OSS Review Toolkit

at COOL COMPACT OSADL ONLINE LECTURES on April 26, 2003

presented by Sebastian Schuberth (Founder of ORT, IT Consultant)

About Myself

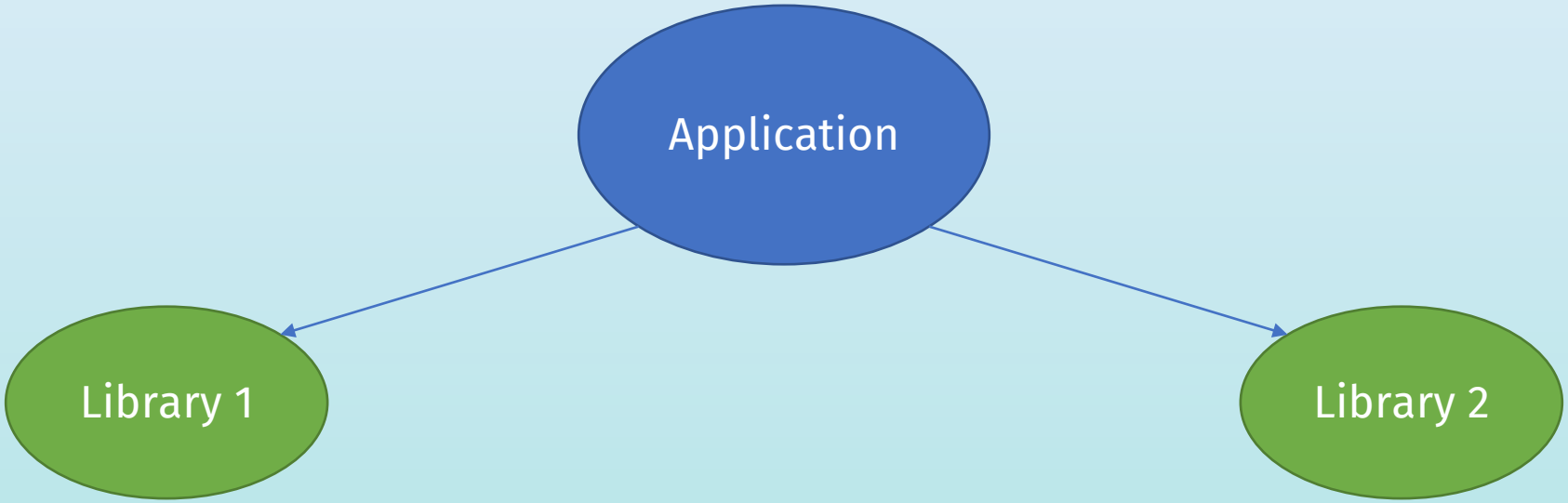
- Master of Computer Science (German Diploma)
  - Studied at TU Braunschweig and FU Berlin
  - Focus on Computer Graphics, Robotics, and Cryptology
- Open Source enthusiast
  - Started with fixing bugs in software that I use
  - Contributing to Open Source for decades
    - ~ 3400 GitHub contributions / year
  - Wide network of trust
    - Reviewing code in about a dozen GitHub organizations

About Myself (continued)

