ARRAY

Linaro Connect
MADRID 2024 | MAY 12-17 2024

Implementing an Openchain Compliant Policy and Best Practice at Linaro

by Alberto Pianon and Carlo Piana, Array

Madrid, May 15, 2023

Context

What and Who (1)

- an Eclipse Foundation project for an all-scenario embedded OS platform:
  - multiple hw targets, multiple kernels to support big and small smart and IoT devices
  - initially based on Yocto, now moving to a GN/Ninja-based build system (~AOSP)
  - twin project: OpenHarmony, hosted by the OpenAtom Foundation (China)
  - next milestone: a developer phone powered by Oniro, with a react-native based framework to develop apps

What and Who (2)

oniro Compliance Toolchain

an Eclipse Foundation project, too, led by Array and Noi Techpark within the Eclipse Oniro WG

ARRAY

law firm specialized in IT Law and Open Source; OpenChain partner; Carlo Piana (founding partner) is Chair of OSI

TECHPARK SÜDTIROL / ALTO ADIGE

The science and technology park of South Tirol (Italy); it hosts research institutes (Fraunhofer and Eurac), University Faculties, scientific laboratories, companies and startups.

A Journey in Compliance

- compliance is a journey, not a destination
  - from policy…
  - …to process…
  - …to toolchain and CI
- challenges, solutions, some lessons learned

How it started

2020

- Development of an operating system platform for connecting big and small devices, fully open source and vendor-neutral
- integration of hundreds of third party OSS components
- need to draft an OSS compliance policy from the very beginning
  - to comply and avoid passing downstream non-compliances
  - to ease OSS compliance for downstream users, by providing reference compliance artifacts
  - to do OSS compliance in the Open Source way (upstream first, reuse, transparency, collaboration)

Flashback: Legal Background

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

Back to Our Story: Enter OPENCHAIN

2020

- we needed a recognized standard to build our OSS policy on
- https://www.openchainproject.org: a project by THE LINUX FOUNDATION
  - Rules of the road for being a good member of a supply chain (legal)
- A natural choice:
  - OpenChain was already an industry standard, and it was right about to become an ISO standard (ISO/IEC 5230:2020)
  - we at Array were already OpenChain Partners

OpenChain, in a Nutshell

- *Main purpose*: build trust in the software supply chain
  - by giving proof (compliance artifacts) that you have a system to ensure that you comply
  - so your partners and customers can rely on your efforts
- *Main artifact*: SBOM in a standard, machine-readable format such as SPDX (ISO/IEC 5962:2021) → standards reuse standards
- Requires processes, clear definition of roles, adequate education
- Not one off, revised on a rolling basis, as any quality system
- Self-assessment or even third-party certified
- *(now covering also cybersecurity with the new ISO/IEC 18974:2023: we will come to that later)*

So We Have an OpenChain-based Policy. And Now?

