DSOMM

Metric Collector and Analyzer

Timo Pagel

/bin/whoami

- Freelance IT Security Architect/Trainer/Strategist
- Open Source/Knowledge Fan
- University Lecturer

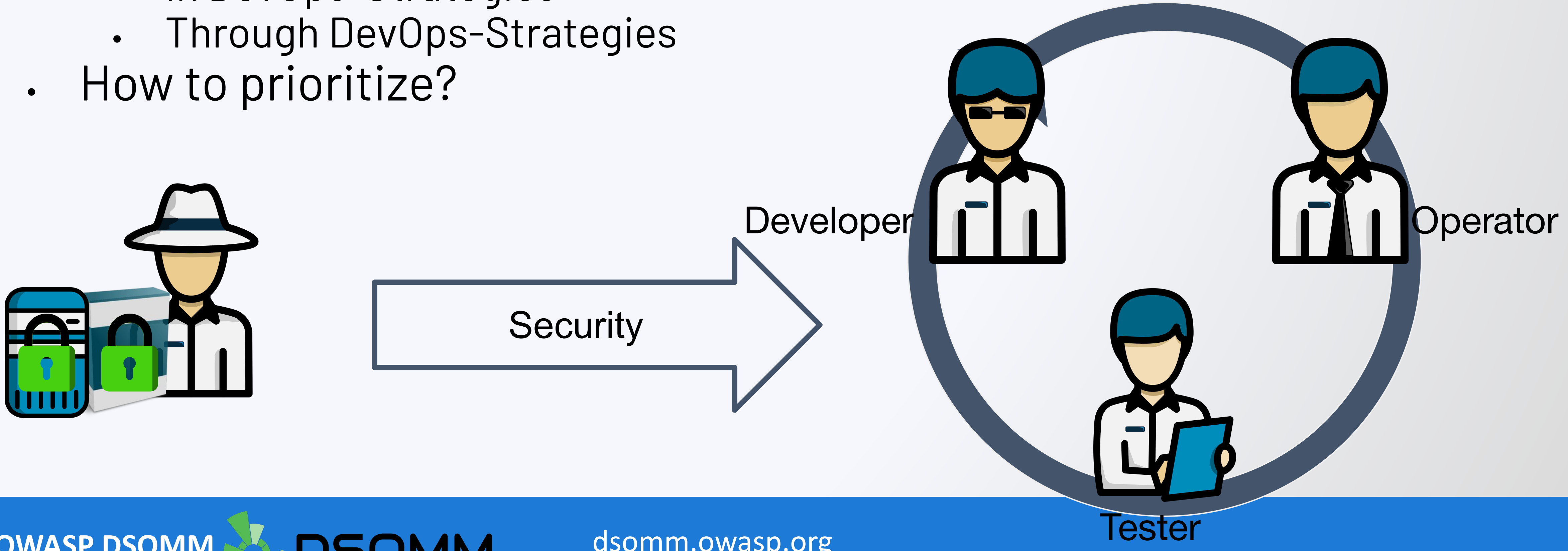
Timo Pagel

Agenda

- Introduction
- MetricaCA Architecture
- Outlook

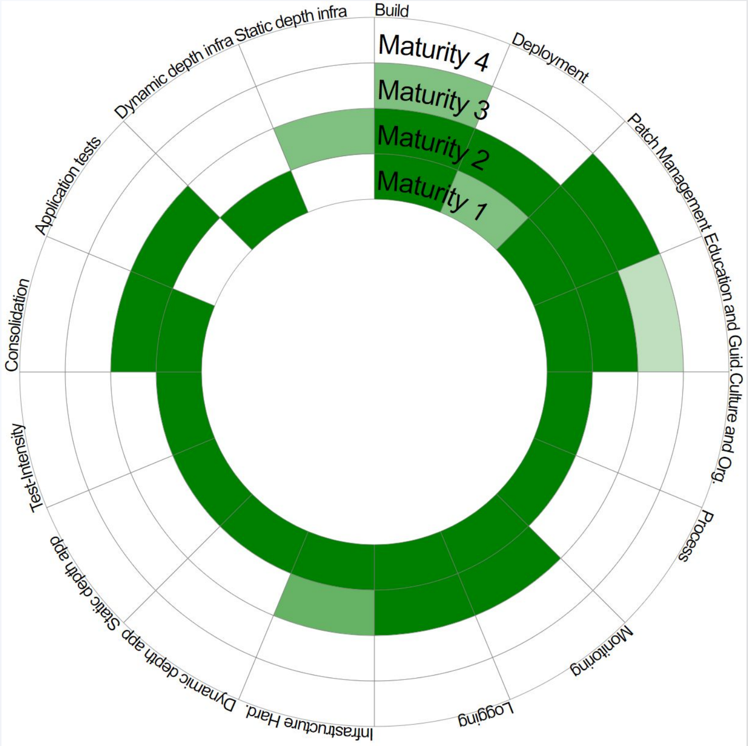
DevSecOps Maturity Models

- How to enhance security?
    - In DevOps-Strategies
    - Through DevOps-Strategies
- How to prioritize?