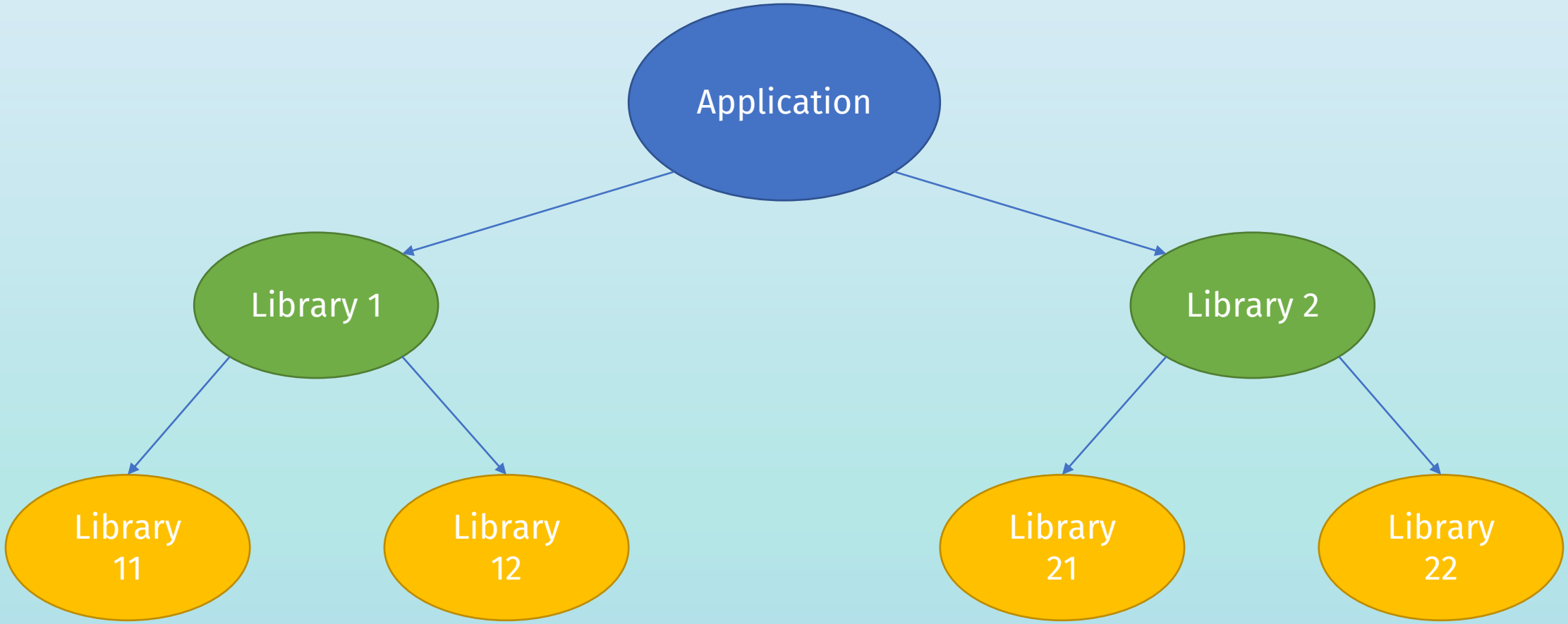
- Long experience in developing proprietary software
  - From high-performance computing to low-power mobile apps
- Automation “freak”
  - Configuration as Code instead of GUIs
- Motivation
  - Take out the process pain of software development
  - Bridge the gap between Open Source and proprietary software development
  - Advocate for an open and transparent engineering culture

Problem Statement

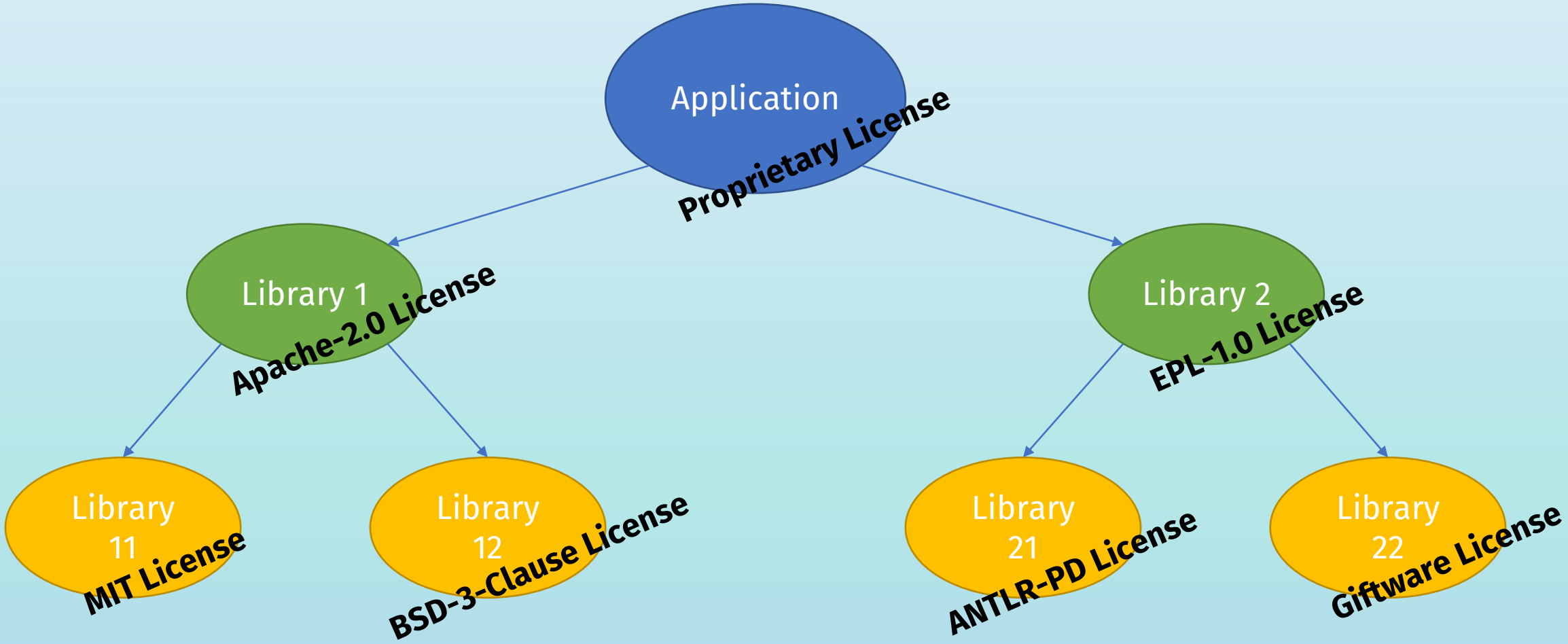
Problem Statement (continued)



