

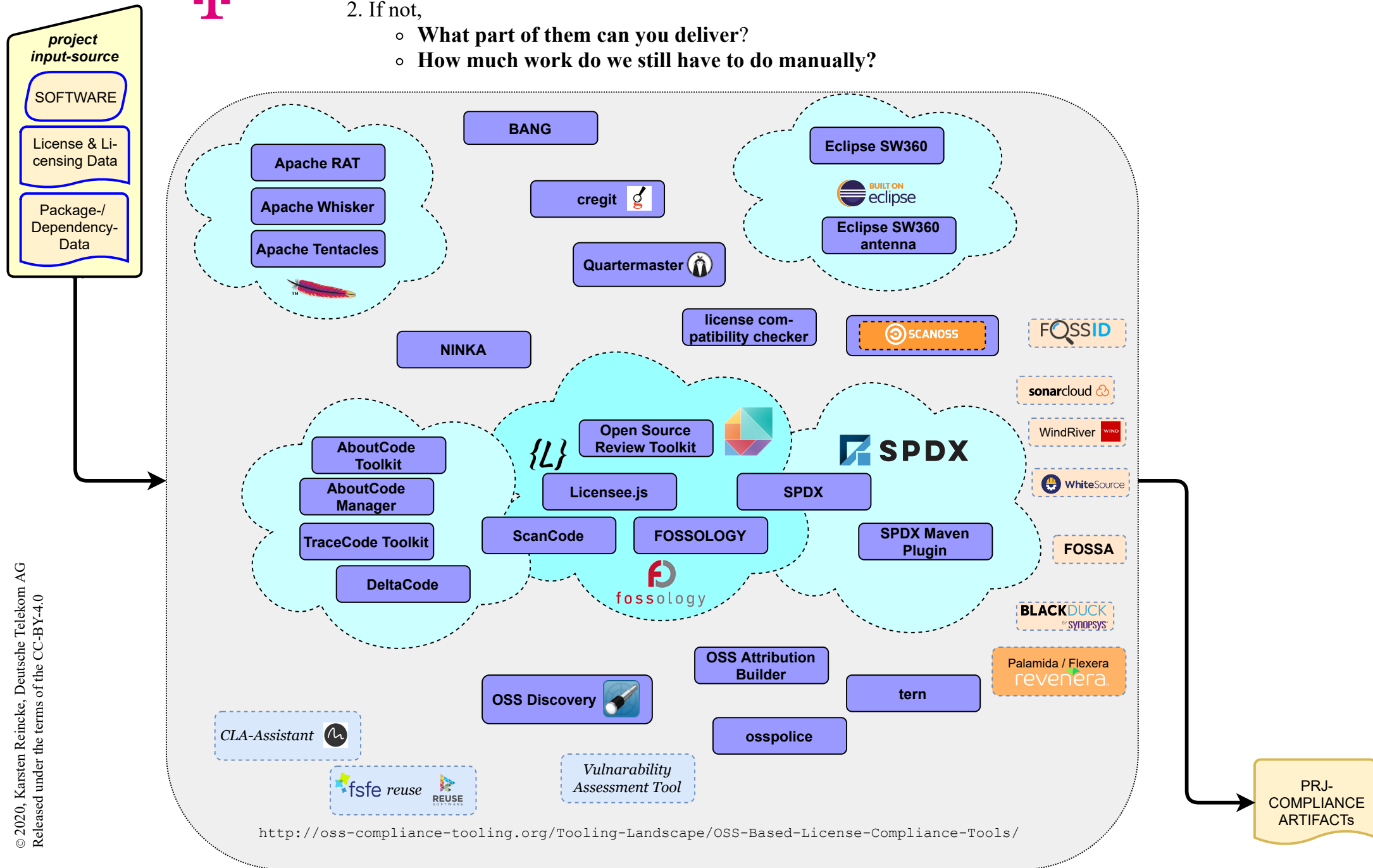
The 3 simple questions from Deutsche Telekom AG to all FOSS compliance tools and companies:



1. Do you deliver us the FOSS compliance artifacts we really need?

2. If not,

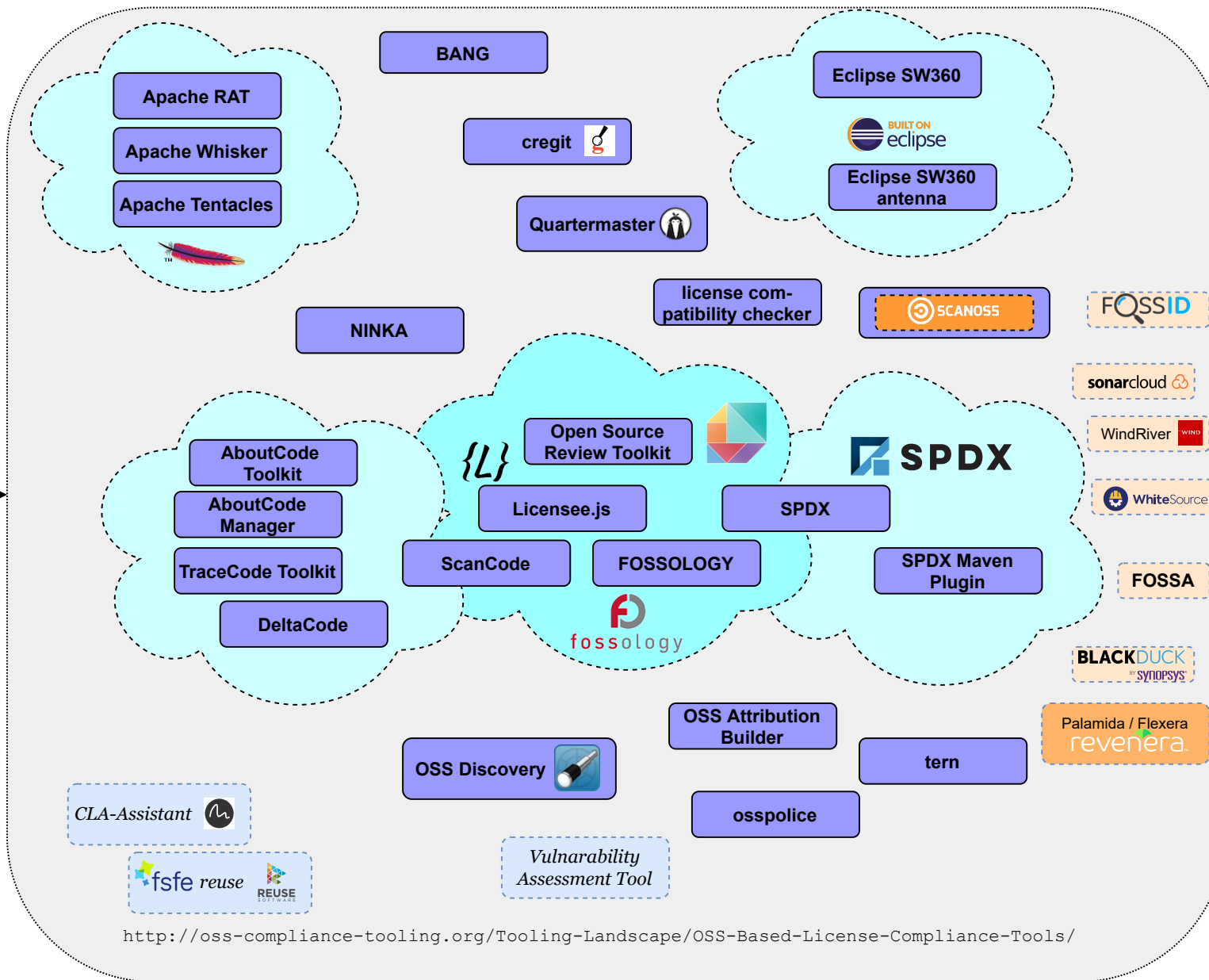
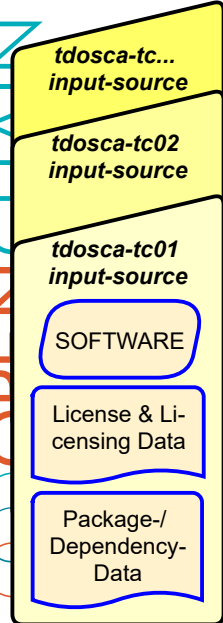
- What part of them can you deliver?
- How much work do we still have to do manually?



1. Do you deliver us the FOSS compliance artifacts we really need?

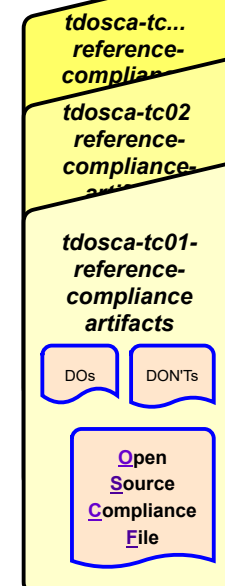
2. If not,

- What part of them can you deliver?
- How much work do we still have to do manually?



*semantically
equivalent

* ? \triangle ?



