ARRAY

TECHPARK SÜDTIROL / ALTO ADIGE

KYCS – Know Your Code Sources

(and let it be known)

Alberto Pianon, Carlo Piana – Array

SFScon @Noi Techpark 2023

Image by Brigipix from Pixabay

CRA

Cyber

Resilience

Act

Under CRA

- You Must Know Your (Upstream) Code
- You Must Let It Be Known
- Which Means:
  - provenance
  - integrity
  - security
  - maintenance

This is not just a friendly advice

For instance:

10.4

For the purposes of complying with the obligation laid down in paragraph 1, ***manufacturers shall exercise due diligence when integrating components sourced from third parties*** *in products with digital elements. They shall ensure that such components do not compromise the security of the product with digital elements.*

Annex I

(2) Products with digital elements shall be delivered without **any known exploitable vulnerabilities**

Annex II

As a minimum, the product with digital elements shall be accompanied by:

[…]

6. if and, where applicable, where the software bill of materials can be accessed;

Annex V

The technical documentation [...] shall contain at least the following information

[...]

where applicable, the software bill of materials [...], further to a reasoned request from a market surveillance authority provided that it is necessary in order for this authority to be able to ***check compliance with the essential requirements*** *set out in Annex I.*

Stricter requirements for

- Critical products with digital elements
- High-risk AI systems

Therefore...

Just take the plunge!

Image by DaveMeier from Pixabay

Where We Have Done It

ECLIPSE FOUNDATION oniro v2.0

- Embedded Operating System platform, with multiple build targets
- an integrated Continuous Compliance process managed through a dedicated toolchain
  - based on existing OSS tools fossology ScanCode
  - + a set of custom tools (aliens4friends)
- using SPDX across the whole workflow, to get machine readable **SBOM**

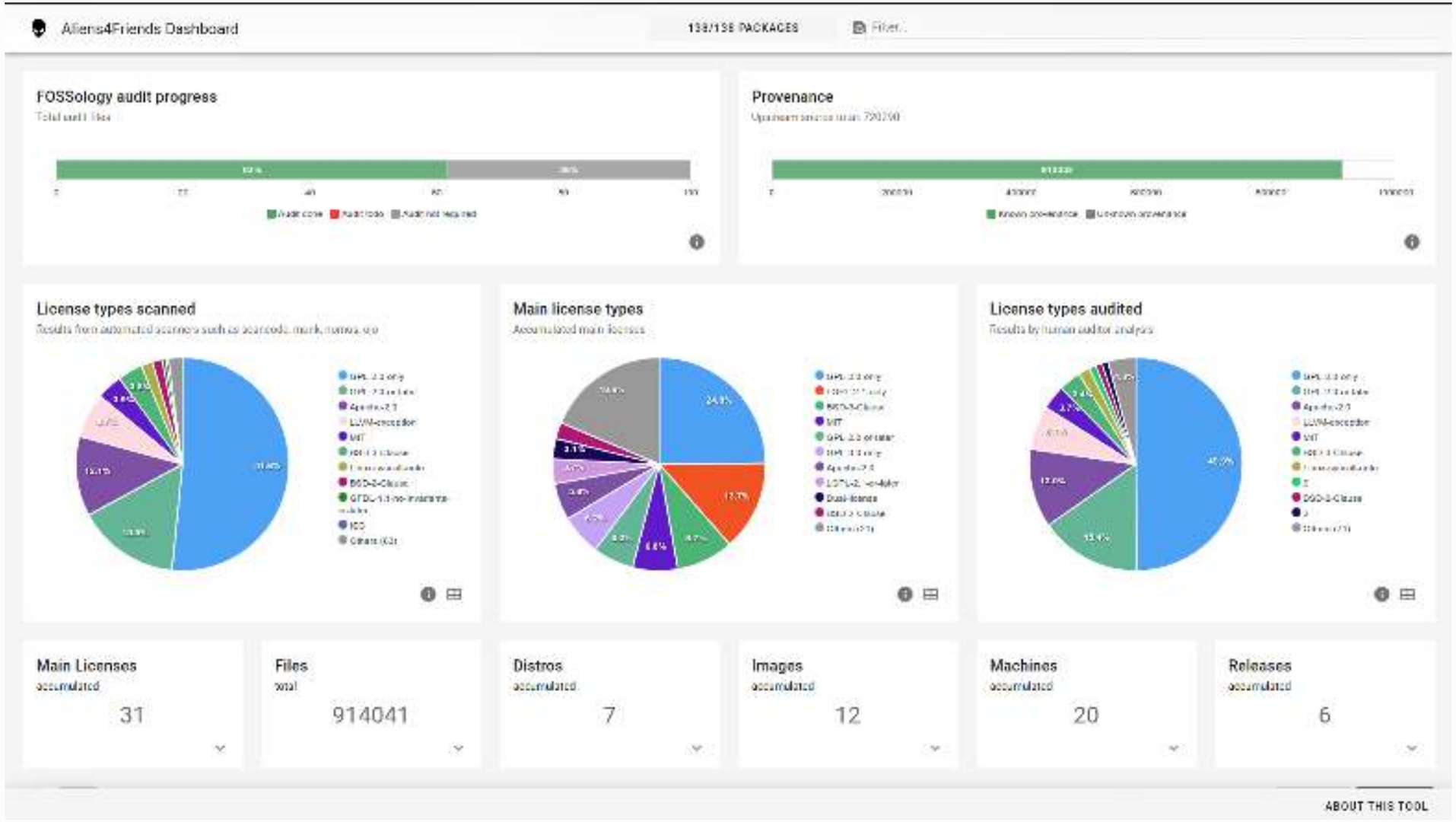
Who Did It?

