Towards a DevSecOps approach in the CICD pipelines for the INFN sysinfo apps

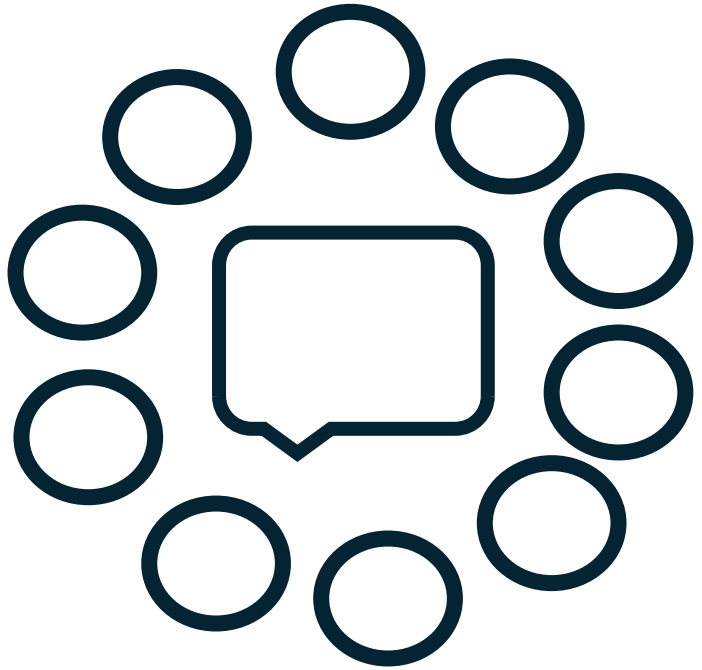
Stefano Bovina

Guido Guizzunti

Giuseppe Misurelli (speaker)