PRJ-
COMPLIANCE
ARTIFACTs TPL

tdosca-tc...
input-source

tdosca-tc02
input-source

tdosca-tc...
reference-compliance

tdosca-tc02
reference-compliance-artifacts

tdosca-tc01
input-source

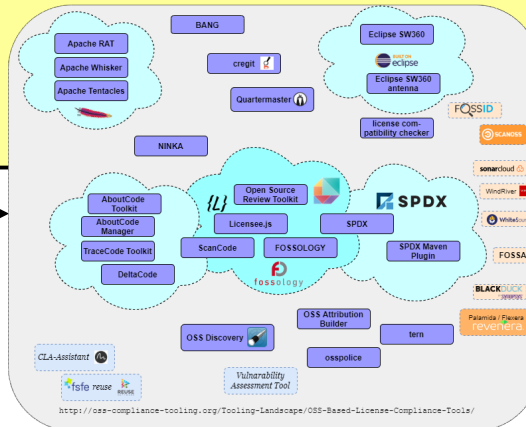
- **README.md** describes in accordance to GitHub tradition
 - the program,
 - installation
 - its usage
 - ...
- **HEALTH** documents in accordance to the programming style
- **License** documents in accordance to the license tradition
- **(Compilable) SOURCE CODE**

tdosca-tc01-meta-documents

- **README.md** describes
 - the testcase
- **compliance-traps.md** defines
 - particular challenges embedded into the test case

tdosca-tc01-reference-compliance artifacts

- **bom.csv** = bill of material FOSS embedded into the package
- **dos.yaml** = still open tasks to get a compliant package
- **donts.yaml** = forbidden actions to get a compliant package
- **oscf.md** = Open Source Compliance File which makes the package compliant*
- **prerequisites.csv** = list of the FOSS which must be pre-installed but is not part of the package



*semantically
equivalent

* ? ≡ ?

*An unfortunately not entirely appropriate, but very illustrative and stimulating example:

https://www.cisco.com/c/dam/en_us/about/doing_business/open_source/docs/CiscoJabberforWindows-128-1578365187.pdf