AppSec Programs

- analyze current software security practices
- plan security activities
- implement and measure improvements



DevSecOps Assessment

Assessments performed only quarterly/yearly/bi-yearly

As a product team, I want fast feedback for performed (or gone missing) security activities to stay motivated

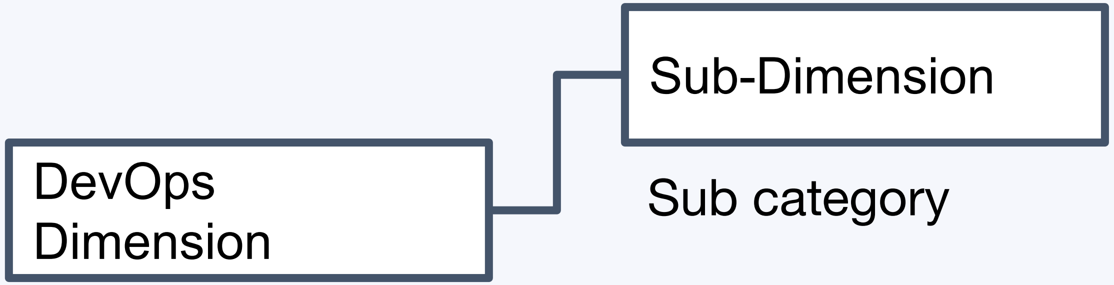
Solution: metricCA

DSOMM Structure

DevOps Dimension

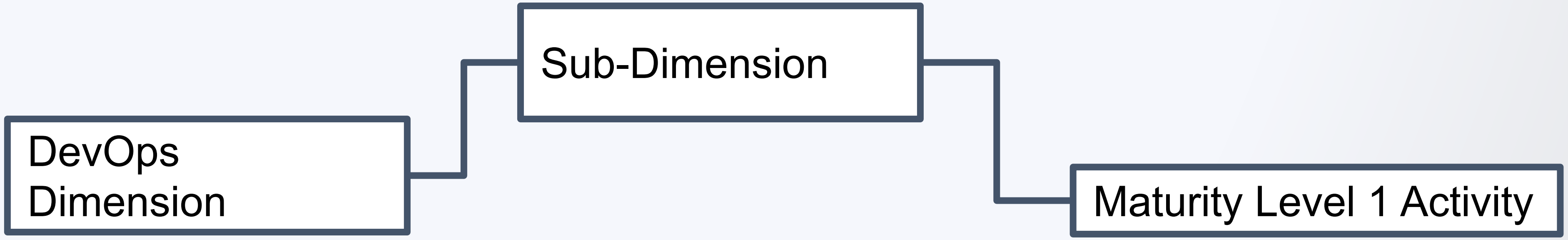
Category

DSOMM Structure



Sub category

DSOMM Structure



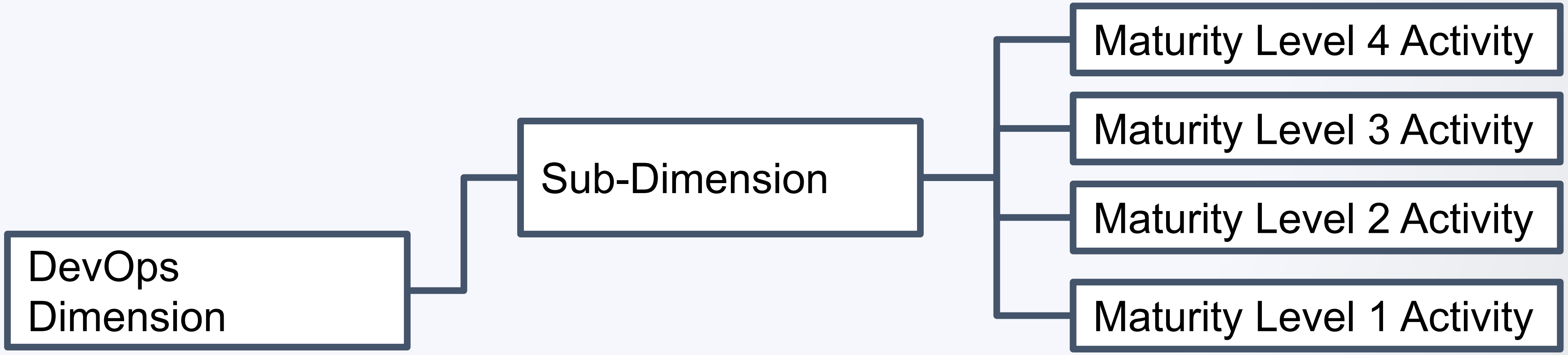
DevOps Dimension

Sub-Dimension

Maturity Level 1 Activity

Higher is better

DSOMM Structure



DevOps Dimension

Sub-Dimension

Maturity Level 4 Activity

Maturity Level 3 Activity

Maturity Level 2 Activity

Maturity Level 1 Activity

Higher is better

DevSecOps Dimensions

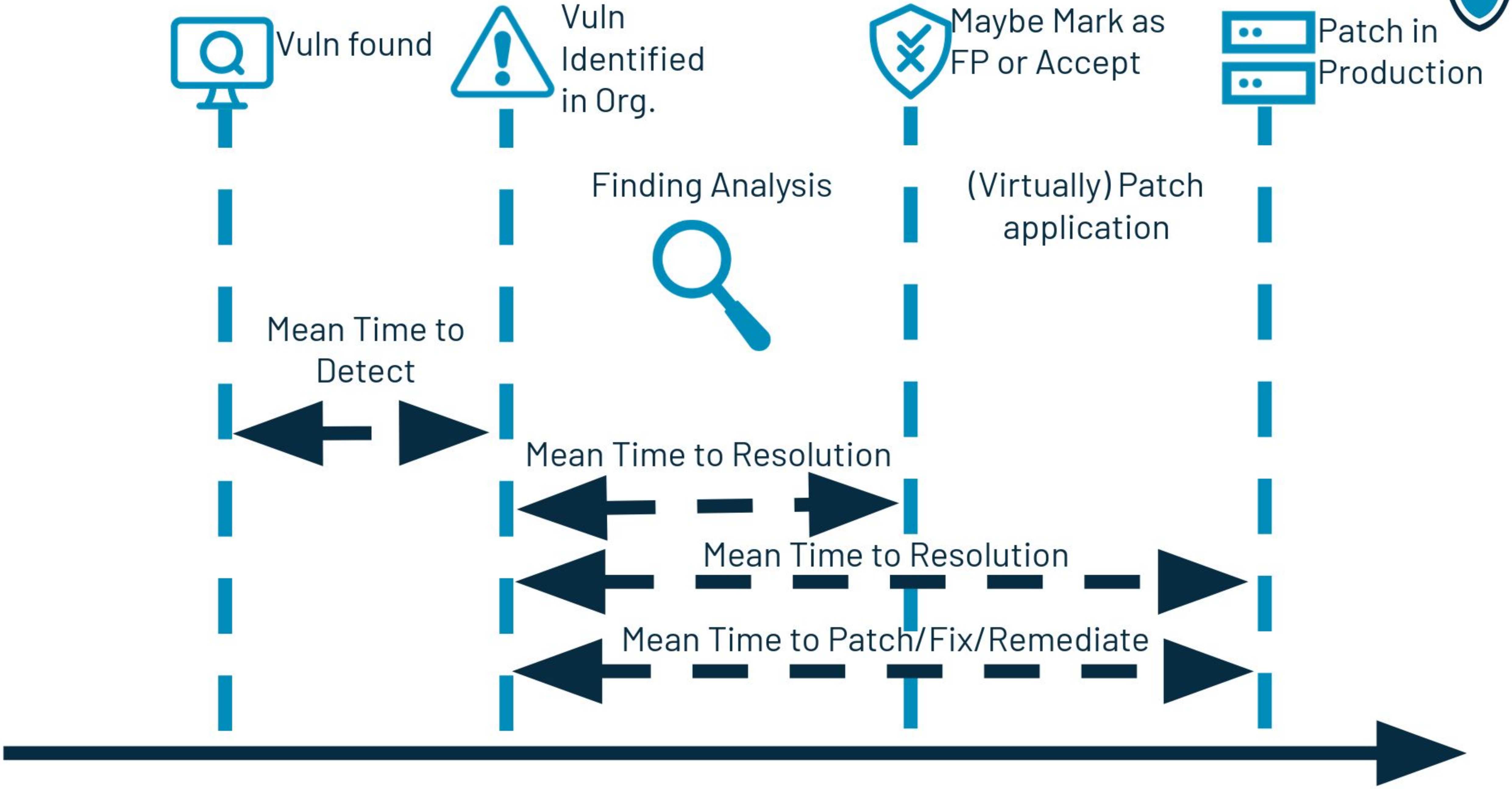
- Build and Deployment
- Culture and Organisation
- Information Gathering
- Hardening
- Test and Verification

Where to apply measurements? Culture and Organization

- Each team has a security champion
- Slack channel #security exchange rate per team
- Threat modeling frequency per team
- Threat modeling quality per team
- Creation of abuse stories in a requirements/planning tool
- Hours of security training per team

MTTR (Mean Time to Resolution) vs MTTF (Mean Time to Fix)

- **MTTR** measures how quickly issues are resolved from the user's perspective. It demonstrates that the **entire vulnerability management process** is functioning effectively. Additionally, MTTR provides insight into how **noisy the security tools** being used are, as frequent false positives can inflate this metric.
- While **MTTF** offers a comprehensive view, Mean Time to Fix represents the **final goal in addressing vulnerabilities** or issues.



Vuln found

Vuln Identified in Org.

