Abstracting a Snippet Scanner in a Multi-Company Setup with ORT

Vladimir Slavov

Nicolas Nobelis

ORT Community Days, Berlin, March 2024

BOSCH

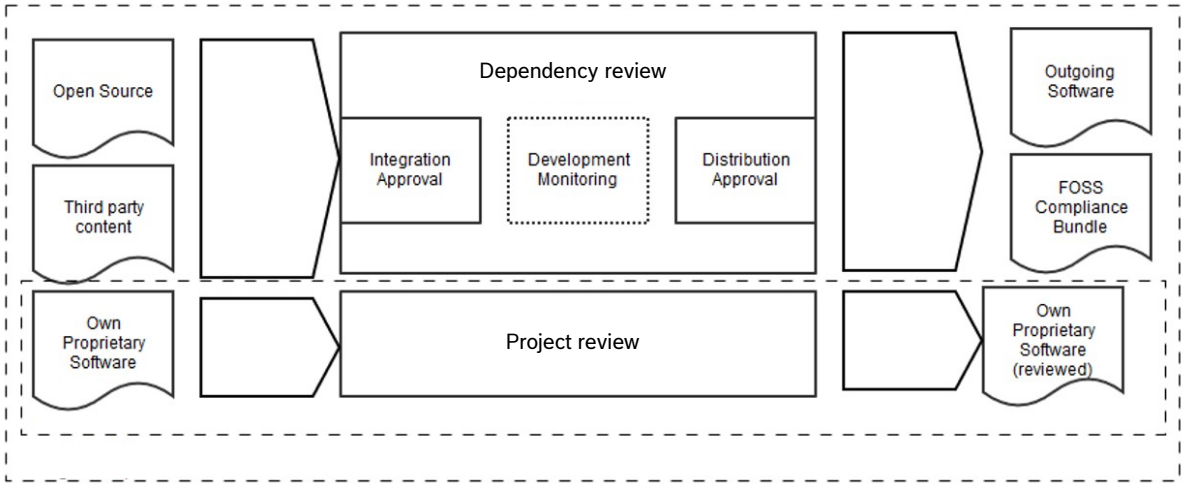
Overview

1. Typical OSM Setup
2. What changed in the joint project
3. Effects of Snippet Choice via Configuration-as-Code
4. Technical presentation

BOSCH

Typical OSM Setup

- Project Review
  - Only done on own code
  - Performed via snippet scanner (e.g. FossID, ScanOSS)
  - Expectation: no real snippet findings
- Dependency Review
  - Only done on dependencies
  - Performed via a dependency scanner (e.g. ORT Scanner)
  - Expectation: all open source introduced as dependencies



Challenges in the multi-company project
Problem

- No possibility to remotely use a snippet scanner via a UI
  → Snippet handling work should be non-interactive
- Snippet handling needs to be reproducible outside of the joint-project
  → Configuration of snippet handling should be accessible locally

Challenges in the multi-company project
The ort.yml file

Repository Configuration `(.ort.yml)`

The items below can be configured by adding an `.ort.yml` file to the root of the source code repository. All configurations in this file apply only to this Project's context. Usually the global context is preferred for an increased degree of automation and local configurations should only be done if there are good reasons.

- excludes - Mark files, directories or package manager scopes as not included in released artifacts.
- curations - Overwrite package metadata, set a concluded license or correct license findings.
- resolutions - Resolve any issues or policy rule violations.
- license choices - Select a license for packages which offer a license choice.

