Eclipse Apoapsis - Open Source based Software Composition Analysis at scale

Marcel Kurzmann, Robert Bosch GmbH

FOSSNorth 2024

BOSCH

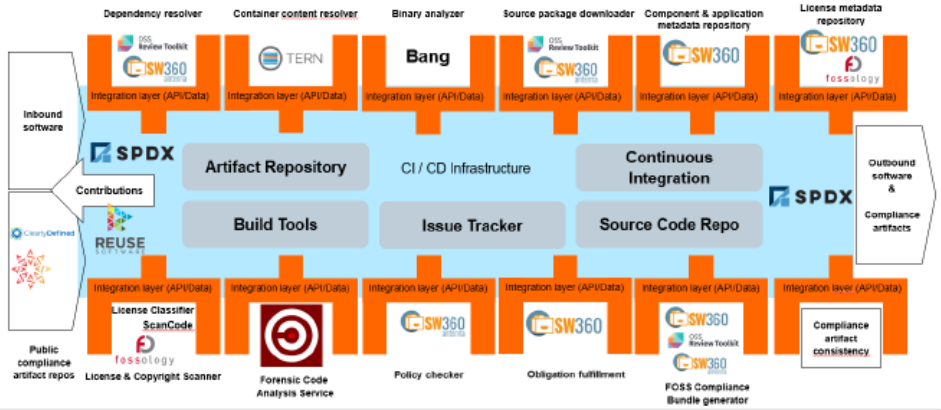
Reference Tooling Work Group

We are building an open source compliance toolchain ecosystem with open source tools as an open source project.
To accomplish this we:

- Use existing independent tooling projects
- Provide reference workflows to allow their adoption
- Provide the concepts and glue to ensure easy interoperability and integration in existing environments
- Provide reference turnkey toolchains that can be used without fees by anybody

World-Wide Collaboration, World-Wide Availability

Example Automation Implementation Using Open Source Tools



Join Us in Creating a New Era for Open Source Compliance

Mailing List: oss-based-compliance-tooling@groups.io

Subscription page: https://groups.io/g/oss-based-compliance-tooling

Online meetings: Bi-weekly - Invitations are sent to the mailing list

Website: https://oss-compliance-tooling.org/

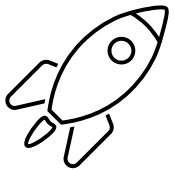
And of course we are on GitHub:

https://github.com/Open-Source-Compliance/Sharing-creates-value

Oliver Fendt

Background

Background
Our journey – the beginning



Mission: Open Source Management automation for JAVA/Maven projects.

Target Fact Sheet (simplified) - JAVA/Maven

Environment Parameters

- Business context: Server-based applications, fat clients
- Distribution context: hosted/distributed
- Development context: explorative / deterministic
- Development Mode: Agile / classic using agile methods
- Build mode: CI/CD, Jenkins

Open Source Parameter

- Open Source Use: only permissive licenses
- Open Source snippets: forbidden
- OSM Concept: binary identification via hashes, hash matching
- Package identification: package manager
- Component paradigm: 1 component ⇔ 1 source
- Metadata Source: central (commercial) database

Mission completed?

https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html

Earth Fact Sheet

Bulk parameters

Mass (10^{24} kg)	5.9722
Volume (10^{10} km^3)	108.321
Equatorial radius (km)	6378.137
Polar radius (km)	6356.752
Volumetric mean radius (km)	6371.000
Core radius (km)	3485
Ellipticity (Flattening)	0.003353
Mean density (kg/m^3)	5513
Surface gravity (mean) (m/s^2)	9.820
Surface acceleration (eq) (m/s^2)	9.780
Surface acceleration (pole) (m/s^2)	9.832
Escape velocity (km/s)	11.186
GM (x 10^6 km^3/s^2)	0.39860
Bond albedo	0.294
Geometric albedo	0.434
V-band magnitude V(1,0)	-3.99
Solar irradiance (W/m^2)	1361.0
Black-body temperature (K)	254.0
Topographic range (km)	20.4
Moment of inertia (I/MR^2)	0.3308
J_2 (x 10^{-6})	1082.63
Number of natural satellites	1
Planetary ring system	No

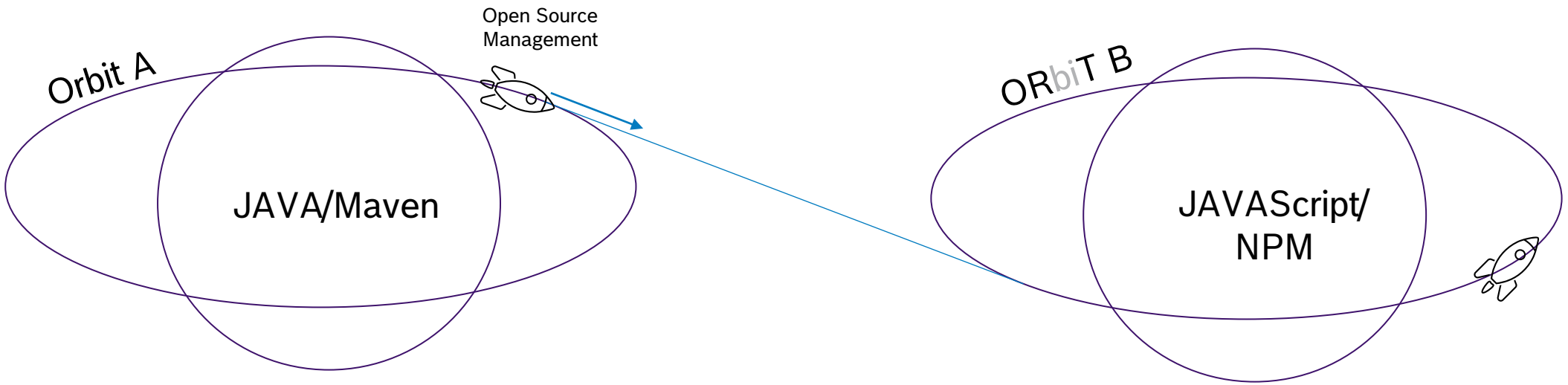
Orbital parameters

Semimajor axis (10^6 km)	149.598
Sidereal orbit period (days)	365.256
Tropical orbit period (days)	365.242
Perihelion (10^6 km)	147.095
Aphelion (10^6 km)	152.100

Source: https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html

...

Background
Our journey – orbit transfer



Background

Our journey – the next mission

Open Source Management automation for JAVAscript/NPM projects.

Target Fact Sheet (simplified) - JAVAScript/NPM

Environment Parameters

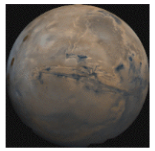
- Business context: Web applications
- Distribution context: distributed
- Development context: explorative / deterministic
- Development Mode: Agile / classic using agile methods
- Build mode: CI/CD, Jenkins

Open Source Parameter

- Open Source Use: only permissive license
- Open Source snippets: forbidden
- OSM Concept: binary identification via hashes, hash matching => recursive dependency resolution
- Package identification: package manager
- Component paradigm: 1 component ⇔ 1 source => n:m; download sources and scan
- Metadata Source: central (commercial) database => local database with scan results and curations

https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html

Mars Fact Sheet



Mars/Earth Comparison

Bulk parameters

	Mars	Earth	Ratio (Mars/Earth)
Mass (10^{24} kg)	0.64169	5.9722	0.107
Volume (10^{10} km^3)	16.312	108.321	0.151
Equatorial radius (km)	3396.2	6378.1	0.532
Polar radius (km)	3376.2	6356.8	0.531
Volumetric mean radius (km)	3389.5	6371.0	0.532
Core radius (km)	1830**	3485	0.525
Ellipticity (Flattening)	0.00589	0.00335	1.76
Mean density (kg/m^3)	3934	5513	0.714
Surface gravity (mean) (m/s^2)	3.73	9.82	0.380
Surface acceleration (eq) (m/s^2)	3.69	9.78	0.377
Surface acceleration (pole) (m/s^2)	3.73	9.83	0.379
Escape velocity (km/s)	5.03	11.19	0.450
GM (x 10^6 km^3/s^2)	0.042828	0.39860	0.107
Bond albedo	0.250	0.294	0.850
Geometric albedo	0.170	0.434	0.392
V-band magnitude V(1,0)	-1.60	-3.99	-
Solar irradiance (W/m^2)	586.2	1361.0	0.431
Black-body temperature (K)	209.8	254.0	0.826
Topographic range (km)	30	20	1.500
Moment of inertia (I/MR^2)	0.366	0.3308	1.106
J_2 (x 10^{-6})	1960.45	1082.63	1.811
Number of natural satellites	2	1	
Planetary ring system	No	No	

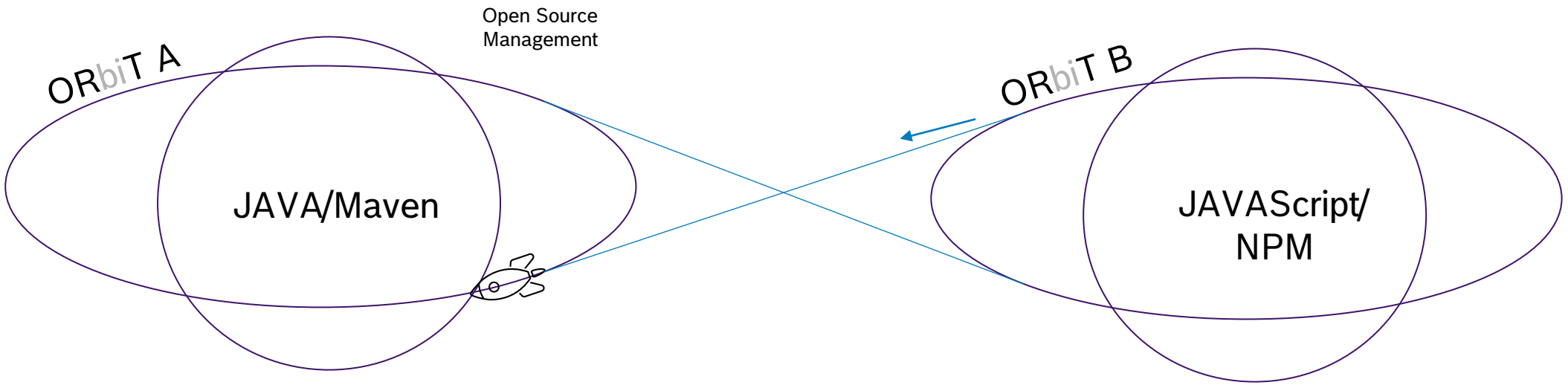
** Recent results indicate the radius of the core of Mars may only be 1650 - 1675 km.

Orbital parameters

	Mars	Earth	Ratio (Mars/Earth)
Semimajor axis (10^6 km)	227.956	149.598	1.524
Sidereal orbit period (days)	686.980	365.256	1.881
Tropical orbit period (days)	686.973	365.242	1.881
Perihelion (10^6 km)	206.650	147.095	1.405
Aphelion (10^6 km)	249.261	152.100	1.639

Source: https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html

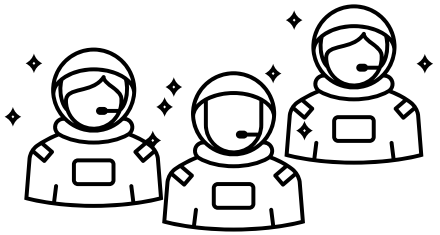
Background
Our journey – transfer of learnings



Background

Our journey – utilizing the momentum

Open Source Management automation for Embedded systems.