Maybe Mark as FP or Accept

Patch in Production

Finding Analysis

(Virtually) Patch application

Mean Time to Detect

Mean Time to Resolution

Mean Time to Resolution

Mean Time to Patch/Fix/Remediate

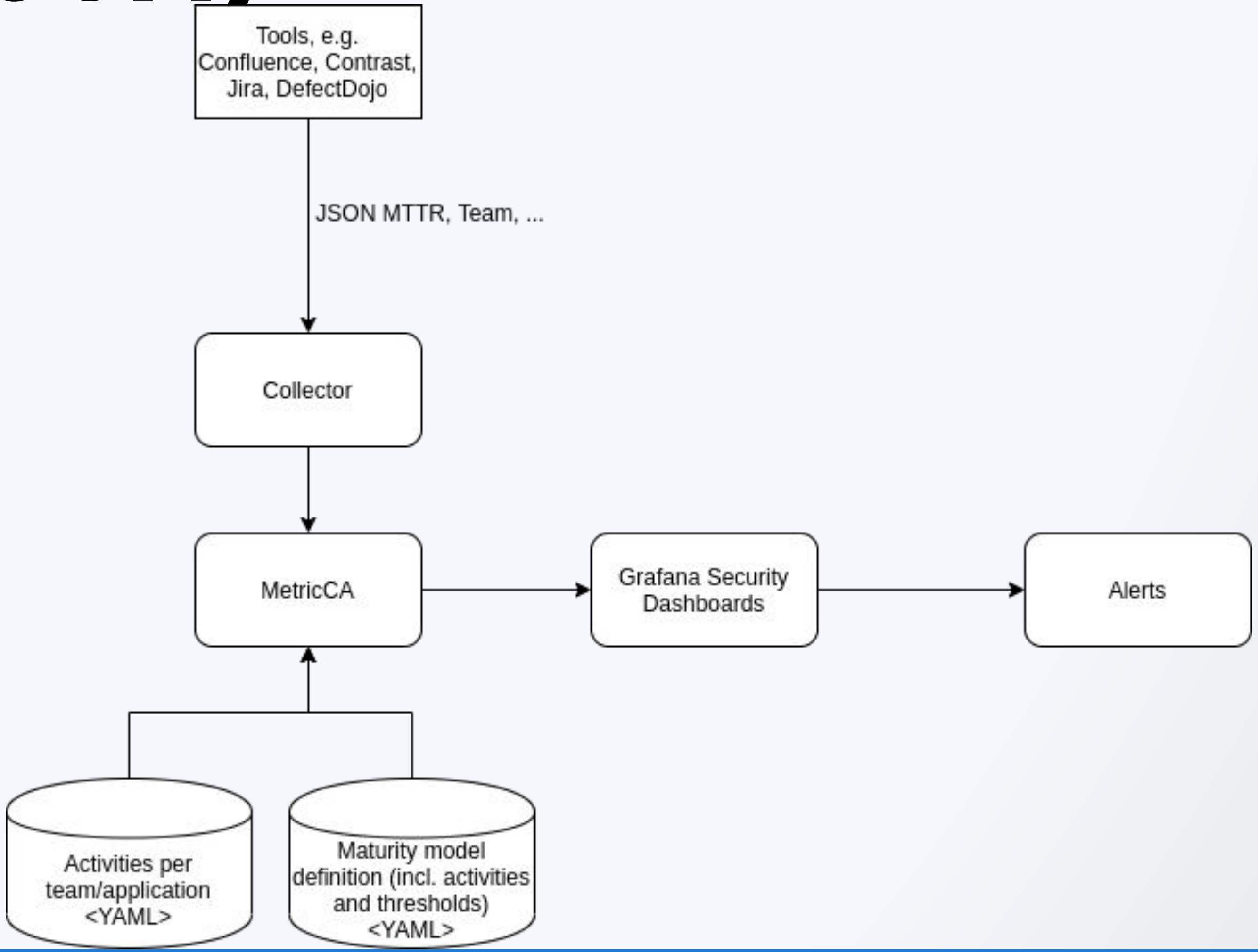
Agenda

- Introduction
- MetricaCA Architecture
- Outlook

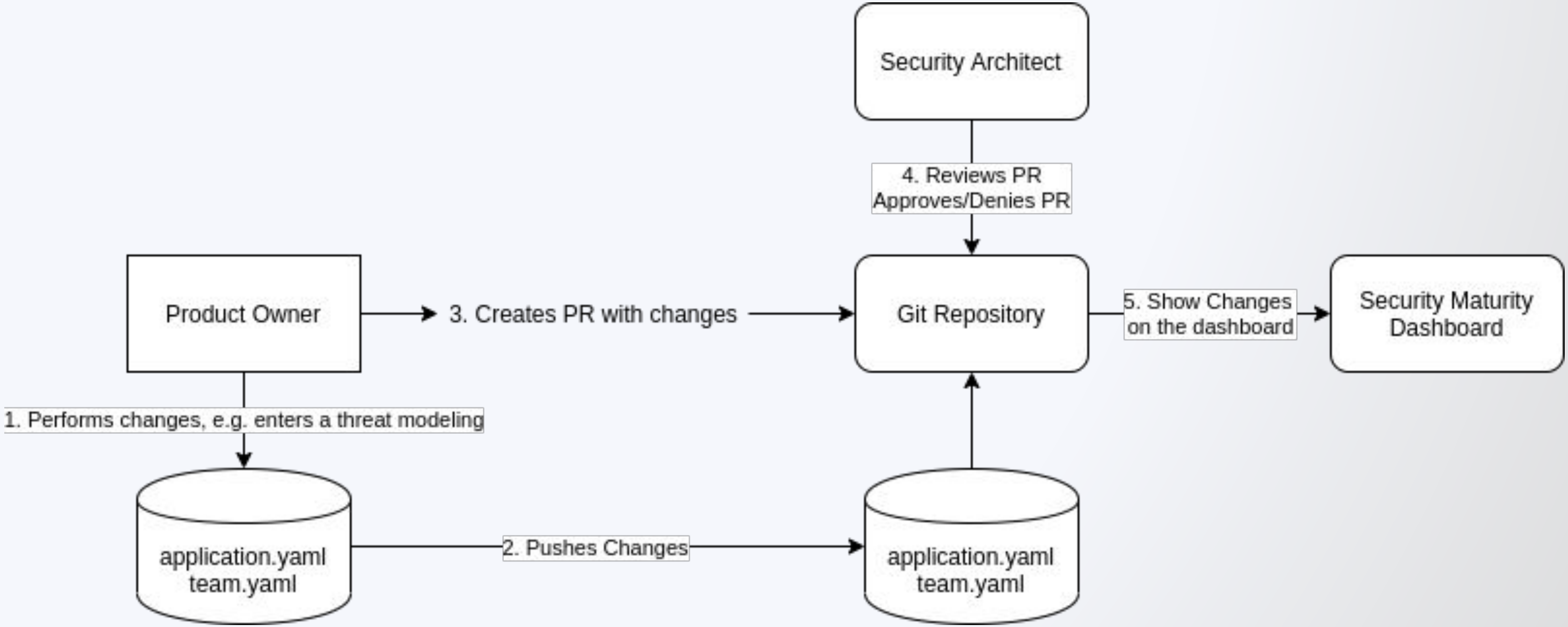
Sources

- Manual assessment placed in YAML
- Automatic assessment fetched from tools (e.g. jira, security tools, ...)