Source: https://oss-review-toolkit.org/ort/docs/configuration/ort-yml

Challenges in the multi-company project
Solution

- Wrap away the snippet scanner and use ORT as the interface
- Perform the snippet handling work via Configuration-as-Code in the .ort.yml file

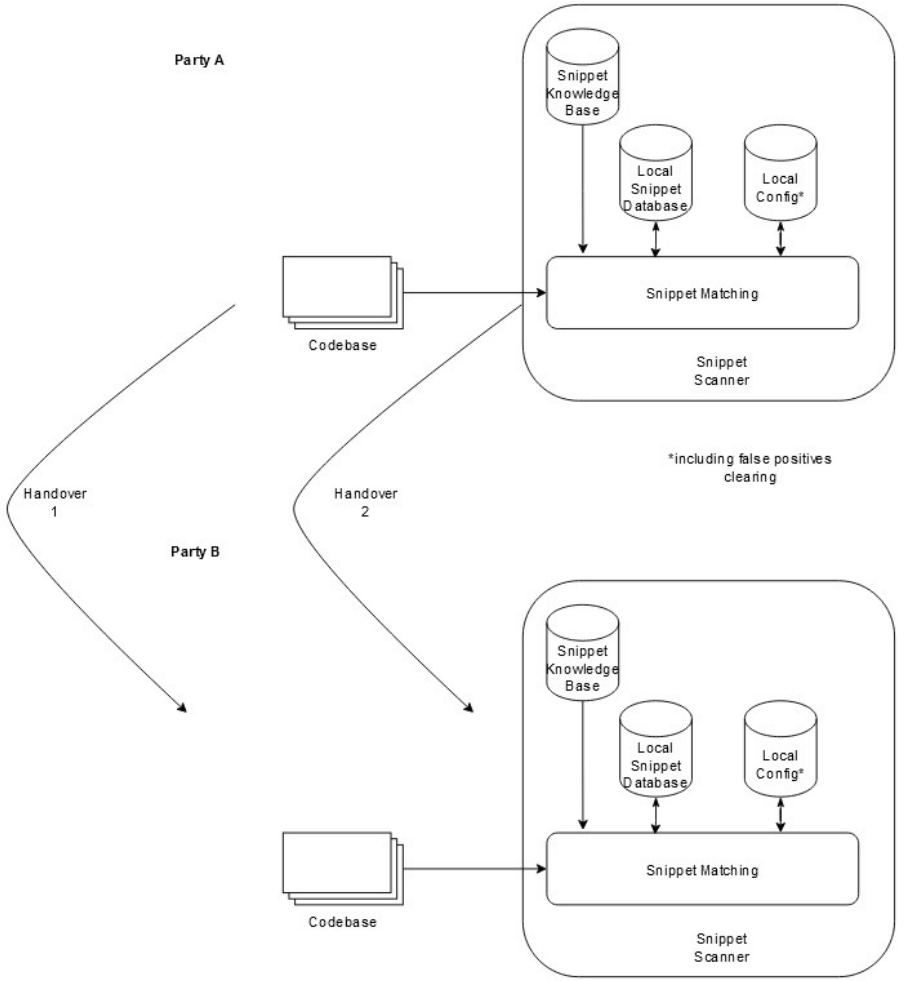
Effects of doing snippet choice via .ort.yml

- All the benefits of Config as Code
  - Flexibility
  - Version control
  - Reuse
  - etc.
- Abstracting away and decoupling from the underlying snippet scanner → avoid vendor lock-in
- Data stored locally