Team consisting of Open Source Office members and automation developers

Target Fact Sheet (simplified) – Embedded C / Conan

Environment Parameters

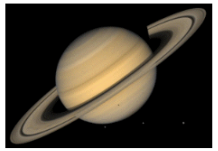
- Business context: Embedded Software for devices
- Distribution context: distributed
- Development context: deterministic
- Development Mode: scaled agile framework
- Build mode: regular incremental builds, Github action, limited scaling options => ORT-Server

Open Source Parameter

- Open Source Use: permissive licenses, weak copyleft licenses
- Open Source snippets: forbidden, use with exception
- OSM Concept: project.spdx.yml-files combined with snippet and license and copyright scanning
- Package identification: manually maintained project spdx.yml-files
- Component paradigm: 1 source ⇔ different binaries
- Metadata Source: source code

https://nssdc.gsfc.nasa.gov/planetary/factsheet/saturnfact.html

Saturn Fact Sheet



Saturn/Earth Comparison

Bulk parameters

	Saturn	Earth	Ratio (Saturn/Earth)
Mass (10^{24} kg)	568.32	5.9722	95.16
Volume (10^{10} km^3)	82,713	108.321	763.59
Equatorial radius (1 bar level) (km)	60,268	6,378.1	9.449
Polar radius (1 bar level) (km)	54,364	6,356.8	8.552
Volumetric mean radius (km)	58,232	6,371.0	9.140
Ellipticity (Flattening)	0.09796	0.00335	29.24
Mean density (kg/m^3)	687	5,513	0.125
Gravity (mean, 1 bar) (m/s^2)	11.19	9.82	1.140
Acceleration (eq., 1 bar) (m/s^2)	8.96	9.78	0.916
Acceleration (pole, 1 bar) (m/s^2)	12.14	9.83	1.235
Escape velocity (km/s)	35.5	11.19	3.172
GM (x 10^6 km^3/s^2)	37.931	0.39860	95.16
Bond albedo	0.342	0.294	1.16
Geometric albedo	0.499	0.434	1.15
V-band magnitude V(1,0)	-8.91	-3.99	-
Solar irradiance (W/m^2)	14.82	1,361.0	0.011
Black-body temperature (K)	81.0	254.0	0.319
Moment of inertia (I/MR^2)	0.210	0.3308	0.635
J_2 (x 10^{-6})	16,298.	1082.63	15.054
Number of natural satellites	146	1	
Planetary ring system	Yes	No	

Orbital parameters

	Saturn	Earth	Ratio (Saturn/Earth)
Semimajor axis (10^6 km)	1,432.041	149.598	9.573
Sidereal orbit period (days)	10,759.22	365.256	29.457
Tropical orbit period (days)	10,746.94	365.242	29.424
Perihelion (10^6 km)	1,357.554	147.095	9.229
Aphelion (10^6 km)	1,506.527	152.100	9.905

Source: https://nssdc.gsfc.nasa.gov/planetary/factsheet/saturnfact.html

Going back in time in: https://github.com/oss-review-toolkit/ort/

Supported package manager

Currently, the following package managers / dependencies:

- Gradle
- Maven
- SBT
- NPM
- PIP

JAN 2018

Supported package managers

Currently, the following package managers / build s dependencies:

- Bower (JavaScript)
- Bundler (Ruby)
- dep (Go)
- Glide (Go)
- Godep (Go)
- Gradle (Java)
- Maven (Java)
- NPM (Node.js)
- Composer (PHP)
- PIP (Python)
- SBT (Scala)
- Stack (Haskell)
- Yarn (Node.js)

JAN 2019

Currently, the following package managers are supported:

- Bower (JavaScript)
- Bundler (Ruby)
- Cargo (Rust)
- Conan (C / C++, *experimental* as the VCS locations oft
- dep (Go)
- DotNet (.NET, with currently some limitations)
- Glide (Go)
- Godep (Go)
- GoMod (Go, *experimental* as only proxy-based source
- Gradle (Java)
- Maven (Java)
- NPM (Node.js)
- NuGet (.NET, with currently some limitations)
- Composer (PHP)
- PIP (Python)
- Pipenv (Python)
- Pub (Dart / Flutter)
- SBT (Scala)
- Stack (Haskell)
- Yarn (Node.js)

JAN 2020

Currently, the following package managers are supported:

- Bower (JavaScript)
- Bundler (Ruby)
- Cargo (Rust)
- Carthage (iOS / Cocoa)
- Composer (PHP)
- Conan (C / C++, *experimental* as the VCS locations oft #2037)
- dep (Go)
- DotNet (.NET, with currently some limitations)
- Glide (Go)
- Godep (Go)
- GoMod (Go, *experimental* as only proxy-based source
- Gradle (Java)
- Maven (Java)
- NPM (Node.js)
- NuGet (.NET, with currently some limitations)
- PIP (Python)
- Pipenv (Python)
- Pub (Dart / Flutter)
- SBT (Scala)
- SPDX (SPDX documents used to describe projects or p
- Stack (Haskell)
- Yarn (Node.js)

JAN 2021

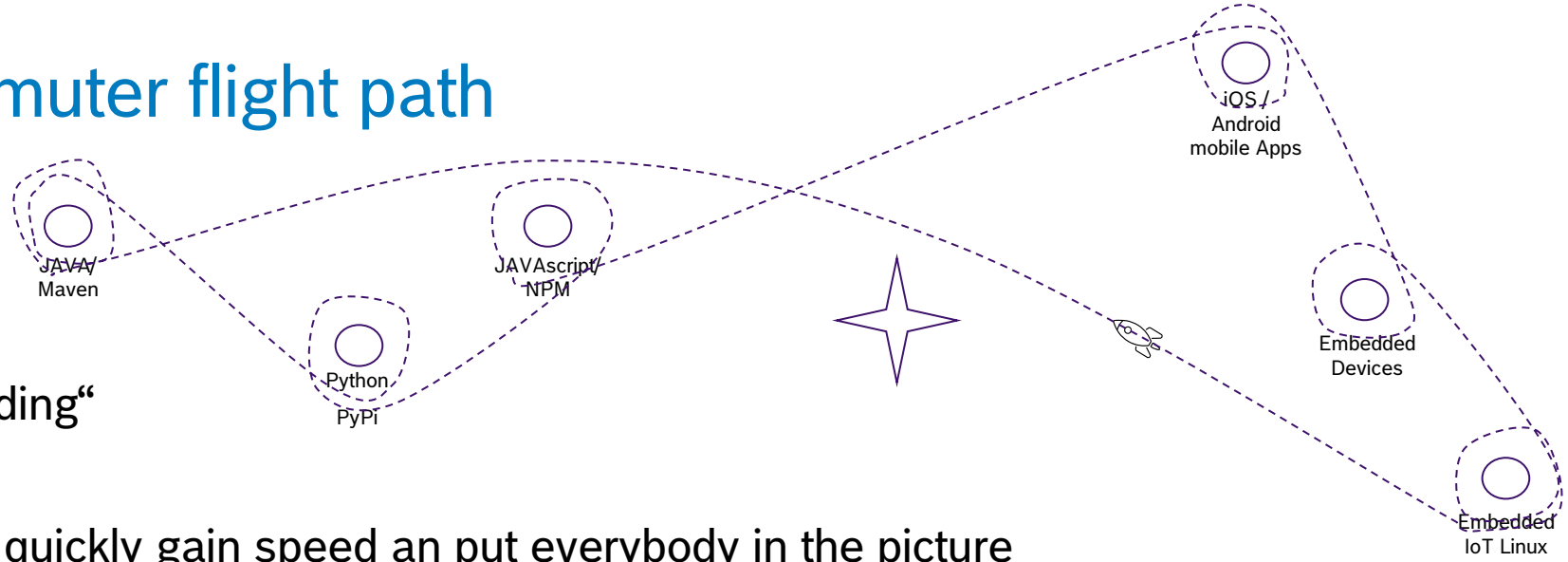
Currently, the following package managers (grouped by the programming language with) are supported:

- C / C++
  - Conan (limitations: receipe vs. source repository)
  - Also see: SPDX documents
- Dart / Flutter
  - Pub
- Go
  - dep
  - Glide
  - Godep
  - GoMod (limitations: no `replace` directive)
- Haskell
  - Stack
- Java
  - Gradle
  - Maven (limitations: default profile only)
- JavaScript / Node.js
  - Bower
  - NPM (limitations: no scope-specific registries, no peer dependencies)
  - Yarn (limitations: no Yarn 2 / 3 support)
- .NET
  - DotNet (limitations: no floating versions / ranges, no target framework)
  - NuGet (limitations: no floating versions / ranges, no target framework)
- Objective-C / Swift
  - Carthage (limitation: no `cartfile.private`)
  - CocoaPods (limitations: no custom source repositories)
- PHP
  - Composer
- Python
  - PIP (limitations: Python 2.7 or 3.6 and PIP 18.1 only)
  - Pipenv (limitations: Python 2.7 or 3.6 and PIP 18.1 only)
- Ruby
  - Bundler (limitations: restricted to the version available on the host)
- Rust
  - Cargo
- Scala
  - SBT

Today

BOSCH

Background
„at scale“ – commuter flight path



Experience from „Onboarding“

- „Fact sheets“ helpful to quickly gain speed an put everybody in the picture
  - For new team members
  - For the „customer“ development teams that needed support

- Mandatory concept documentation based on standardized template accelerated evolution
  - Initial documentation => reuse => iterative improvement => standardization => automation
  - Find reusable solutions faster by reducing search range with the help of „fact sheets“

Background

Our journey – the next stop

Open Source Management automation for Embedded IoT Linux systems.

Target Fact Sheet (simplified) – Embedded IoT LINUX

Environment Parameters

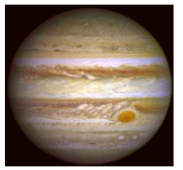
- Business context: Internet of things
- Distribution context: distributed
- Development context: deterministic
- Development Mode: classic using agile methods
- Build mode: development builds/release builds

Open Source Parameter

- Open Source Use: copyleft license
- Open Source snippets: forbidden
- OSM Concept: SBOM generated by build, component scanning or matching against database
- Package identification: purl, hashes, ...
- Component paradigm: source2binary-files, recipes, ...
- Metadata Source: collaboratively maintained public database; upstream first

https://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html

Jupiter Fact Sheet



Jupiter/Earth Comparison

Bulk parameters

	Jupiter	Earth	Ratio (Jupiter/Earth)
Mass (10^{24} kg)	1,898.13	5.9722	317.83
Volume (10^{10} km^3)	143,128	108.321	1321.33
Equatorial radius (1 bar level) (km)	71,492	6,378.1	11.209
Polar radius (km)	66,854	6,356.8	10.517
Volumetric mean radius (km)	69,911	6,371.0	10.973
Ellipticity	0.06487	0.00335	19.36
Mean density (kg/m^3)	1,326	5,513	0.241
Gravity (mean, 1 bar) (m/s^2)	25.92	9.82	2.640
Acceleration (eq., 1 bar) (m/s^2)	23.12	9.78	2.364
Acceleration (pole, 1 bar) (m/s^2)	27.01	9.83	2.748
Escape velocity (km/s)	59.5	11.19	5.32
GM (x 10^6 km^3/s^2)	126.687	0.39860	317.83
Bond albedo	0.343	0.294	1.17
Geometric albedo	0.538	0.434	1.24
V-band magnitude V(1,0)	-9.40	-3.99	-
Solar irradiance (W/m^2)	50.26	1361.0	0.037
Black-body temperature (K)	109.9	254.0	0.433
Moment of inertia (I/MR^2)	0.254	0.3308	0.768
J_2 (x 10^{-6})	14,736	1082.63	13.611
Number of natural satellites	95	1	
Planetary ring system	Yes	No	

Orbital parameters

	Jupiter	Earth	Ratio (Jupiter/Earth)
Semimajor axis (10^6 km)	778.479	149.598	5.204
Sidereal orbit period (days)	4,332.589	365.256	11.862
Tropical orbit period (days)	4,330.595	365.242	11.857
Perihelion (10^6 km)	740.595	147.095	5.035
Aphelion (10^6 km)	816.363	152.100	5.367

Source: https://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html

Background
Goals and needs

- Find match: Map your needs and
  - … find existing solutions
  - … find birds of a feather

Fact sheets

- Share and reuse

Generic architecture model

Standardized representation

- Standardizing while keeping flexibility

Example: Finding clothes online

1st limitation of search range

Women OR Men OR Kids

2nd limitation of search range

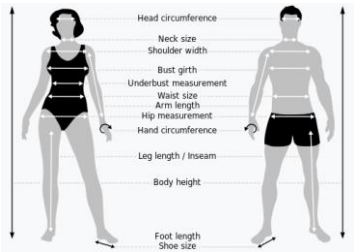
Clothing OR Shoes OR Sportswear OR …

3rd limitation of search range

Jackets OR T-Shirts OR Pants OR …

4th limitation of search range

Size ? Determine parameters



„Fact sheet“

Size Chart
XS, S, M, L, XL

Source: https://commons.wikimedia.org/wiki/File:Body_measures_SVG.svg

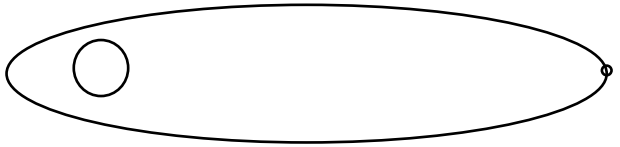
Get overview of all clothes matching to your parameters

BOSCH

Eclipse Apoapsis

BOSCH

Eclipse Apoapsis
New project proposal



apoapsis noun

apo·apsis ¦apō +

plural **apoapses** *or* **apoapsides** " +

: the apsis that is farthest from the center of attraction **:** the high point in an orbit

Source: https://www.merriam-webster.com/dictionary/apoapsis

apoapsis [ăp′ō-ăp′sĭs]

Plural apoapsides (ăp′ō-ăp′sĭ-dēz′)