Metric Collector and Analyzer (MetricCA)


Tools, e.g. Confluence, Contrast, Jira, DefectDojo
JSON MTTR, Team, ...
Collector
MetricCA
Grafana Security Dashboards
Alerts
Activities per team/application <YAML>
Maturity model definition (incl. activities and thresholds) <YAML>

Manual Overview



Security Architect

4. Reviews PR Approves/Denies PR

Product Owner

3. Creates PR with changes

Git Repository

5. Show Changes on the dashboard

Security Maturity Dashboard

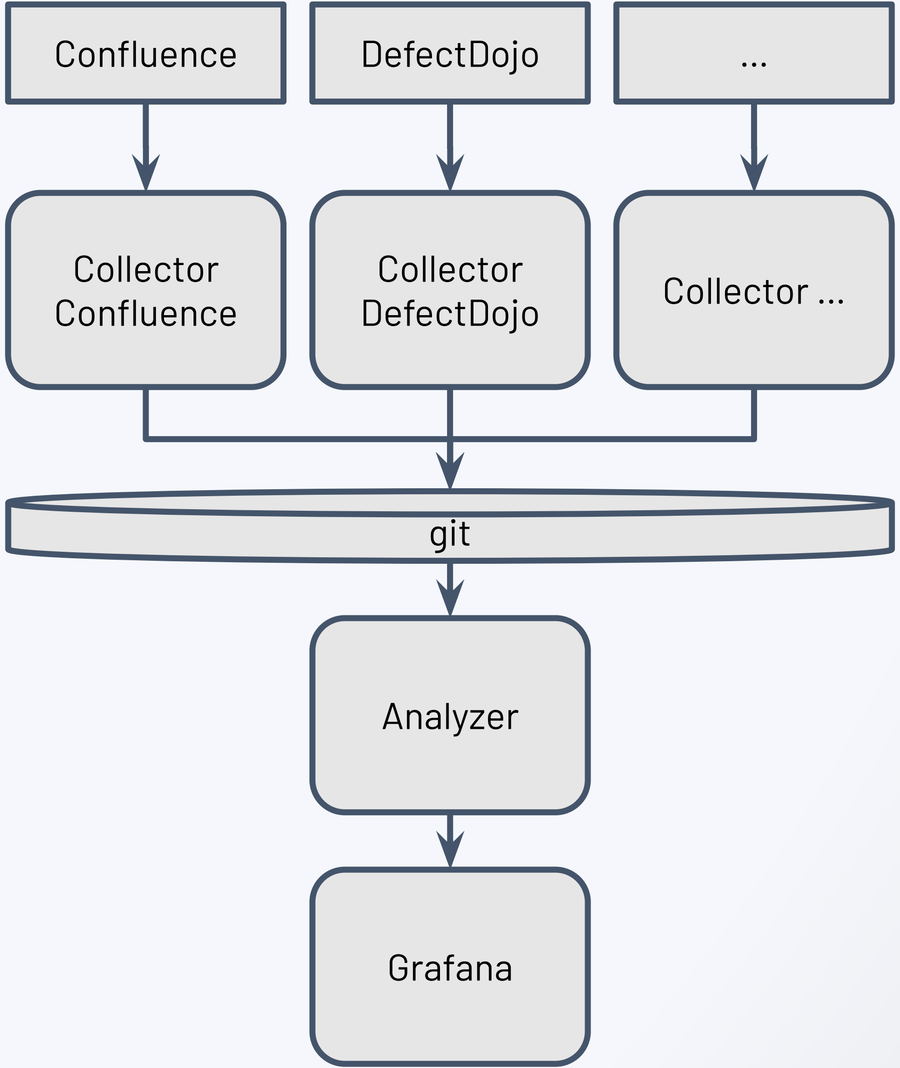
1. Performs changes, e.g. enters a threat modeling

application.yaml team.yaml

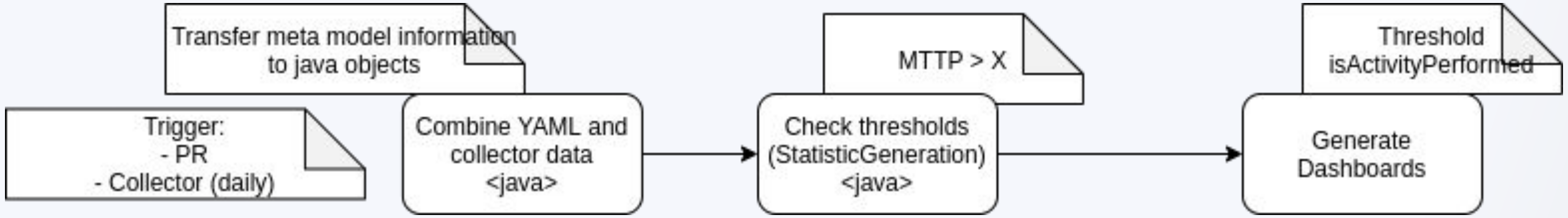
2. Pushes Changes

application.yaml team.yaml

Architecture as Dataflow



Threshold Flow



Transfer meta model information to java objects

Trigger:
- PR
- Collector (daily)

Combine YAML and collector data <java>

MTTP > X

Check thresholds (StatisticGeneration) <java>

Threshold isActivityPerformed

Generate Dashboards

YAML

Elastic definition of activities (e.g. threshold)