2021-2022

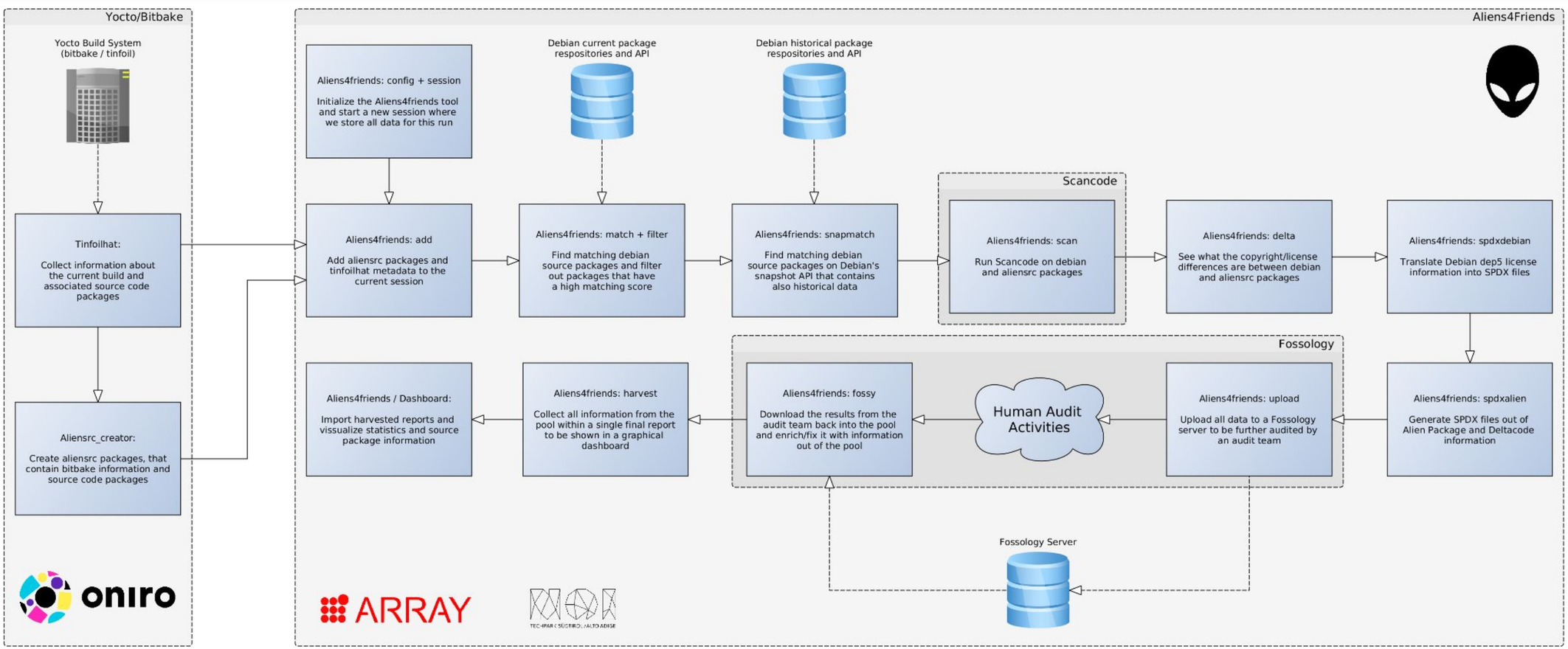
- SCA in embedded Linux projects is hard: custom distros, no prepackaged software, license metadata often unrefined, imprecise, or missing → we need license scanners, like ScanCode
- But license scanners return too many false positives and false negatives → we need human review, too! Enter fossology
- But it's too much work! How can we handle it? → The open source way! Reuse others' work: enter debian
  - a trusted friend that vouches for "alien" (third-party) OSS software
  - curated, detailed (file-level), machine-readable license metadata

OSS compliance for Oniro 2.0 (1)

2022 (end of)

- Tooling: Aliens4friends (A4F):
  - integration with yocto PROJECT (metadata and upstream sources),
  - Integration with fossology and ScanCode (license scan and review)
  - debian matcher and reuser (using Fossology API)
- Process design (tools + human work) and parallel (async) CI pipelines → continuous compliance
- dedicated dashboard to monitor audit progress and analyze results

Workflow Overview



CI Pipeline Overview

Pipeline | Needs | Jobs 43 | Tests 0

Group jobs by: Stage | Job dependencies

prepare
- cleanup
- workspace

build
- build: [linux, gcc, seco-imx8mm-c61-2gb, oniro-image-base , ACCEPT_FSL_EULA = "1"]
- build: [linux, gcc, seco-imx8mm-c61-4gb, oniro-image-base , ACCEPT_FSL_EULA = "1"]
- build: [linux, qemuarm64-efi, clang, oniro-image-base]
- build: [linux, qemuarm64-efi, gcc, oniro-image-base]
- build: [linux, qemuarm-efi, clang, oniro-image-base]
- build: [linux, qemuarm-efi, gcc, oniro-image-base]
- build: [linux, qemux86, clang, oniro-image-base]
- build: [linux, qemux86, gcc, oniro-image-base]
- build: [linux, qemux86-64, clang, oniro-image-base]
- build: [linux, qemux86-64, gcc, oniro-image-base]
- build: [linux, raspberrypi4-64, clang, oniro-image-base]
- build: [linux, raspberrypi4-64, gcc, oniro-image-base]
- build: [linux, seco-intel-b68, clang, oniro-image-base]
- build: [linux, seco-intel-b68, gcc, oniro-image-base]
- build: [linux, seco-px30-d23, clang, oniro-image-base]
- build: [linux, seco-px30-d23, gcc, oniro-image-base]
- build 4