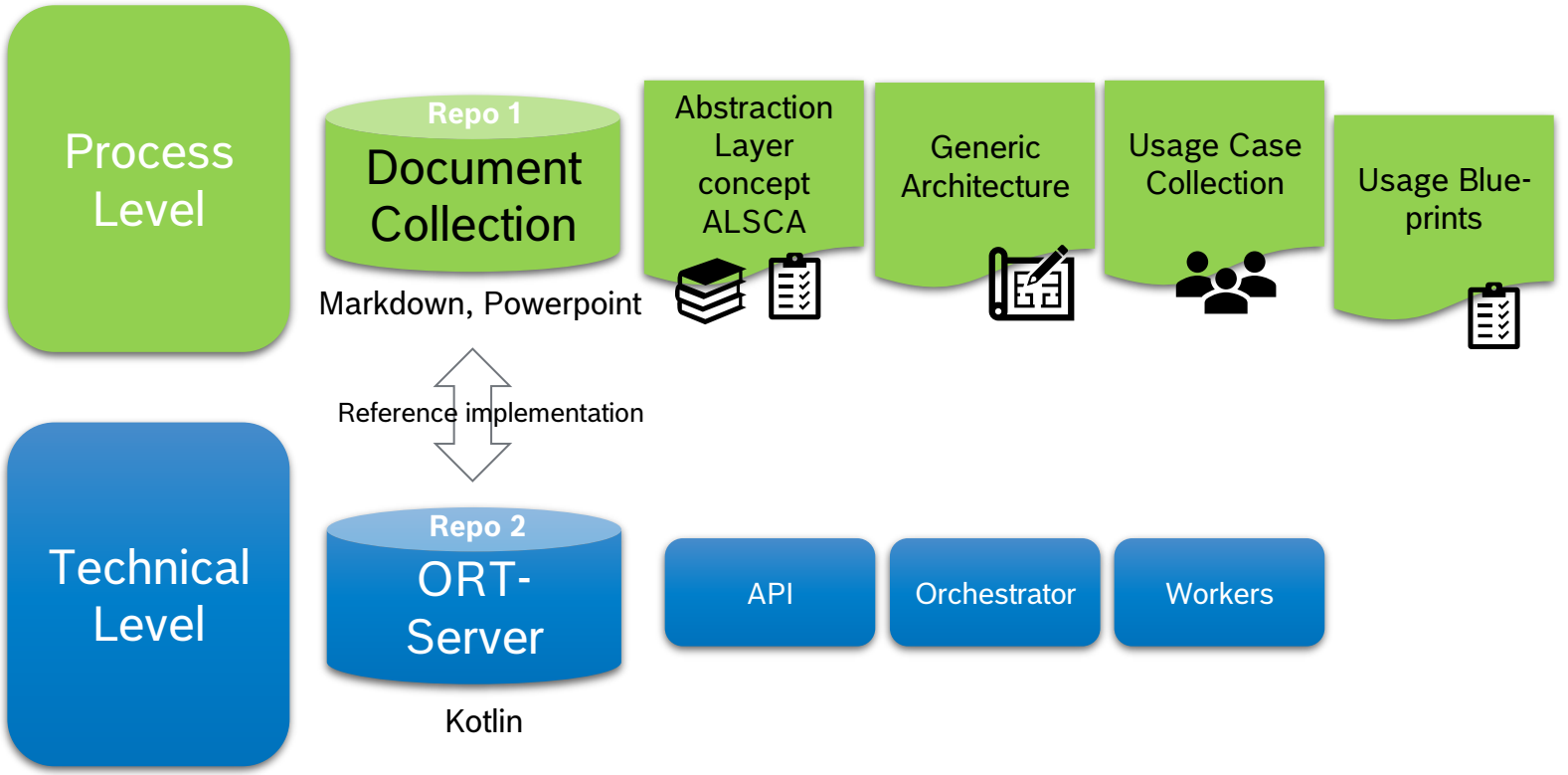
The point at which an orbiting object is farthest away from the body it is orbiting.

Source: https://www.dictionary.com/browse/apoapsis

- Apoapsis is a good opportunity, if you want to transfer to another object‘s orbit.

- Details see
- https://projects.eclipse.org/proposals/eclipse-apoapsis

Eclipse Apoapsis
Overview and planned Outputs



Match your needs to available solutions and jump-start process definition

Build your own recursive Dependency Resolution Service

Eclipse Apoapsis
Dependencies

SPDX

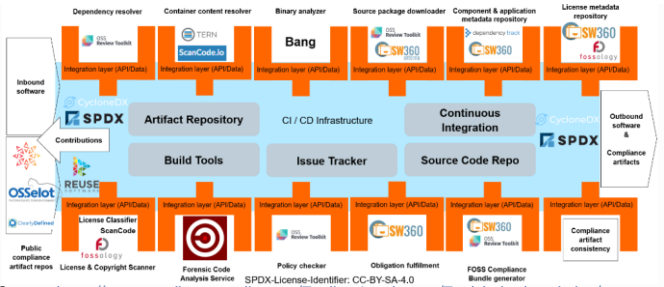
OPENCHAIN

Open Source Tooling Group

SPDX3.x Operations Profile

Capability Map

Process Level



Source: https://oss-compliance-tooling.org/Tooling-Landscape/Toolchain-description/

Technical Level

OSS Review Toolkit

PostgreSQL

APACHE ACTIVEMQ

KEYCLOAK

Kotlin

RabbitMQ

docker

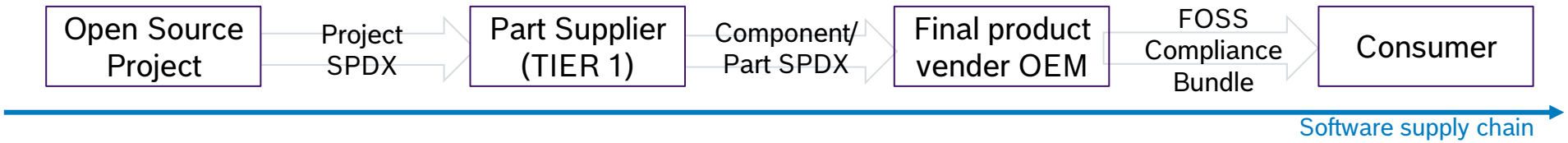
Ktor

kubernetes

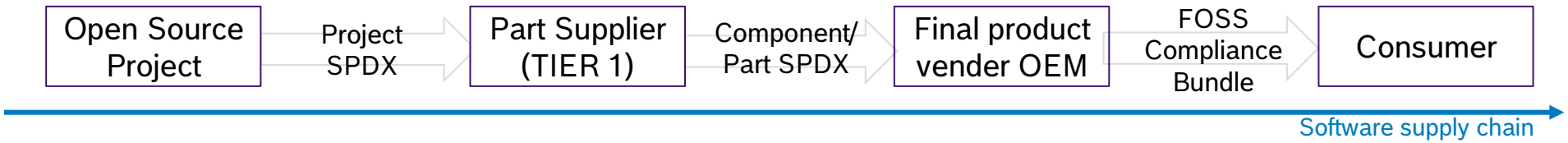
BOSCH

Process Level Outputs

Software Supply Chain Model (simplified)



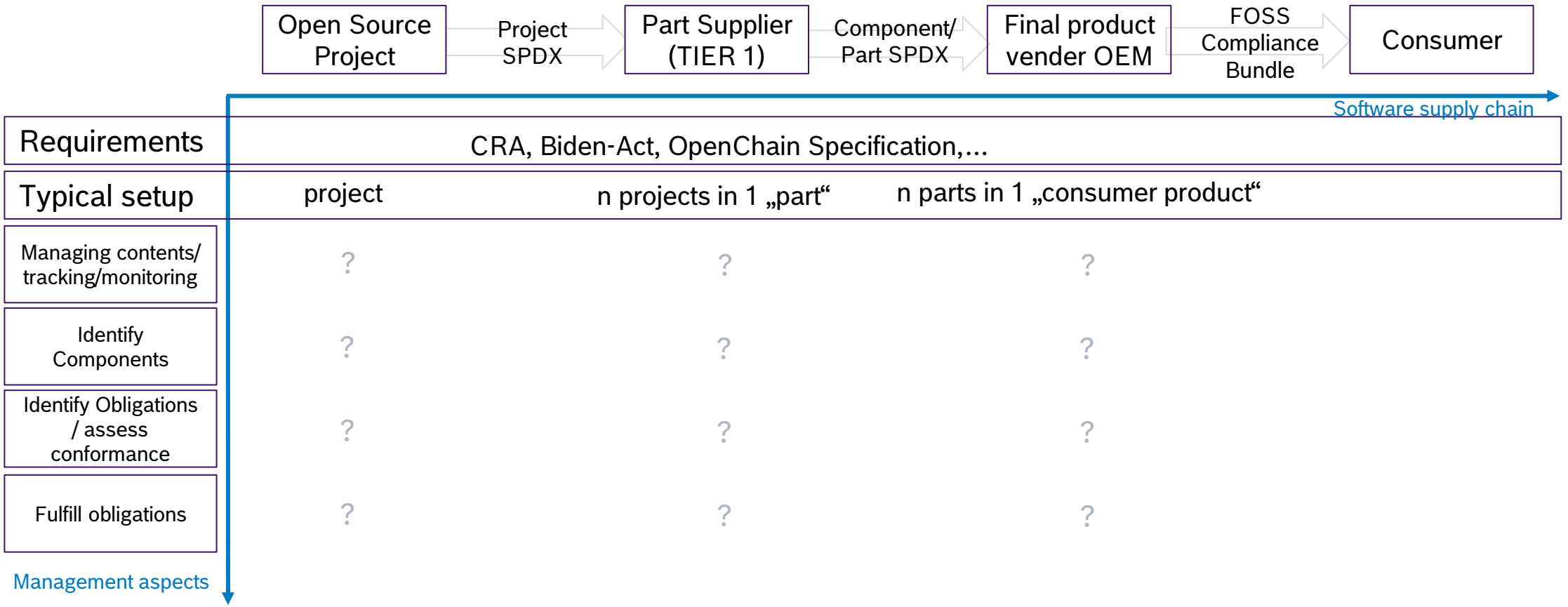
Software Supply Chain Model (simplified)



Supply Chain Simulation with Software Management Dummy Repositories ?

Details see :
Sharing OSM Test Dummies
(https://github.com/Open-Source-Compliance/Sharing-creates-value/tree/master/Meeting-Material/Meeting-20231206)

Software Supply Chain Model (simplified) – Management aspects



Software Supply Chain Model (simplified) – Generic process

Open Source Project → Project SPDX → Part Supplier (TIER 1) → Component/ Part SPDX → Final product vender OEM → FOSS Compliance Bundle → Consumer

Software supply chain

Requirements	CRA, Biden-Act, OpenChain Specification,…
Typical setup	project — n projects in 1 „part“ — n parts in 1 „consumer product“

Managing contents/ tracking/monitoring

Identify Components

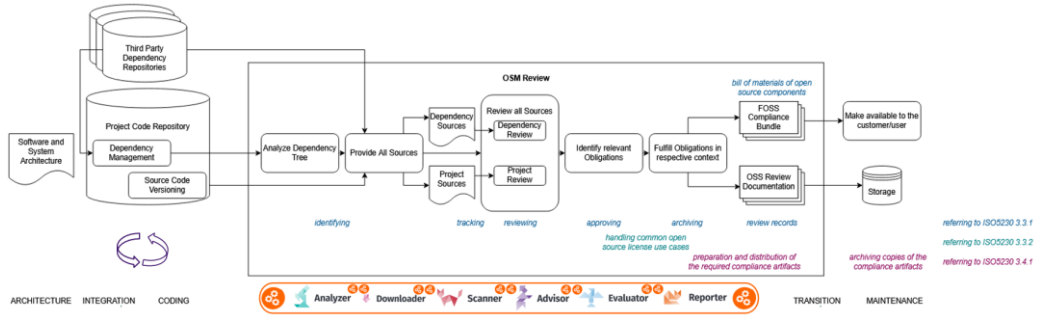
Identify Obligations / assess conformance