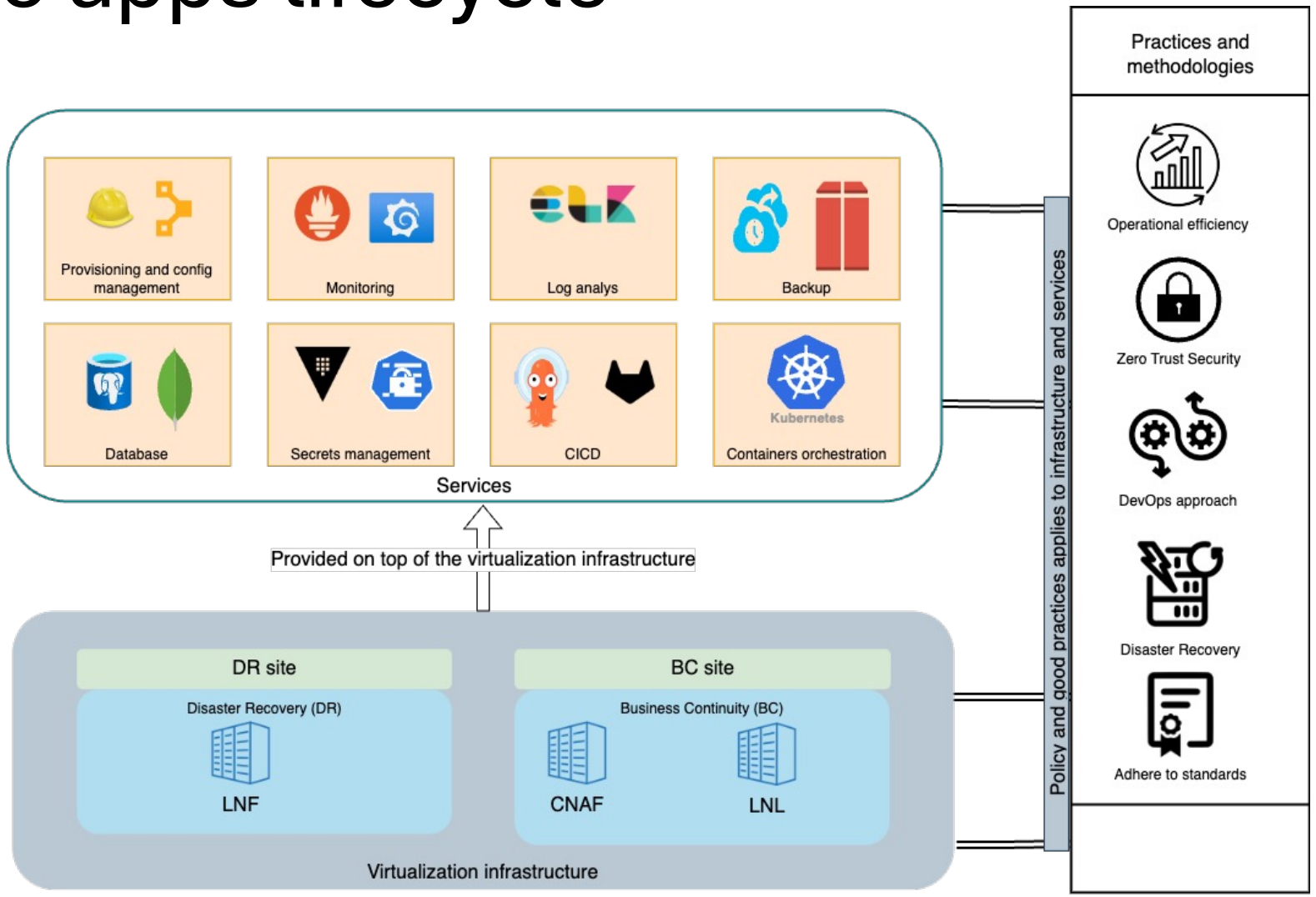
The INFN sysinfo apps

Ecosystem of apps serving INFN people (business trips, buying computing facilities, managing recruitment process, accounting, payrolls).



Enabling sysinfo apps lifecycle

