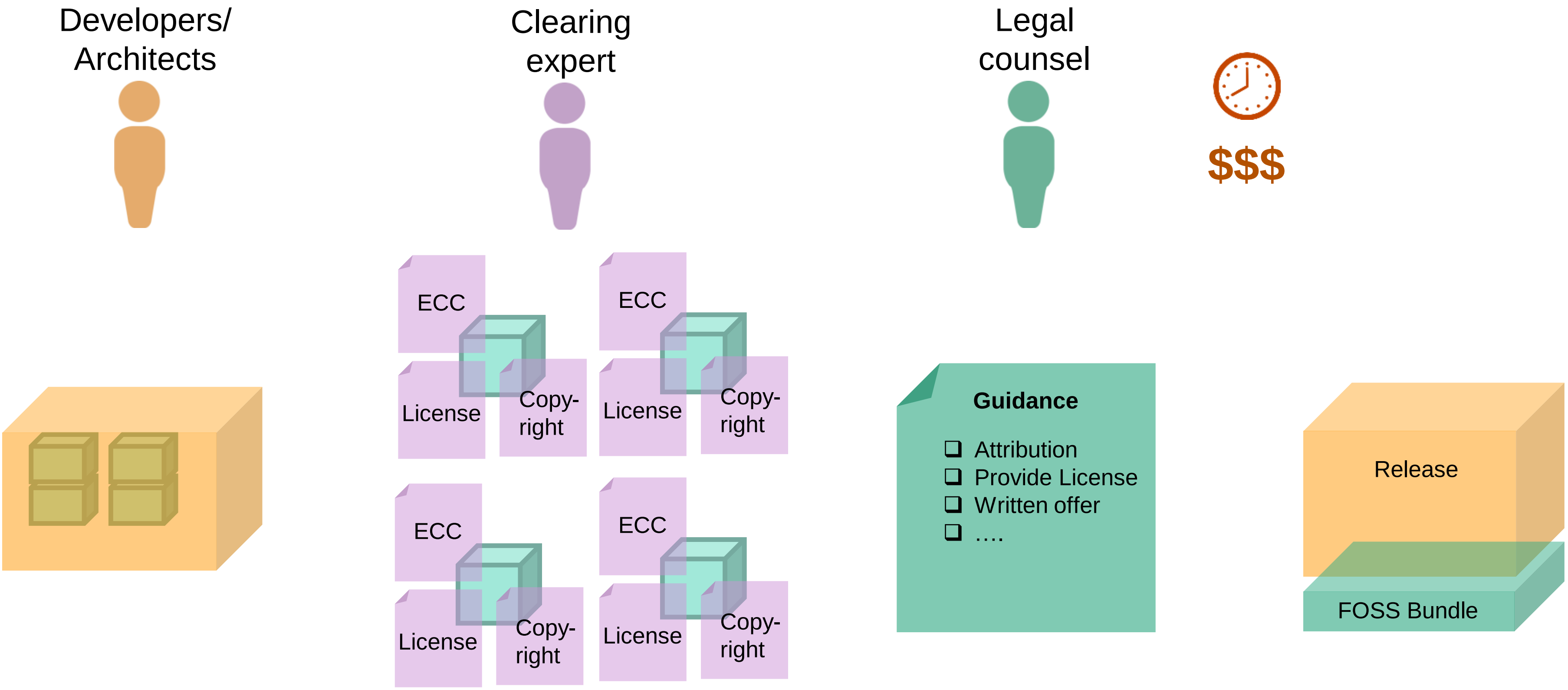
Lessons Learned from Automated License Compliance

Johannes Kristan (Bosch SI GmbH)
Michael C. Jaeger (Siemens AG)