PRJ-
COMPLIANCE
ARTIFACTs TPL



Applying **TDOSCA** to Open-Source Review Toolkit

tdosca-tc...
input-source

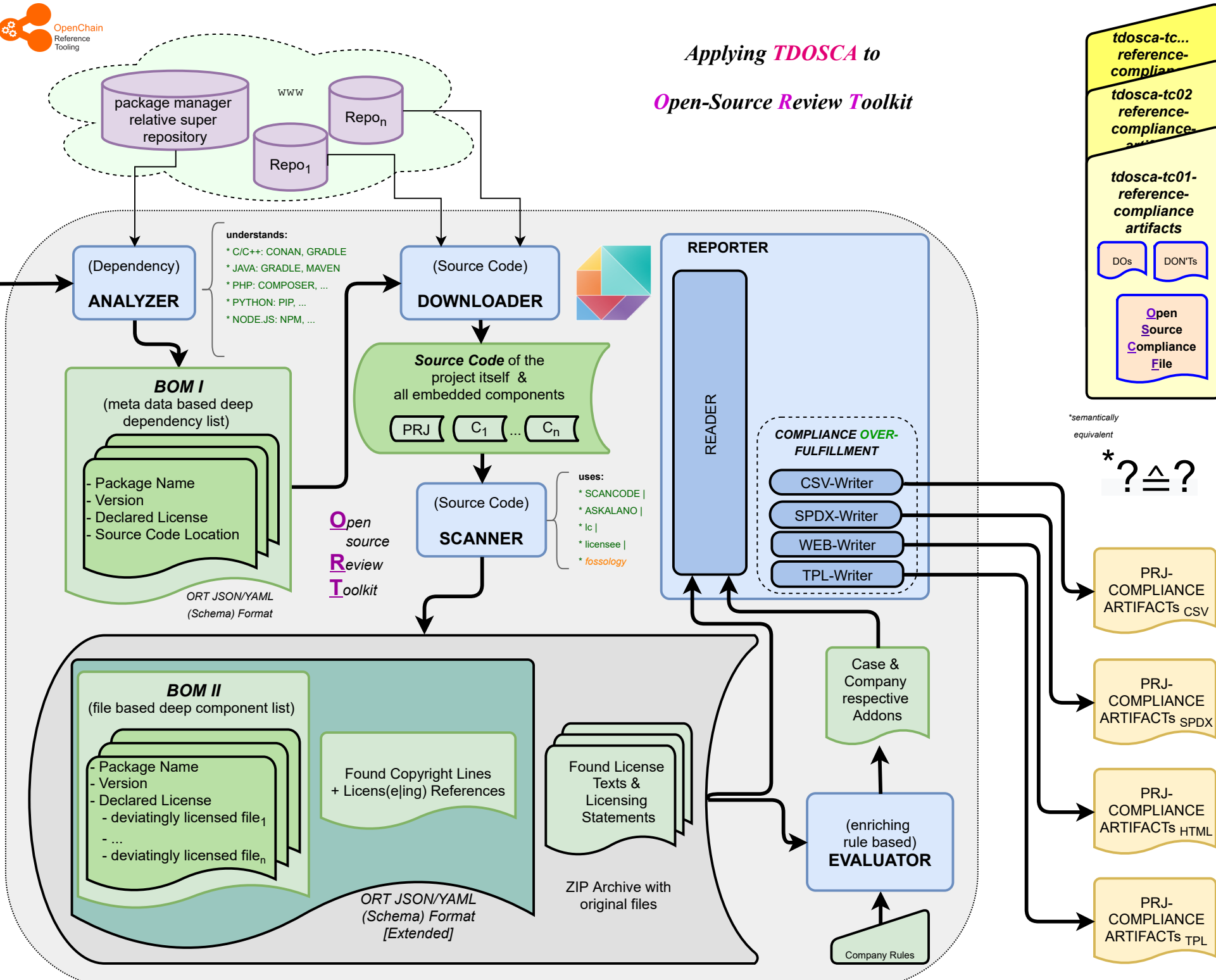
tdosca-tc02
input-source

tdosca-tc01
input-source

SOFTWARE

License & Li-
censing Data

Package-/
Dependency-
Data



*semantically equivalent

* ? ≐ ?