collect
- aliensrc
- tinfoilhat

analyze-a4f
- add
- delta
- fossy-manual-update
- harvest
- match
- scan
- session-lock
- session-unlock
- session-variants
- snapmatch
- spdxalien
- spdxdebian
- upload

sbom
- fossy-spdx
- generate-sbom 4

Dashboard: General Overview

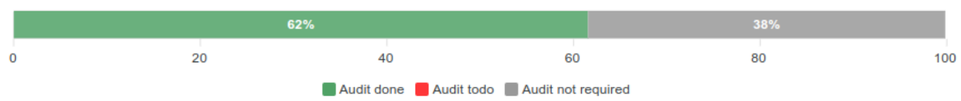
Aliens4Friends Dashboard

138/138 PACKAGES

Filter...

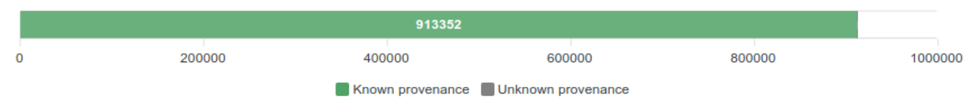
FOSSology audit progress

Total audit files



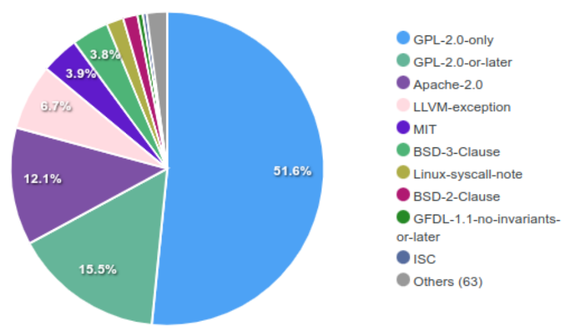
Provenance

Upstream source total: 720290



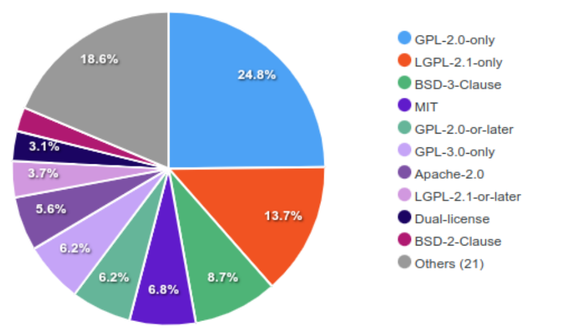
License types scanned

Results from automated scanners such as scancode, monk, nomos, ojo



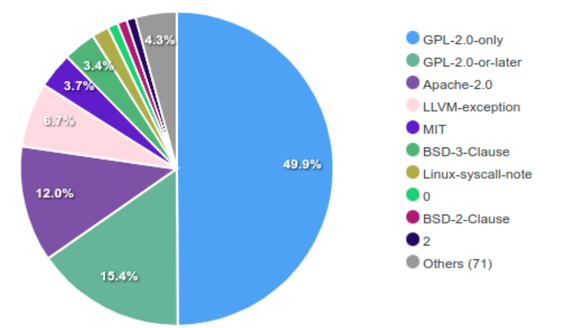
Main license types

Accumulated main licenses



License types audited

Results by human auditor analysis



Main Licenses	Files	Distros	Images	Machines	Releases
accumulated	total	accumulated	accumulated	accumulated	accumulated
31	914041	7	12	20	6

ABOUT THIS TOOL

Dashboard: Component Details

Aliens4Friends Dashboard

138/138 PACKAGES

Filter...