Fulfill obligations

Management aspects

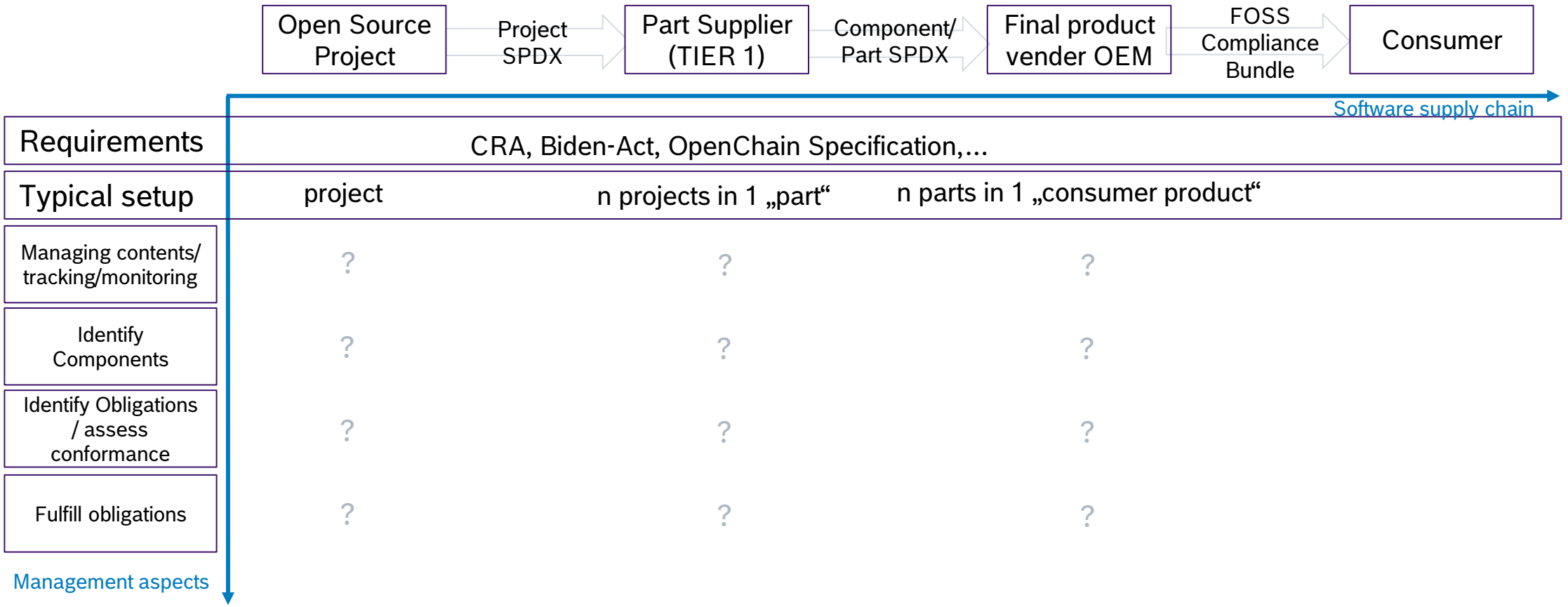
Generic process

Generic Architecture

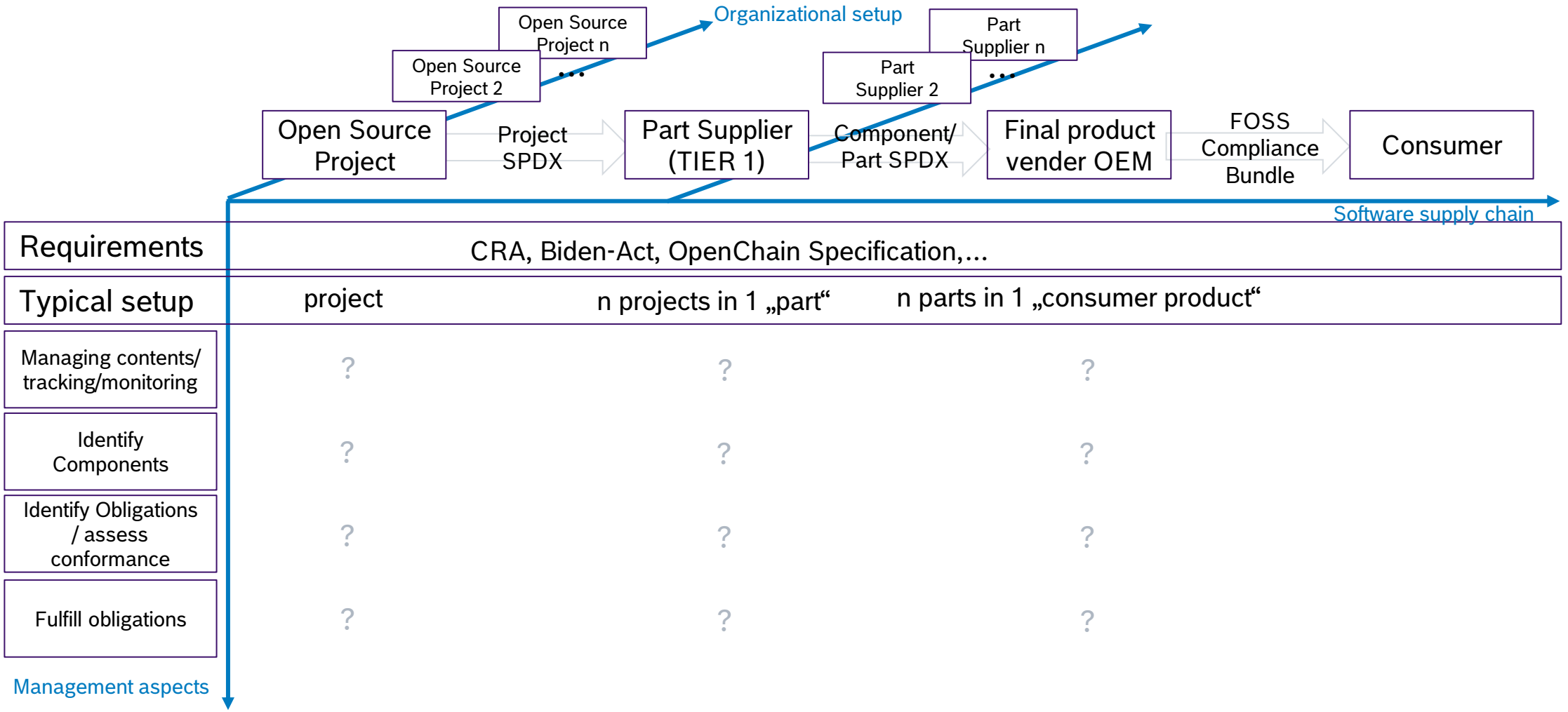


Details will be provided soon

Software Supply Chain Model (simplified)



Software Supply Chain Model (simplified) – Organizational Setup



Software Supply Chain Model (simplified)

Organizational setup

Open Source Project n
Open Source Project 2
...
Part Supplier n
Part Supplier 2
...

Open Source Project → Project SPDX → Part Supplier (TIER 1) → Component/ Part SPDX → Final product vender OEM → FOSS Compliance Bundle → Consumer

Software supply chain

Requirements

Typical setup

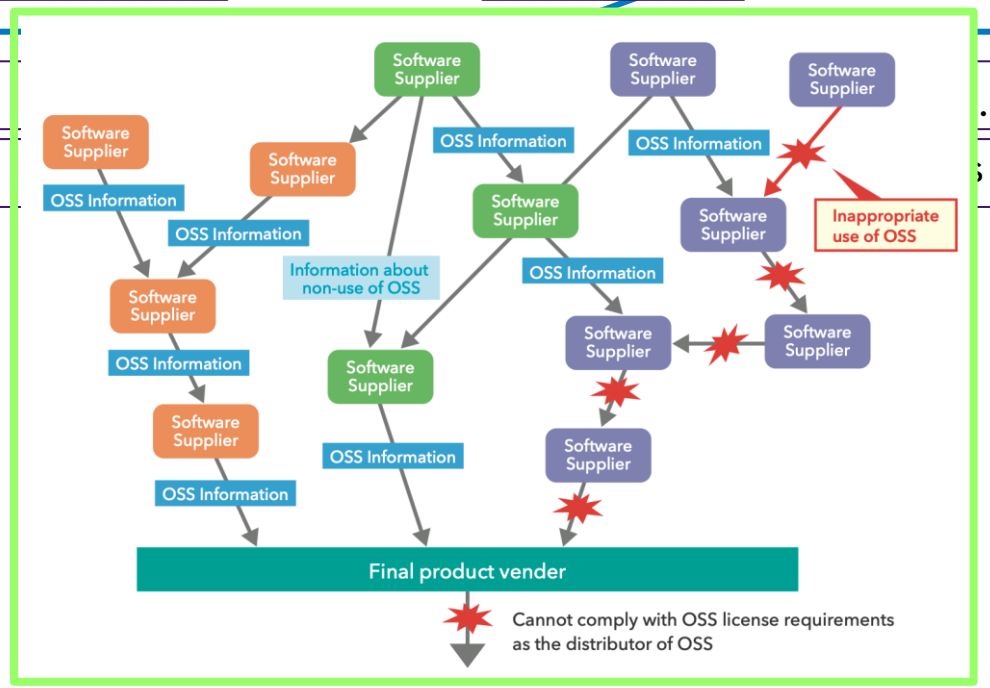
Managing contents/ tracking/monitoring

Identify Components

Identify Obligations / assess conformance

Fulfill obligations

Management aspects



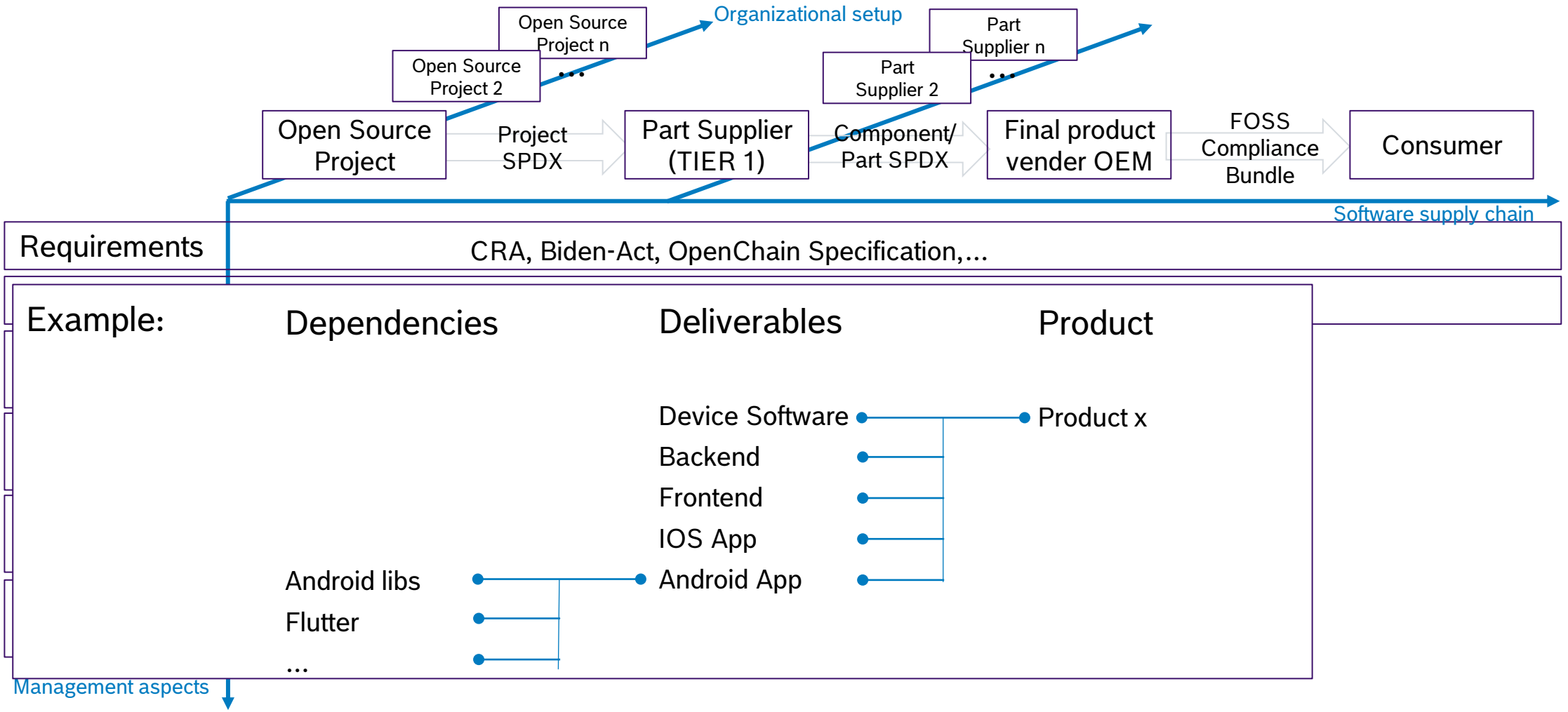
... in 1 „consumer product“

?
?
?

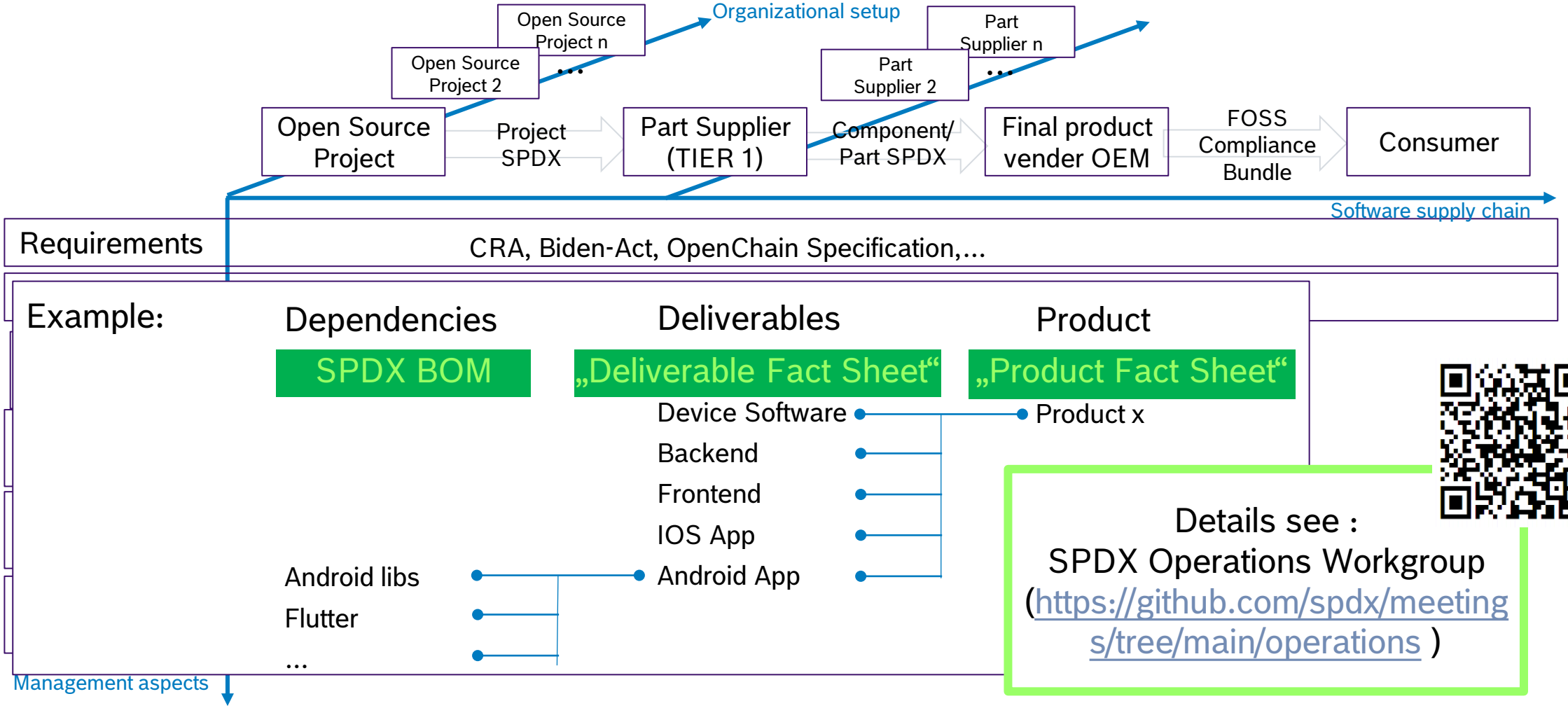
Supplier education leaflet

Source: https://github.com/OpenChain-Project/Reference-Material/blob/master/Education-For-Suppliers/Supplier-Education-Leaflet/supply-chain-education-leaflet-version-2-2024/supply-chain-education-leaflet-version-2_en.md