Effects of doing snippet choice via .ort.yml

Supply chain handover

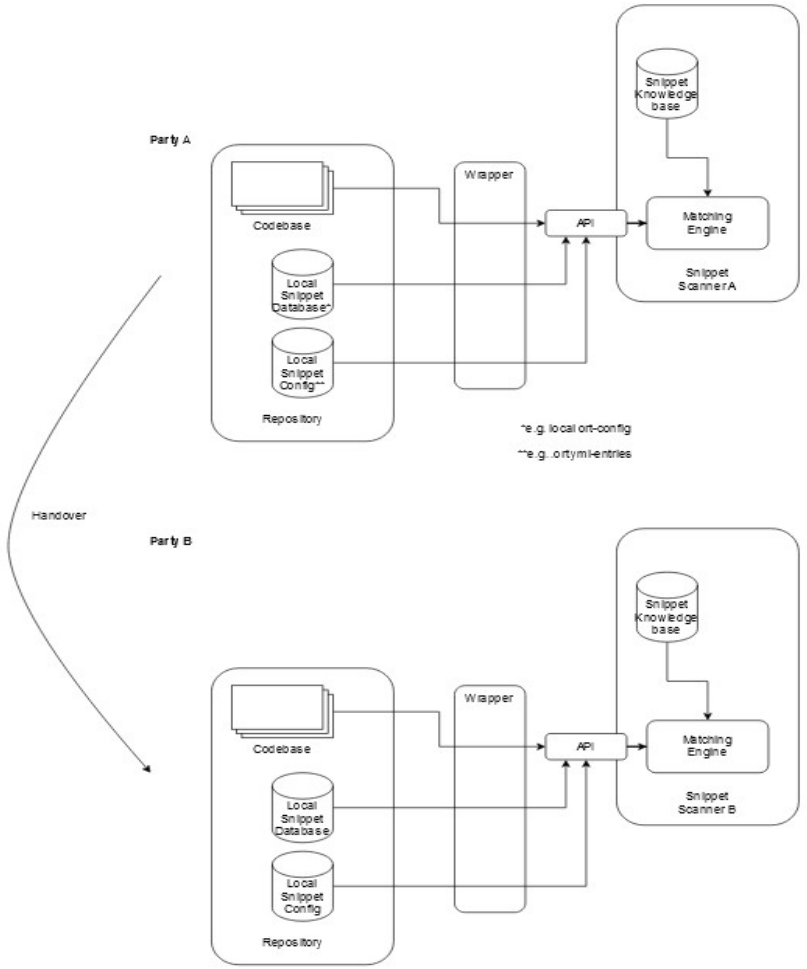
Classic Approach



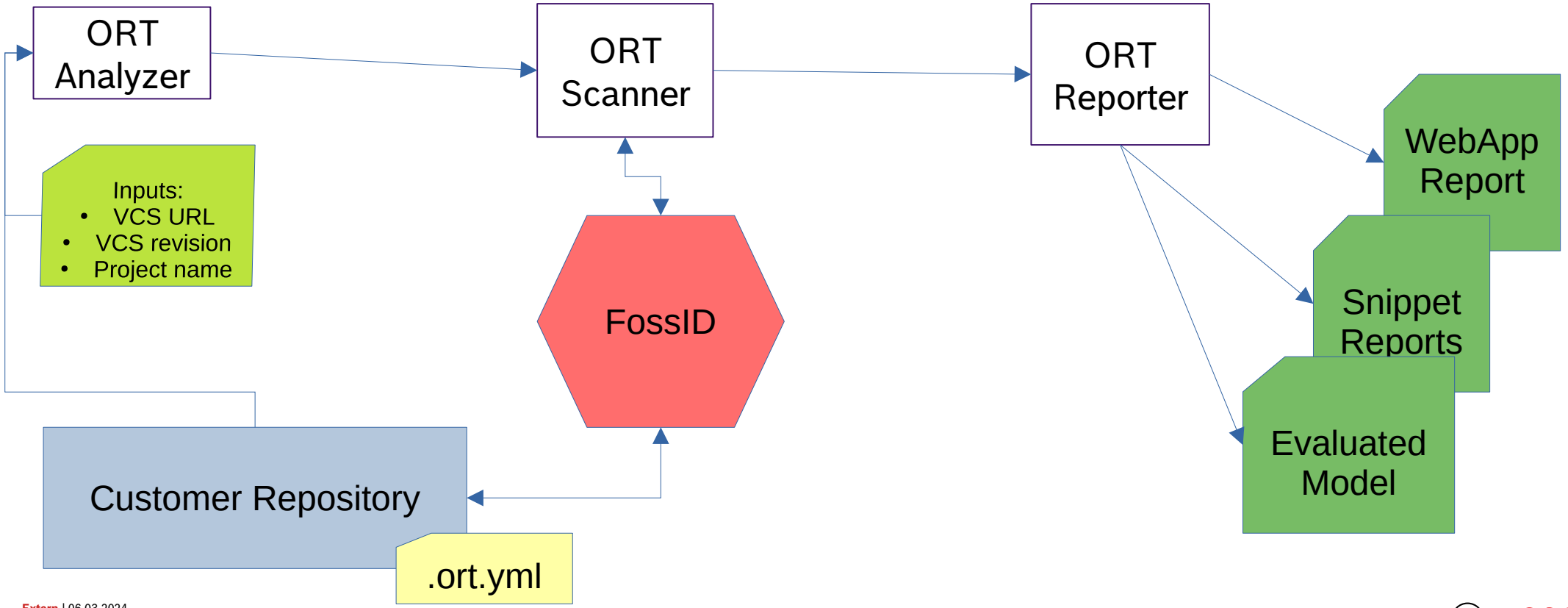
Effects of doing snippet choice via .ort.yml

