

Open Source distributions in a cloud-native world

From a technical to a legal point of view

Dr. Lina Böcker
Angelika Wittek

October 17, 2023



Who are we?



Dr. Lina Böcker
Rechtsanwältin / Partnerin
Fachanwältin für Informationstechnologierecht
Osborne Clarke Germany
lina.boecker@osborneclarke.com



Angelika Wittek
Independent OSS Consultant
angelika.wittek@konteno.de

What is License Compliance all about?

- Software is protected under Intellectual Property (IP) Law
 - Includes: copyrights, patents, business secrets (know-how) ...
- Each developer holds the IP rights to their code: § 69a Sec. 2 of the German Copyright Act
- Yes: YOU hold the IP rights to whatever YOU code. Does not need to be a fully fledged contribution or “aesthetic” or “good” (§ 69a Sec. 3 of the German Copyright Act)
- Protection of this intellectual property: If anyone wants to use your code, they will need permission
 - statutory
 - contractual

What is License Compliance all about?

- Statutory permission: e.g. for certain educational or journalistic purposes
- Contractual permission: license

Licensing = Allowing others to use your work for certain purposes under certain conditions

- Adhering to these conditions = License Compliance

Why do I have to think about License Compliance?

- Respecting the wishes of the authors of code
- Violating the terms of a license can result in copyright infringement, which is regulated by law
- If the intellectual property owner learns that you have not complied with their license, you may lose the right to use the software and may be held liable
- If you run a business, you may have a special legal duty to ensure that all legal and contractual obligations are met

What kinds of Open Source Licenses are there?

- Strong Copyleft (GPL) / weak Copyleft (EPL) / Permissive (Apache)
 - Some licenses are mutually exclusive
 - The same software code may use Open Source code components from different Open Source libraries. These components may be under different licenses with different license terms and conditions
- ⇒ To avoid copyright infringement, only code components under licenses that are compatible with each other should be used

And do I always have to think about it, or only under certain circumstances?

- Holy grail: **distribution**
 - Technical term from Copyright Law
 - **Caution: May vary in different jurisdictions**

- What IS distribution?
 - Classical sense: when the code leaves your computer
 - “Making available to the public”, e.g. cloud hosting + remote access (e.g. SaaS)

And do I always have to think about it, or only under certain circumstances?

- “Distribution” according to the Open Source Community:

OSI FAQ	“usually understood to mean ‘deliver copies in source code and / or binary form’”
EPL	“the acts of a) distributing or b) making available in any manner that enables the transfer of a copy”
Apache, GPL	no definition → must be interpreted according to applicable Copyright Law

And do I always have to think about it, or only under certain circumstances?

- New (sort of) kids on the block: **to propagate & to convey**
- Idea: more precise than the ambiguous “distribute”
- Legal certainty: terms defined in the license itself rather than interpreting them
- Most common definition: GPL v.3.0, also used in SSPL

And do I always have to think about it, or only under certain circumstances?