Software Supply Chain Model (simplified) - Example

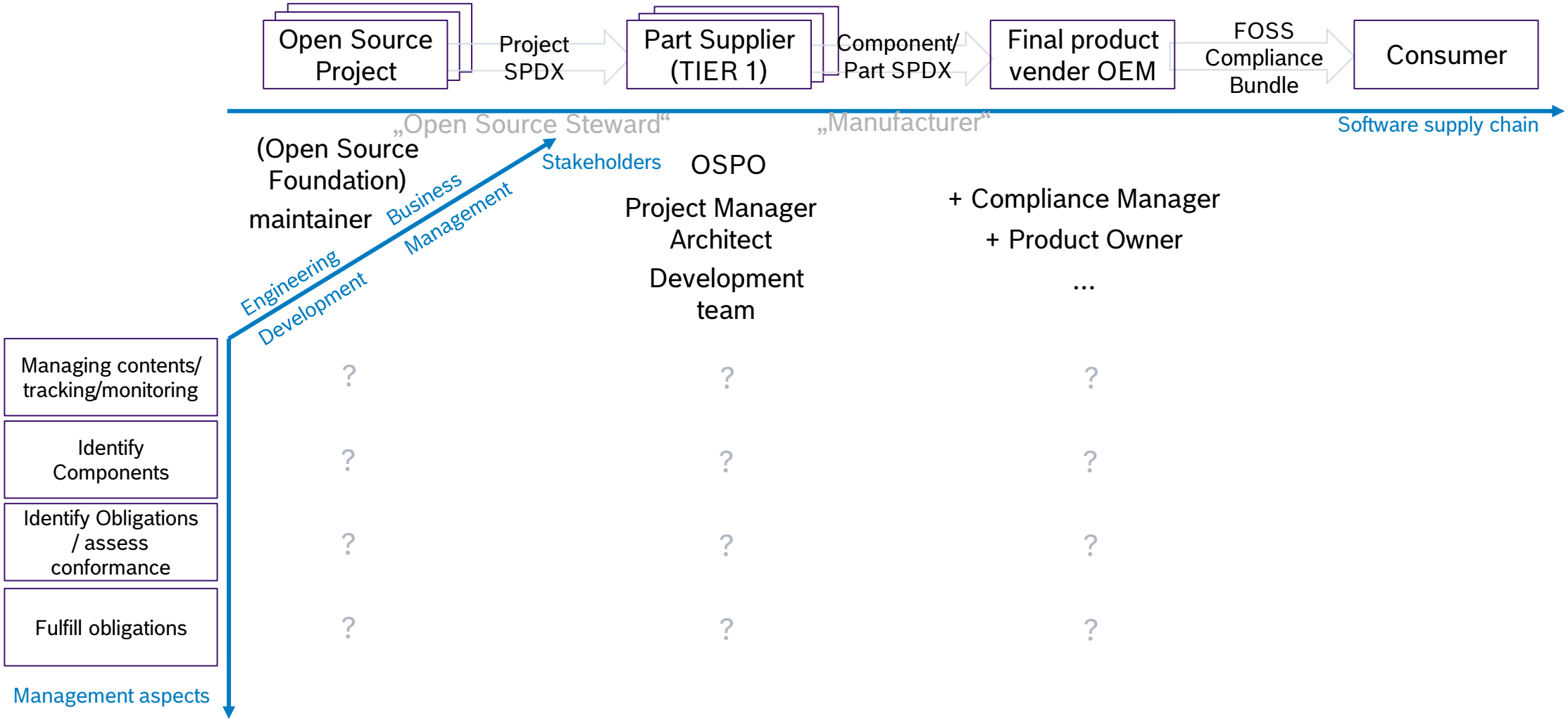


Software Supply Chain Model (simplified) – SPDX Operations



Details see :
SPDX Operations Workgroup
(https://github.com/spdx/meetings/tree/main/operations)

Software Supply Chain Model (simplified) - Stakeholders



Software Supply Chain Model (simplified) - Stakeholders

(Open Source Foundation) maintainer

Stakeholders

OSPO

Project Manager
Architect
Development team

+ Compliance Manager
+ Product Owner
...

Business Management

Engineering Development

Managing contents/ tracking/monitoring

Identify Components

Identify Obligations / assess conformance

Fulfill obligations

Management aspects

? ? ?

? ? ?

? ? ?

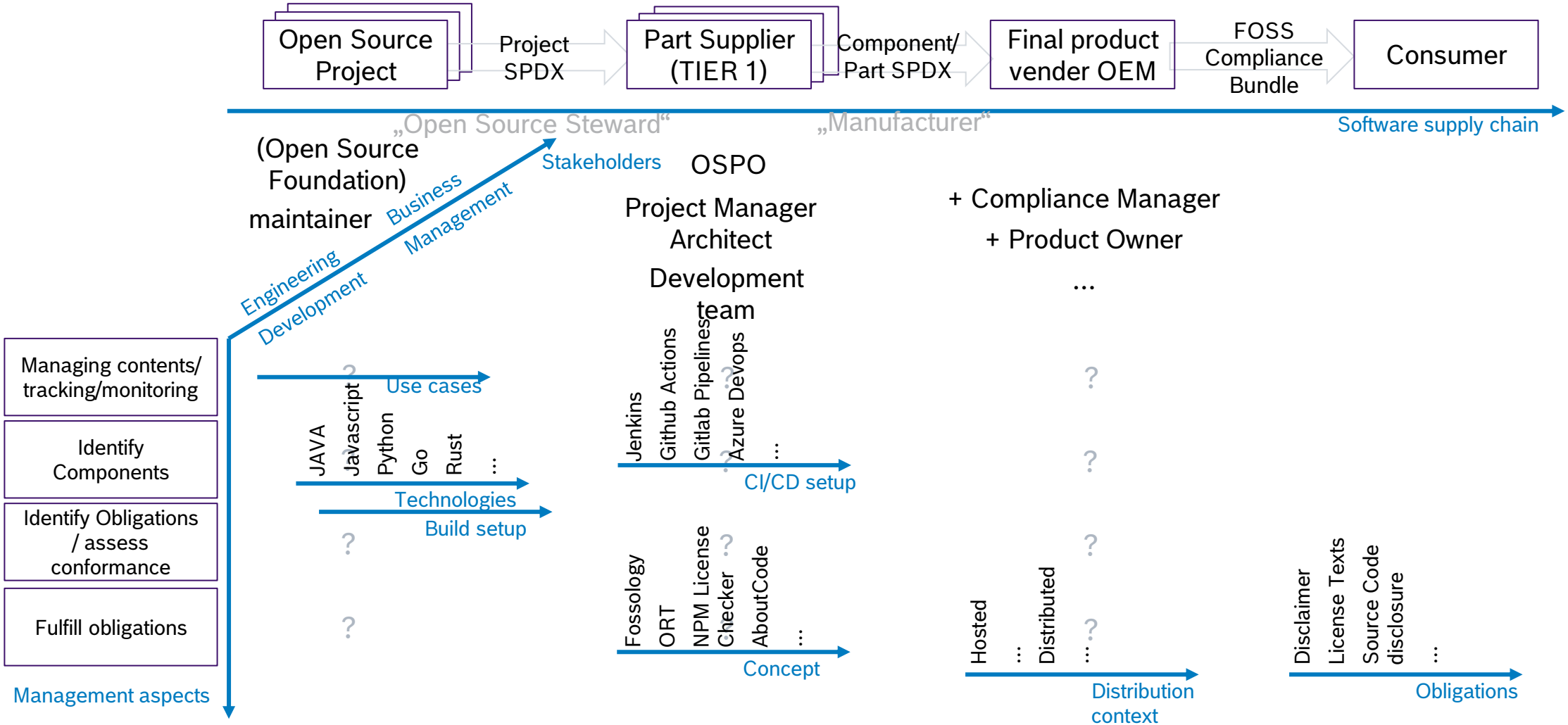
OpenChain Automation Workgroup User Stories

Usage Case* Collection

Details see :
OpenChain AutomationWorkgroup
(https://github.com/Open-Source-Compliance/Sharing-creates-value/tree/master/User-Stories)

**“Usage“ in contrast to „Contribution“ and „Champion“*

Software Supply Chain Model (simplified) – Further dimensions



Software Supply Chain Model (simplified) – Further dimensions

Open Source → Project → Part Supplier → Component/ DX → Final product vender OEM → FOSS Compliance Bundle → Consumer

Software supply chain

Example: Finding clothes online

1st limitation of search range

Women OR Men OR Kids

2nd limitation of search range

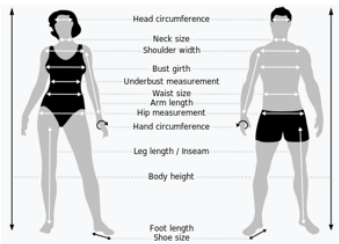
Clothing OR Shoes OR Sportswear OR ...

3rd limitation of search range

Jackets OR T-Shirts OR Pants OR ...

4th limitation of search range

Size ? Determine parameters



Source: https://commons.wikimedia.org/wiki/File:Body_measures_SVG.svg

„Fact sheet“

Size Chart
XS, S, M, L, XL

Get overview of all clothes matching to your parameters

Managing conten tracking/monitori

Identify Components

Identify Obligatio / assess conformance

Fulfill obligation

Management aspe

+ Compliance Manager

+ Product Owner

How to find the right solution?

Usage Blue-prints

?

Details will be provided soon
Blueprints = how to combine the different tools from the „toolkit“ to fit the needs best

Hoste … Distri … ?