Package Explorer

State	ID	Aud. Progress	Aud. Workload	Main licenses	Distros	Images	Machines	Releases	Scan	Audit All
	acl-2.3.1-r0-963c29b5+a4f	100% 166 Files done	166 Files total	LGPL-2.1-only	linux, linux-clang, linux-gcc	image-base, image-base-dev, image-base-seco-px30-d23-emmc.manifest, image-base-tests, image-base-tests-seco-px30-d23-emmc.manifest, image-extra, image-extra-dev, image-extra-seco-px30-d23-emmc.manifest, image-extra-tests, image-extra-tests-seco-px30-d23-emmc.manifest	qemuarm-efi, qemuarm64-efi, qemux86, qemux86-64, raspberrypi4-64, seco-imx8mm-c61-2gb, seco-imx8mm-c61-4gb, seco-intel-b68, seco-px30-d23	kirkstone-v2.0.0-beta, kirkstone-v2.0.0-beta-109-g2f5fa92	32 (21, 5, 3, 3) GPL-2.0-or-later, FSFULLR, GPL-3.0-or-later, Others (3)	32 (20, 5, 5, 2) GPL-2.0-or-later, Autoconf-exception, GPL-3.0-or-later, Others (4)
	acpid-2.0.33-r0-c931e98d+a4f	100% 49 Files done	49 Files total	GPL-2.0-only	linux, linux-clang, linux-gcc	image-base, image-base-dev, image-base-tests, image-extra, image-extra-dev, image-extra-tests	seco-intel-b68	kirkstone-v2.0.0-beta, kirkstone-v2.0.0-beta-109-g2f5fa92	7 (3, 3, 1) GPL-2.0-or-later, GPL-2.0-only, X11	11 (3, 3, 3, 2) GPL-2.0-or-later, GPL-2.0-only, Autoconf-exception, Others (2)
	alsa-lib-1.2.6.1-r0-b83ad896+a4f	100% 244 Files done	244 Files total	LGPL-2.1-only	linux, linux-clang, linux-gcc	image-base, image-base-dev, image-base-seco-px30-d23-emmc.manifest, image-base-tests, image-extra, image-extra-dev, image-extra-tests	qemuarm-efi, qemuarm64-efi, raspberrypi4-64, seco-imx8mm-c61-2gb, seco-imx8mm-c61-4gb, seco-intel-b68, seco-px30-d23	kirkstone-v2.0.0-beta, kirkstone-v2.0.0-beta-109-g2f5fa92	15 (5, 5, 3, 2) GPL-2.0-or-later, FSFULLR, GPL-3.0-or-later, Others (2)	23 (5, 5, 5, 8) GPL-2.0-or-later, FSFULLR, Autoconf-exception, Others (5)
	alsa-plugins-1.2.6-r0-47385cc5+a4f	100% 68 Files done	68 Files total	LGPL-2.1-only	linux, linux-gcc	image-base, image-base-dev, image-base-tests, image-extra	seco-imx8mm-c61-2gb, seco-imx8mm-c61-4gb	kirkstone-v2.0.0-beta, kirkstone-v2.0.0-beta-109-g2f5fa92	17 (5, 5)	27 (4, 4, 15)

ABOUT THIS TOOL

OSS compliance for Oniro 2.0 (2)

- audit guidelines for human validation → consistency, transparency → reusability
- final output:
  - fixed issues in the Oniro project (by removing offending components)
  - fixed issues in third-party components, (by removing offending files or fixing license conditions or wrong license references)
  - reported outstanding issues to users to enable the latter to handle them

https://gitlab.eclipse.org/eclipse/oniro-compliancetoolchain/toolchain/docs/-/tree/main/audit_workflow
https://oniroproject.readthedocs.io/en/latest/releases/2.0/2.0.0/ip_compliance_note.html