Problem Statement (continued)



Problem Statement (continued)



License Obligations (OSADL matrix)

Compatibility*	OBSD	AFL-2.0	AFL-2.1	AFL-3.0	AGPL-3.0-only	AGPL-3.0-or-later	Apache-1.0	Apache-1.1	Apache-2.0	Artistic-1.0	Artistic 1.0-Per
OBSD	Same	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
AFL-2.0	Yes	Same	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
AFL-2.1	Yes	Yes	Same	Yes	No	No	Yes	Yes	Yes	Yes	Yes
AFL-3.0	Yes	Yes	Yes	Same	No	No	Yes	Yes	Yes	Yes	Yes
AGPL-3.0-only	Unknown	Unknown	Unknown	Unknown	Same	Yes	No	No	Yes	Unknown	Unknow
AGPL-3.0-or-later	Unknown	Unknown	Unknown	Unknown	No	Same	No	No	Yes	Unknown	Unknow
Apache-1.0	Yes	Yes	Yes	Yes	No	No	Same	Yes	Yes	Yes	Yes
Apache-1.1	Yes	Yes	Yes	Yes	No	No	Yes	Same	Yes	Yes	Yes
Apache-2.0	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Same	Yes	Yes
Artistic-1.0	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Same	Yes
Artistic-1.0-Perl	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Same
Artistic-2.0	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
blessing	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-1-Clause	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-2-Clause	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-2-Clause-Patent	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-3-Clause	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-4-Clause	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-4-Clause-UC	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSD-Source-Code	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
BSL-1.0	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
bzip2-1.0.5	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
bzip2-1.0.6	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
CC0-1.0	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes
CDDL-1.0	Unknown	Unknown	Unknown	Unknown	No	No	Unknown	Unknown	Unknown	Unknown	Unknow
CDDL-1.1	Unknown	Unknown	Unknown	Unknown	No	No	Unknown	Unknown	Unknown	Unknown	Unknow
CPL-1.0	Unknown	Unknown	Unknown	Unknown	No	No	Unknown	Unknown	Unknown	Unknown	Unknow
curl	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes

Common Pitfalls

- Which pairs of licenses do I need to look up?
- Caring about direct dependencies only is enough
  - No, transitive dependencies might be distributed as well
- Relying on the declared license is sufficient
  - No, you have invest “best effort” to double-check / detect licenses
- This is just about some crazy company policies
  - No, this is actually about law (German “Urheberrecht”)

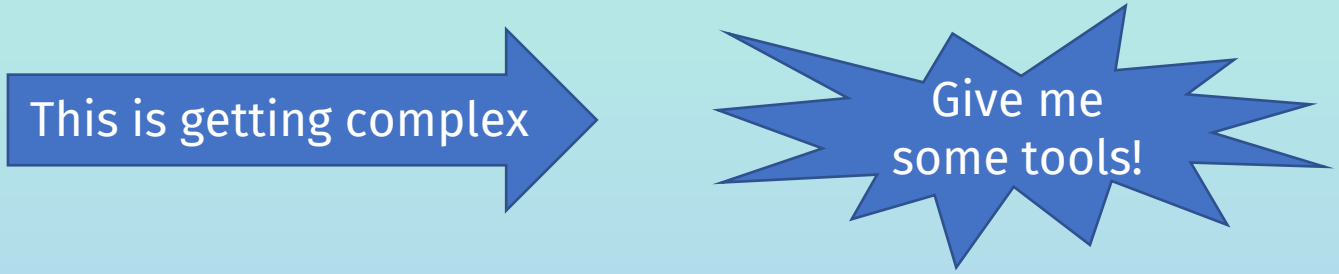
Common Pitfalls (continued)