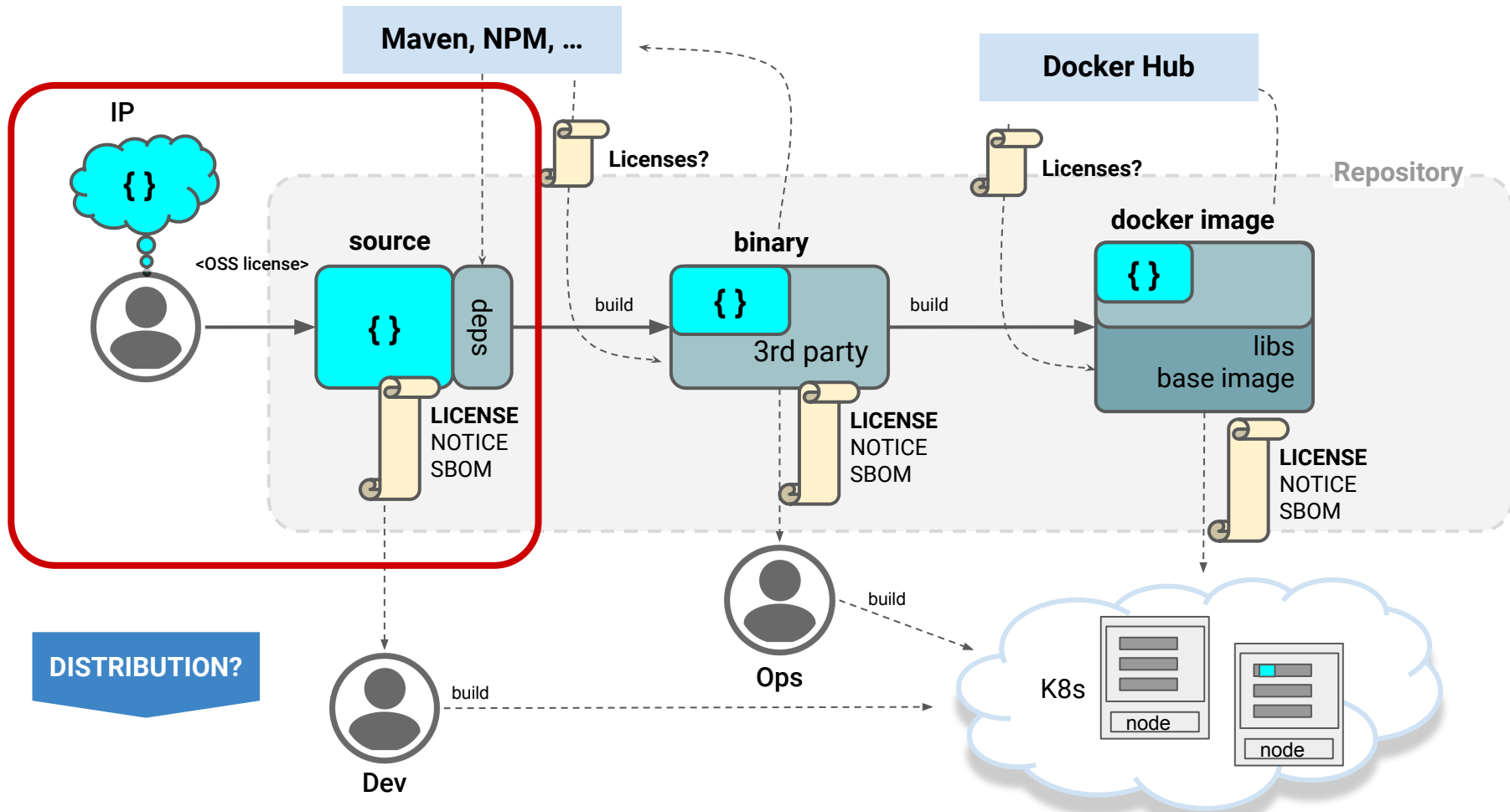
GPL v.3.0:

- **to propagate** = to do anything with a work that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law
 - includes: copying, distributing with or without modification, making available to the public, ...
 - not: executing the work on a computer, modifying a private copy (create backup, install on multiple computers...)
- **Means:** you can do with your own copy in private whatever you want. As soon as you leave your private space or give others access to it, you are propagating the work and must obey the license

And do I always have to think about it, or only under certain circumstances?

