Software Composition Analysis of Docker and container images. Summer 2022

Introduction: Philippe Ombredanne

- Weird facts and claims to fame
- Signed off the **largest deletion of source lines in the linux kernel** (but these were only license comments)

- Maintainer of FOSS tools for FOSS code origin, license, security and quality analysis aka. SCA "**Software Composition Analysis"**
- ScanCode, VulnerableCode and AboutCode tools, LicenseDB, Package URLs
- Co-founder of SPDX, ClearlyDefined, long time GSoC/GSoD mentor, contributor to Open Chain Reference Tooling group & several FOSS projects
- Co-founder and CTO of nexB Inc. SCA tools and services
- pom@nexb.com or pombredanne@gmail.com

Agenda

- ▷ The container challenge
- ▷ The ScanCode.io solution
- ▷ Who is using ScanCode.io for containers?
- ▷ ScanCode.io user flow & pipeline details
- ▷ Demo
- ▷ Status
- ▷ Architecture
- ▷ Next Steps

The container challenge (1)

- ▷ A Container is like a VM with a twist
  - Multiple slices of root filesystems
  - No kernel
  - Commonly include multiple Linux distros
- ▷ **Many packages**, mostly pre-built binaries
  - No kernel, BUT **10x to 100x** more packages
  - 10x to 100x **more licenses** :|
- ▷ Package **metadata are not enough** (too little or too much)
  - The declared license is often incorrect or misleading
- ▷ Not everything is a package
  - Extra files COPY'ied, download and ADD'ed to Docker image

The container challenge (2)

- Dynamic analysis (e.g. **running tools inside**) is problematic because you modify what you are analyzing (observer effect)
- No scriptable, customizable and open source solution that provides an acceptable quality of license detection
  - Most tools focused only on surface package scans with minimal cross-checks
- 100x more packages: But how to avoid doing **100x more compliance** work?
- And still get high quality composition analysis?

Why ScanCode.io?

- **Easy end-to-end analysis**, press of a button analysis
- **Static analysis** e.g. do not run container to analyze it
- **Guarantee** that ALL files in an image are vetted
  - Not a mere inventory of packages and their licenses
- **Scriptable pipelines aka. ScanPipes** easy to customize
  - Not limited to containers, also any rootfs or any code
    - e.g. Full VM images or OpenWRT-based devices
  - Integration platform: can add analysis steps to run any other tools
  - Installable locally
- **Open source** and best in class
  - No other commercial or open source tool has the same capabilities so far
  - Recognized by key users as best in class

Who is using **ScanCode.io** for containers?

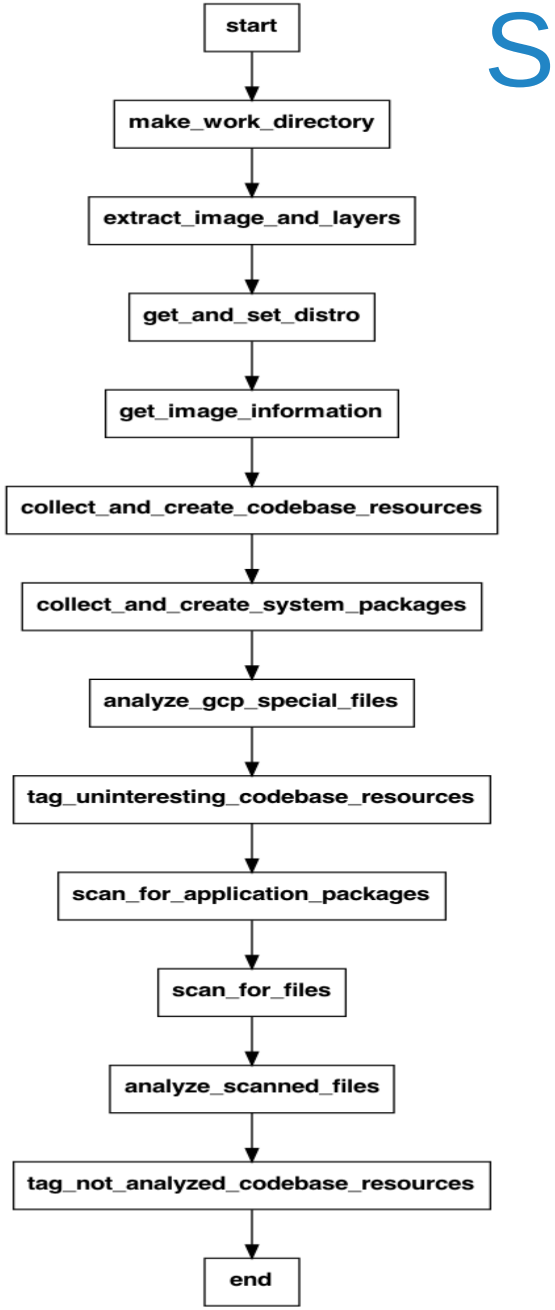
- Two of the largest big tech companies
- A large US device manufacturer
- Three large European industrial companies
- Many more
- nexB professional service team for product release due diligence and M&A audits