Platform team providing infrastructure and services to enable the software development lifecycle.

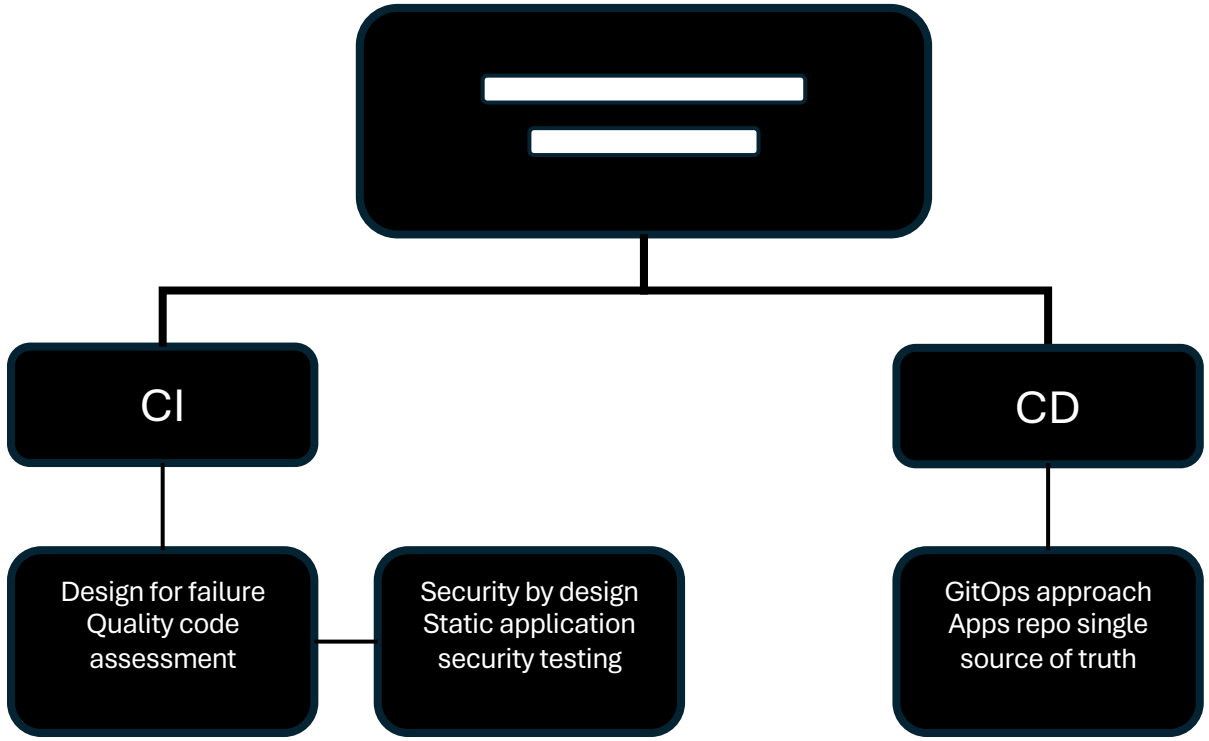


Re-architecting sysinfo apps

Legacy applications to entirely replace/re-think.