GPL v.3.0:









- **to convey** = any kind of propagation that enables other parties to make or receive copies
 - includes: remote access
 - e.g.: uploading to GitHub
- **Means:** additional protection against the work being obtained or modified without the owner's knowledge or consent



Example: Eclipse project repository











eclipse-tractusx

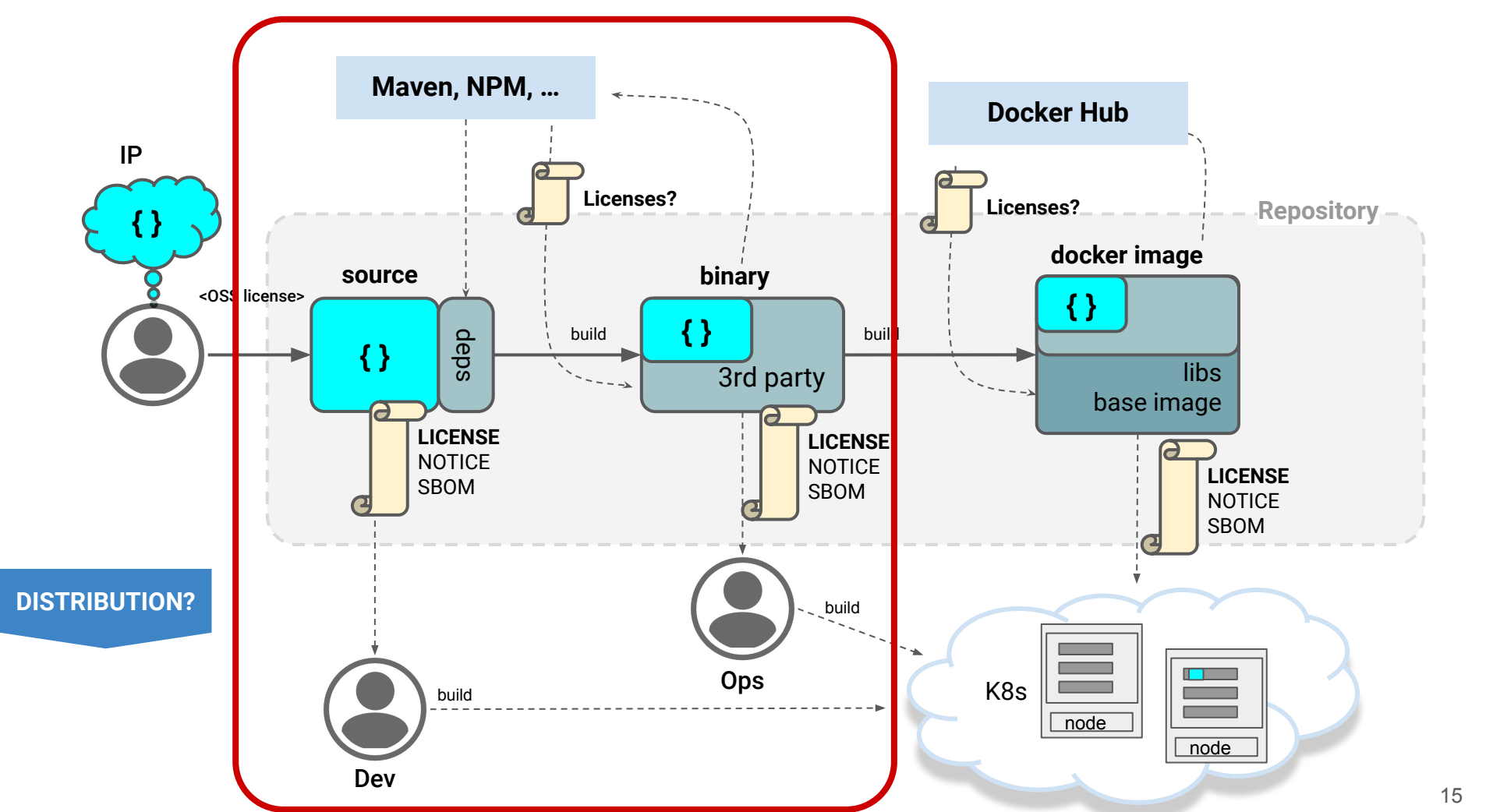
 AUTHORS.md	docs: remove/fix broken links
 CODE_OF_CONDUCT.md	cho
 CONTRIBUTING.md	doc
 DEPENDENCIES	cho
 LICENSE	cho
 NOTICE.md	cho
 README.md	Upd
 SECURITY.md	Rev