Alternatives

- Commercial tools are focused primarily on Security
  - Shallow support for licensing using only (weak) metadata
- Open source tools include Tern and few others
  - Use techniques of dynamic analysis
  - Package-level only, not vetting **all** the files
  - Fixed analysis process
  - Some use ScanCode Toolkit to provide better results

Docker image user flow

- Upload (or fetch) a Docker image to **your** ScanCode.io server
- ScanCode.io analyzes the image and collects structured license and provenance of:
  - All system packages
  - All application packages
  - All files not part of packages are scanned in details
  - (optional: Your own Custom steps)
- Fetch results as JSON, XSLX, Browse online, JSON REST API access



start

make_work_directory

extract_image_and_layers

get_and_set_distro

get_image_information

collect_and_create_codebase_resources

collect_and_create_system_packages

analyze_gcp_special_files

tag_uninteresting_codebase_resources

scan_for_application_packages

scan_for_files

analyze_scanned_files

tag_not_analyzed_codebase_resources

end

ScanCode.io Docker pipeline details

- ▷ Fetch then Prepare **image archive**
- ▷ For each image **layer**: scan **system packages**
  - ○ Find their file and check if modified
- ▷ For remaining files: scan **application packages**
  - ○ All ScanCode-supported package types (npm, maven, composer, etc.)
- ▷ For remaining files: **scan files**
  - ○ All files, including binaries
- ▷ For remaining files: analyze and tag
  - ○ Dispose of temp and transient or log files and more
- ▷ *Add your own special step*
- ▷ Assemble results from DB and return JSON

Live Demo

- ▷ A Debian-based Redis image
  - Get it from docker://redis:buster
- ▷ A problematic Alpine image
  - Get it from docker://quay.io/wire/alpine-deps
    - and https://quay.io/repository/wire/alpine-deps
  - Contains native **GPL-3.0licensed binary** built on the fly, no origin, no source, no license!
    - /usr/lib/libcryptobox.so happens to be a "random" GPL-3.0-licensed binary built on the fly and added to the image
    - https://github.com/wireapp/wire-server/blob/8d8525b30a5eb33557cb2c8a0f21a8aa2ea63999/build/alpine/Dockerfile.deps#L8
    - https://github.com/wireapp/cryptobox-c

Architecture

Scan Pipelines execute in ScanCode.io server

- ○ Python, Django, PostgreSQL

▷ Each focused composition analysis script is a pipeline

- ○ Flexible and clear scripting, customizable, resume/restart

▷ JSON API, Web UI, reporting

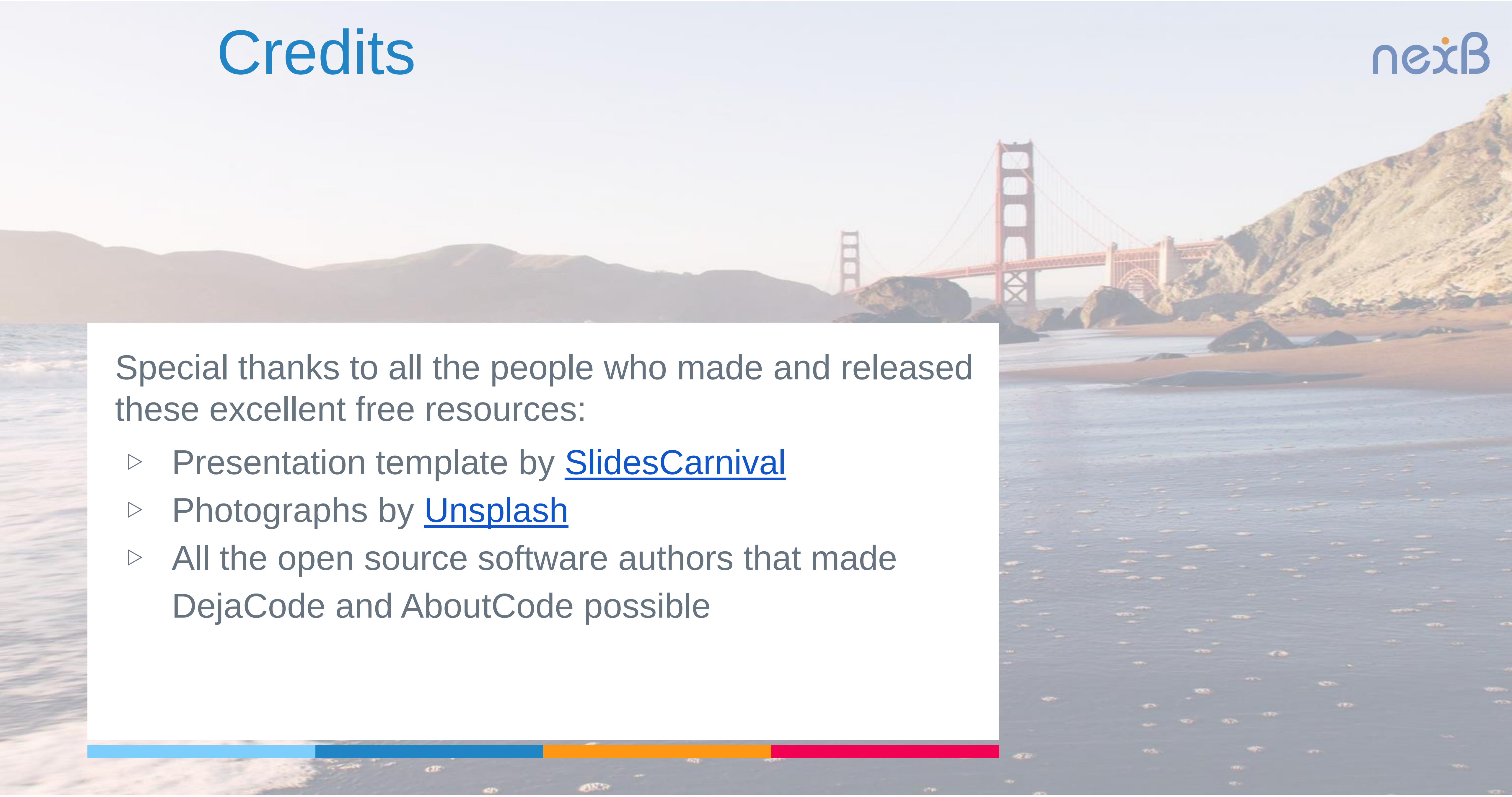
▷ Inside:

- ○ **ScanPipe** for end-to-end scripting and pipeline documentation
- ○ **ScanCode** toolkit for license and application package parsing
- ○ **container-inspector** library for container image processing
- ○ **debian-inspector** for debian
- ○ ScanCode for Alpine and RPMs, and distroless for system package

Status and plans

- Support for all main Linux distro is available for Docker and OCI containers:
  - Debian/Ubuntu, RedHat/Suse RPM-based, Alpine and Distroless
  - And Windows containers too!
  - Support all common VM image formats
- Upcoming
  - Major improvement on license detection accuracy
  - Smart ML-based analysis of detected licenses and automated active learning
  - Policies and efficient handling of TODOs for human review
  - New one off license scans pipeline
  - New Android app scan pipeline
- Building a library of pre-scanned base images and layers
  - e.g. SCAN and REVIEW ALL THE PUBLIC CONTAINERS

Credits



Special thanks to all the people who made and released these excellent free resources:

- ▷ Presentation template by SlidesCarnival
- ▷ Photographs by Unsplash
- ▷ All the open source software authors that made DejaCode and AboutCode possible

nexB Solutions Overview

- **SCA and Audit Services**
  - Enabled and accelerated by our free ScanCode and AboutCode tools
  - http://www.nexb.com/services.html
- **AboutCode** - Open source for open source analysis
  - Recognized as best-in-class tools
  - nexB provides professional services to accelerate or customize implementation
  - **ScanCode** TK, ScanCode.io, ScanCode WB, AttributeCode TK, DeltaCode, TraceCode and other tools available at https://aboutcode.org and https://github.com/nexB
- **DejaCode** - Compliance application for legal and management teams
  - A central system of records to aggregate and manage all your software products, components, licenses and policies https://dejacode.com

Related FOSS projects

- **AttributeCode** TK - Auto generate attribution notices
- **TraceCode TK** - trace your build to find deployed code
- **VulnerableCode** - The free correlated vulnerabilities DB (startup funding from the EU and NLnet)
- DeltaCode - compare two scans
- **Container-Inspector** - Static Docker images analysis - low level library
- **Debian-Inspector** - Debian packages analysis
- AboutCode - Data models (used in Libraries.io and ORT)
- ScanCode Workbench - Desktop app for Scan review
- license expression - parse, combine, simplify
- Package URL (purl) - used in OWASP, Sonatype

Contact us

- Contact persons
  - Michael Herzog
    mjherzog@nexb.com+ 1 650 380 0680
  - Philippe Ombredanne
    pombredanne@nexb.com+ 1 650 799 0949
- More information
  - https://www.nexb.com/

About nexB

- Focused on **Software Composition Analysis** (SCA) and FOSS Compliance since 2007
- Software provenance experts
  - 500+ SCA projects completed to-date
  - 100% customer satisfaction
- Authors of ScanCode - industry-leading FOSS-SCA toolset
- Industry thought leaders
  - Co-founders of SPDX
  - Co-founders of Package URLs

nexB in the SCA domain

- Software Composition Analysis comprises four dimensions of managing your software:
  - Identification of **software origin**
  - Identification of software **licensing**
  - Identification of software **vulnerabilities**
  - Quantification of software **quality**
- nexB currently offers:
  - Leading solution for identification of software **origin and license** in sources and binaries
  - Emerging solution for **vulnerabilities**

SCA and FOSS

- Newer software products and systems comprise **80% or more FOSS**
  - Many products include hundreds or thousands of FOSS components
- FOSS licensing is a **higher risk now** than in the recent past
  - Explosion of the number of package dependencies and their rate of change
  - More **“dual” licensing models** - e.g. MongoDB, Redis, Elastic - that blur the lines between FOSS and proprietary software
- FOSS Compliance is focused on identifying licensing and complying with license conditions