- Which pairs of licenses do I need to look up?
- Caring about direct dependencies only is enough
  - No, transitive dependencies might be distributed as well
- Relying on the declared license is sufficient
  - No, you have invest “best effort” to double-check / detect licenses
- This is just about some crazy company policies
  - No, this is actually about law (German “Urheberrecht”)

This is getting complex

Common Pitfalls (continued)

- Which pairs of licenses do I need to look up?
- Caring about direct dependencies only is enough
  - No, transitive dependencies might be distributed as well
- Relying on the declared license is sufficient
  - No, you have invest “best effort” to double-check / detect licenses
- This is just about some crazy company policies
  - No, this is actually about law (German “Urheberrecht”)



Evaluation of Tools (back in 2016)

- No single good fit for all requirements
- Dependencies not correct
  - Not fully transitive
  - Scopes not supported
  - Version conflict resolution not considered
- Not configurable enough
  - Policy rules
  - Attribution documents
- Commercial
  - Vendor lock-in
  - Black box
  - No data ownership

Evaluation of Tools (continued)

- Decision to build own tooling
  - Fill the gaps, do not reinvent the wheel
  - Abstract away underlying tools to ease comparison / migration
  - “Do one thing and do it well” Unix philosophy
  - Run equally well on CI and locally
  - Simple human- and machine-readable result format
  - Comprehensible results
- Prototype was a collection of scripts
  - One script per language / package manager
- Rewrite as unified tool
  - Use Kotlin for fun and profit ☺

What *is* ORT?

- The OSS Review Toolkit is a suite of tools that „*aims to assist with the tasks that commonly need to be performed in the context of license compliance checks*“
  - Focus is on applications
- Used / evaluated by several bigger companies
  - Alliander, Bosch, BMW Car IT, CARIAD, EPAM, HERE Technologies, Lectra, Porsche, and more
- Scope is continuously being extended
  - Security, commercial license management, Open-Sourceing checks

What is ORT *not* (yet)?

- The OSS Review Toolkit isn’t a legal workflow tool
- No GUI (yet) to *interactively* work with the results
  - There is much more data than shown in any of the reports
- ORT was not designed for checking compliance of Docker images or Linux distributions

Goal and Workflow

Highly automate Open Source compliance process

↓

Fulfill Open Source license obligations

(redistribution of source code, attribution to Copyright holders, license compatibility)

↓

Analyze project dependencies

↓

Download the source code

↓

Goal and Workflow (continued)

Scan for Copyright holders

↓

Scan for licenses used

(declared vs. detected licenses)

↓

Evaluate rules

(license compatibility, company-internal policies)

↓

Report about the outcome

ORT Building Blocks

- Analyzer
  - Software Composition Analysis (SCA) tool to determine dependencies of software projects
- Advisor
  - Find known security vulnerabilities or defects
- Scanner / Downloader
  - Extract licenses / copyrights from downloaded source code
- Evaluator
  - Implement the actual checks and policy rules
- Reporter
  - Create attribution documents and assets like source code bundles

The ORT Analyzer (design decisions)

- Inspect projects from the “outside”, no build system plugins
  - No changes to the projects required
    - No vendor lock-in
    - Can analyze projects not under your control (like Open Source projects)
- First analyze everything and filter data later
  - No reanalysis needed when distribution scope changes
- Take version conflict resolution into account
  - Static parsing of “definition files” (like `pom.xml`) is not enough
- Supports ~20 package managers / ecosystems
  - Conan, Go, Java, JavaScript / Node, .NET, Python, Ruby, Rust and more
- Fallback to SPDX documents if no package manager available