About

 eclipse-tractusx.github.io/sig-release

-  Readme
 -  Apache-2.0 license
 -  Code of conduct
 -  Security policy
 -  Activity
 -  1 star
 -  9 watching
 -  5 forks
- Report repository

every file (where applicable):
copyright & license header

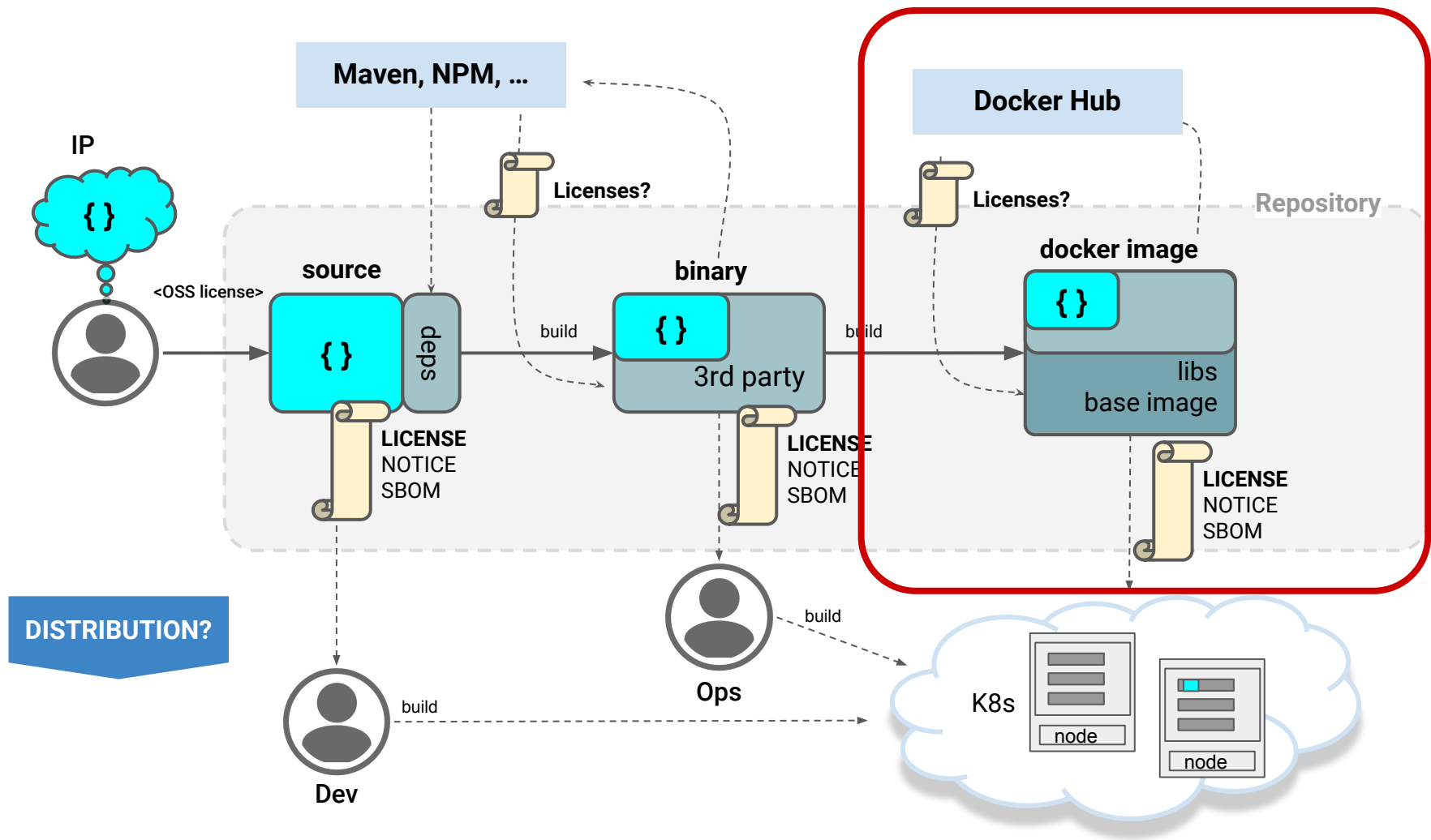


What should you consider when publishing artifacts?

