RSA Conference 2018
San Francisco | April 16 – 20 | Moscone Center

NOW MATTERS

SESSION ID: DEV-R14R

OPEN SOURCE IN SECURITY-CRITICAL ENVIRONMENTS

James Zemlin

Executive Director
Linux Foundation
@jzemlin

Open Source is here to stay in security critical environments and every place software is used

Linux has grown into the most important open source project in the world

100%	82%	2nd	90%	90%	62%	#1
Supercomputer Market	Smartphone Market Share	To Windows in Enterprise	Mainframe Customers	Public Cloud Workload	Embedded Systems Market	Internet Client

Every market Linux has entered it eventually dominates

Linux Evolves Faster Than Ever

4,300
Contributors From 450 Organizations

10,000
Lines of Code Added Daily

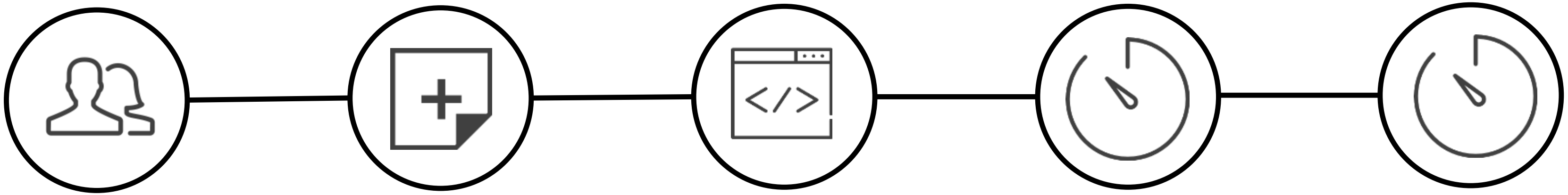
2,000
Lines of Code Modified Daily

2,500
Lines of Code Removed Daily

8.5
Changes Per Hour

Open Source Development is Accelerating

23M+
Open Source Developers



78M+
Repositories on Github

41B+
Lines of Code

1,100
New Projects a Day

10,000+
New Versions per day

Sources: Sourceclear, Sonatype, Github

It’s actually open source software that’s eating the world.

- Venturebeat 2015

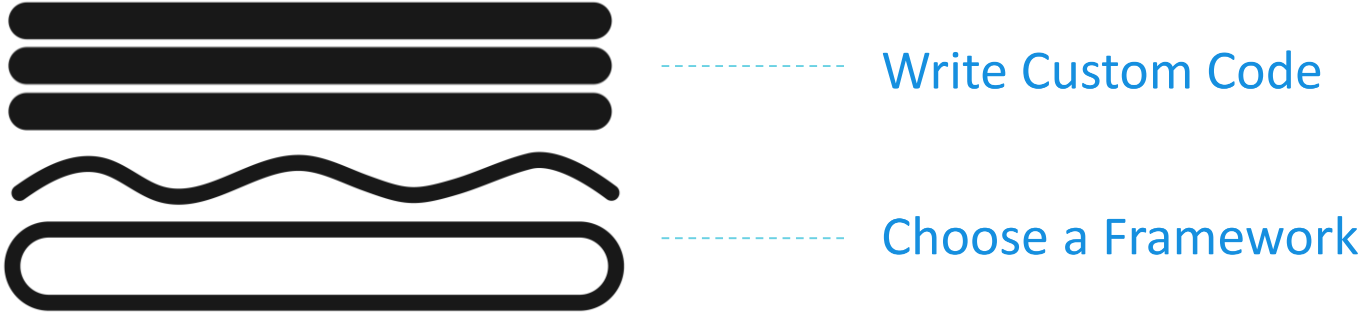
Creating Applications these days is like making a sandwich

Code Club (Sandwich)

Code Club (Sandwich)

Choose a Framework

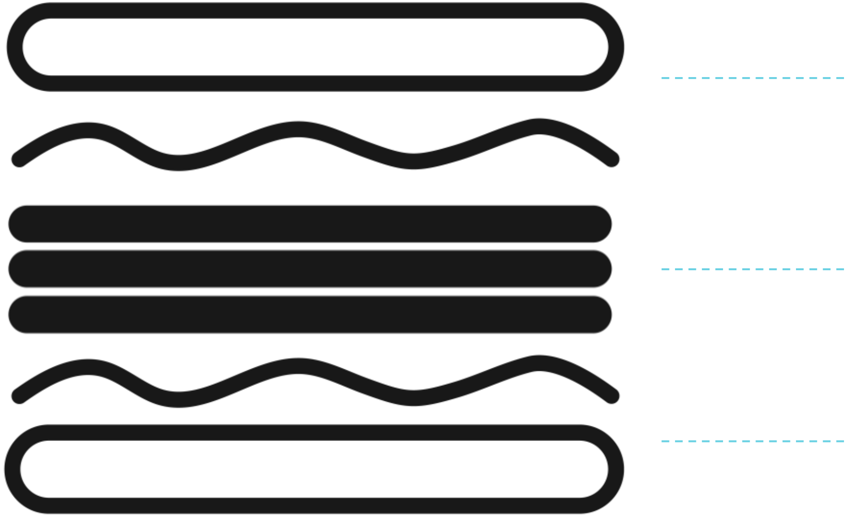
Code Club (Sandwich)