	NOI Techpark	https://noi.bz.it/en
	Array	https://array.eu

ARRAY

TECHPARK SÜDTIROL / ALTO ADIGE

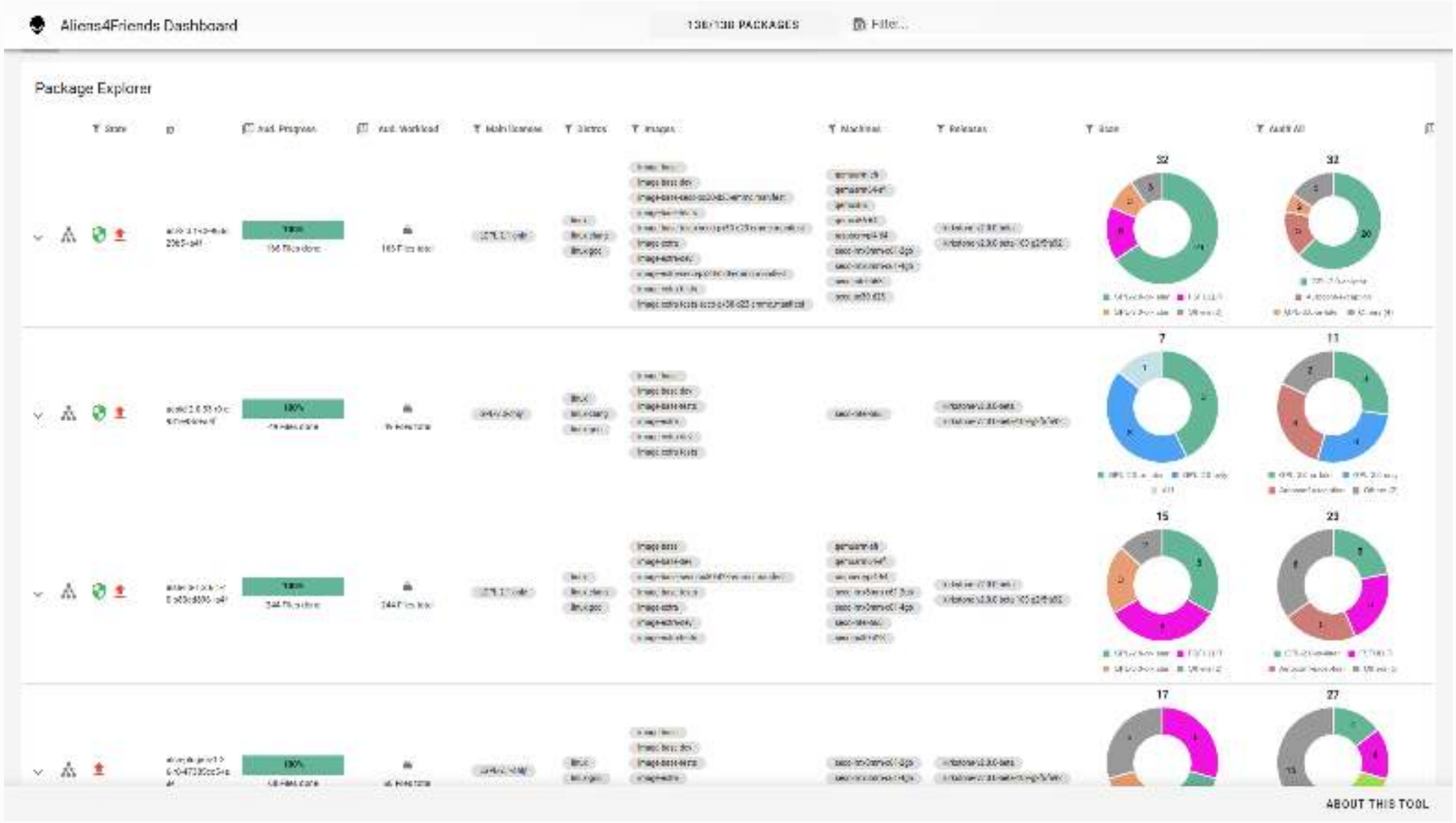
sca.software.bz.it



ARRAY

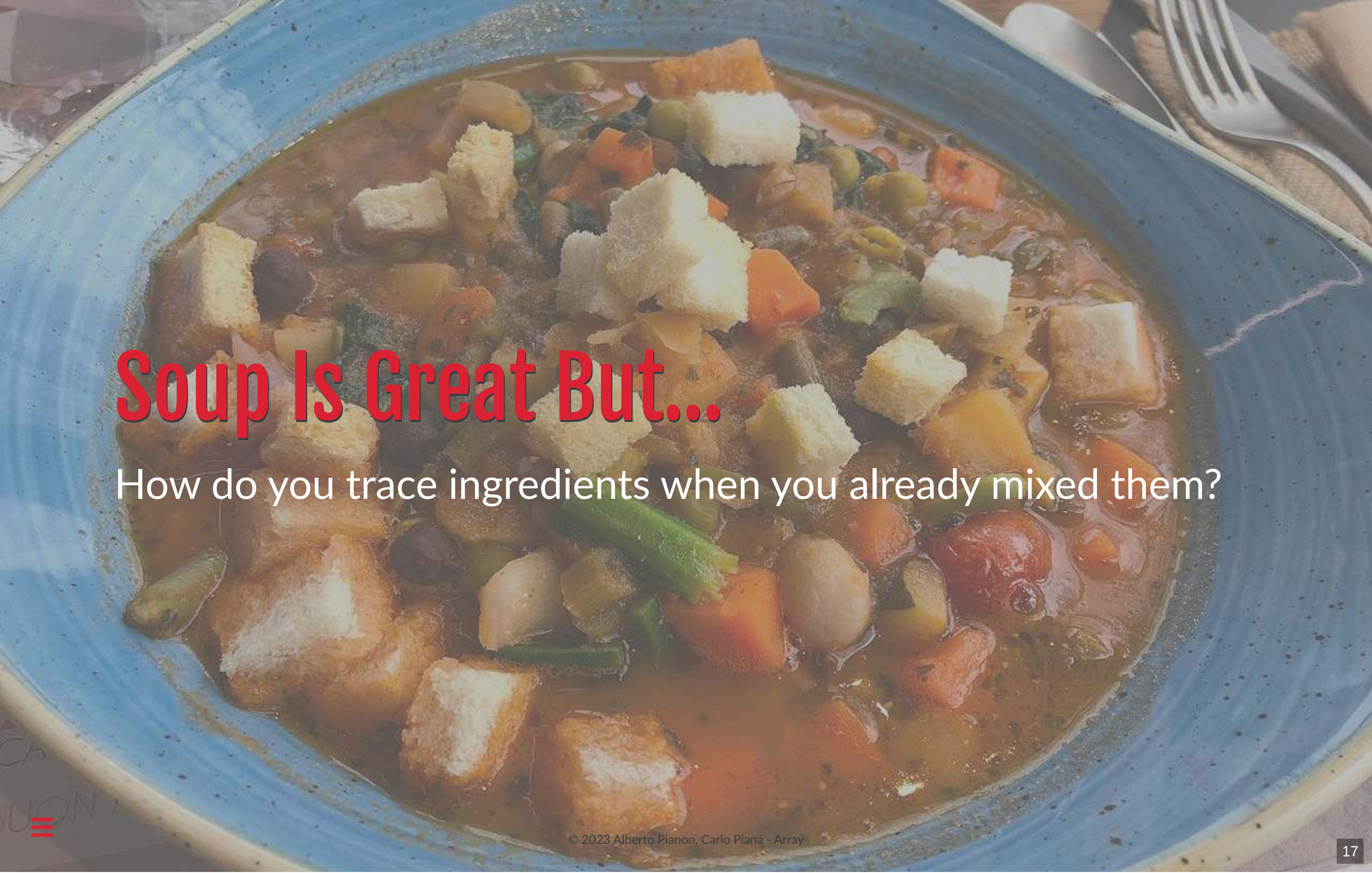
TECHPARK SÜDTIROL / ALTO ADIGE

sca.software.bz.it



The Problem We Had To Solve

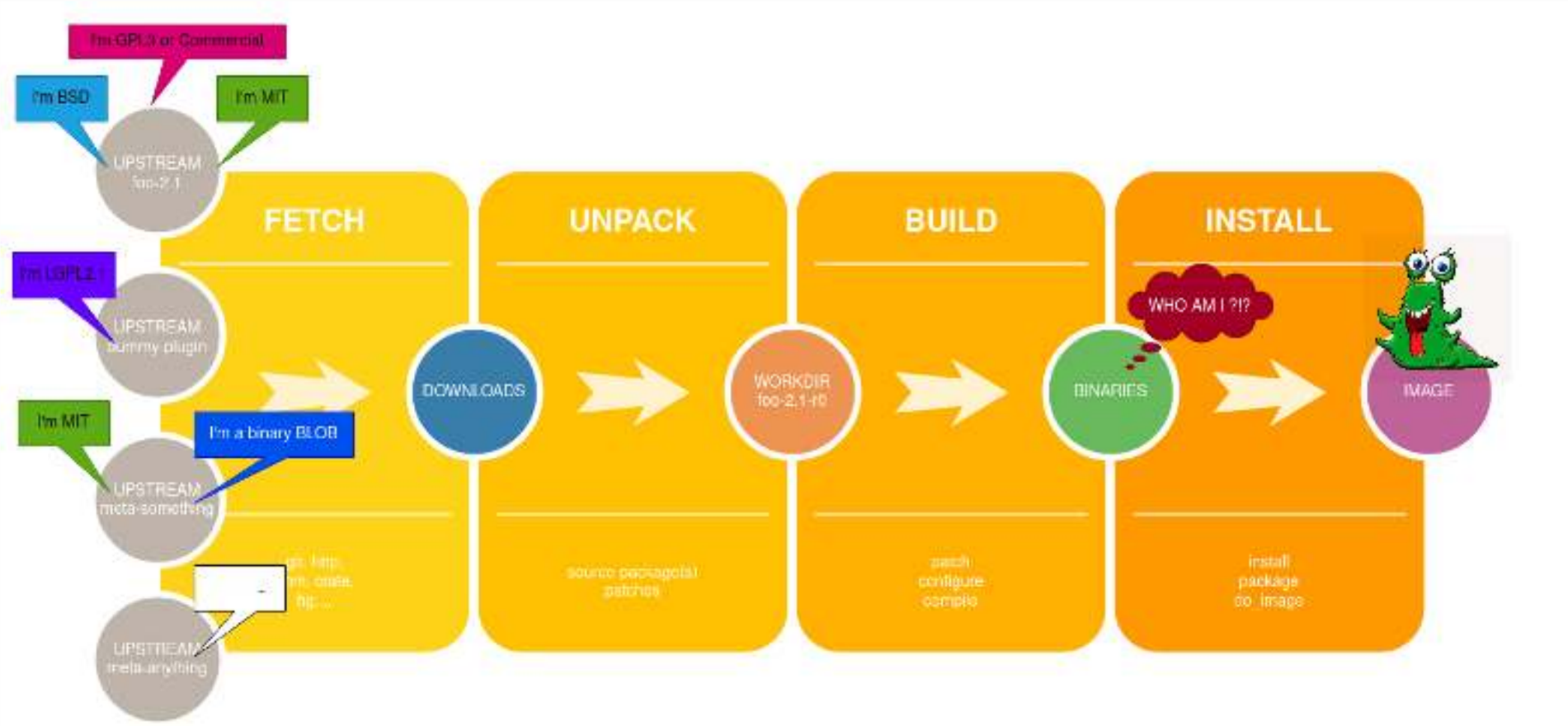
- In embedded operating systems, hardware integration is a PITA
- to handle it, you need to bake your own custom linux distribution: yocto PROJECT is *the* tool
- mixing ingredients by using recipes, layers and overrides offers great flexibility and eases hardware integration…
- …but it brings PITA when you need to generate SBOM



Soup Is Great But...