- Have processes for OSS Governance and implement them in your project
 - Eclipse Foundation Development Process
 - Eclipse Foundation Project Handbook
- Scan your code / perform IP checks for 3rd party content (transitive closure)
 - Eclipse IP GitLab, Eclipse Dash Tool
- Add legal information to distributions
- Add legal notice for end user content

What should you consider when publishing artifacts?

- Use (release) tags for traceability
- Build frequently
- Automate build process
- Using tools like Dependabot:
be aware that updates to 3rd party libs also need IP clearance



Publishing Docker Images - Docker Official Images

The Docker Official Images are a curated set of Docker repositories hosted on Docker Hub.

These images provide essential base repositories that serve as the starting point for the majority of users.

E.g.

- alpine
- eclipse-temurin
- NGINX
- postgres

Example: alpine's License Notice

View [license information](#) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, along with any direct or indirect dependencies of the primary software being contained).

Some additional license information which was able to be auto-detected might be found in [the repo-info repository's alpine/ directory](#).

As for any pre-built image usage, it is the image user's responsibility to ensure that any use of this image complies with any relevant licenses for all software contained within.

What should you consider when publishing docker images?

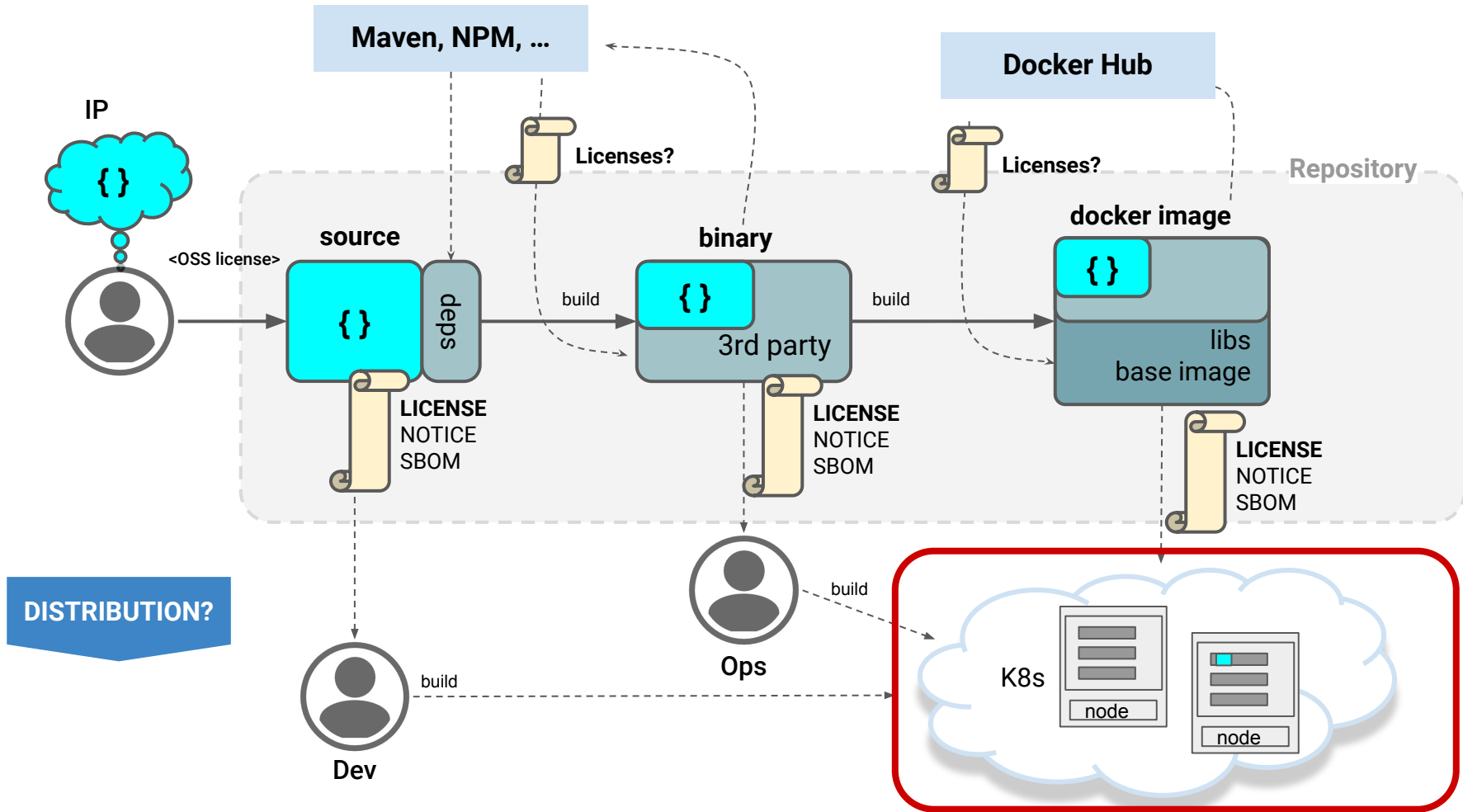
- Why do you need it?
- Define a process and implement it in your project
- Perform image scans and publish the package lists
- Publish a README, add a LICENSE section:
 - References to Dockerfile, package lists, your sources, base image
 - Good example: Official Docker Images
- Recommendation - Pack into your image:
 - README, Dockerfile
 - for better traceability: add your legal documentation (LICENSE, SBOM, NOTICE, SECURITY)