Good chance to architecting focusing on:

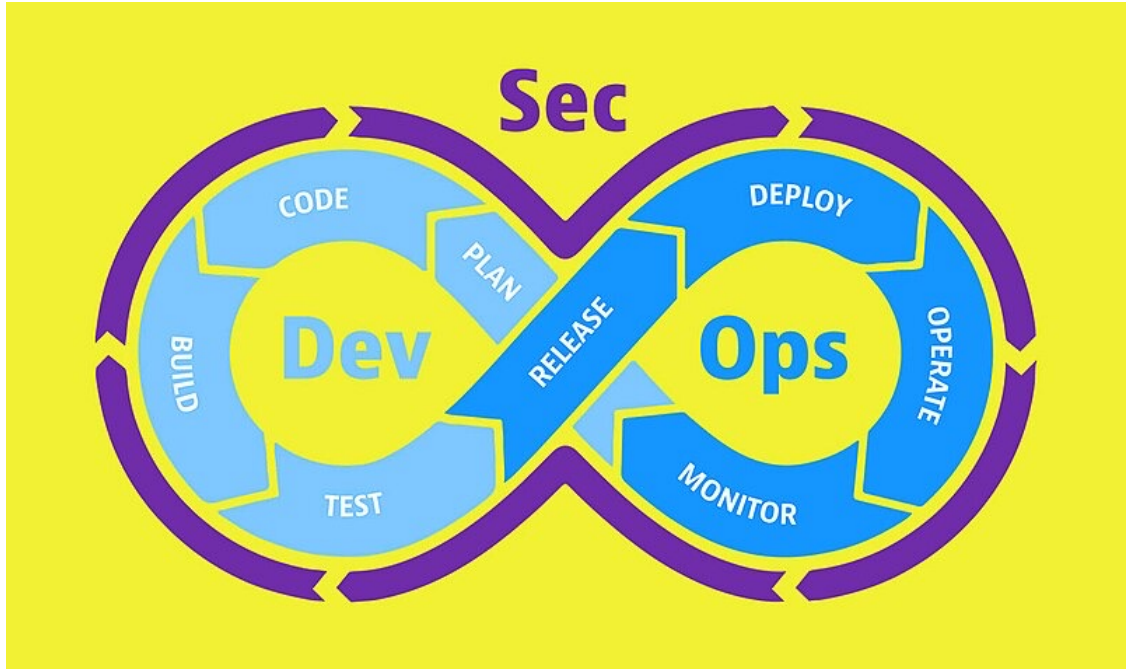
- Produce better software
- Improve security and compliance
- Optimize microservices management



CICD pipelines evolution (>> DevSecOps)

Guiding principles:

- Agree on a contract on how to develop, build, test, deliver and deploy software
- Platform team owner and responsible for the governance and security
- Shift left testing to introduce as early as possible



Automate, Monitor, Apply security at all phases of software development lifecycle.

Guiding principle 1

Platform and Development team to agree on a contract on how to develop, build, test, deliver and deploy software.

Build, metadata, dependencies at project level (build.gradle, project.toml, package.json)

Formalized design patterns (circuit braker, timeout/retry/failure management, integrated monitoring)