Moving forward

2023

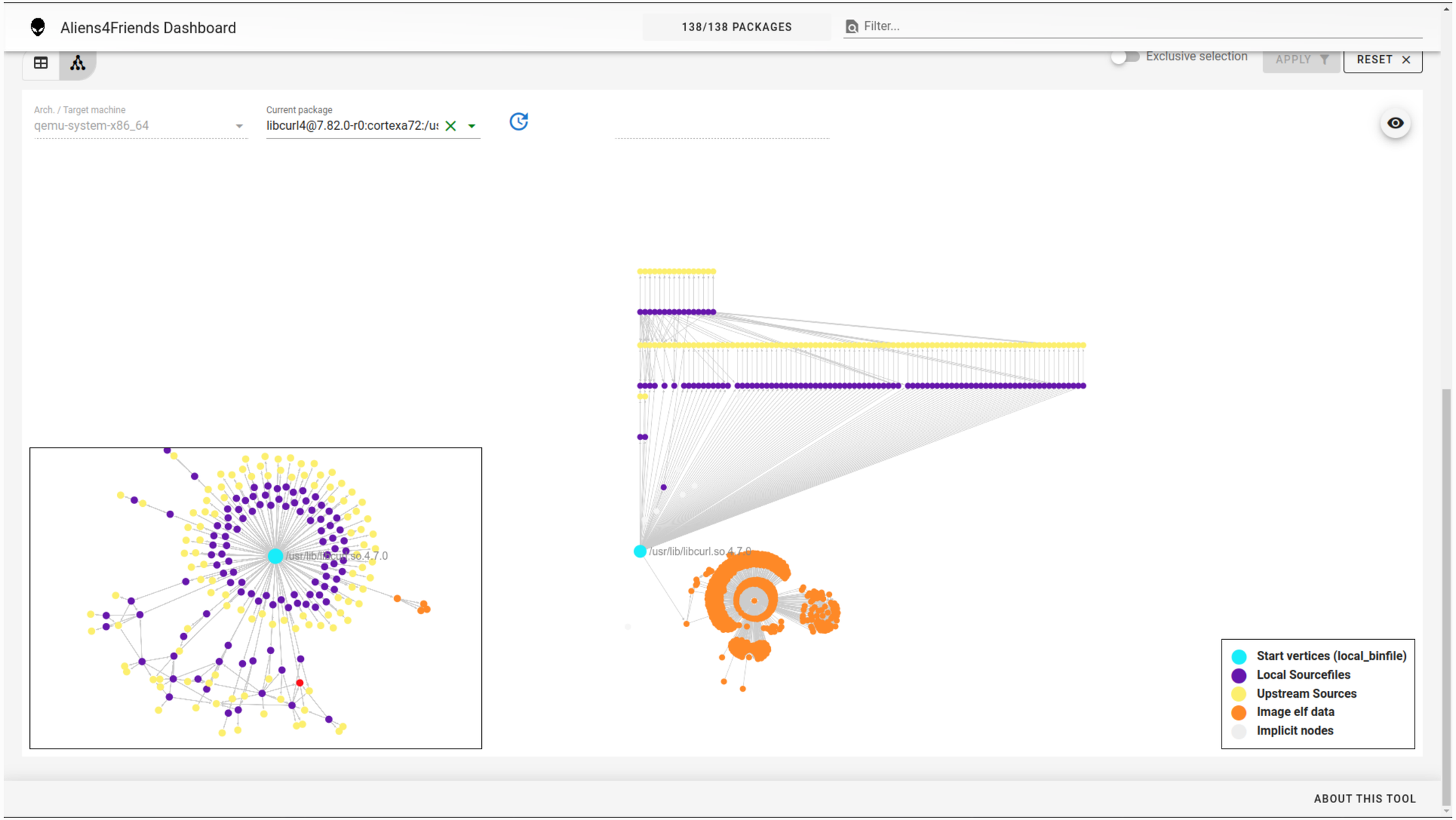
- Upstreaming metadata collection logic to Yocto: added Unpack Tracer API (accepted), meta-bbtracer (WIP)
- Improving: automatically resolve binary file licenses and file-level license incompatibilities by mapping binary files to source files (PoC)
- Scaling out: implement a4f CI pipelines in other operating system projects full implementation on Eclipse Leda (OS for SDV), demo on Linaro TRS