The ORT Analyzer (example output)


```yaml
    - package:
        id: "Maven:org.apache.commons:commons-text:1.1"
        purl: "pkg:maven/org.apache.commons/commons-text@1.1"
        authors:
        - "Benedikt Ritter"
        - "Bruno P. Kinoshita"
        - "Gary Gregory"
        - "Rob Tompkins"
        - "The Apache Software Foundation"
        declared_licenses:
        - "Apache License, Version 2.0"
        declared_licenses_processed:
          spdx_expression: "Apache-2.0"
          mapped:
            Apache License, Version 2.0: "Apache-2.0"
        description: "Apache Commons Text is a library focused on algorithms working\
          \ on strings."
        homepage_url: "http://commons.apache.org/proper/commons-text/"
        binary_artifact:
          url: "https://repo.maven.apache.org/maven2/org/apache/commons/commons-text/1.1/commons-text-1.1.jar"
          hash:
            value: "c336bf600f44b88af356c8a85eef4af822b06a4d"
            algorithm: "SHA-1"
        source_artifact:
          url: "https://repo.maven.apache.org/maven2/org/apache/commons/commons-text/1.1/commons-text-1.1-sources.jar"
          hash:
            value: "f0770f7f0472bf120ada47beecadce4056fbd20a"
            algorithm: "SHA-1"
```

The ORT Advisor

- Advise about
  - security vulnerabilities
  - defects (bugs)
- Providers
  - Google OSV
  - VulnerableCode
  - OSS Index
  - Nexus IQ
  - GitHub


```
advisor:
  start_time: "2021-04-29T14:54:16.562951Z"
  end_time: "2021-04-29T14:54:18.969210Z"
  environment:
    ort_version: "7fcbb3b"
    java_version: "11.0.8"
    os: "Linux"
    processors: 4
    max_memory: 12884901888
    variables:
      JAVA_HOME: "/opt/java/openjdk"
      ANDROID_HOME: "/opt/android-sdk"
      GOPATH: "/go"
    tool_versions: {}
  config:
    nexus_iq:
      server_url: "https://oss-review-toolkit.org"
      browse_url: "https://oss-review-toolkit.org"
      username: "user"
  results:
    advisor_results:
      Maven:junit:junit:4.12:
      - vulnerabilities:
        - id: "CVE-2020-15250"
          references:
          - url: "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-15250"
            scoring_system: "CVSS2"
            severity: "5.5"
        advisor:
          name: "NexusIQ"
        summary:
          start_time: "2021-04-29T14:54:17.322191Z"
          end_time: "2021-04-29T14:54:18.966672Z"
    has_issues: false
evaluator: null
```

The ORT Scanner (introduction)

- Not a scanner by itself
  - Fill the gaps, do not reinvent the wheel
  - Abstraction layer (also eases migration)
- Wraps any configured scanners
  - Unify results (also eases comparison)
  - License scanners: ScanCode, Licensee (GitHub), Askalono
  - Snippet scanners: FossID, SCANOSS
- Uses the downloader programmatically to download source code from VCS or artifacts

The ORT Scanner (example output)


```yaml
    scan_results:
      - id: "Maven:com.vdurmont:semver4j:3.1.0"
        results:
          - provenance:
              vcs_info:
                type: "Git"
                url: "https://github.com/vdurmont/semver4j.git"
                revision: "7653e418d610ffcd2811bcb55fd72d00d420950b"
                path: ""
              resolved_revision: "7653e418d610ffcd2811bcb55fd72d00d420950b"
            scanner:
              name: "ScanCode"
              version: "3.2.1-rc2"
              configuration: "--copyright --license --ignore *.ort.yml --info --strip-root\
            \ --timeout 300 --ignore META-INF/DEPENDENCIES --json-pp"
            summary:
              start_time: "2020-09-30T09:27:12.023451Z"
              end_time: "2020-09-30T09:28:20.525647Z"
              package_verification_code: "48ba11487d53ce933b5d4db1d069b70a803ff19b"
              licenses:
                - license: "BSD-3-Clause"
                  location:
                    path: "pom.xml"
                    start_line: 28
                    end_line: 34
                - license: "MIT"
                  location:
                    path: "LICENSE.md"
                    start_line: 1
                    end_line: 1
```

The ORT Evaluator (introduction)

- Freely programmable via a Kotlin DSL
- Full access to all metadata of an ORT result
- Not limited to *Open Source* license checks
  - Could also do checks unrelated to licenses at all
- Makes use of configured license classifications
  - Do not (only) classify per license category, but per obligation