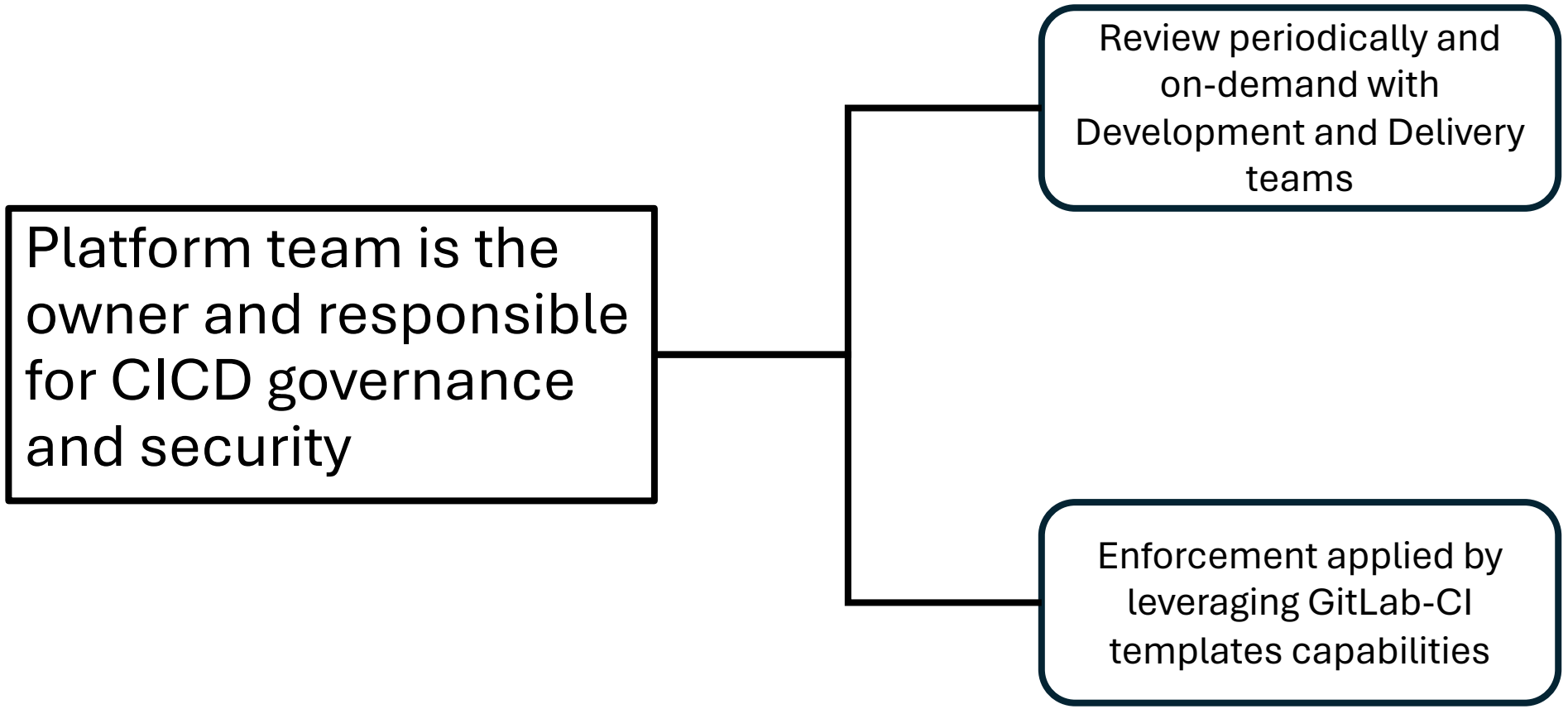
SAST with well identified acceptance thresholds

Security scanning of containers images with well identified acceptance thresholds

Containers image registry publishing, scanning and lifecycle policy

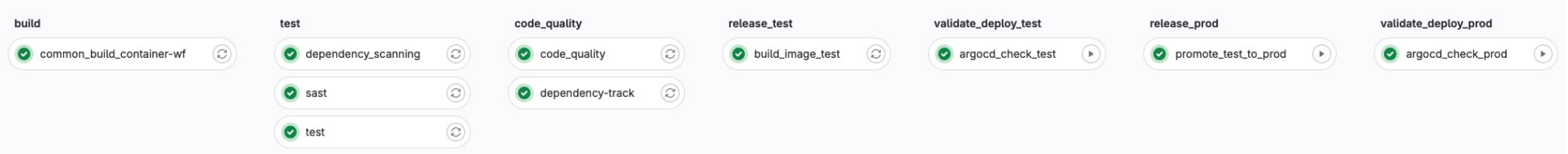
Kubernetes deployment workflow and admission controller policy

Guiding principle 2



Guiding principle 3

Introduce shift left testing as early as possible in the software development lifecycle.



Security assessments early in the software development lifecycle before vulnerabilities find their way into production.

Shift left testing to balance with shift right one.

- Production is where most attacks happen.
- Observing the application when it is running in production.
- Zero day vulnerabilities.