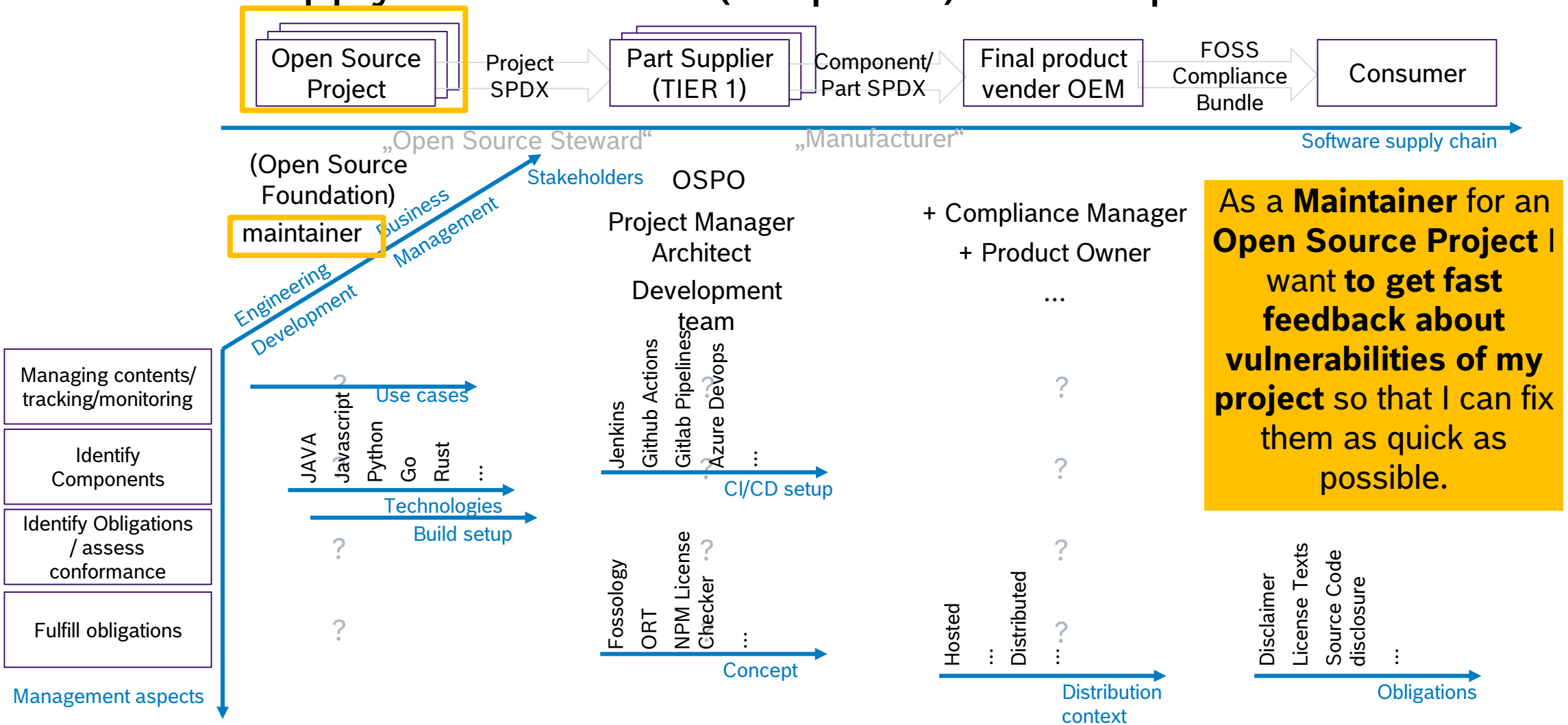
Distribution context

Discla Licen Sourc disclo …

Obligations

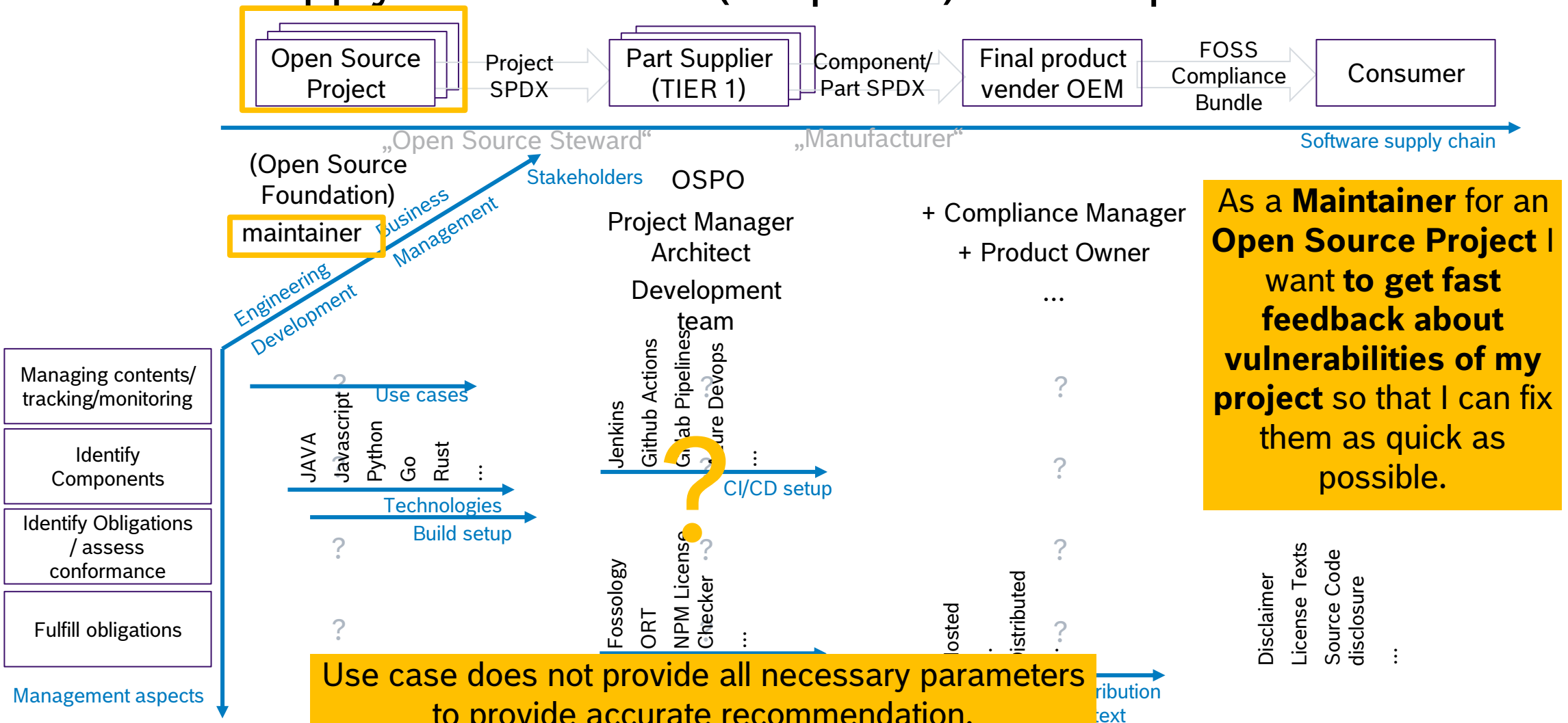
BOSCH

Software Supply Chain Model (simplified) – Example 1 - Need



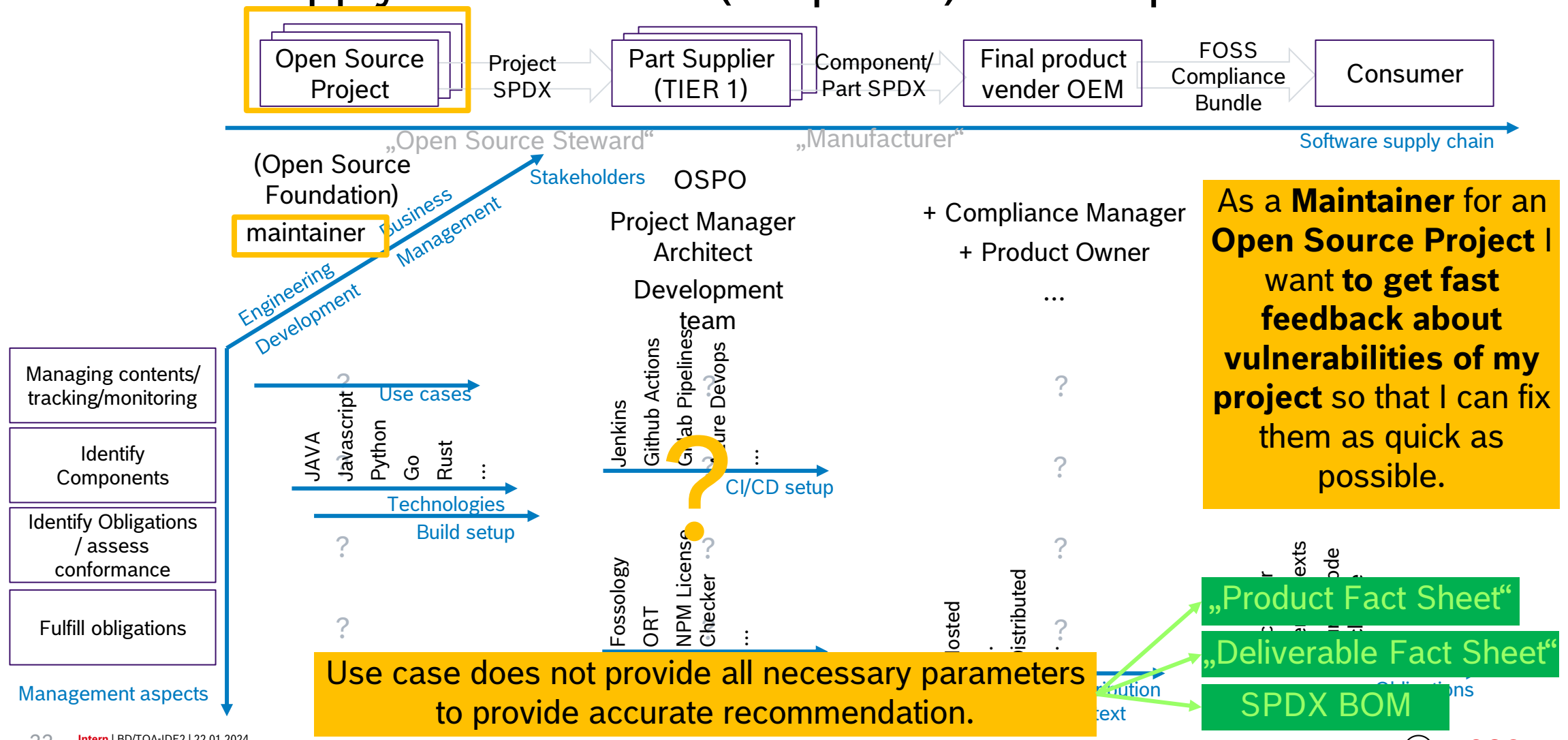
As a **Maintainer** for an **Open Source Project** I want **to get fast feedback about vulnerabilities of my project** so that I can fix them as quick as possible.

Software Supply Chain Model (simplified) – Example 1 - Need



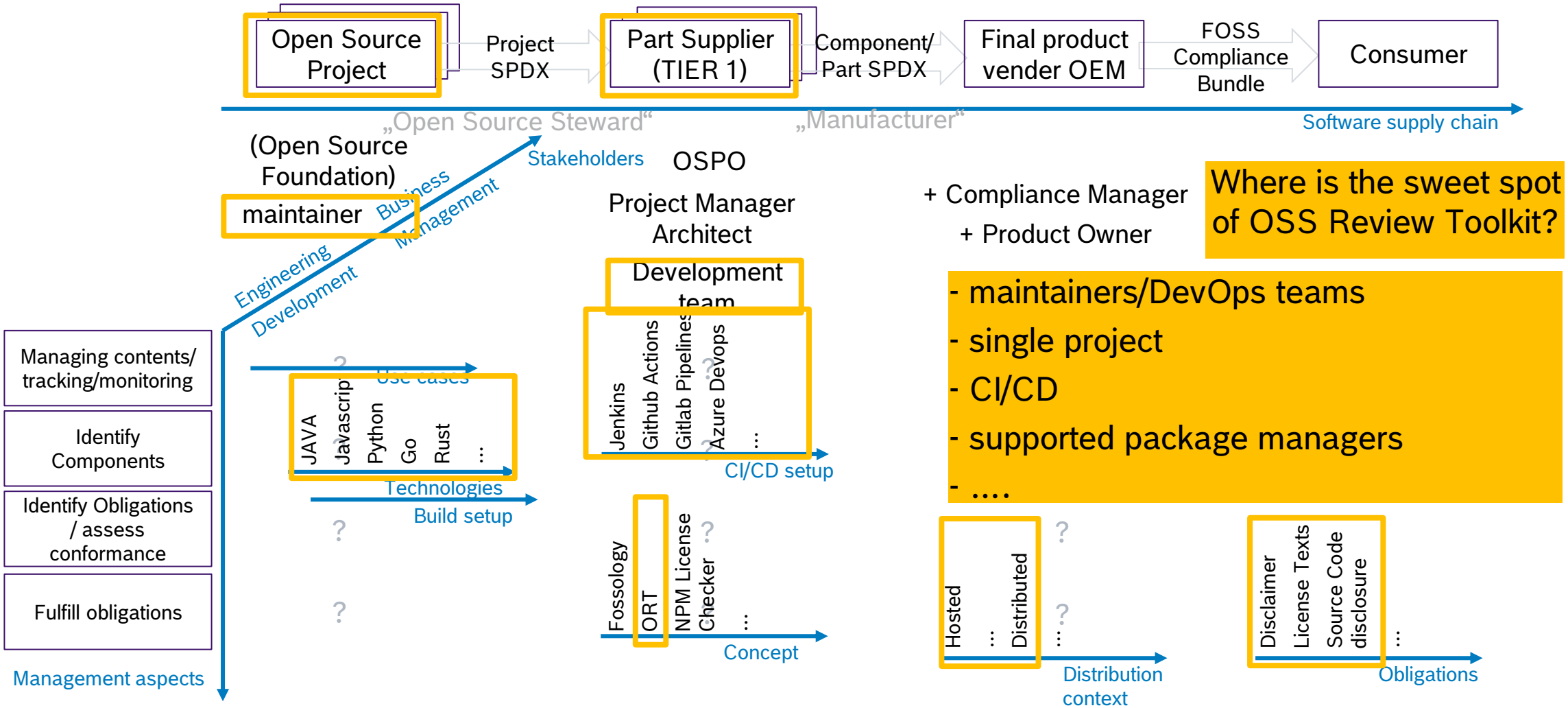
As a **Maintainer** for an **Open Source Project** I want **to get fast feedback about vulnerabilities of my project** so that I can fix them as quick as possible.

