Starting and scaling an Open Source Office

The good, the bad and the ugly

Thomas Steenbergen
Open Source Office - HERE Technologies

Open Source Leadership Summit 2019

About me

Open Source Office at HERE Technologies
global location cloud company

200+

countries mapped

HERE Maps on board of
100M
vehicles and counting

8,000+

employees in 56 countries

Active contributor

OSS Review Toolkit
http://oss-review-toolkit.org

SPDX

OSS Tooling Group

TODO
European Chapter

OPENCHAIN

thomas.steenbergen@here.com
@tsteenbe
linkedin.com/in/tsteenbe

2

© 2019 HERE

here

Work In Progress

Sharing lessons learned
of our ongoing journey
scaling our Open Source office

Story Time

How we started our Open Source Office in 2016

Companies must master open source if they are to master software.

SAMSUNG

Essential Elements to Master Open Source

COMMUNITY

CONTRIBUTION

COMPLIANCE

CONSUMPTION
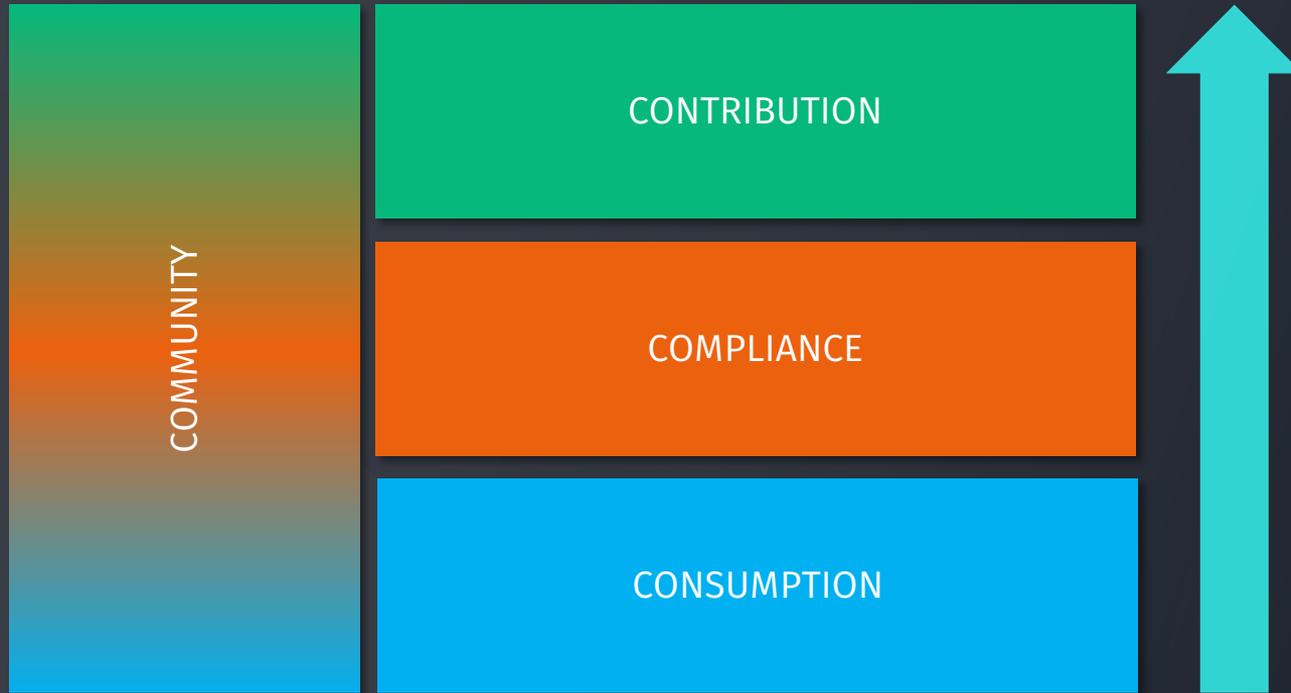
borrowed from SAMSUNG

6

Enable better products

by

Minimizing
risks

&

Maximizing
the benefits of
Open Source

Handling **compliance** and **contributions** from **Day 0** with a **small team**
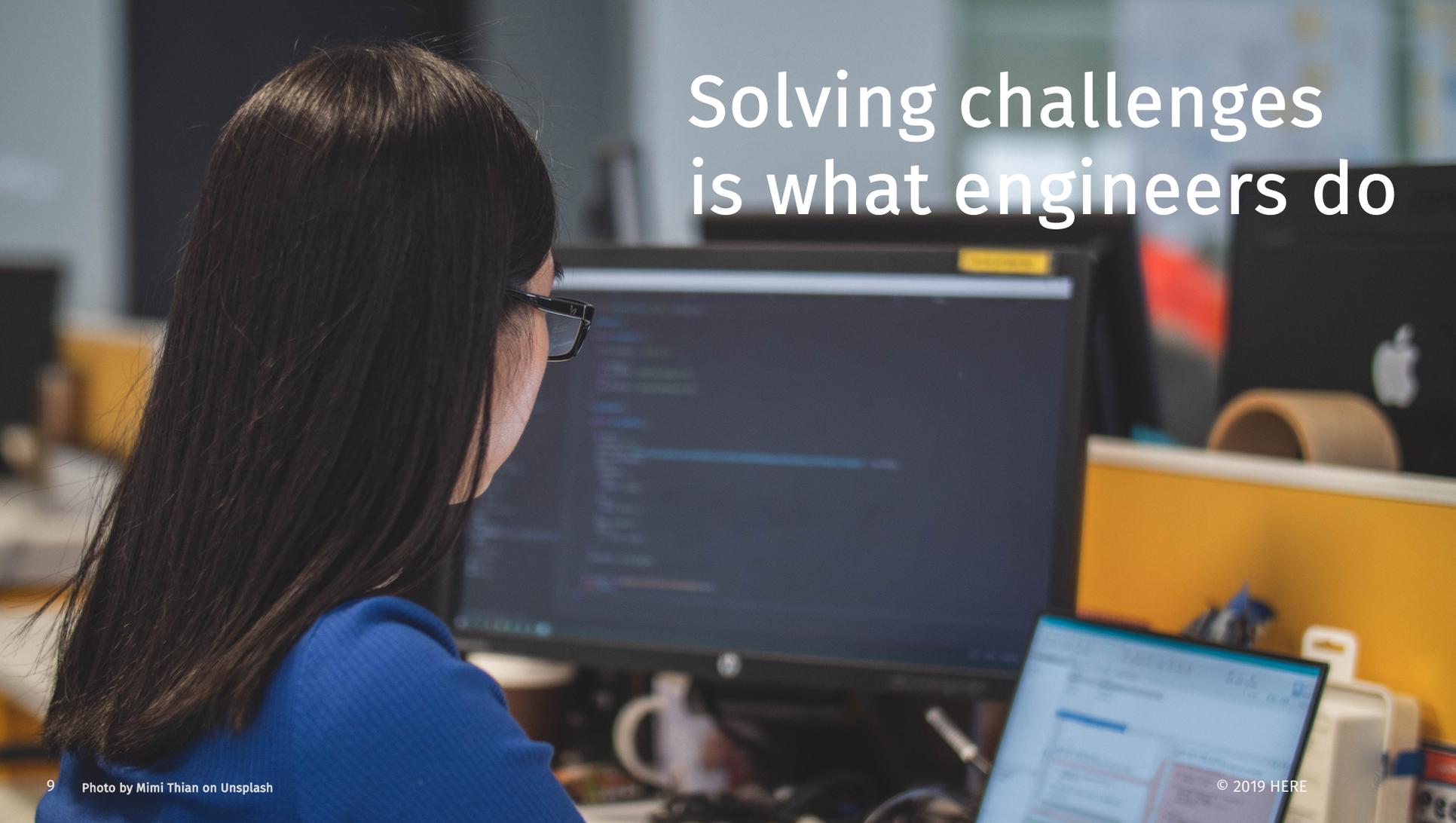
Solving challenges is what engineers do

Translating Scaling Challenges into Process and Tooling Requirements

- Ways of working/shift-left - integrated with SW dev process
- Scales cost-effectively - whether running 10 or 10K reviews
- Self-service - lightweight review process executed by devs
- Low learning curve - build on common dev tools with clear & concise docs
- Trust but verify - able to check each workflow step in detail if needed
- Turn-key integration - no change to SW build needed for review
- Configurable - enable teams to correct to match reality
- Based on standards - for future proofing, think SPDX
- Analytics - insights in SW inventory and OSS tooling
- Accuracy - detection of license context and copyright holders
- Freshness - scan early and often
- Speed - highly automated

here

Early-on Lessons

Good
- Sufficient getting started material available
- Welcomed with open arms by your peers

Bad
- Contributing at scale as a company is complex topic (culture + tools)
- Common misconception: engineers who have contributed before know how to manage Open Source projects

Ugly
- Compliance at scale is fire fighting

Choices to make

Build
Buy
or
Share?