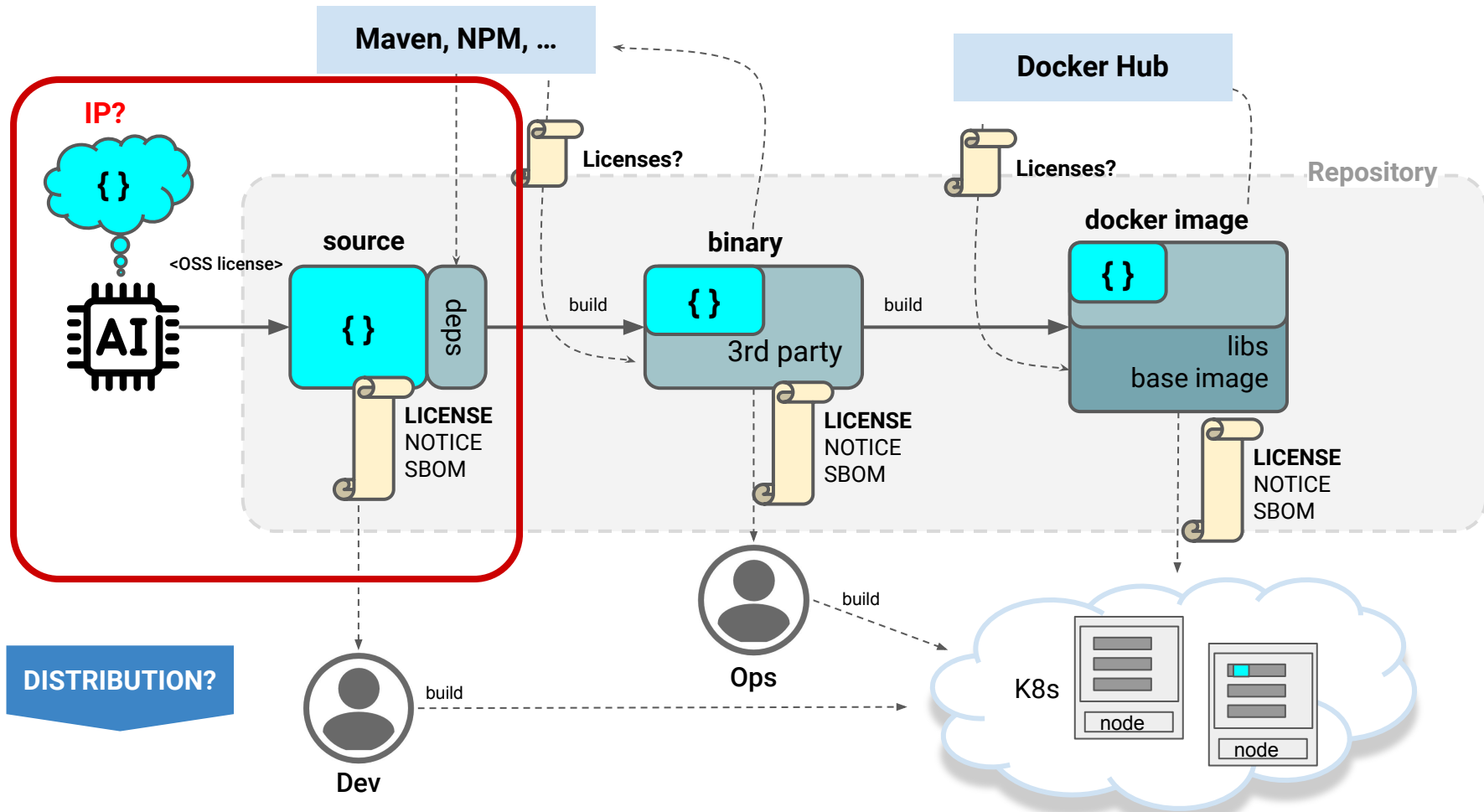
⇒ Make it as transparent and traceable and reproducible as possible!