Two Times Lessons Learned

- General Automation with SW360 REST API
- CI and Build System Integration

Introduction



Developers/ Architects

Clearing expert

Legal counsel

$$$

ECC

License

Copy-right

ECC

License

Copy-right

ECC

License

Copy-right

ECC

License

Copy-right

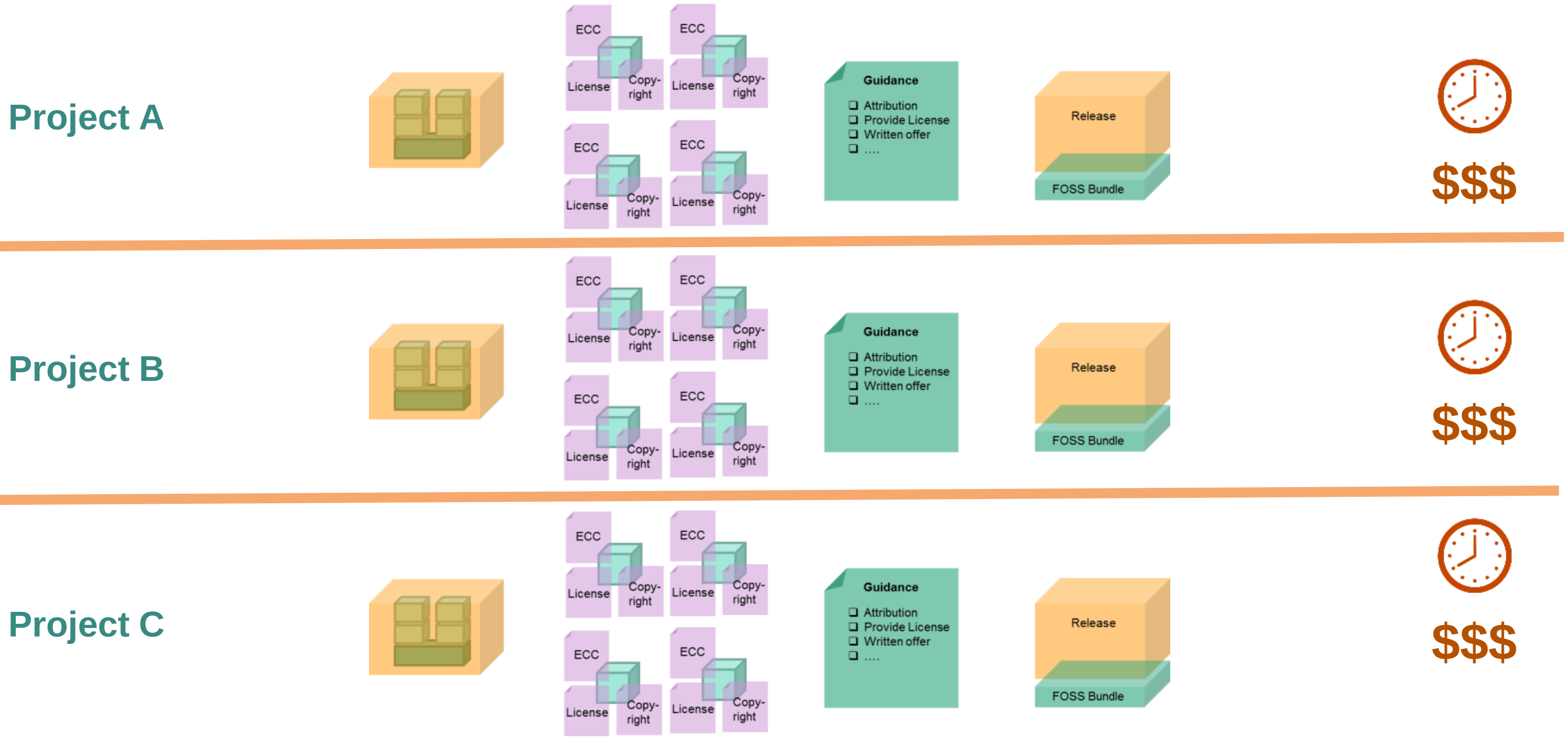
Guidance

❑ Attribution
❑ Provide License
❑ Written offer
❑ ….

Release

FOSS Bundle

Missed Reuse = Waste



Project A

ECC License Copy-right ECC License Copy-right ECC License Copy-right ECC License Copy-right

Guidance
- ❑ Attribution
- ❑ Provide License
- ❑ Written offer
- ❑ ....

Release

FOSS Bundle

$$$

Project B

ECC License Copy-right ECC License Copy-right ECC License Copy-right ECC License Copy-right

Guidance
- ❑ Attribution
- ❑ Provide License
- ❑ Written offer
- ❑ ....

