ORT Server
Progress Update & Demo

Martin Nonnenmacher
09.04.2025

BOSCH

01

## Overview

Motivation
Building a scalable ORT service

Scalability	REST API	Database	Credentials
Decouple from limitations of CI Pipelines	REST API eases integration with other services	Central database for all ORT results	Credential management via API
ORT is currently mainly deployed using CI Pipelines like Jenkins, GitHub Actions, or Azure Pipelines. This limits scaling options as these pipelines are not designed to run a service.	The REST API (with OpenAPI Spec) allows to integrate the ORT Server with other services. For example, the CI integrations can simply call the API instead of implementing a complex ORT workflow.	Instead of separate ORT result files for each run, the server stores all results in a central database. This enables performance optimizations and data analysis, for example, to find out which projects use a specific dependency.	Running a central ORT service also requires to manage the required credentials for analyzing the projects. With ORT Server, users can manage their credentials via the REST API.
✔ 01	✔ 02	✔ 03	✔ 04

BOSCH

02

Architecture

BOSCH

Architecture
Components

22

BOSCH
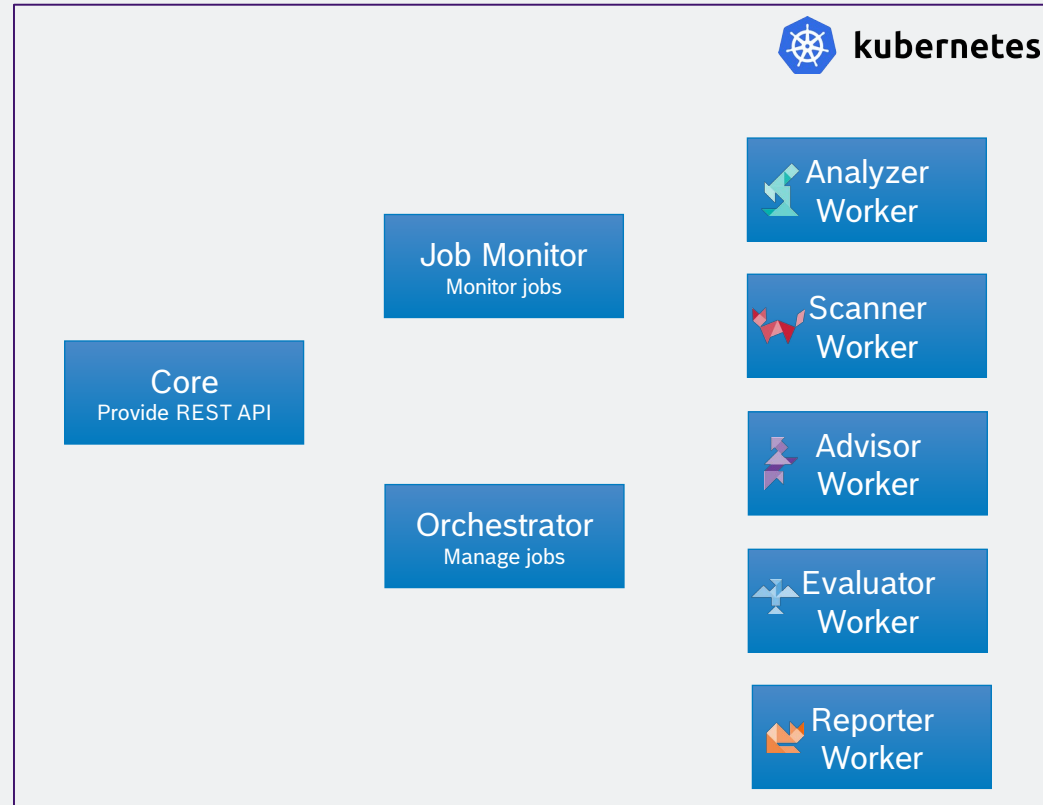
03

Progress Update

BOSCH

Progress Update
Documentation Website

https://eclipse-apoapsis.github.io/ort-server/
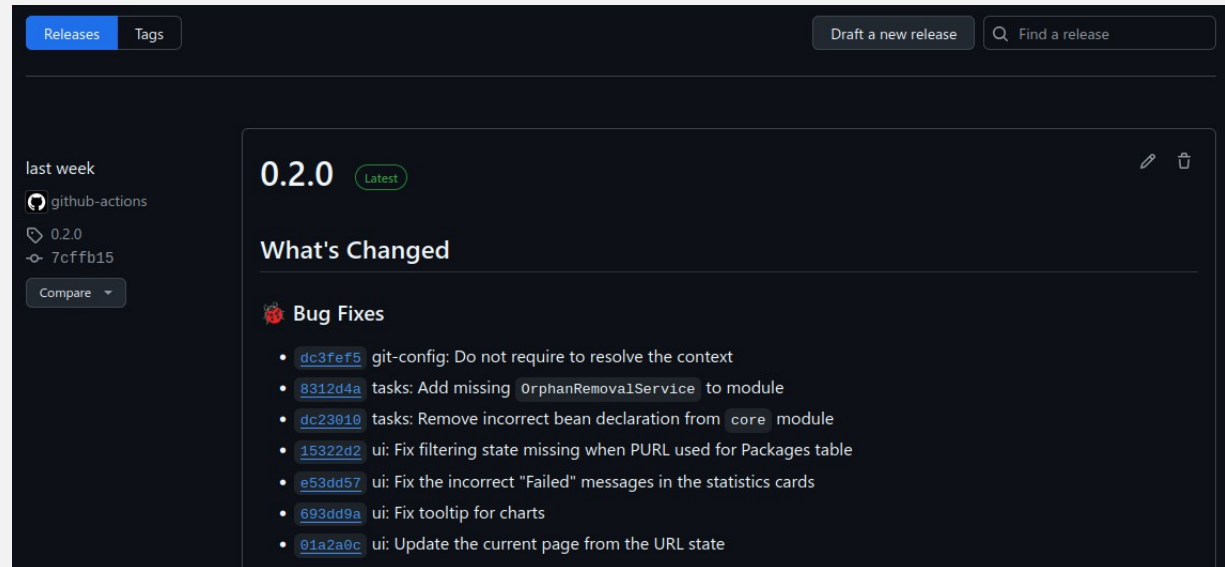
22

BOSCH
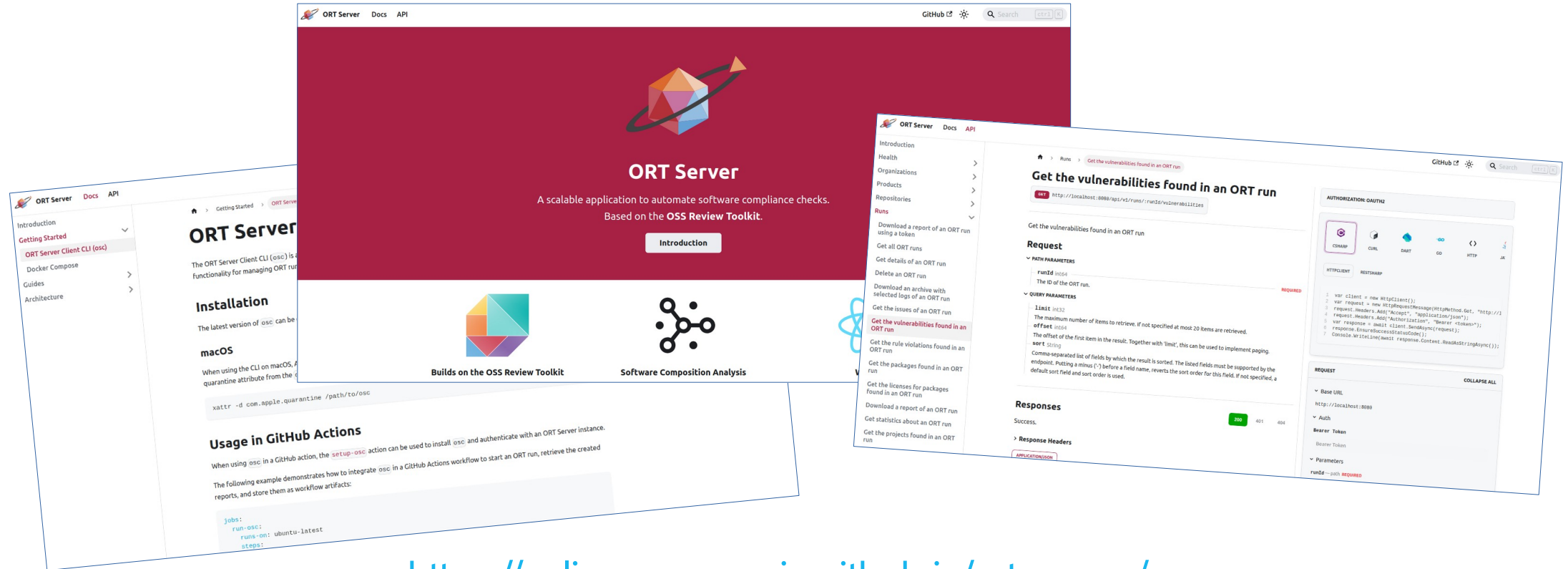
Progress Update
Passed Eclipse Foundation Progress Review