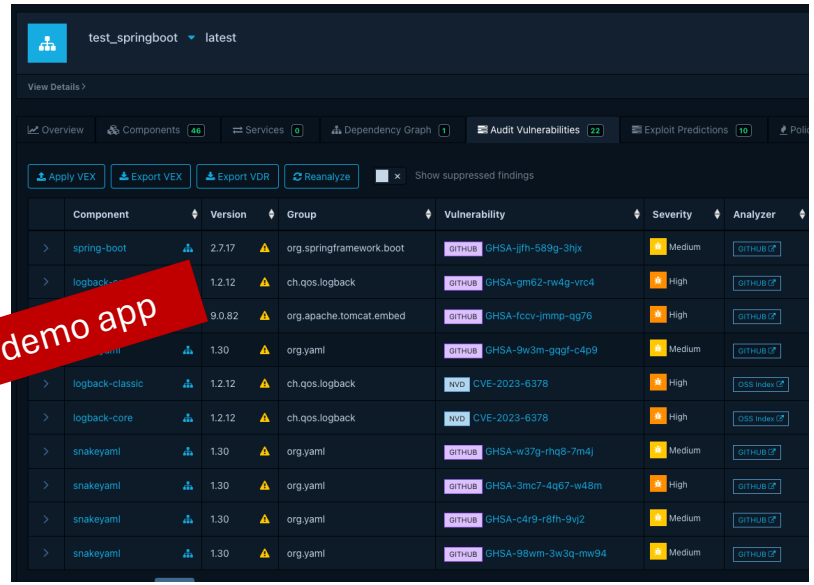
Our shift left strategy

- Main goal to satisfy SLSA specification as much as possible.
- Find and prevent defects early in the software delivery process.
- Development teams are aware of the security constraints and best practices.
- CI pipelines create SBOM files (Software Bills of Material) to:

Spot problematic code and automatically block the pipeline.

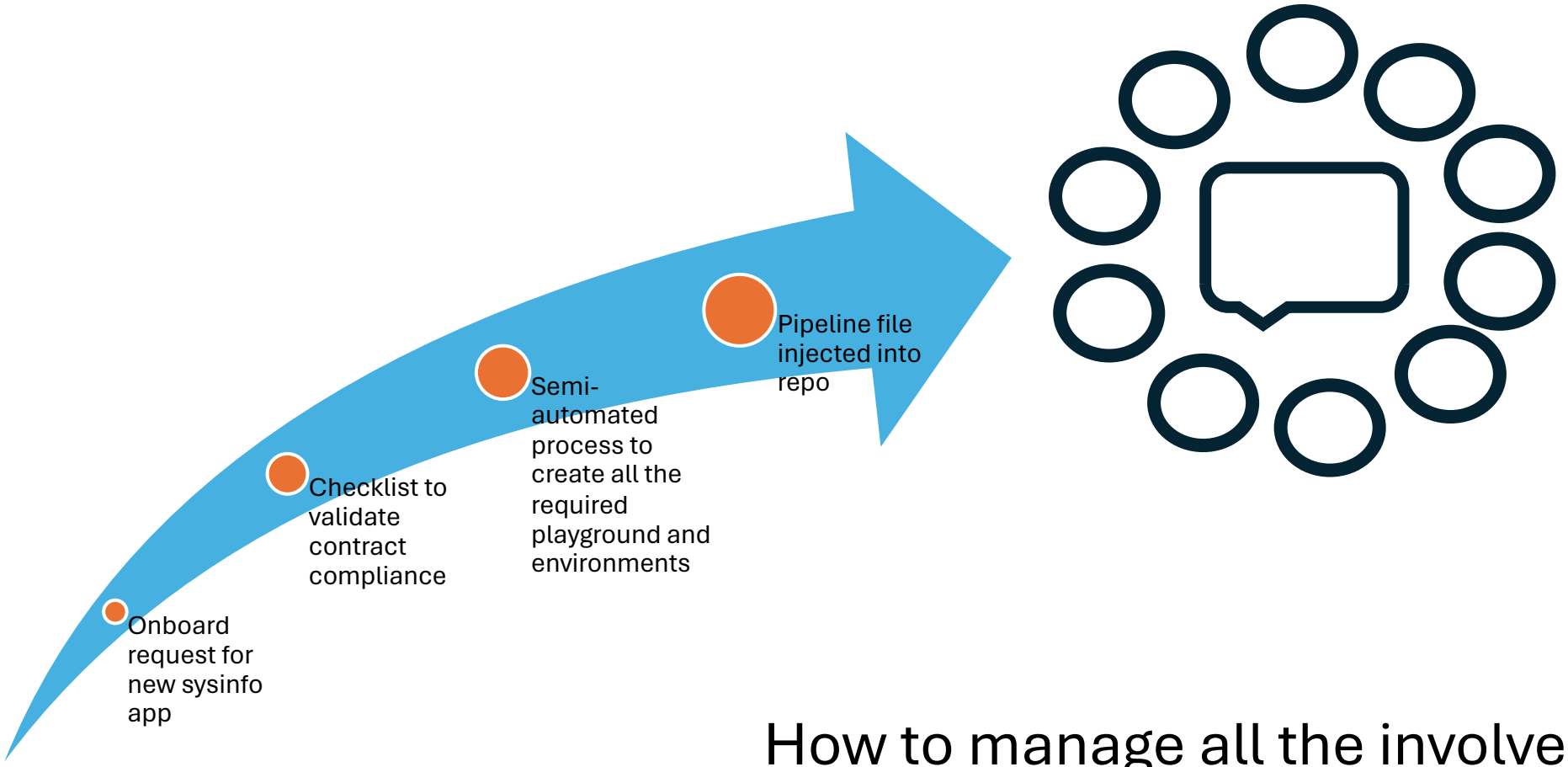
```
Generating report for project test_springboot...
Only vulnerabilities with a severity >= high will trigger an error
---------------------------------------------------------------------------------------
| Dependency                                                | CVE # |  Severity |
---------------------------------------------------------------------------------------
| test_springboot.jar: jackson-databind-2.13.5.jar          |     1 |    MEDIUM |
---------------------------------------------------------------------------------------
| test_springboot.jar: logback-core-1.2.12.jar              |     2 |      HIGH |
---------------------------------------------------------------------------------------
| test_springboot.jar: snakeyaml-1.30.jar                   |     7 |  CRITICAL |
---------------------------------------------------------------------------------------
| test_springboot.jar: spring-boot-2.7.17.jar               |     1 |    MEDI
---------------------------------------------------------------------------------------
| test_springboot.jar: spring-boot-jarmode-layertools-2.7.17.jar |     1 |
---------------------------------------------------------------------------------------
| test_springboot.jar: spring-web-5.3.30.jar                |     2 |  CRITICAL |
---------------------------------------------------------------------------------------
| test_springboot.jar: tomcat-embed-core-9.0.82.jar         |     1 |      HIGH |
---------------------------------------------------------------------------------------
Exit code 1...see dependency-check report for more details.
```