Release

FOSS Bundle

$$$

Project C

ECC License Copy-right ECC License Copy-right ECC License Copy-right ECC License Copy-right

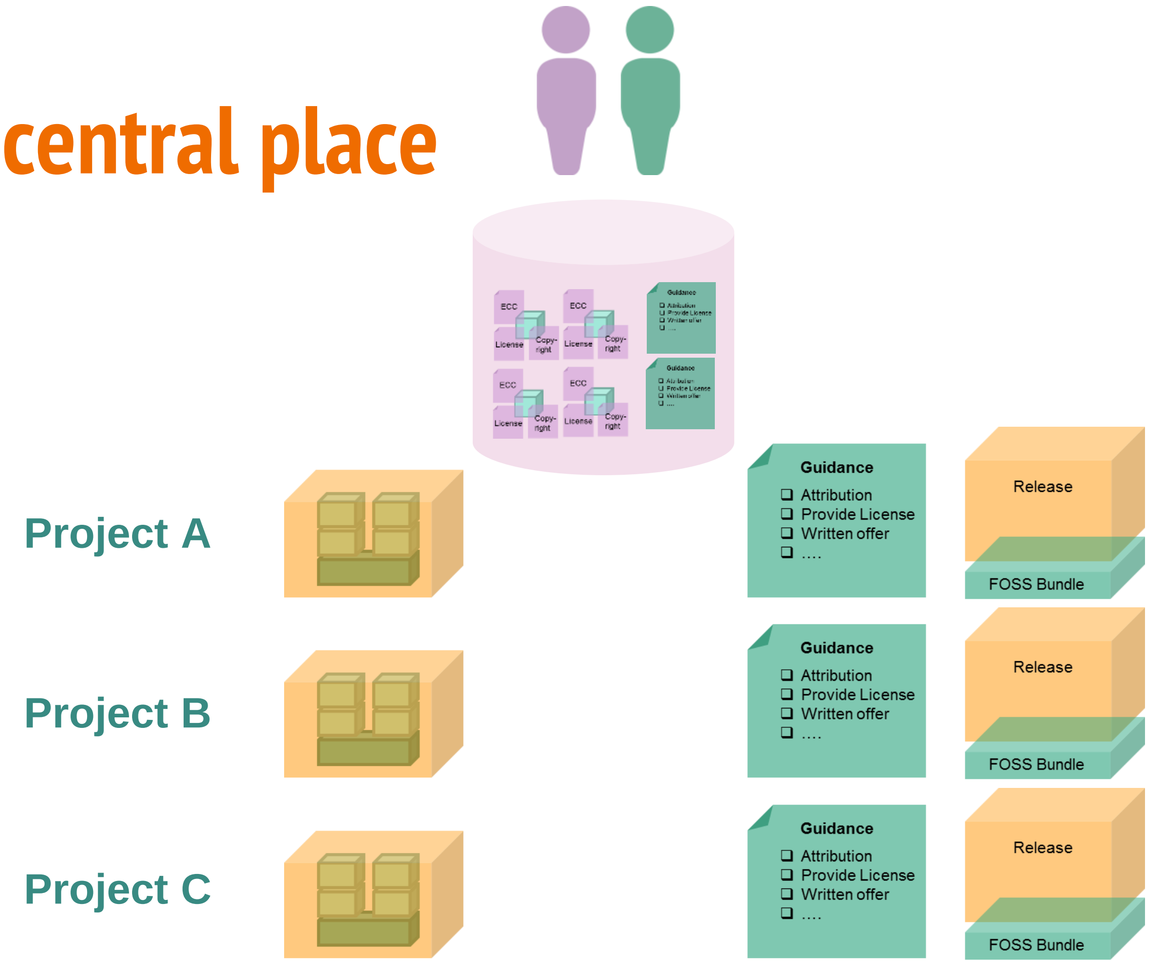
Guidance
- ❑ Attribution
- ❑ Provide License
- ❑ Written offer
- ❑ ....