- The project passed the Eclipse Foundation progress review and can now make official releases
- Weekly releases on Tuesday, integrating the ORT release from the week before
- Expect breaking changes until release 1.0.0, current version is 0.2.0

BOSCH

Progress Update
Data Retention

- Automatic deletion of ORT runs after configurable time

    - For example, delete all runs older than 90 days

    - Ensure that database does not keep growing

- Will be improved for more granular control

    - For example, to always keep the results for releases

22

BOSCH

Progress Update
CLI & CI Integration

- Added a native CLI for Linux, Mac, and Windows

  - Trigger and get status of ORT runs

  - Download logs and reports

- Added a GitHub action to setup the CLI

  - Makes it easy to write GitHub workflows that interact with the server

```yaml
jobs:
  run-osc:
    runs-on: ubuntu-latest
    steps:
      - name: Setup OSC
        uses: eclipse-apoapsis/setup-osc@main
        with:
          osc-version: 0.1.0-RC16
          base-url: https://ort-server.example.com
          token-url: https://auth.example.com/realms/prod/protocol/openid-connect/token
          client-id: ort-server-prod
          username: user
          password: ${{ secrets.ORT_SERVER_PASSWORD }}

      - name: Start ORT run
        run: |
          osc runs start --parameters '{
            "revision": "${{ github.head_ref || github.ref_name }}",
            "jobConfigs": {
              "analyzer": {},
              "advisor": {},
              "evaluator": {},
              "scanner": {},
              "reporter": {},
              "notifier": {}
            }
          }'
        env:
          OSC_REPOSITORY_ID: 42
          OSC_RUNS_START_WAIT: true # Block the runner until the

      - name: Download Reports
        run: |
          osc runs download reports
        env:
          OSC_DOWNLOAD_REPORTS_FILE_NAMES: 'scan-report-web-app.
          OSC_DOWNLOAD_REPORTS_OUTPUT_DIR: './reports'
```

```
) osc runs info --run-id=154034
ORT Run Information
  ID                  154034
  Repository ID       1321
  Index               1375
  Status              FINISHED_WITH_ISSUES
  Created             2025-03-31T08:26:13Z
  Revision            main
  User                Marcel Bochtler (          )

Job Status
  Analyzer Status     FINISHED
  Advisor Status      FINISHED
  Scanner Status      FINISHED
  Evaluator Status    FINISHED_WITH_ISSUES
  Reporter Status     FINISHED
  Notifier Status     FINISHED
```

BOSCH

Progress Update
Web Frontend

Demo

BOSCH

04

Questions?

BOSCH