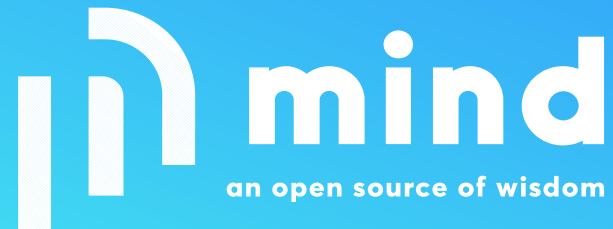


Build systems, traceability and compliance

Arnout Vandecappelle

LPC 2024 – Build System MC



Who is Arnout

Embedded software architect

Focus on Linux OS integration

Mind consultant since 2008

Worked for 40+ customers in multimedia, security, home automation, satellite, telecom, chips, ...

Buildroot maintainer (team of 5)



Agenda



- What is traceability and compliance and why do we need it?
- What can we (build systems) do to improve?

Compliance = knowing what is in the software you distribute



Software Distribution Triggers Obligations

If you integrate OSS in a product that you *distribute*, you must comply with the *inbound* OSS licenses and with the Law:

- know the OSS you are using (Software Composition Analysis)
- identify the *inbound* licenses and calculate possible compatible *outbound* license(s),
- identify and handle legal risks and obligations related to OSS licenses, IP rights, cybersecurity regulations (soon, CRA), etc.
- put in place a process to ensure continuous compliance
- produce artifacts to comply and to demonstrate compliance to your downstream customers in the supply chain and to the authorities

More complex if you have a downstream



Current: produce an image → give to end user

Not sufficient if image is re-composed by downstream

- **A** - Platform image to which downstream adds software
- **B** - Container image composed with others
- **C** - SDK to build appliance
- **D** - ...

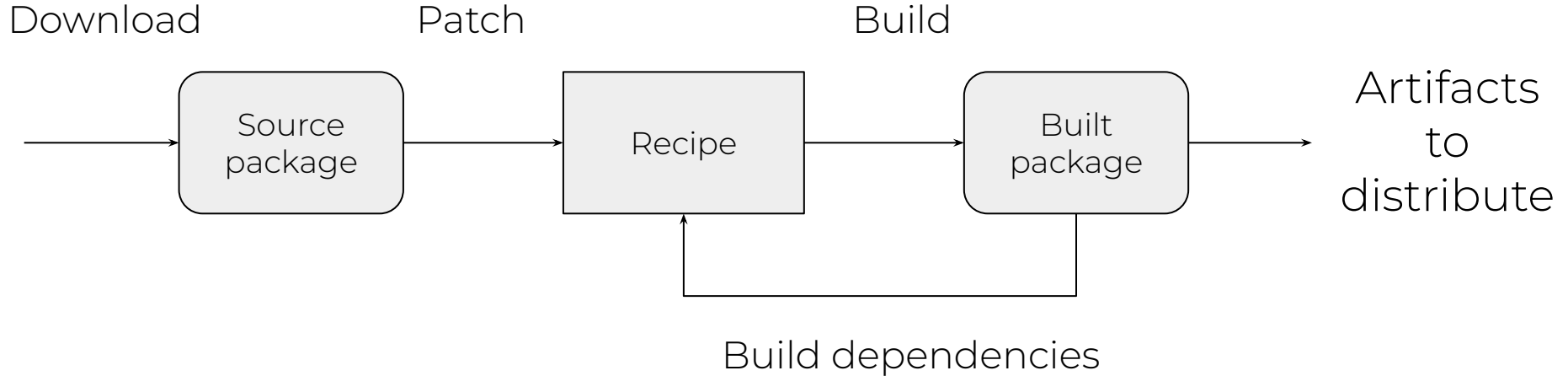
Who sees one of the use cases above?

What does this mean concretely?



- Manifest that describes distributed components
- Decomposition
 - E.g. image consists of packages; package consist of files
- Provenance of each component
 - E.g. built from these sources
 - Sources downloaded from that URL at that time
 - Patched with these patches
- License of each component
 - License information is only accurate for source files
 - Concluded license needs some form of review
- Traceability of who concluded this