Release

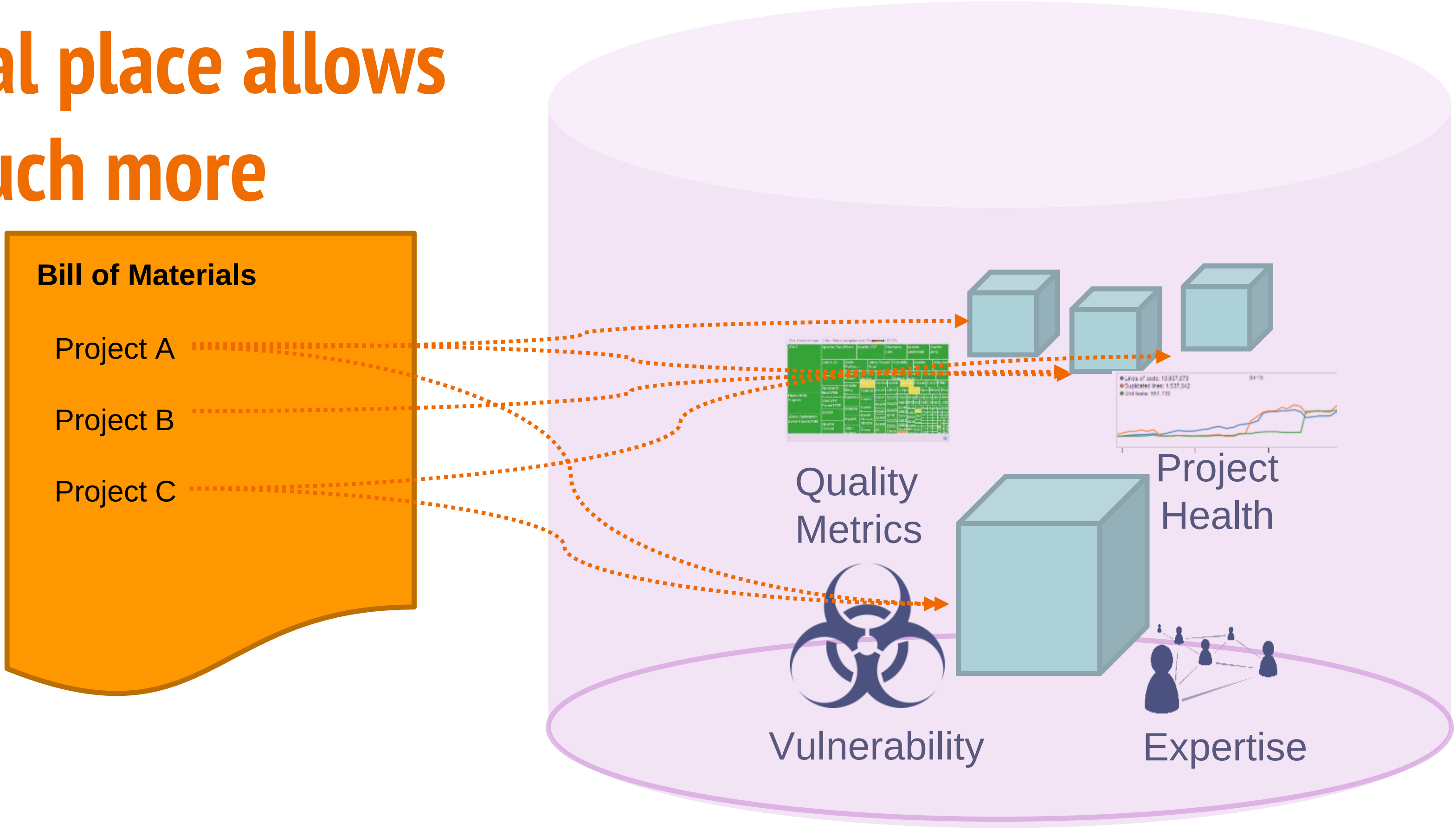
FOSS Bundle

$$$

Share at central place to reuse



Central place allows for much more



Bill of Materials

Project A

Project B

Project C

Quality Metrics

Project Health

Vulnerability

Expertise

The SW360 Automation: A REST API

… well it is a REST API

- Hypermedia interface
- Authentication with spring-security, JWT
- fully integrated into SW360
- For:
  - CI Build System Integration
  - Other software component managing systems
  - Vulnerability sync, for example: Whitesource integration

Lessons Learned
Part I

- Roll Out
- Security
- Endpoint Design
- Data Hygiene

Lessons Learned: Roll Out

Problem:

- REST API = code from others that access your system
- How to prevent client from entering bad data
- Or, compromising the system because of programming mistakes

Solution:

- **dev** playground as self service
- **stage** for guided testing of solutions
- after “probation” allow access for **productive**

➔ also operations needs to learn what happens at client use

Lessons Learned: Security

Problem:

- Security for state-of-the-art Web apps is highly developed
- REST API needs to have same security strength as browser login

Part of the Problem:

- State of the art using spring-security and Oauth legacy workflow
- Authorization server based auth is good for security but not quick
- Clumsy to implement in quick scripts

Solutions:

- Change to token vending approach
- Allow for easy READ access for component catalogue (ie. licensing information)

Lessons Learned: Endpoints Design

Problem: Designing good endpoints is hard for new use cases

- Actually use cases: you understand them when programming then
- You think it is easy, how do you find your information?
- As a result, clients query a lot of information
- Getting all components and iterating by them one by one

➔ a lot of load on the servers

Solution:

- Optimize use cases with REST API users
- Use case tailored endpoints help
- Filtering, querying, paging, optimizing the returned data
- Buffering in REST endpoint implementation or Web server level

Lessons Learned: Data Hygiene I

Problem:
Field entry is not uniform

- Different users, different data
- Referencing info (e.g. Maven id)
- Expressing approvals
- What is a component actually
  - (folder from an OSS project …)

➔Querying the REST API results in lack of uniformity immediately is visible.

Solutions

- Education and documentation
- Admin UI for data hygiene in tabular form
- Merge feature to merge duplicate component data sets
- More constraints on data entry required
- Provide drop down lists with data as desired (or how it makes sense) in the UI

SW360 antenna

SW360 in the Build Process

Eclipse SW360antenna

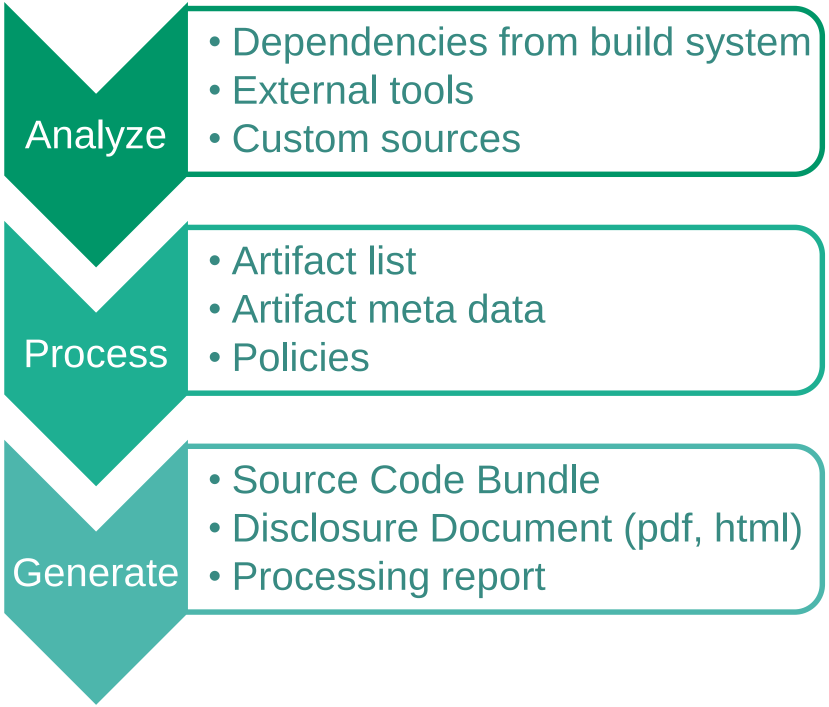
Automate OSS Management in your Builds

Integrates into your build process

- Analyze dependencies
- Synchronize BOMs with SW360
- Enforce policies
- Generate FOSS bundle

Project provides

- Frontends to Maven, Gradle, CLI
- Extensibility via plugin mechanism
- Staged configuration for standardized processes