Assessing Market Solutions

Tested more than 20+ commercial and FOSS solutions on the market which include review workflows, tooling, etc. ...

You can just buy solutions,
build few missing pieces
and be done, right?

Modern SW dev practices
are **not fully supported**
by current FOSS review tooling
available on the market

Introducing ORT - OSS Review Toolkit (late 2017)

Developer preview released at **github.com/heremaps/oss-review-toolkit**

Written in Kotlin, Apache-2.0 licensed. Welcome your feedback and contributions

SW project
source code

analyzer

downloader

scanner

file
results

gradle

npm

PiP

maven

sbt

deps

ABCD-style YAML
dependency graph

src

src

src

src

Retrieves sources for
all packages specified
within YAML file

vendor A

vendor B

ScanCode

API

wraps underlying
license / copyright
scanners with
a common API

ABCD-style YAML
dependency graph

ScanCode
JSON/HTML/SPDX
scan results

here
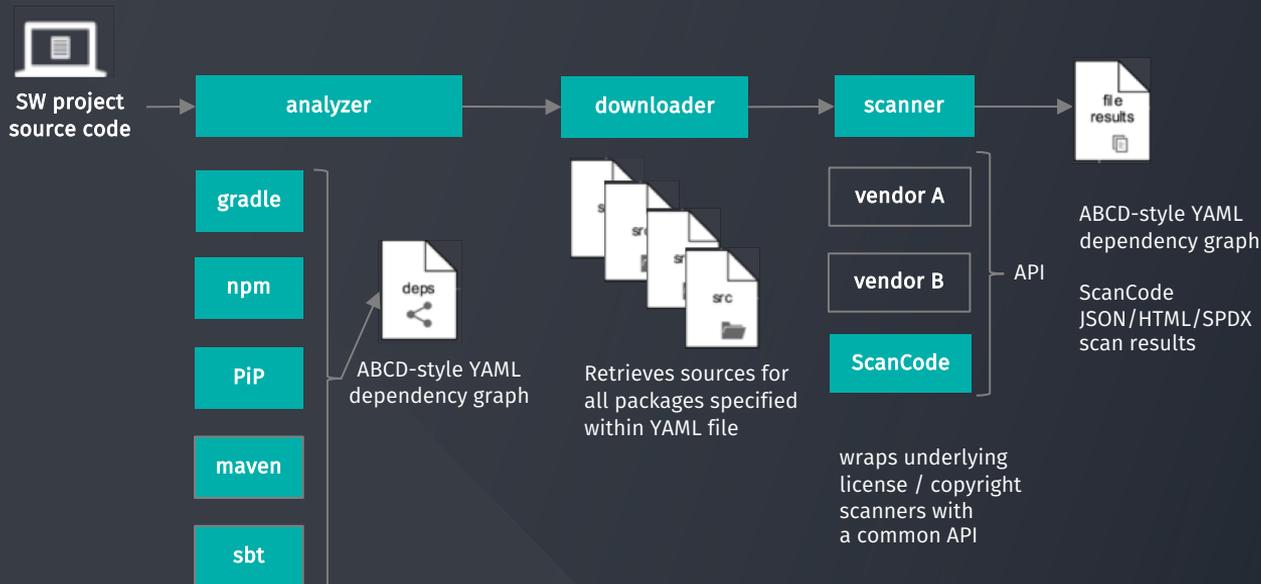
Open-sourcing ORT lead to participating in ClearlyDefined (early 2018)

