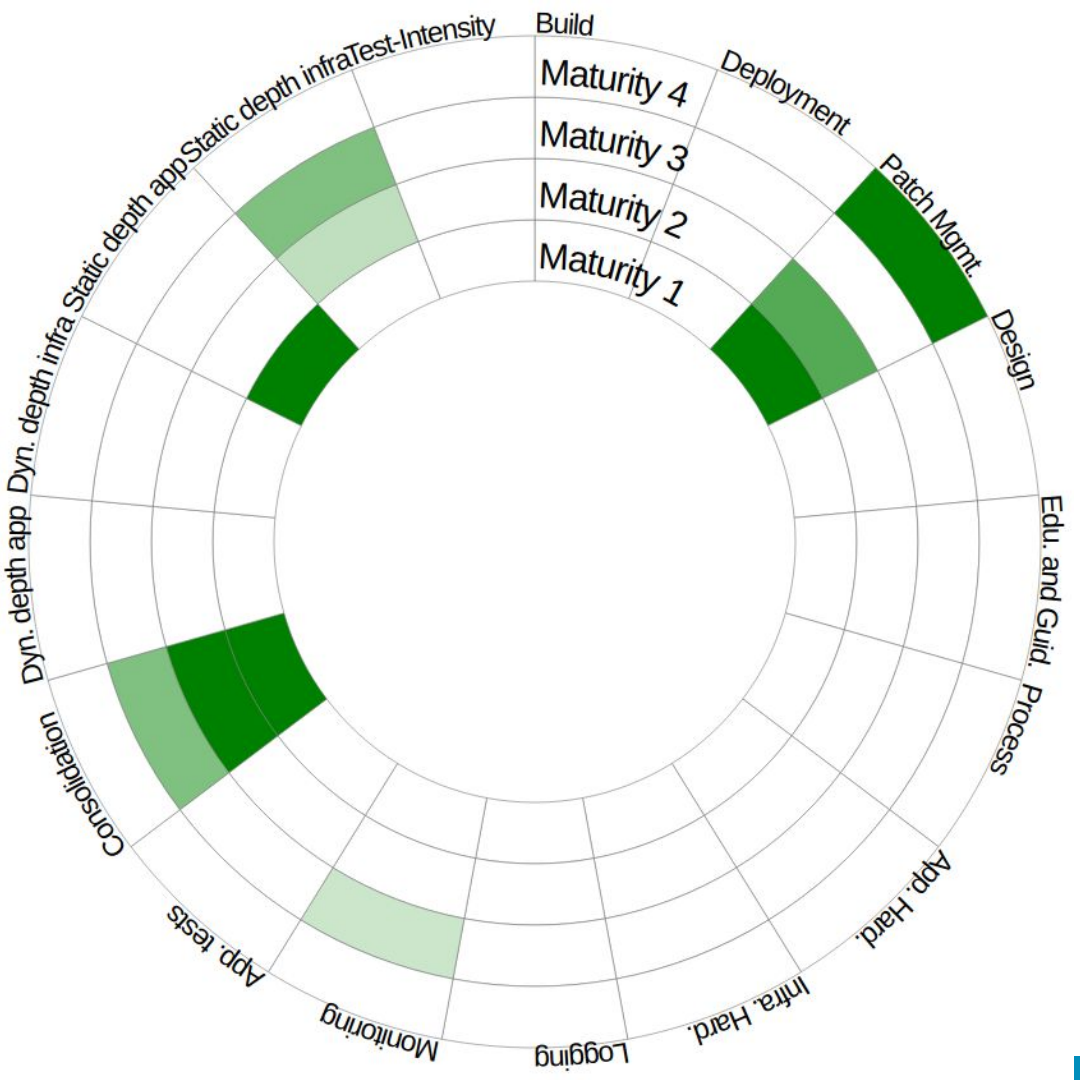
Conclusion



Cluster Scanner

+

DefectDojo



Conclusion

SDA SE
OPEN INDUSTRY SOLUTIONS

Vulnerability Management via OWASP DefectDojo

Thank you
Questions?

CLUSTER **SCANNER**

timo.pagel.ext@sda.se
clusterscanner@pagel.pro

work@pagel.pro for Jobs

Outlook

Mid Term:

- ore Scans
- Plain Docker Collector

Long Term: Dependency Track