Analyze

- Dependencies from build system
- External tools
- Custom sources

Process

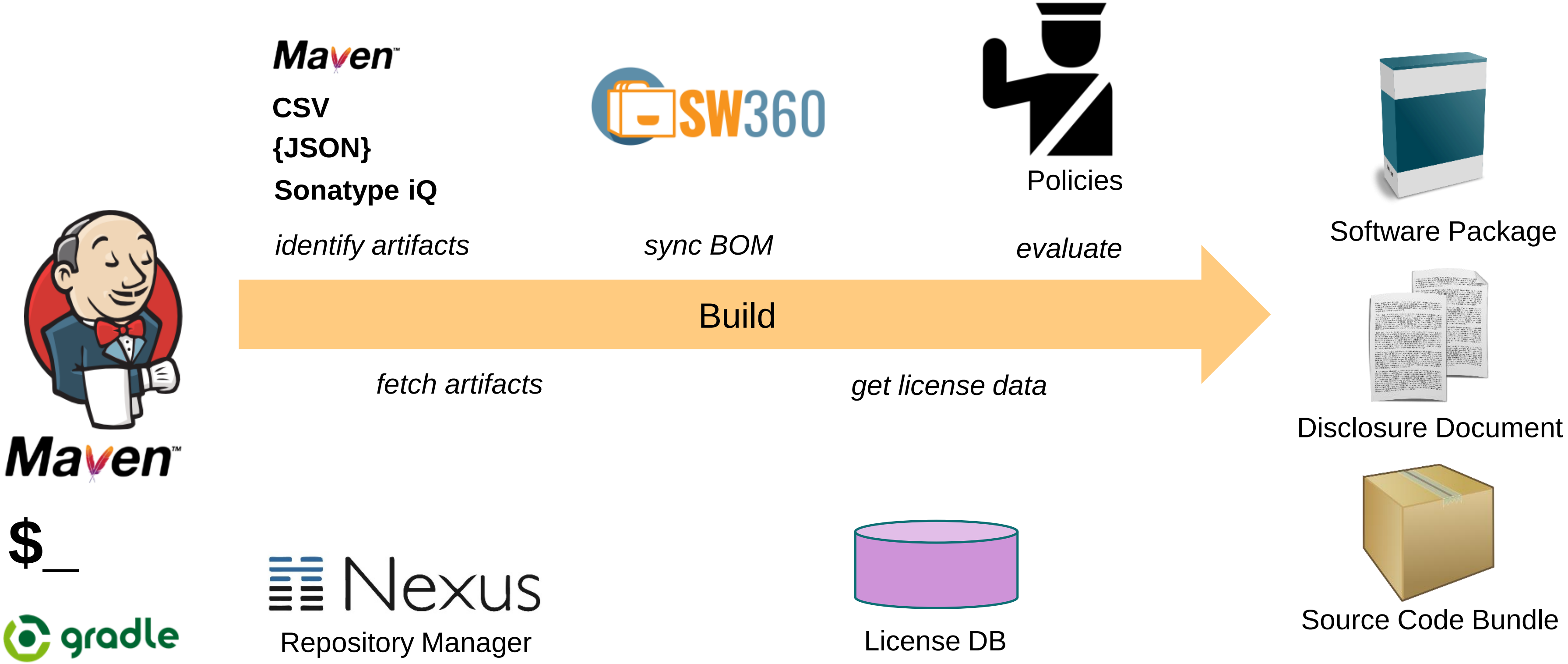
- Artifact list
- Artifact meta data
- Policies

Generate

- Source Code Bundle
- Disclosure Document (pdf, html)
- Processing report

Join and find out more here: https://eclipse.org/antenna

Compliance Mngmt Integrated into Build Process



Maven

CSV

{JSON}

Sonatype iQ

SW360

Policies

Software Package

identify artifacts

sync BOM

evaluate

Build

fetch artifacts

get license data

Disclosure Document

Maven

$_

gradle

Nexus

Repository Manager

License DB

Source Code Bundle

Lessons Learned

- Customization required
- Automated policy evaluation requires up-to-date data
- Only have on place to find out about component use
- Give user feedback in their working environment

Lesson Learned: Customizability Required

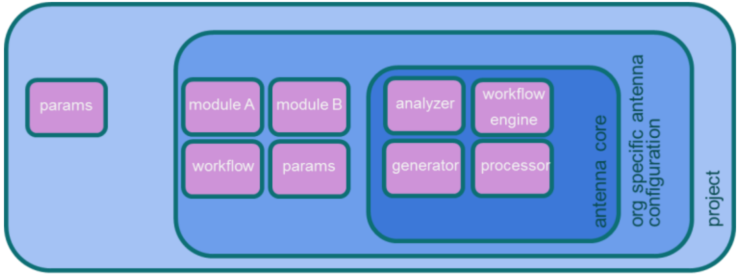
Problem

- To strict policies can block projects
- Technology is evolving
- Unforeseen corner cases

Provide means to customize the tool to fit project team needs.