The ORT Evaluator (example script)

```
135     packageRule("COPYLEFT_IN_SOURCE") {
136         require {
137             -isExcluded()
138         }
139
140         licenseRule("COPYLEFT_IN_SOURCE", LicenseView.CONCLUDED_OR_DECLARED_AND_DETECTED) {
141             require {
142                 -isExcluded()
143                 +isCopyleft()
144             }
145
146             val message = if (licenseSource == LicenseSource.DETECTED) {
147                 "The ScanCode copyleft categorized license $license was ${licenseSource.name.lowercase()} " +
148                         "in package ${pkg.id.toCoordinates()}."
149             } else {
150                 "The package ${pkg.id.toCoordinates()} has the ${licenseSource.name.lowercase()} ScanCode copyleft " +
151                         "catalogized license $license."
152             }
153
154             error(message, howToFixDefault())
155         }
156
```

The ORT Reporter (introduction)

- Creates various assets for both distribution and internal use
  - Themed PDF attribution documents
  - NOTICE files
  - HTML reports
  - (S)BOM formats (SPDX, CycloneDX)
  - JSON output for further processing
- Support for format-specific options

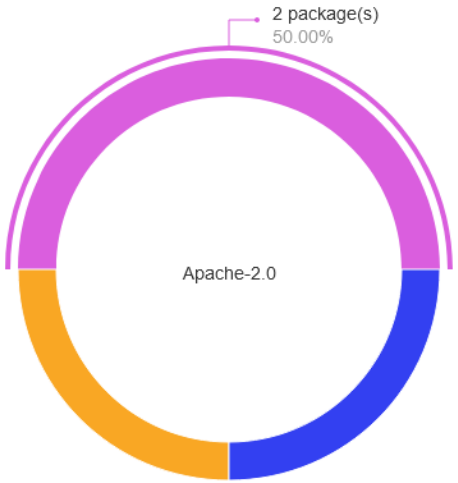
The ORT Reporter (example output)

Summary | Table | Tree

- Scanned revision of Git repository
- Found 1 files defining 5 unique dependencies within 2 scopes and 2 dependency levels
- Detected 3 declared licenses
- Completed scan with 1 unresolved issue

Issues (1) | Declared Licenses (3)

License	Packages
Apache-2.0	2
BSD-3-Clause	1
EPL-1.0	1



Integrations

- Jenkins
  - Declarative pipeline with Docker build steps

Stage View

	Configure pipeline	Build ORT Docker image	Clone project	Clone ORT configuration	Run ORT analyzer	Run ORT scanner	Run ORT advisor	Run ORT evaluator	Run ORT reporter
Average stage times:	685ms	2min 47s	9s	0ms	9s	8s	9s	5s	3s
#3: Maven:com.vdurmont:semver4j:3.1.0 Jul 14 18:13 4 commits	772ms	2min 44s	9s		9s	8s	9s	10s	7s

- GitLab CI
  - Tutorial video available
- GitHub Action

The ORT Workbench

ORT Workbench

OSS Review Toolkit

- Summary
- Packages
- Dependencies
- Issues
- Rule Violations
- Vulnerabilities

Settings

ORT version bd6e4668d4

Filter | Severity | License | License Source | Rule | Package | Resolution

NO_GPL Maven:junit:junit:4.12 — LGPL-2.1-or-later Source: DETECTED
The package 'Maven:junit:junit:4.12' has the DETECTED license 'LGPL-2.1-or-later'.
Remove the dependency on this package.

NO_GPL Maven:org.mockito:mockito-all:1.10.19 — LGPL-2.1-or-later Source: DETECTED
The package 'Maven:org.mockito:mockito-all:1.10.19' has the DETECTED license 'LGPL-2.1-or-later'.
Remove the dependency on this package.

NO_GPL Maven:org.mockito:mockito-all:1.10.19 — LGPL-3.0-or-later Source: DETECTED
The package 'Maven:org.mockito:mockito-all:1.10.19' has the DETECTED license 'LGPL-3.0-or-later'.
Remove the dependency on this package.

NO_GPL Maven:org.mockito:mockito-all:1.10.19 — LGPL-3.0-only Source: DETECTED
The package 'Maven:org.mockito:mockito-all:1.10.19' has the DETECTED license 'LGPL-3.0-only'.
Remove the dependency on this package.

The ORT Server

- Backend / frontend architecture in addition to CLI
  - Main goal is scalability
  - Will also simplify the setup / workflow
- Work has started
  - Source code not publicly available yet

Questions & Answers

Thank you!
Any questions?

Reach out to me via:
sschuberth@gmail.com
https://www.linkedin.com/in/sschuberth
https://github.com/sschuberth