How do you trace ingredients when you already mixed them?

Yocto Workflow (Simplified)



How We Want To Collect Required Data On Yocto Side

- Map upstream source files to local workdir source files to binary files
  - consume metadata coming from Yocto
  - trace unpacking of upstream source packages and of downstream patches (the ingredients), separately from each other, and map them to local workdir files (the soup pan)

Downstream Or Upstream?

- In a first PoC we did that with an external, post-mortem script using Yocto/bitbake libraries to parse build directories and reproduce the unpack process…
- but the solution needed to be pushed **upstream**!

Finally, We Made It!

- Patch has been recently **accepted in Bitbake** and Poky repositories and **will be part** of the upcoming Yocto release
- It exposes an “unpack tracer” API where one can plug-in upstream source tracing logic directly into the core of bitbake
- Trace ingredients before making the soup!
- A first API implementation (WIP) is already available at https://gitlab.eclipse.org/eclipse/oniro-compliancetoolchain/toolchain/next/meta-bbtracer
- We will work on back-porting the patch to older Yocto LTS releases

Q&A: We Need Your Feedback!

Thank You For Your Attention

https://array.eu

https://projects.eclipse.org/projects/oniro.oniro-compliancetoolchain

https://gitlab.eclipse.org/eclipse/oniro-compliancetoolchain/toolchain

---

Array Alberto Pianon Carlo Piana