Write Custom Code

Choose a Framework

Code Club (Sandwich)



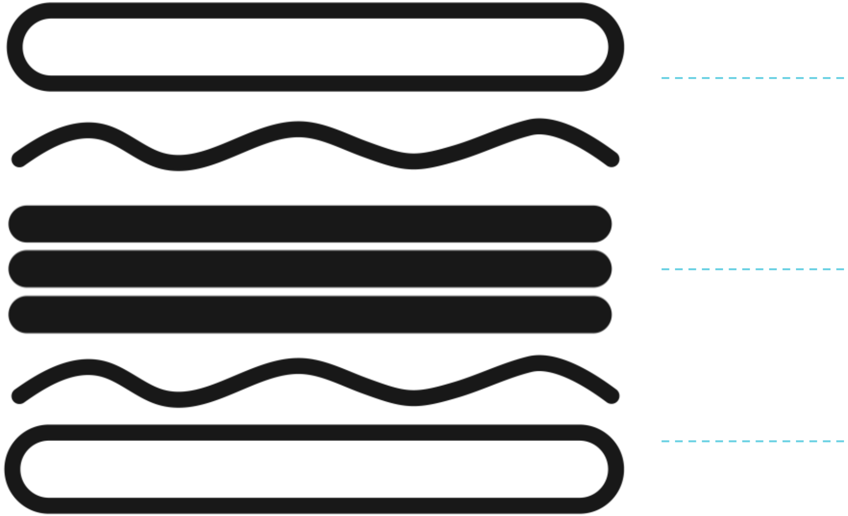
Use Open Source Libraries to Solve Problems

Write Custom Code

Choose a Framework

Code Club (Sandwich)

Open Source Code = ~ 90%



Use Open Source Libraries to Solve Problems

Open Source Code (~70%)

Write Custom Code

Custom Code (~10%)

Choose a Framework

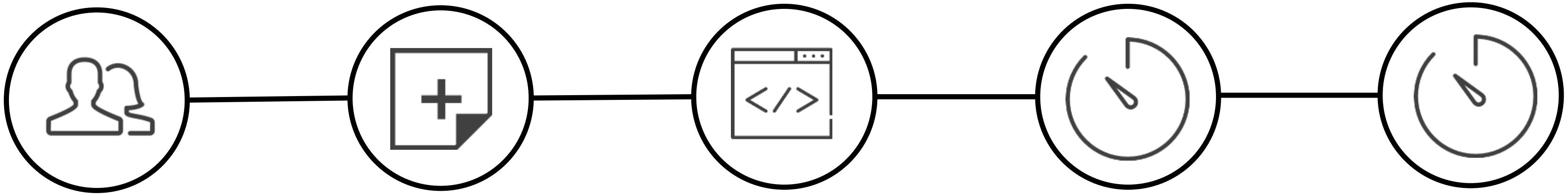
Open Source Code (~20%)