Dockerfiles best practices

- Keep your images small
- Use “trusted” base images
- Avoid using e.g. *apt update/upgrade*
- Avoid using *latest* tag

⇒ Make it as transparent and traceable and reproducible as possible!





Use of AI generated code in OSS projects

Statement from organizations (October 2023):

- GitLab: *"Due to restrictions in third-party AI vendor agreements, we cannot accept AI-generated contributions at this time."* <https://about.gitlab.com/community/contribute/dco-cla/>
- Eclipse Foundation: *"Note that our current policy is that we do not accept contributions generated by an AI as doing so is – at least – in conflict with the ECA/DCO (note that an AI cannot sign the ECA)."*
- Stack Overflow: *"Why posting GPT and ChatGPT generated answers is not currently acceptable"* <https://stackoverflow.com/help/gpt-policy>
- Investigation on GitHub Copilot <https://githubcopilotlitigation.com/>

CRA & PL (October 2023)

- Cyber Resilience Act (EU): European cyber security law in the making
- Expected to include regulations regarding Open Source Software
- However: extent remains unclear, FOSS-exception still vague

Questions?

Contact us:

Dr. Lina Böcker lina.boecker@osborneclarke.com

Angelika Wittek angelika.wittek@konteno.de

THANK YOU FOR ATTENDING!

Evaluate the Sessions

- Please help by leaving feedback on the sessions you attend!
- To rate a session, you must be registered for it in Swapcard BEFORE the talk starts.
- Swapcard will prompt you to leave feedback after the end of each session.
- You may also rate a talk by locating the session from the “Agenda” or “My Event” buttons on the Swapcard Event Home page. Click on the session and look for the “Give your feedback” box.