Solution

- Provide staging mechanism for tool configuration
- Only decide basics on org level
- Allow for a stepwise refinement



Credentials
Naming
Sources

Org specific templates
Org specifc policies
Org specific params

Build system integration
Workflow engine
Core modules
Basic configuration

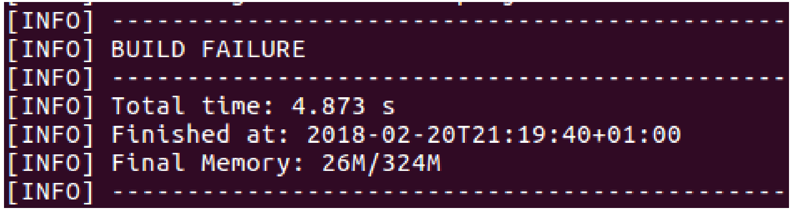
Lesson Learned: Maintain Local DB

Problem

- External database which cannot be updated within your organization can lead to severe delays
- Manual intervention and bypassing the process required

Solution

- At least fallback to provide own data
- External data as additional source of information



```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.873 s
[INFO] Finished at: 2018-02-20T21:19:40+01:00
[INFO] Final Memory: 26M/324M
[INFO] ------------------------------------------------------------------------
```

If you check policies automatically ensure your database is up-to-date and can quickly be extended.

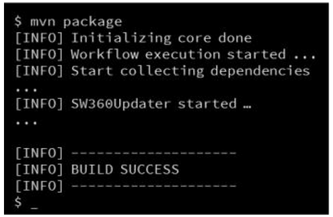
Lesson Learned: Data Access at one Place

Problem

- Data integration is hard
- Inconsistent naming
- Different identification mechanisms

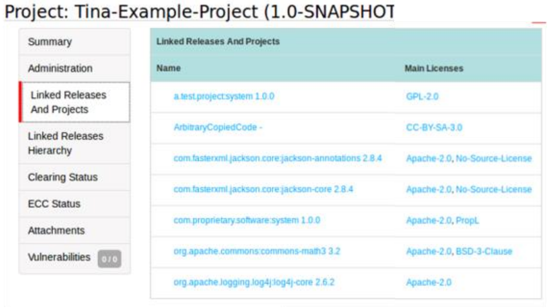
Solution

- Central place for data integration



```
$ mvn package
[INFO] Initializing core done
[INFO] Workflow execution started ...
[INFO] Start collecting dependencies
...
[INFO] SW360Updater started …
...

[INFO] --------------------
[INFO] BUILD SUCCESS
[INFO] --------------------
$ _
```



Project: Tina-Example-Project (1.0-SNAPSHOT

Linked Releases And Projects

Name	Main Licenses
a.test.project:system 1.0.0	GPL-2.0
ArbitraryCopiedCode -	CC-BY-SA-3.0
com.fasterxml.jackson.core:jackson-annotations 2.8.4	Apache-2.0, No-Source-License
com.fasterxml.jackson.core:jackson-core 2.8.4	Apache-2.0, No-Source-License
com.proprietary.software:system 1.0.0	Apache-2.0, PropL
org.apache.commons:commons-math3 3.2	Apache-2.0, BSD-3-Clause
org.apache.logging.log4j:log4j-core 2.6.2	Apache-2.0

If you want to reverse lookup component usages make sure to have the data at one place.