So much code – so little time

23M+
Open Source Developers

41B+
Lines of Code

10,000+
New Versions per day

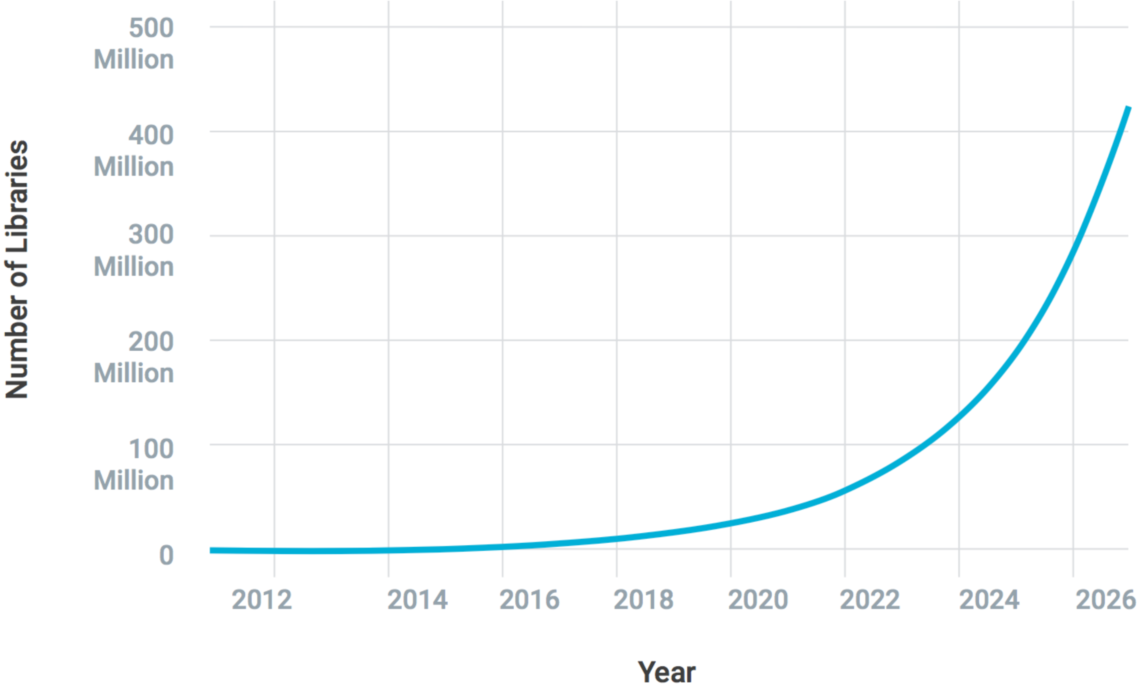


78M+
Repositories on Github

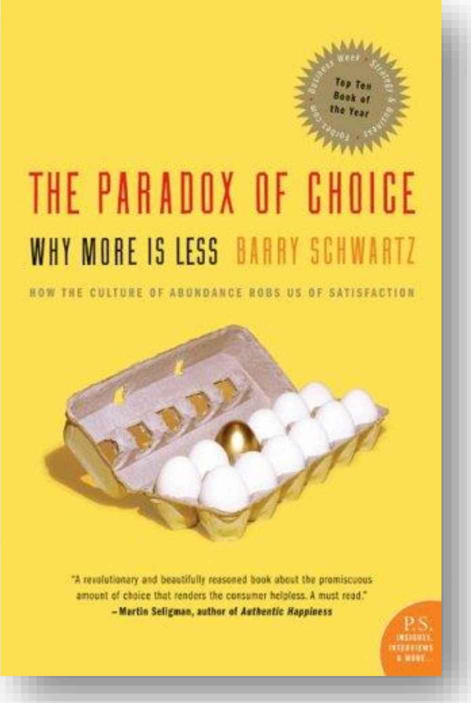
1,100
New Projects a Day

Sources: Sourceclear, Sonatype, Github

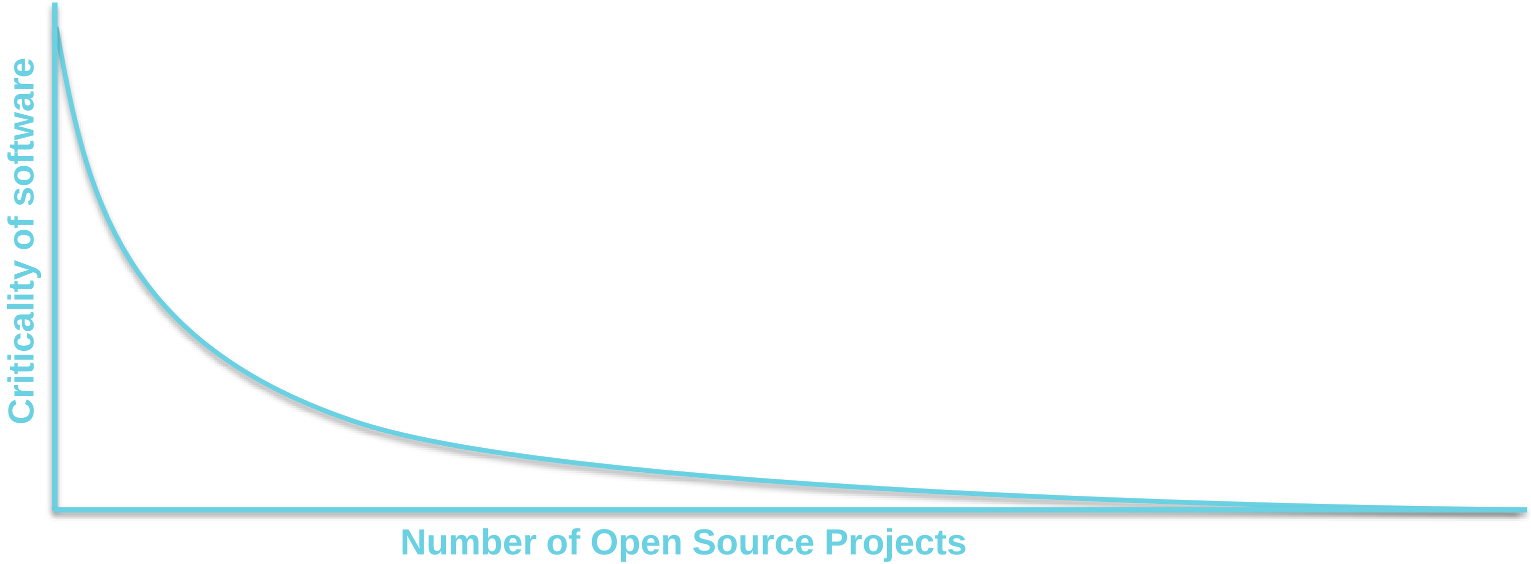
Open source isn't slowing down any time soon