Enable continuous scanning of released software (using dependency track tool)



Data related to demo app

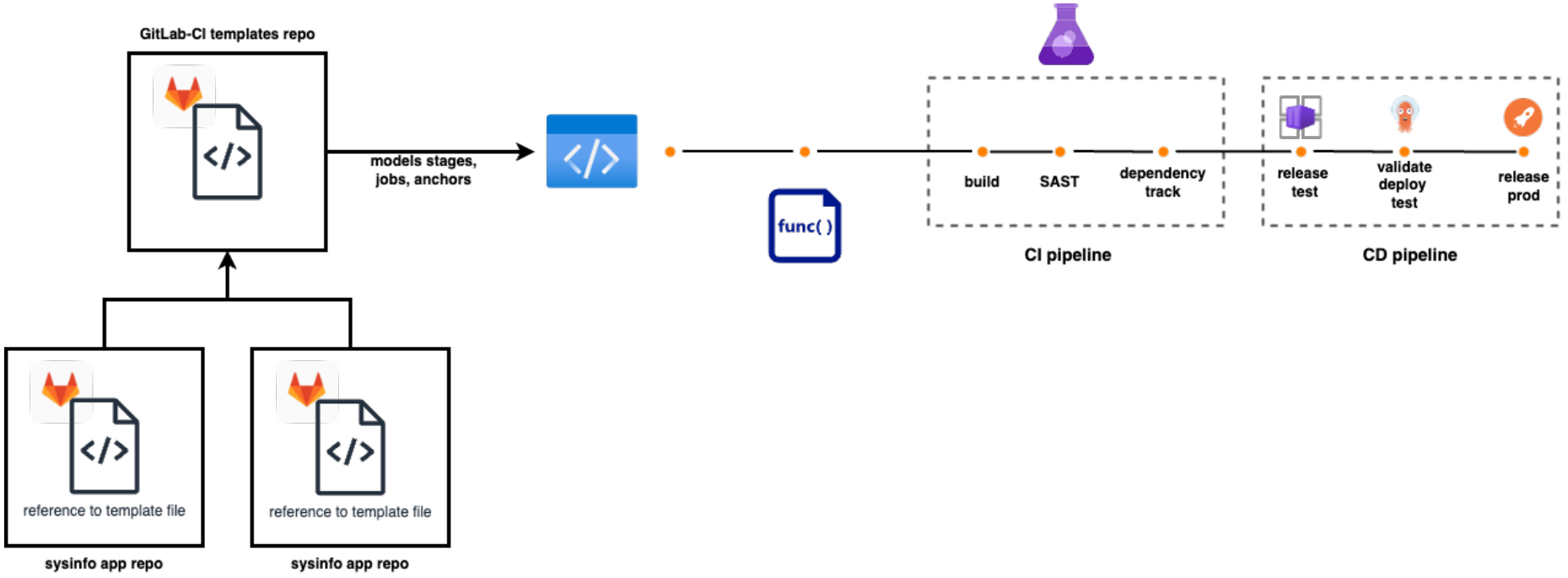
CICD onboarding



How to manage all the involved pipelines?

Pipelines governance and security

GitLab-CI templating is our engine to facilitate the CICD workflows standardization, accelerate the implementations of pipelines end reduce their duplicated code.



Sysinfo apps deployment in k8s

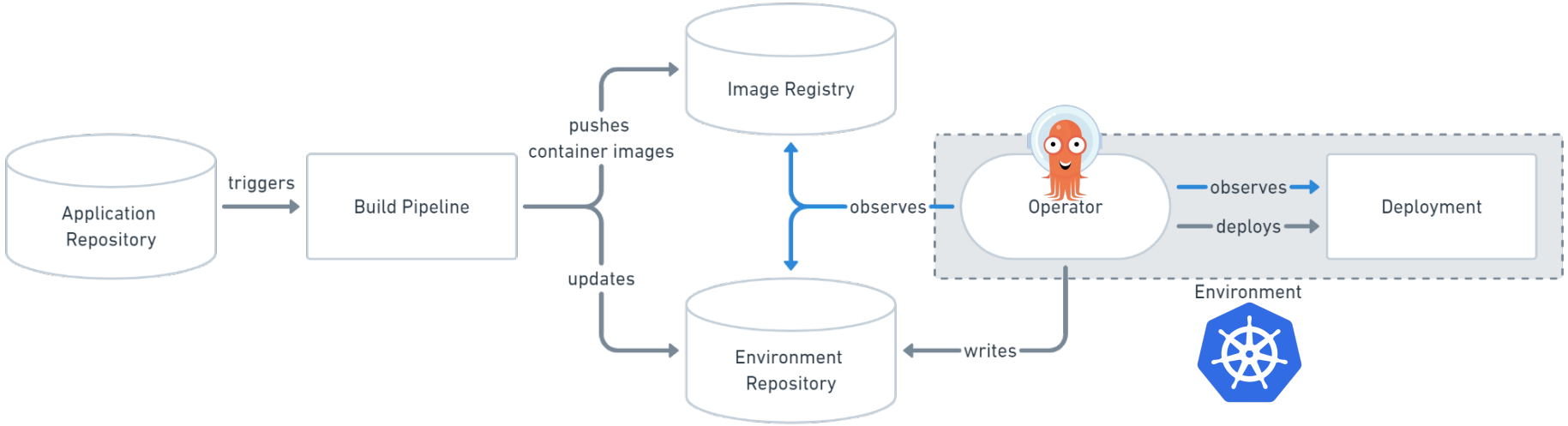
GitOps approach driven by ArgoCD tool.

- No k8s client to autorize – everything happens in the cluster (only admins has access to k8s clusters from outside).
- Apply what is declared in the repo (and the container registry) in the current deployment status to avoid drifts.

ArgoCD and GitOps

GitOps - versioned CICD on top of declarative infrastructure.

- A Git repository that always contains declarative descriptions of the desired infrastructure state.
- Infrastructure state versioned in Git.
- Automated processes to make the production environment match the described state in the repository.



Final considerations

- Shift left testing in the CICD pipelines integrates security in the DevOps process for the sysinfo apps (>> DevSecOps).
- Intercept and keep track of security risks in the early stages of the pipelines allow us to reduce risks and improve compliance.
- A formal agreement, between involved stakeholders, on how to develop, build, test, deliver and deploy software is paramount.
- Our GitOps approach provides governance over the sysinfo apps continuos deployment avoiding drifts.
- Shift right testing still crucial (producton is were real things happen). Don’t forget day-2 operations.
- Improving the developer experience with more self-service capabilities is our next challenge.