Per application/team:

- activities.yaml
- team.yaml

Generic:

- configuration.yaml:

configuration.yaml

```yaml
applicationId: X
activities:
  conduction of simple threat modeling on a technical level:
    level: Level 2
    # TODO: threshold
  data privacy requirements:
    level: Level 1
    # TODO: threshold
```

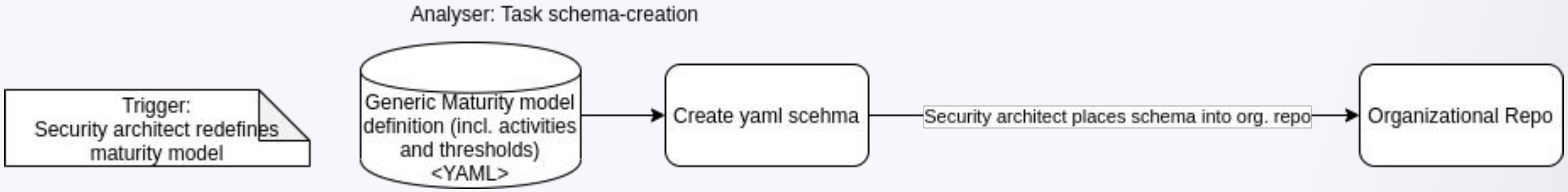
activities.yaml

```
applicationId: X

activities:
  conduction of simple threat modeling on a technical level:
        components:
        - title: string
        - conduction date: date

  data privacy requirements:
        components:
        - date: date
```

Analyzer Task: Schema Creation

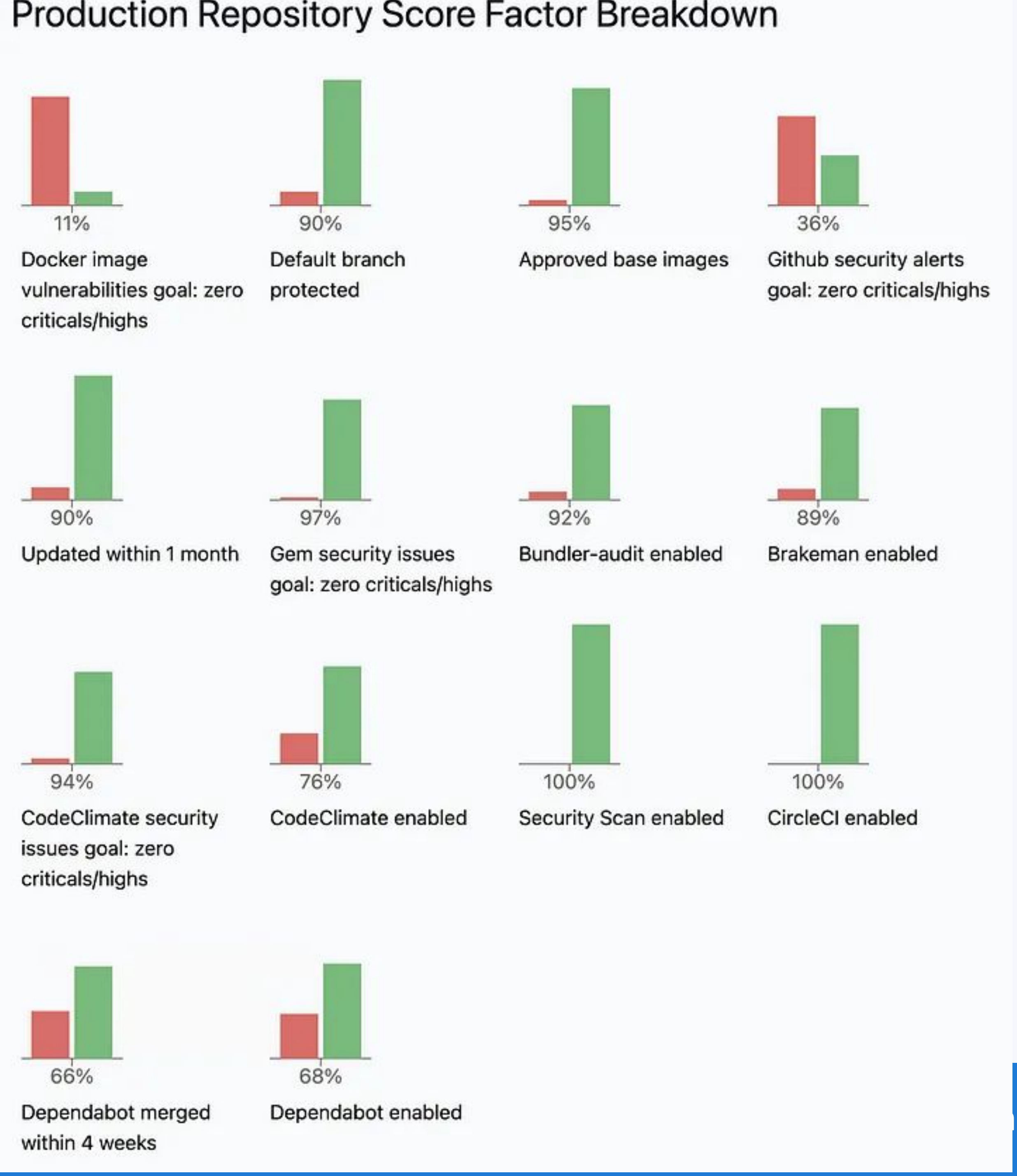

Analyser: Task schema-creation
Trigger:
Security architect redefines maturity model
Generic Maturity model definition (incl. activities and thresholds) <YAML>
Create yaml scehma
Security architect places schema into org. repo
Organizational Repo