Software Supply Chain Model (simplified) – Example 1 - Need

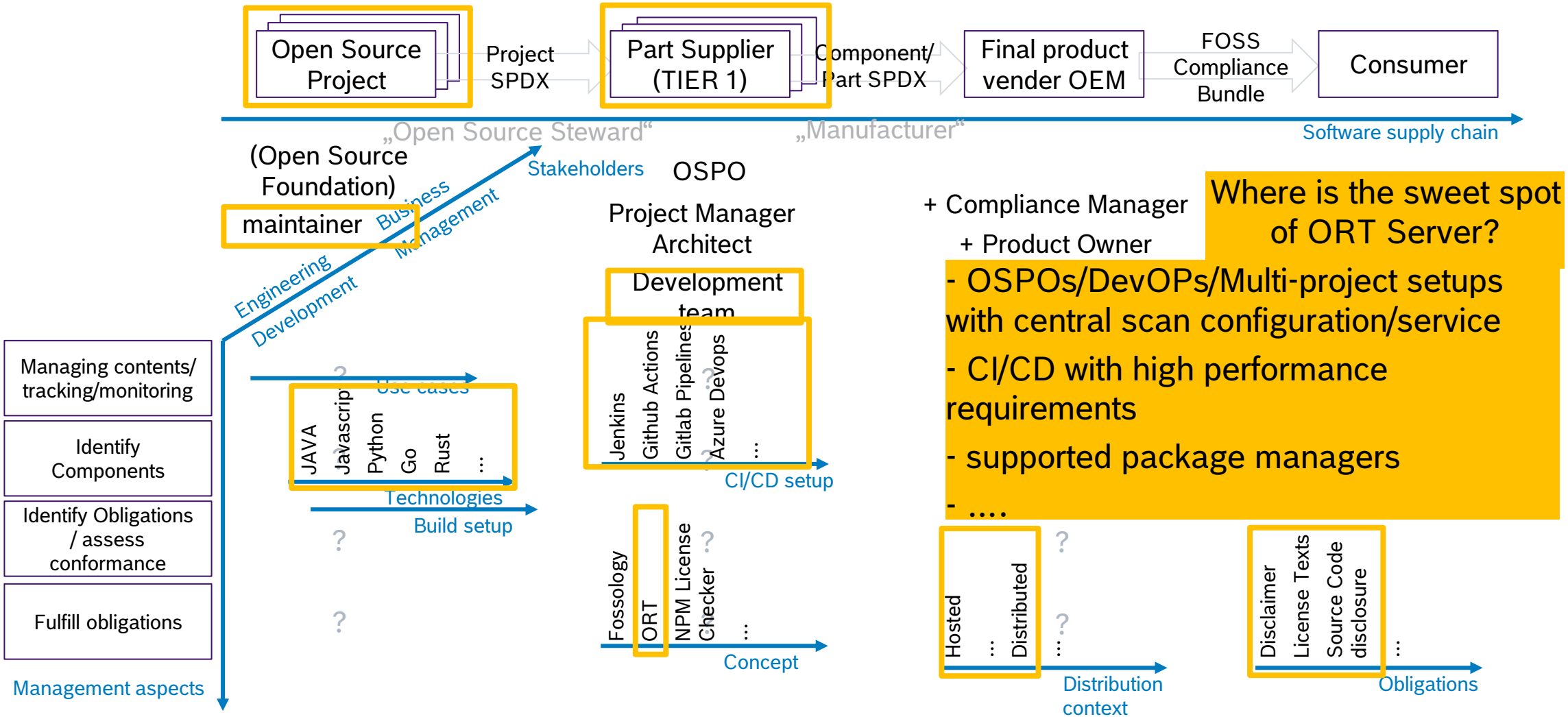


As a **Maintainer** for an **Open Source Project** I want **to get fast feedback about vulnerabilities of my project** so that I can fix them as quick as possible.

Software Supply Chain Model (simplified) – Example 2- Solution



Software Supply Chain Model (simplified) – ORT-Server

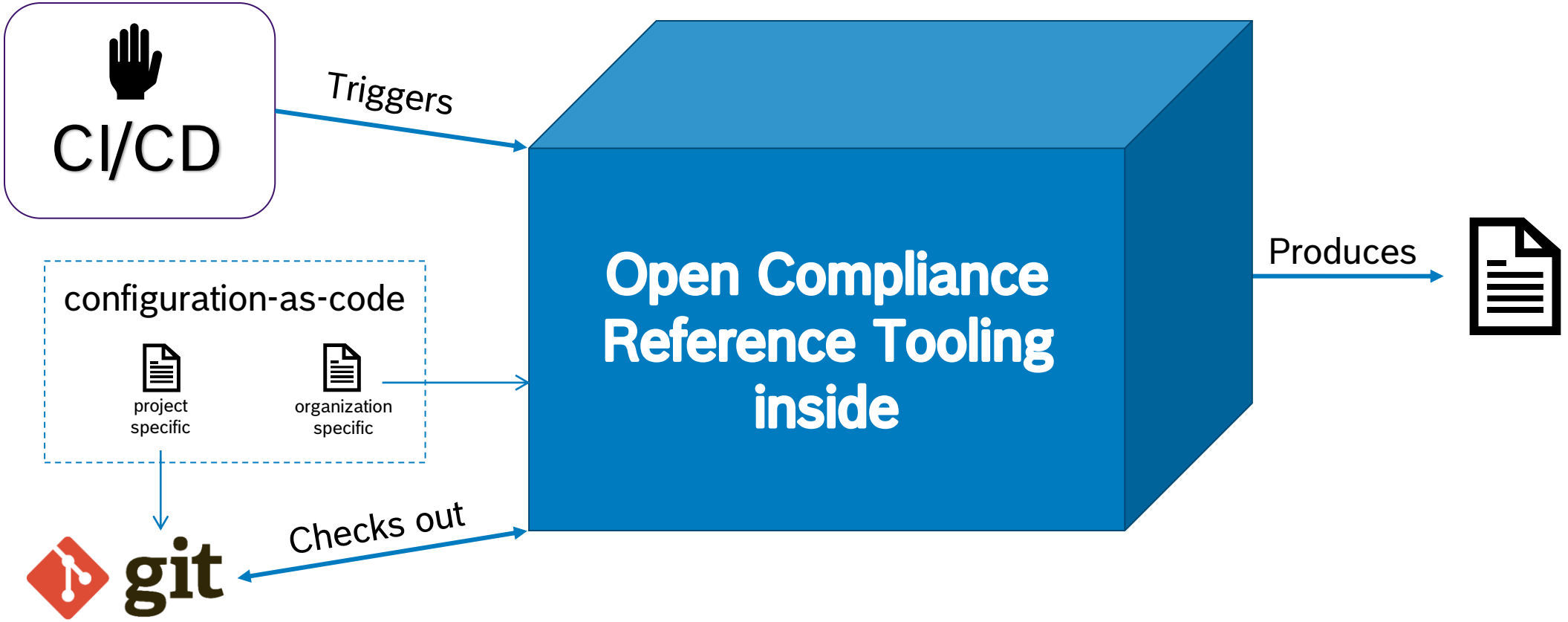


The Open Compliance Reference Tooling setup needs to fit for the respective development context .

Otherwise: “If you have a hammer, every screw looks like a nail.”

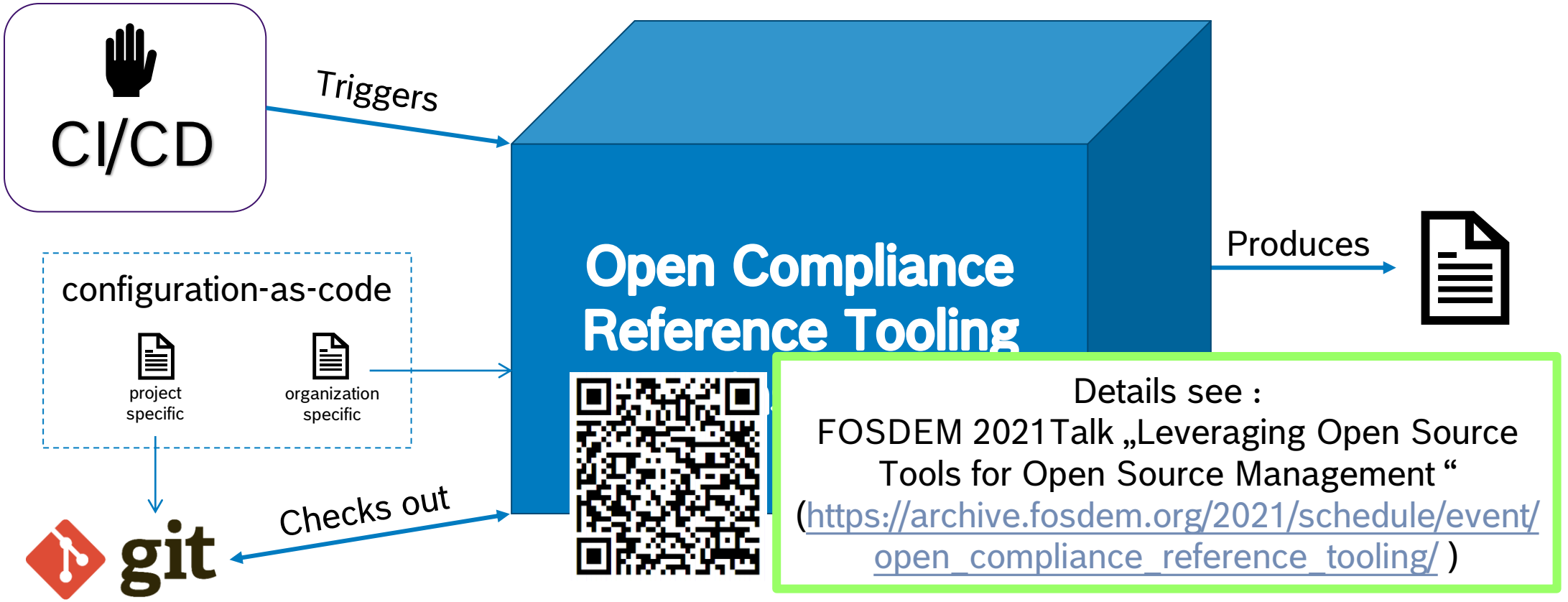
Technical Level Outputs – ORT Server

Open Compliance Reference Tooling recap
Central pipeline - How does it work?



[1] https://freebiesupply.com/logos/git-logo/

Open Compliance Reference Tooling recap
Central pipeline - How does it work?



Details see :
FOSDEM 2021Talk „Leveraging Open Source Tools for Open Source Management “ (https://archive.fosdem.org/2021/schedule/event/open_compliance_reference_tooling/)

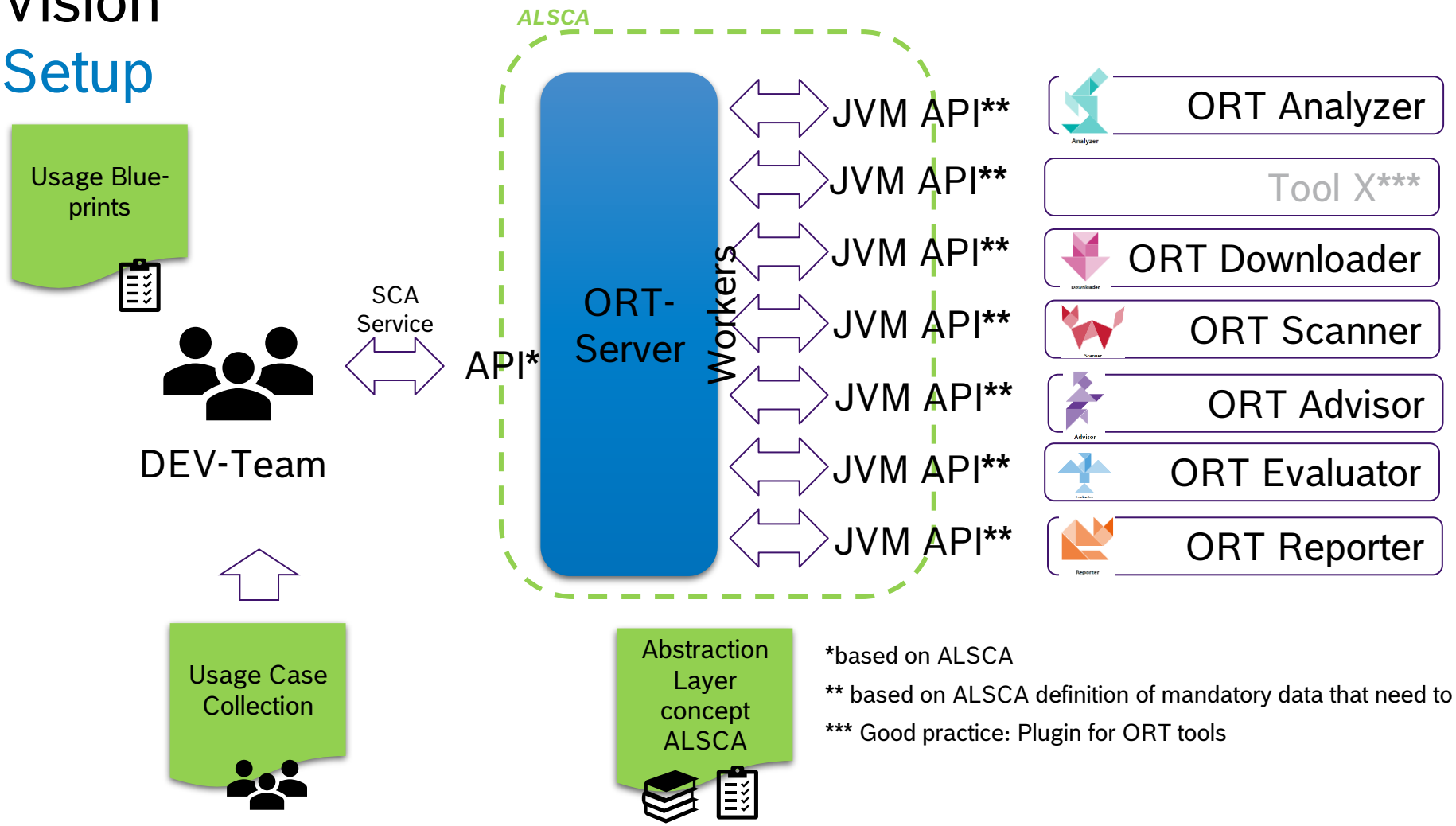
[1] https://freebiesupply.com/logos/git-logo/

Vision
ORT Server Goals

- API (REST)
- Scalable (cloud agnostic)
- Easy setup and integration
- Keep flexibility
- Web frontend => see Outlook
- Access management
- Inventory management