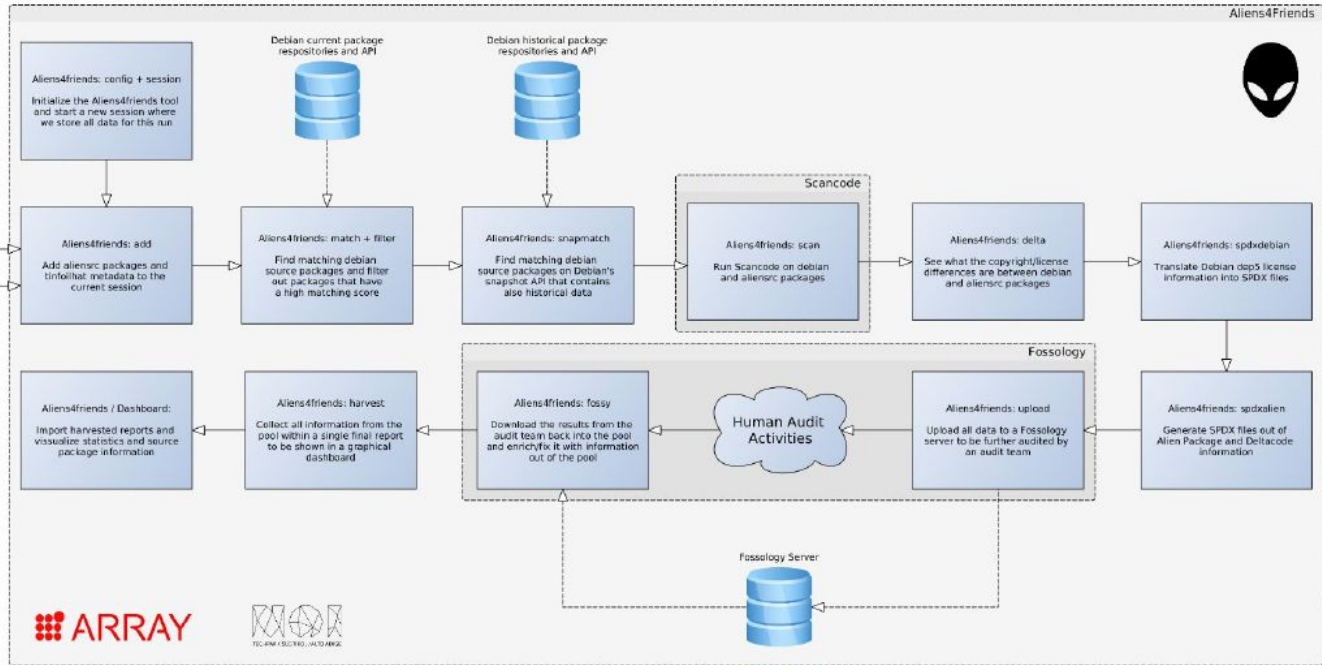
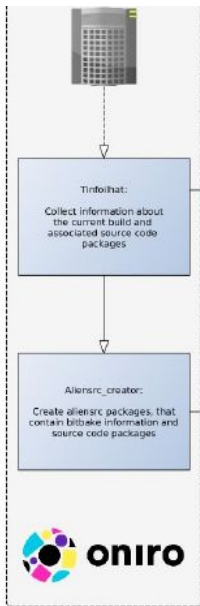
Build system terminology



Additional workflow on top of yocto for full compliance



yocto
PROJECT



What can we do to improve the situation



- Collect per-file info
 - Every source file with its license
 - Every compiled / copied file with its (concluded) license
- Use upstream package's info directly
 - Instead of license info in recipe
 - E.g. cargo-spdx, reuse spdx
- Include per-file info in recipe
 - E.g. in SPDX or DEP5 or REUSE format
- Remove unused sources
 - To make concluded license more believable

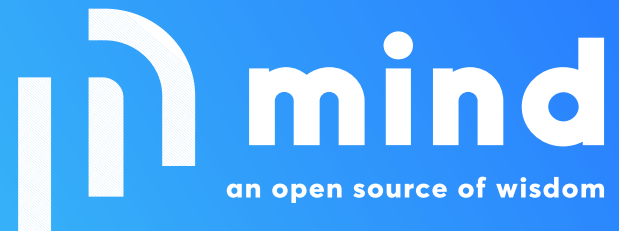
What can we do to improve the situation



A group of five people are gathered around a large monitor in a meeting room. The monitor displays a web page with the word "bootlin" and a list of code snippets. The people are looking at the screen with interest. The room has large windows in the background.

Thank you for listening.

Questions ?





Division of **Essensium nv**

Arenberg Science Park
Gaston Geenslaan 10
3001 Leuven
Belgium
+32 16 28 65 00

General enquiries
info@mind.be

Employment enquiries
jobs@mind.be