Collectors

- **Confluence Collector**: Threat modelings and penetration tests by frequency (implemented)
- HTTP Collector: Collects last change of a website, e.g. github release page for last patch
- Github Collectors:
  - **Collects Security Settings like Branch Protection enabled**
  - Collects open time of automatic created patch pull requests (e.g. from renovate) to calculate Mean Time to Patch
- DefectDojo Collector: Collects Mean Time to Response
- Excel Collector: Collects penetration test by frequency
- Teams Collector: Collects attendee rate for security trainings

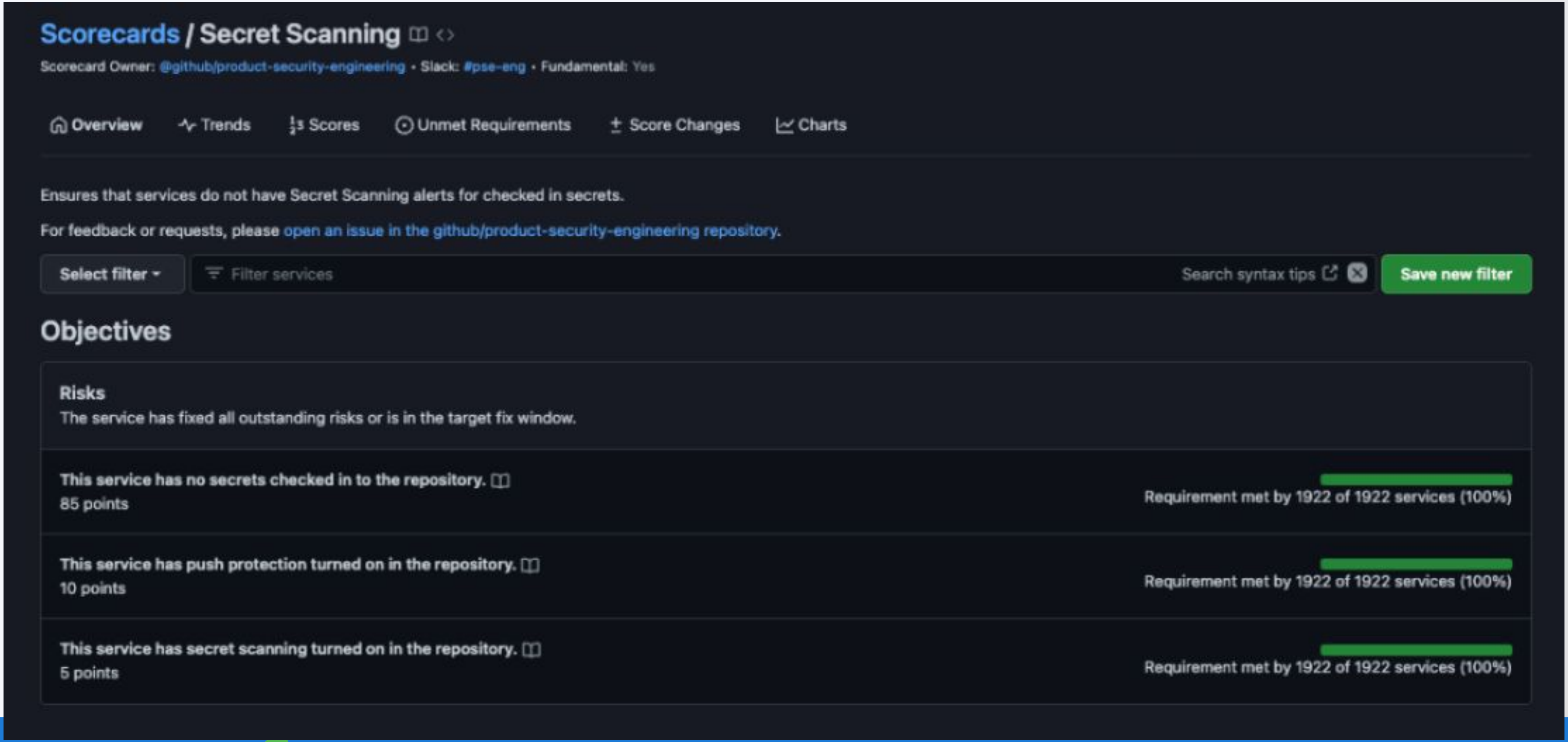
Demo

Other Examples



Production Repository Score Factor Breakdown

Source
https://medium.com/life-at-chime/monocle-how-chime-creates-a-proactive-security-engineering-culture-part-1-dedd3846127f



Scorecards / Secret Scanning

Scorecard Owner: @github/product-security-engineering • Slack: #pse-eng • Fundamental: Yes

Ensures that services do not have Secret Scanning alerts for checked in secrets.

For feedback or requests, please open an issue in the github/product-security-engineering repository.

Objectives

Risks
The service has fixed all outstanding risks or is in the target fix window.