Vision

Setup



*based on ALSCA

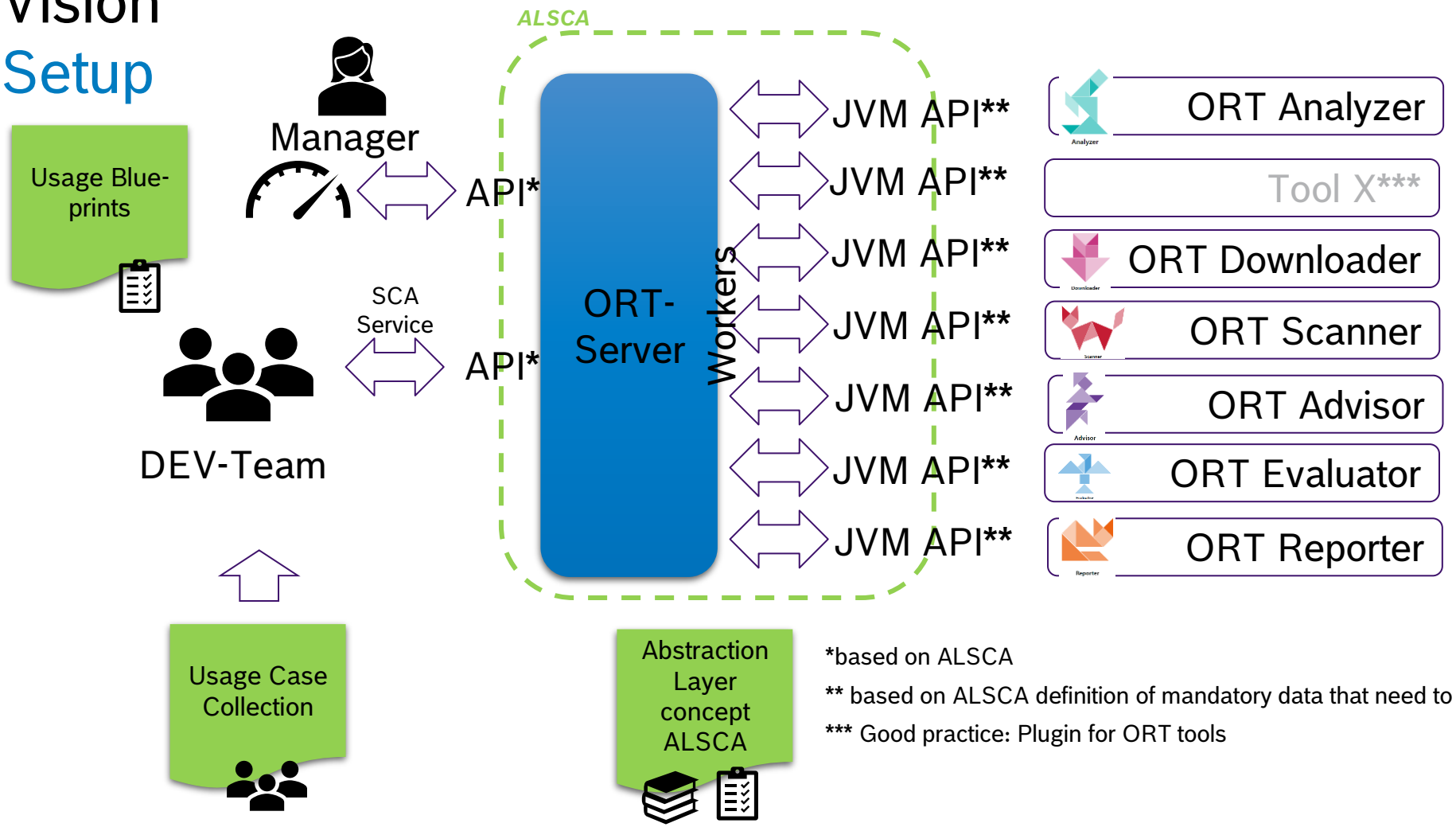
** based on ALSCA definition of mandatory data that need to be handed over

*** Good practice: Plugin for ORT tools

*Abstraction Layer for Software Composition Analysis

Vision

Setup



*based on ALSCA

** based on ALSCA definition of mandatory data that need to be handed over

*** Good practice: Plugin for ORT tools

*Abstraction Layer for Software Composition Analysis

MVP

Project Hierarchy

- Organizations
  - Products
    - Repositories
- Access management
- User management
- User configuration
  - Credential management

REST API

- Manage project hierarchy
- Trigger runs
  - Flexible configuration
- Status updates
- Generate reports
- Query data

Components

- API
- Orchestrator
  - Manage jobs
  - Prevent duplicate work
- Workers (analyzer, scanner, ...)
  - Run individual tools
  - Separate Docker images

Integrations

- Kubernetes 
- Github Action 
- OpenAPI 

MVP
Test setup with test dummies

1. Test-repositories for supported ORT-package managers
2. Schedulers by Github Actions using ORT-Server API to perform nightly scans
3. Results by Mail
4. Results via API stored in folder for post-processing
    1. Uploading SBOM to dependency track
    2. Uploading SBOM to other SBOM-consumers
    3. Creating simple „self made“ dashboards

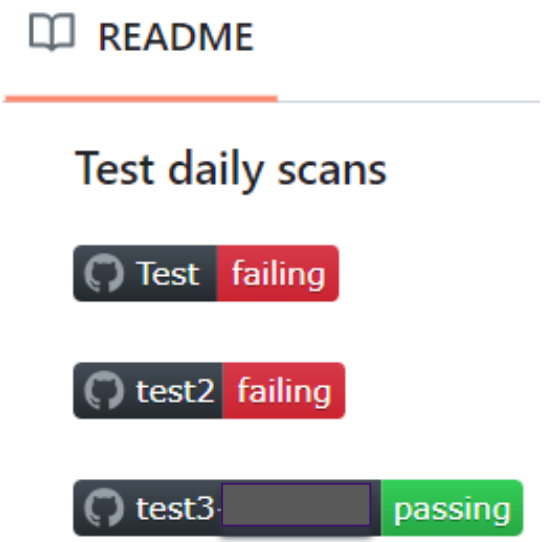
MVP
Test setup with test dummies – 1/4

1. Test-repositories for supported ORT-package managers
2. Schedulers by Github Actions using ORT-Server API to perform nightly scans
3. Results by Mail
4. Results via API stored in folder for post-processing
    1. Uploading SBOM to dependency track
    2. Uploading SBOM to other SBOM-consumers
    3. Creating simple „self made“ dashboards

SWM_C-Dummy	This project holds source code for C based Example project
SWM_CSharp-Dummy	This project holds source code for CSharp based Example project
SWM_Go-Dummy	
SWM_Java-Dummy	This project holds source code for Java based Example project
SWM_Javascript-Dummy	This project holds source code for Javascript based Example project
SWM_Python-Dummy	
SWM_Rust-Dummy	
SWM_SPDX-Dummy	OCaaS-Example-Projects-C-with-SPDX

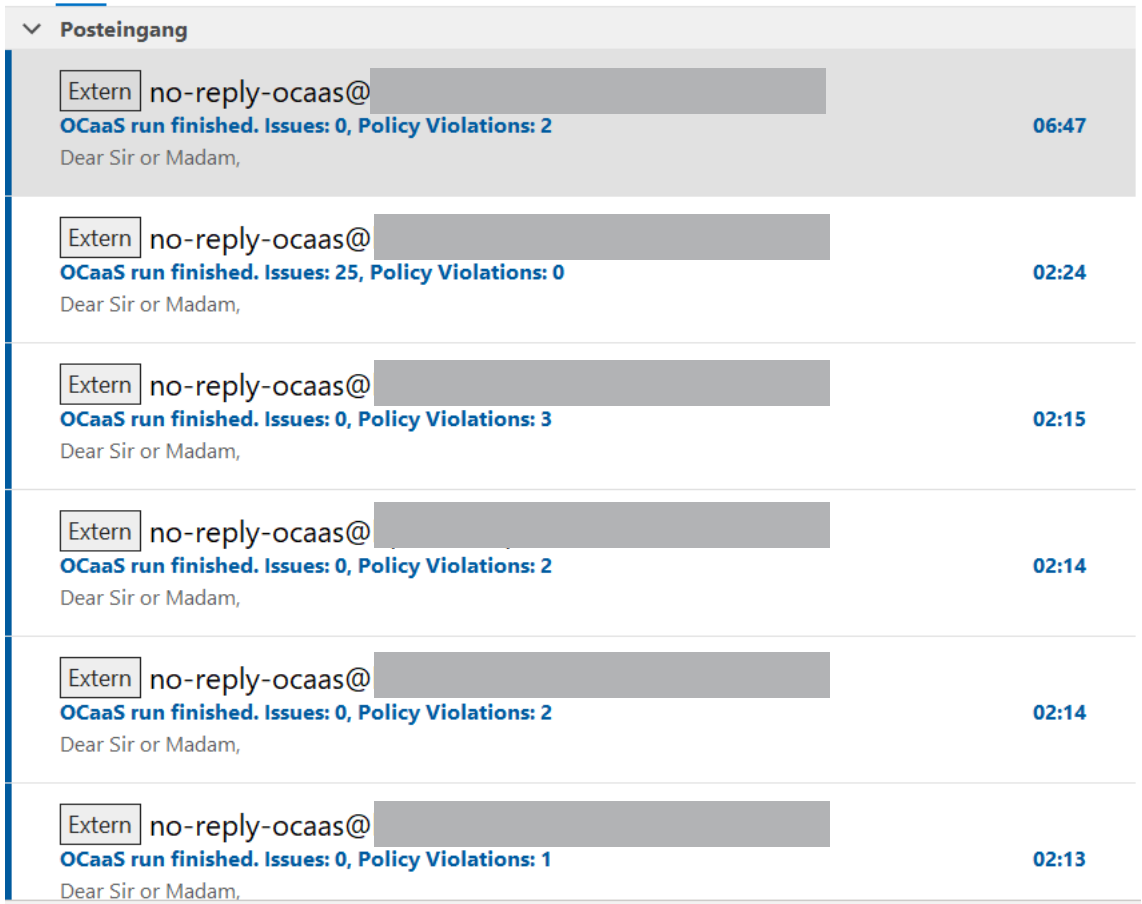
MVP
Test setup with test dummies – 2/4

1. Test-repositories for supported ORT-package managers
2. Schedulers by Github Actions using ORT-Server API to perform nightly scans
3. Results by Mail
4. Results via API stored in folder for post-processing
    1. Uploading SBOM to dependency track
    2. Uploading SBOM to other SBOM-consumers
    3. Creating simple „self made“ dashboards



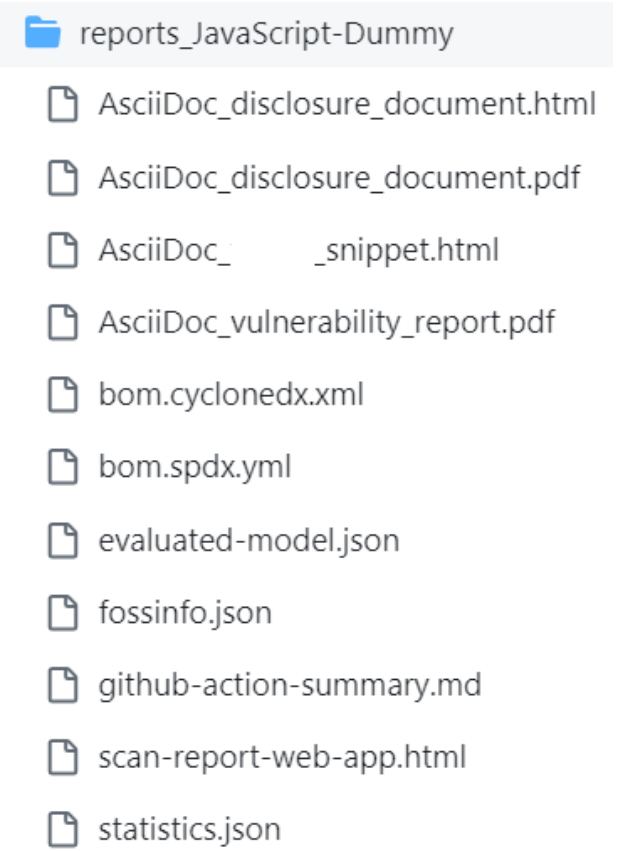
MVP
Test setup with test dummies – 3/4

1. Test-repositories for supported ORT-package managers
2. Schedulers by Github Actions using ORT-Server API to perform nightly scans
3. Results by Mail
4. Results via API stored in folder for post-processing
    1. Uploading SBOM to dependency track
    2. Uploading SBOM to other SBOM-consumers
    3. Creating simple „self made“ dashboards



MVP
Test setup with test dummies – 4/4

1. Test-repositories for supported ORT-package managers
2. Schedulers by Github Actions using ORT-Server API to perform nightly scans
3. Results by Mail
4. Results via API stored in folder for post-processing
    1. Uploading SBOM to dependency track
    2. Uploading SBOM to other SBOM-consumers
    3. Creating simple „self made“ dashboards



Next steps

- ORT-Server
  - Update/Refine of ORT-server documentation for easy onboarding and adoption
- Process level documents
  - Provision of process level documentation in dedicated repository
  - Alignment with SPDX Operations Workgroup, OpenChain Automation Workgroup and ORT-Community about working modes
- Preparation of updates for events in autumn

Outlook:

- Frontend

THANK YOU!

Open Source Tooling Group

Join Us in Creating a New Era for Open Source Compliance

Mailing List: oss-based-compliance-tooling@groups.io

Subscription page: https://groups.io/g/oss-based-compliance-tooling

Online meetings: Bi-weekly – see OpenChain Global Calendar
https://www.openchainproject.org/participate

Website: https://oss-compliance-tooling.org /

And of course we are on GitHub:
https://github.com/Open-Source-Compliance/Sharing-creates-value

BOSCH