Experiences with **TDOSCA** and the **Open-Source Review Toolkit**

tdosca-tc...
input-source

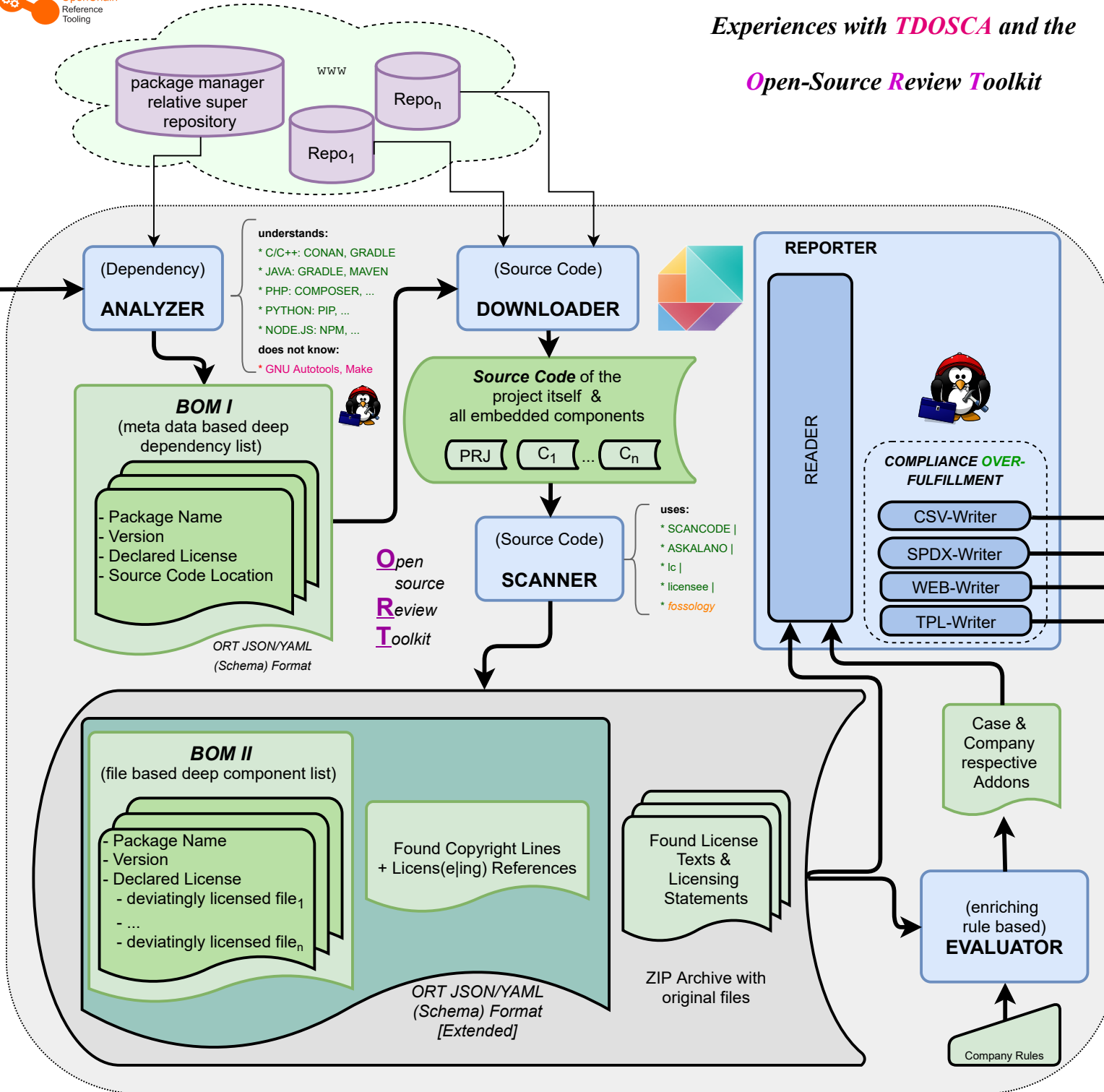
tdosca-tc02
input-source

tdosca-tc01
input-source

SOFTWARE

License & Li-
censing Data

Package-/
Dependency-
Data



tdosca-tc...
reference-compliance

tdosca-tc02
reference-compliance

tdosca-tc01-
reference-compliance
artifacts

DOs

DON'Ts

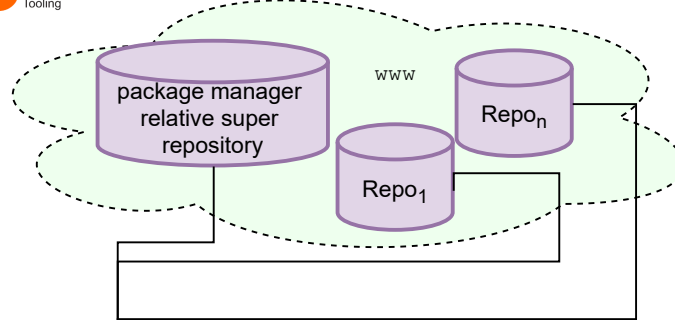
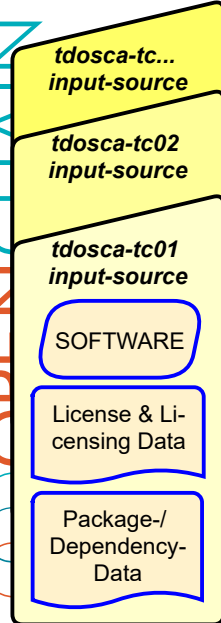
Open
Source
Compliance
File

*semantically equivalent

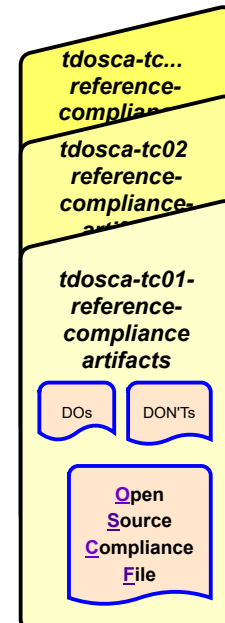
* ? ≙ ?



Hence: Create a multi-dimensional set of test cases

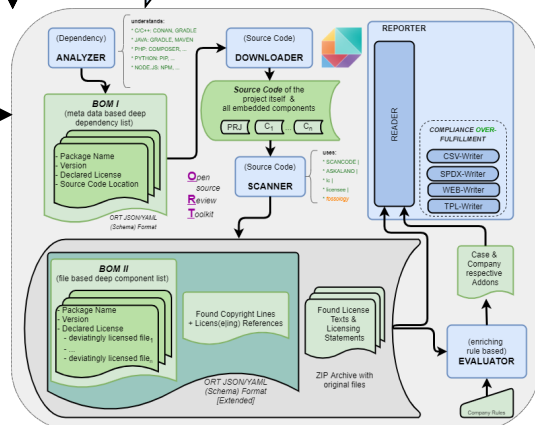


	C/C++	C/C++	C/C++	JAVA	JAVA	PHP	PYTHON	NODE.Js
	Gradle	Conan	GNU autotools	Gradle	Maven	Composer	PIP	NPM
simple hello world			tc01					
plain hello mworld			tc02					
complex hello world								



*semantically equivalent

* ? \triangle ?



PRJ-COMPLIANCE ARTIFACTs CSV

PRJ-COMPLIANCE ARTIFACTs SPDX

PRJ-COMPLIANCE ARTIFACTs HTML

PRJ-COMPLIANCE ARTIFACTs TPL