Harvest

Curate

Upstream

ClearlyDefined
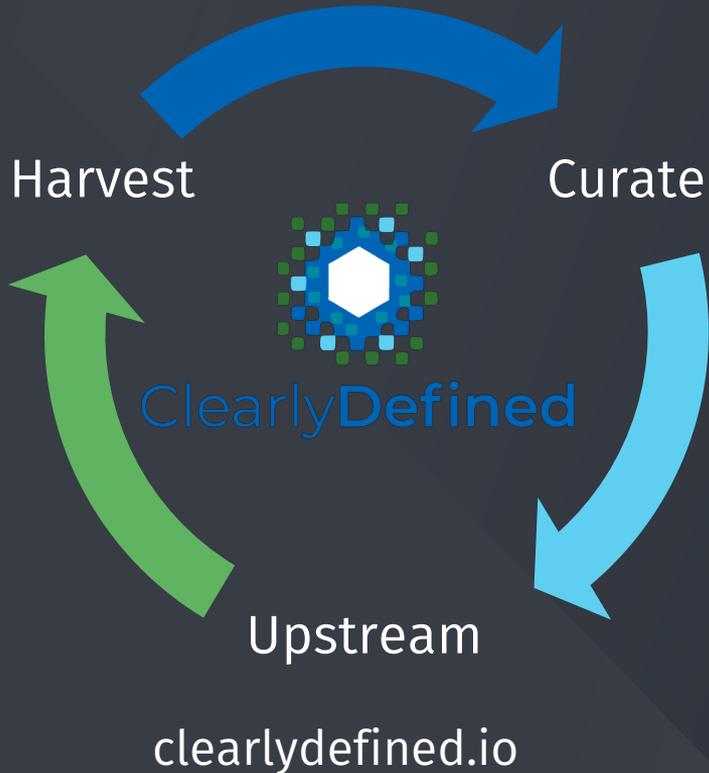
clearlydefined.io

Enabling FOSS project success through clearly defined license and security information

- Community solution to a community problem
- Automated scanning of released components
- Crowd-sourcing curation of ambiguous or missing information
- Contribute updates to upstream projects
- Immediate focus on license, source location, and attribution parties
- Longer-term interest in security, accessibility, localization

- Open Source Initiative incubator project
- Partners: Amazon, Eclipse, Google, HERE, nexB, Microsoft, Qualcomm, Software Heritage, SAP

github.com/clearlydefined

here

OSS
Review Toolkit : scaling OSS reviews in CI/CD (early 2019)

Goal: enable review during source creation by providing
easy, open-source & scalable tooling for developers
to do basic compliance
and share results in open standard formats

Vulnerabilities databases

local + ClearlyDefined

Security advisories

Fix package metadata
and licenses
found in SW project

local + ClearlyDefined

Curated FOSS metadata

advisor

File defining not
distributed OSS

SW project
source code
change

| .ort.yml | → | analyzer | → | downloader | → | scanner | → | evaluator | OK → | Reporter | → |

NOT OK

Maven	Maven
Gradle	Gradle
sbt	GoDep
More coming …	

Retrieve included
OSS dependencies from
dependency managers

Dependencies
graph

Retrieve all
source code

ScanCode

+

ClearlyDefined

Detect licenses / copyrights
in source code

issues
found ⚠

OSS used
OK or NOT OK
based on
computable
rules

OK   NOT OK

ISSUE TRACKER

Ticket based
review process

Open Source notices
SW deliverable

+

Dependency graph
Review results as
HTML / SPDX / JSON

documenter → SPDX

16

Open-sourced & released at github.com/heremaps/oss-review-toolkit

© 2019 HERE

Organizational Lessons Learned

- Open source solutions to solve problems coming from FOSS community
  is the way forward
- People need to see scale of faced challenges to understand them
- Management buy-in requires turning open source challenges
  into revenue growth/cost savings

Still learning

- Scaling our Open Source Office with business needs
- Managing the compliance challenge: Balance vs. Sustainable vs. Maintainable
- Best way to educate ORT users how to resolve license issues
- Building up a open source friendly culture
- Contributing at scale as a company

Process Lessons Learned

- Only culture, data and risk driven processes scale
- Scaling requires a trust but verify approach
- Shift-left compliance is a no brainer but difficult to achieve
- People do not expect Agile in compliance processes
- Do OSS reviews for internal SW but with more relax checks
- Respect TL;DR – show only what is applicable in bit size chunks
- People trust bots in processes more than people
- Usage approval = code context + license context + product context

Still learning
- Designing human centric + highly automated processes

Tooling **Lessons Learned**

- Compliance tools are by default behind build tools changes
- Two questions to ask your compliance vendors
  - How do you determine included OSS?
  - How do you conclude applicable licenses?
- Automation requires more data incl. package level, scope and tree
- Better to be fully in control than to move fast but be restricted in functionality and scaling
- Scaling automation requires setting engineering standards, growing the right culture and participating in the FOSS community

Still learning
- How to fill the gaps between the various OSS-based compliance tools

Thank you

Contact

Thomas Steenbergen
HERE Open Source Office

✉ thomas.steenbergen@here.com
🐦 @tsteenbe
💼 linkedin.com/in/tsteenbe

OSS Review Toolkit

https://github.com/heremaps/oss-review-toolkit

Welcome your feedback and contributions