The Future: Next Steps

2024-…

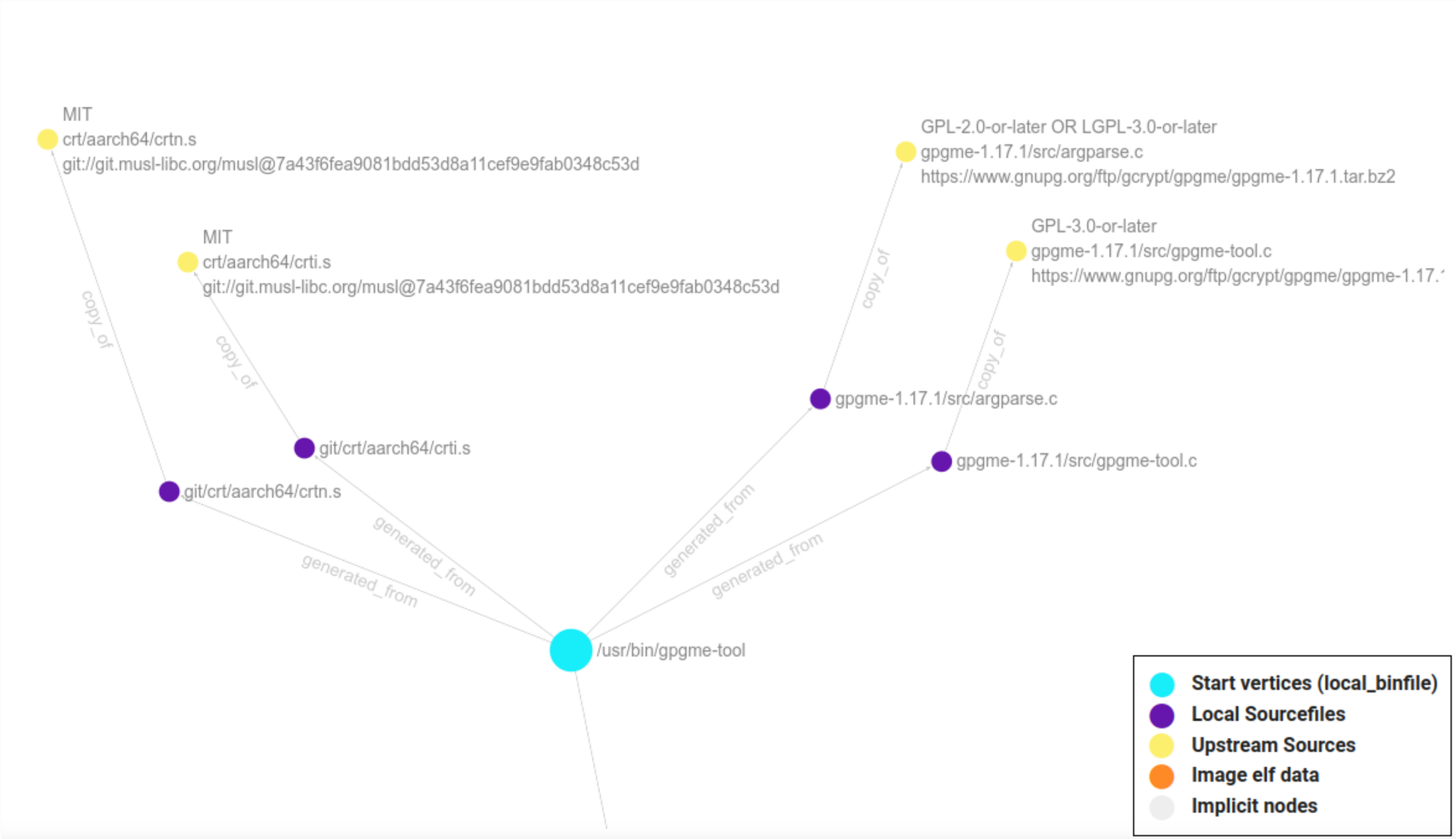
- integrate other tools (ORT, SW360)
- make Aliens4friends fully build-system-independent (support Yocto, GN/Ninja → Oniro 3.0, OpenHarmony, AOSP)
- automatically resolve binary file licenses and file-level license incompatibilities by mapping binary files to source files through a graph database
- *integrate security compliance (CRA) in the toolchain (we will come to that later)*

Dashboard NextGen: Graph Database (1)



ARRAY

Dashboard NextGen: Graph Database (2)



Lessons Learned

Lesson Learned #1: Reuse, Why and How

Automation is key, but human review of automated scan results is also key, especially in the embedded Linux space (no package manager, etc.).