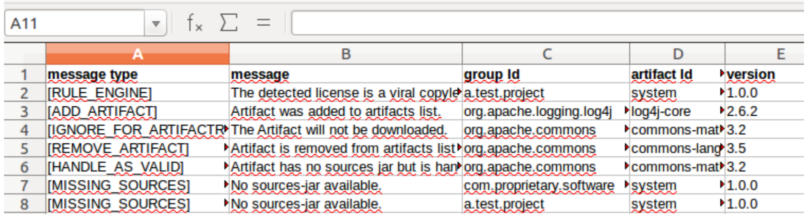
Lesson Learned: Feedback in User’s Context

Problem

- Switching systems leads to missing notifications

Solution

- Build breaker in case of problems
- Processible reports



	A	B	C	D	E
1	**message type**	**message**	**group Id**	**artifact Id**	**version**
2	[RULE_ENGINE]	The detected license is a viral copyle	a.test.project	system	1.0.0
3	[ADD_ARTIFACT]	Artifact was added to artifacts list.	org.apache.logging.log4j	log4j-core	2.6.2
4	[IGNORE_FOR_ARTIFACTR	The Artifact will not be downloaded.	org.apache.commons	commons-mat	3.2
5	[REMOVE_ARTIFACT]	Artifact is removed from artifacts list	org.apache.commons	commons-lang	3.5
6	[HANDLE_AS_VALID]	Artifact has no sources jar but is han	org.apache.commons	commons-mat	3.2
7	[MISSING_SOURCES]	No sources-jar available.	com.proprietary.software	system	1.0.0
8	[MISSING_SOURCES]	No sources-jar available.	a.test.project	system	1.0.0

Don’t force your users to look into another system to get information about processing.

We are on GitHub!

ECLIPSE™
FOUNDATION

github.com/eclipse/sw360

Michael C. Jaeger
Siemens AG Corporate Technology
D-80200 Munich, Germany
michael.c.jaeger@siemens.com

Johannes Kristan
Bosch Software Innovations GmbH
D-10785 Berlin, Germany
johannes.kristan@bosch-si.com

Source code repository:

https://github.com/eclipse/sw360

Eclipse project page:

https://eclipse.org/sw360

1. SonarQube TM and the SonarQube Logo are Trademark of SonarSource By SonarSource - SonarSource, CC BY 3.0, https://commons.wikimeddia.org/w/index.php?curid=27076948
2. Apache Tomcat TM and the Tomcat Logo are trademark of the Apache Software Foundation By The Apache Software Foundation http://svn.apache.org/viewvc/jakarta/site/xdocs/images/logos/tomcat.eps, Apache License 2.0, https://commons.wikimedia.org/w/index.php?curid=11302180
3. Apache Lucene TM and the Lucene Logo are trademark of the Apache Software Foundation http://en.wikipedia.org/wiki/Image:Lucene_logo_green_300.png, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=905779
4. Apache CouchDB TM and the CouchDB Logo are trademark of the Apache Software Foundation
5. Apache Thrift TM and the Thrift Logo is trademark of the Apache Software Foundation
6. Docker TM and the Docker Logo are Trademark of Docker Inc. https://www.docker.com/sites/default/files/legal/DockerMarks_may2017.zip
7. JFrog TM and the JFrog Logo are Trademark of JFrog https://www.jfrog.com/brand-guidelines/
8. Liferay TM and the Liferay Logo are Trademark of Liferay, Inc., https://www.liferay.com/de/trademark
9. 'Eclipse', 'Built on Eclipse' and 'Eclipse Ready', 'SW360' are trademarks of Eclipse Foundation, Inc. https://eclipse.org/legal/logo_guidelines.php
10. Sonarqube Dashboard Screenshot by SonarSource - SonarSource, CC BY 3.0, https://commons.wikimedia.org/w/index.php?curid=27076949
11. The Octocat logo is a Registered Trademark of github.com https://github.com/logos
12. https://www.forbes.com/sites/adrianbridgwater/2015/04/24/if-software-is-eating-the-world-then-open-source-will-chew-it-up-and-swallow/#4f4feb933902
13. Jenkins logo by https://jenkins.io/ is licensed under CC BY-SA 3.0
14. Apache Maven™ and its logo are trademark of the Apache Software Foundation
15. Sonatype Nexus™ and its logo are trademark of Sonatype
16. Police man, software package, documents and box are from https://openclipart.org and licensed under Creative Commons Zero 1.0 Public Domain License
17. Gradle and its logo is trademark of Gradle inc. https://commons.wikimedia.org/wiki/File:Gradle_logo_small.png#/media/File:Gradle_logo_small.png