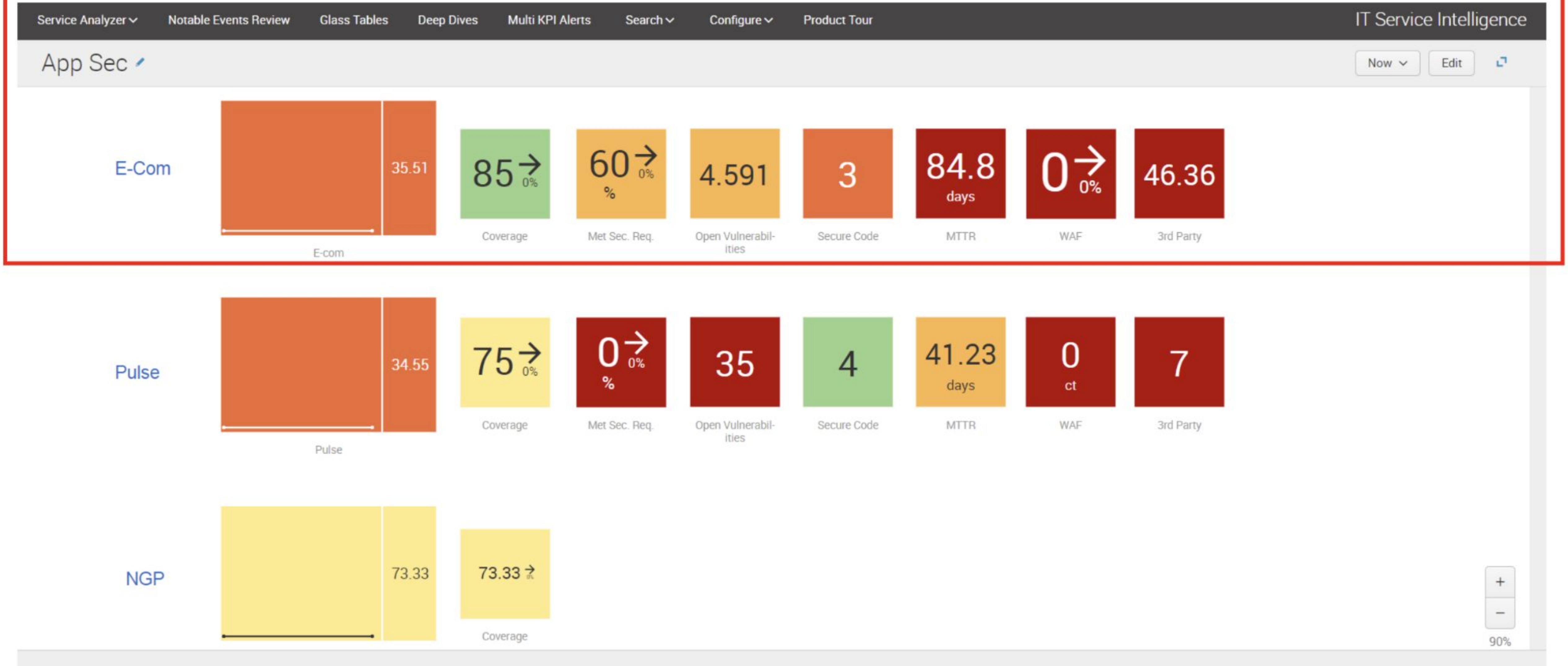
This service has no secrets checked in to the repository.
85 points
Requirement met by 1922 of 1922 services (100%)

This service has push protection turned on in the repository.
10 points
Requirement met by 1922 of 1922 services (100%)

This service has secret scanning turned on in the repository.
5 points
Requirement met by 1922 of 1922 services (100%)

Source: https://github.blog/2024-02-08-githubs-engineering-fundamentals-program-how-we-deliver-on-availability-security-and-accessibility/

AppSec Metrics Dashboard – Executive View



App Sec

		Coverage	Met Sec. Req.	Open Vulnerabil-ities	Secure Code	MTTR	WAF	3rd Party
E-Com	E-com 35.51	85 → 0%	60 → 0% %	4.591	3	84.8 days	0 → 0%	46.36
Pulse	Pulse 34.55	75 → 0%	0 → 0% %	35	4	41.23 days	0 ct	7
NGP	73.33	73.33 →						

https://www.iottechexpo.com/northamerica/wp-content/uploads/2018/12/PodHandler.pdf

Silverbullet?
People and Processes

- Still need to perform activities
- For manual: Need to update YAMLs and understand why

Sec. Dashboard Refs.

https://medium.com/life-at-chime/monocle-how-chime-creates-a-proactive-security-engineering-culture-part-1-dedd3846127f

https://www.youtube.com/watch?v=e6k7DpXTtWA

https://github.blog/2024-02-08-githubs-engineering-fundamentals-program-how-we-deliver-on-availability-security-and-accessibility/

https://www.iottechexpo.com/northamerica/wp-content/uploads/2018/12/PodHandler.pdf

https://tldrsec.com/p/blog-insecure-development-why-some-product-teams-are-great-and-others-arent

https://medium.com/uber-security-privacy/uber-bug-bounty-promotions-1ed7648cf6b0

https://tldrsec.com/p/appsec-a-pragmatic-approach-for-internal-security-partnerships

MetricCA Ref

Documentation:
https://github.com/devsecopsmaturitymodel/metricCA

Main App:

https://github.com/devsecopsmaturitymodel/metricAnalyzer

Collectors:

https://github.com/devsecopsmaturitymodel/collector-confluence

https://github.com/devsecopsmaturitymodel/collector-github

DSOMM User Day

25 September

OWASP Global AppSec

San Francisco

Questions?

Timo's Website

Timo Pagel

Contact: timo.pagel@owasp.org

Business Contact: dsomm@pagel.pro

Business Website AppSec: https://appsec-program.com/

Business Website: https://pagel.pro

Join our community in the OWASP Slack in channel #dsomm

OWASP™