↓

human review is costly and must be made sustainable

↓

To make human review more sustainable, we should be able to reuse the work of others, and others should be able to reuse ours

↓

reuse works well both ways only if certain conditions are met:

(a) we are all reviewing the same thing (original upstream sources)

(b) we can trust each other's work

(process transparency, documented audit criteria)

(c) we work with the upstream every time it is possible

Lesson Learned #2: Continuous Compliance, Why and How

Human review requires substantial time

↓

If we do that only before each release, it turns into a bottleneck

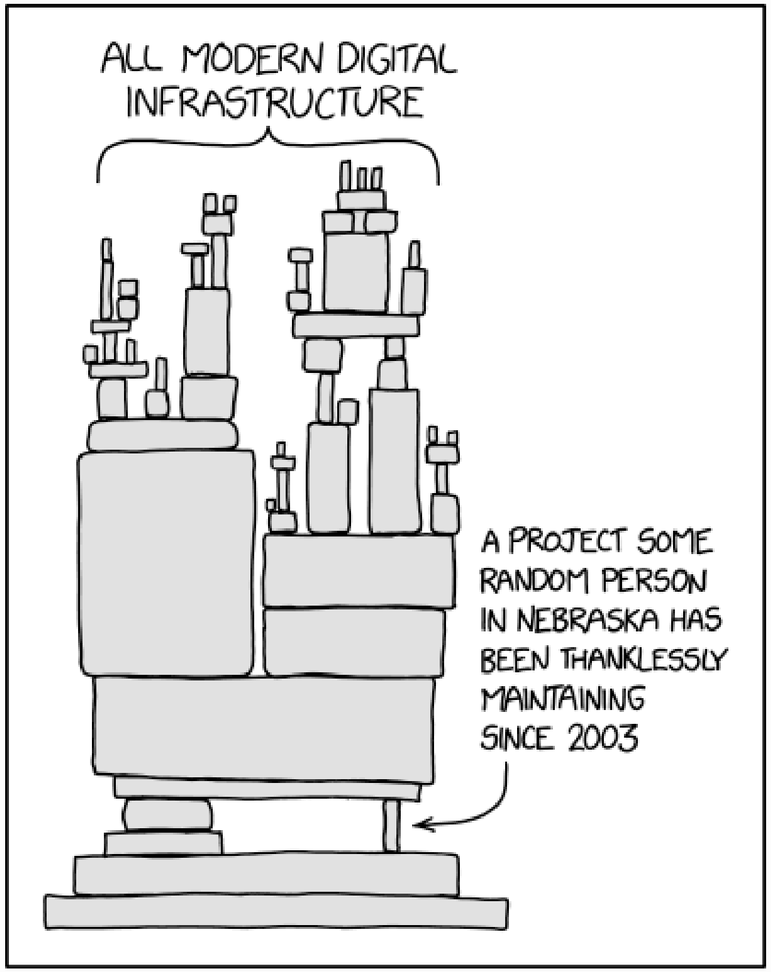
↓

it should be a continuous process, flowing in parallel with the development process (continuous compliance)

↓

parallel (async) CI pipelines, monitoring progress and results

Lesson Learned #3: Compliance and Security Flock Together



- Since we started in 2020, big stuff happened: SolarWinds, Kaseya, Log4j, xz… → POTUS Executive Order on Cybersecurity, EU Cyber Resilience Act (CRA), etc.
- In ~3 years from now:
  - all software will need to be CRA-compliant to be placed on the EU market (CE Mark for PWDE)
  - SBOM and due diligence on OSS components will be mandatory, as well as providing vulnerability fixes for the whole product lifecycle
  - EU Standard Bodies will define the standards for CRA compliance
  - EU Commission will issue guidance and delegated acts to implement CRA
- OpenChain has evolved to cover cybersecurity: OpenChain Security Assurance Specification (ISO/IEC 18974:2023)

Lesson Learned #3: Compliance and Security Flock Together

- In the new regulatory environment, an OSS Policy will need to cover both compliance and security
- Processes and tools for OSS compliance and for OSS security need to be integrated and coordinated, sharing artifacts (SBOM, audit results, etc.)
- Software Composition Analysis (SCA) is the key to both compliance and security: know your software, know your vulnerabilities, know your obligations

TIME FOR DISCUSSION

Discussion

Linaro's Production-Grade Embedded Distros and OpenChain:

- Policy
- Processes
- Tools

Thank you!

https://array.eu
https://projects.eclipse.org/projects/oniro.oniro-compliancetoolchain
https://gitlab.eclipse.org/eclipse/oniro-compliancetoolchain/toolchain

---

Array Alberto Pianon Carlo Piana