Supply chain handover

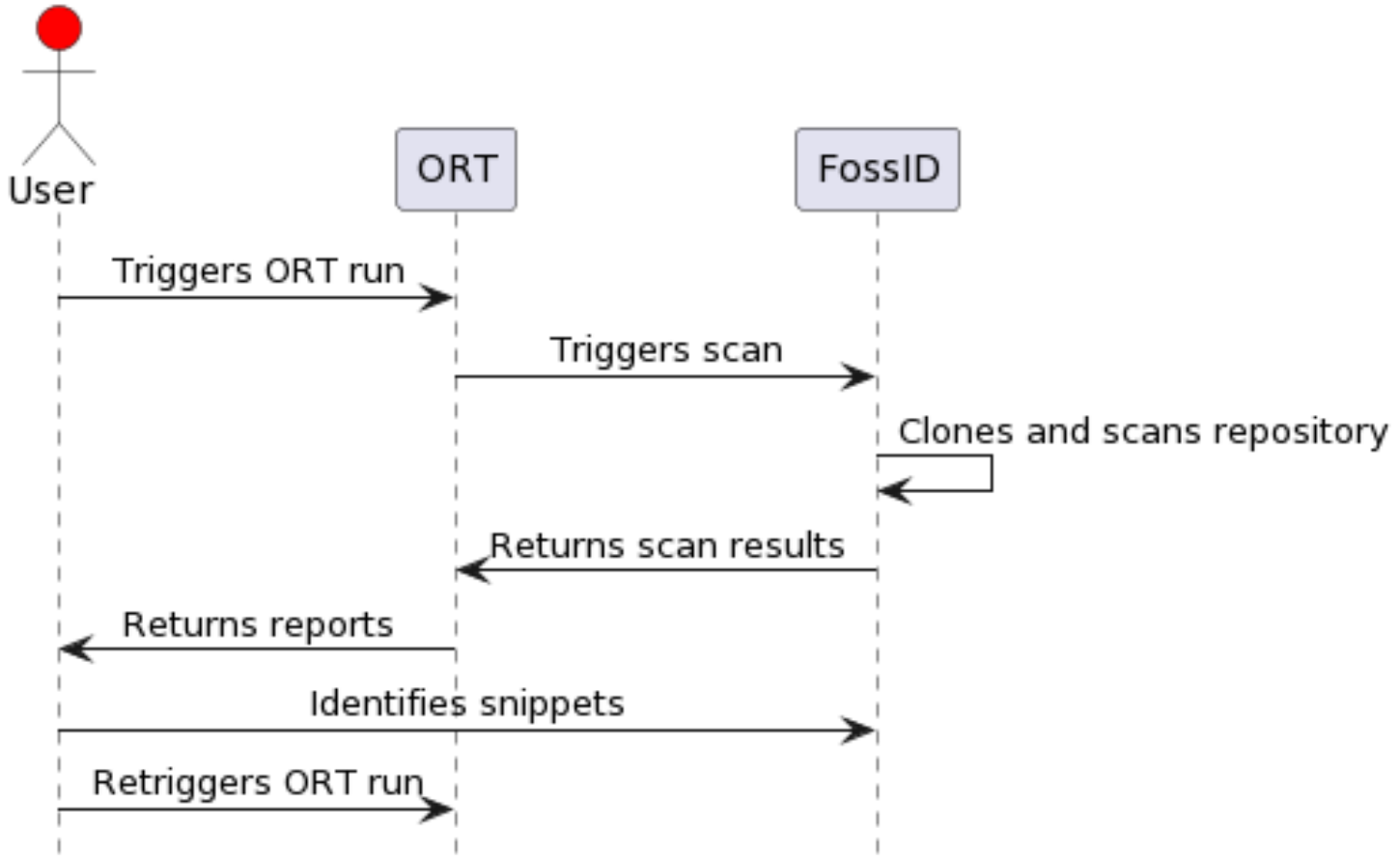
Future Approach



The standard workflow



The standard workflow (cont'd)



FossID UI

VIDEO

ORT Snippets report

src/test/java/com/vdurmont/semver4j/TokenizerTest.java
License(s): MIT, Apache-2.0

Source Location	pURL	License	File	URL	Score	Release Date
Full match	pkg:github/vdurmont/semver4j@3.1.0	MIT	src/test/java/com/vdurmont/semver4j/TokenizerTest.java	URL	1	
	pkg:maven/org.infrastructurebuilder.usurped/semver4j@3.2.0.3	Apache-2.0	com/vdurmont/semver4j/TokenizerTest.java	URL	0.27	2021-03-11

.github/workflows/ci.yml
License(s): MIT, NOASSERTION

Source Location	pURL	License	File	URL	Score	Release Date
Partial match 3-17	pkg:github/v6b/v6@0f154e1ca2d22fc1d3694c794339db13fbf9c80a	MIT	rocket-booster-template/github/workflows/node.yml	URL	0.93	2022-10-05
	pkg:github/RS2007/dotfiles@0384a21038fd2e5befb429d0ca52384172607a6d	NOASSERTION	dot_config/nvim/autoload/plugged/vim-devicons/dot_github/workflows/vint.yml	URL	0.93	2022-09-01
	pkg:github/stianfro/dotfiles@b371008f262377599edac1c8ea23ef53da82f832	NOASSERTION	private_dot_config/nvim/plugged/vim-devicons/	URL	0.93	2022-12-20

Summary

Problems:

- FossID UI is not reachable
- Process must be non-interactive

Solution:

- Drive the snippet identification with the .ort.yml

Snippet choice in the .ort.yml

Choose a snippet:

```yaml
package_snippet_choices:
- provenance:
    url: "https://github.com/vdurmont/semver4j.git"
  choices:
  - given:
      source_location:
        path: ".github/workflows/ci.yml"
        start_line: 3
        end_line: 17
    choice:
      purl: "pkg:github/RS2007/dotfiles@0384a21038fd2e5befb429d0ca52384172607a6d"
      reason: "ORIGINAL_FINDING"
      comment: "Explain why this snippet choice was made"
```

Consequences:

- The license of the chosen snippet is added to the license findings
- Other snippets from the same source location are hidden from the snippet reports

Snippet choice in the .ort.yml (cont'd)

Mark a source location as having only false positives:

```yaml
package_snippet_choices:
- provenance:
    url: "https://github.com/vdurmont/semver4j.git"
  choices:
  - given:
      source_location:
        path: "CHANGELOG.md"
        start_line: 2
        end_line: 5
    choice:
      reason: "NO_RELEVANT_FINDING"
      comment: "Explain why this location has only false positives snippets"
```

Consequences:

- All snippets from this source location are hidden from the snippet reports

Under the hood

- What happen when the snippet choice is deleted from the .ort.yml ?
- What happened when a snippet is chosen and new snippets appears afterwards for the same source location ?
- What happen when the source code has changed and the snippet choice is not valid anymore ?