All this abundance has created anxiety

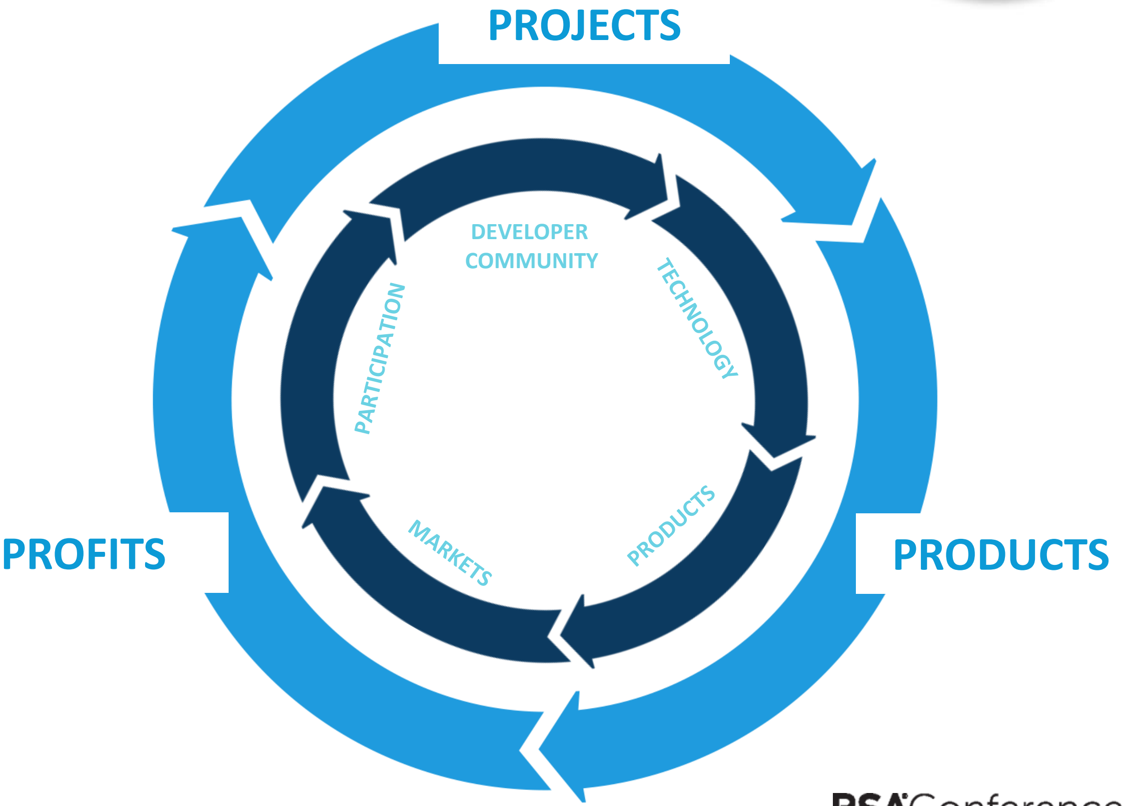

Business Week • Strategy & Business • Forbes.com
Top Ten Book of the Year
THE PARADOX OF CHOICE
WHY MORE IS LESS
BARRY SCHWARTZ
HOW THE CULTURE OF ABUNDANCE ROBS US OF SATISFACTION
"A revolutionary and beautifully reasoned book about the promiscuous amount of choice that renders the consumer helpless. A must read."
– Martin Seligman, author of Authentic Happiness
P.S. INSIGHTS, INTERVIEWS & MORE...

The real question is which projects matter?

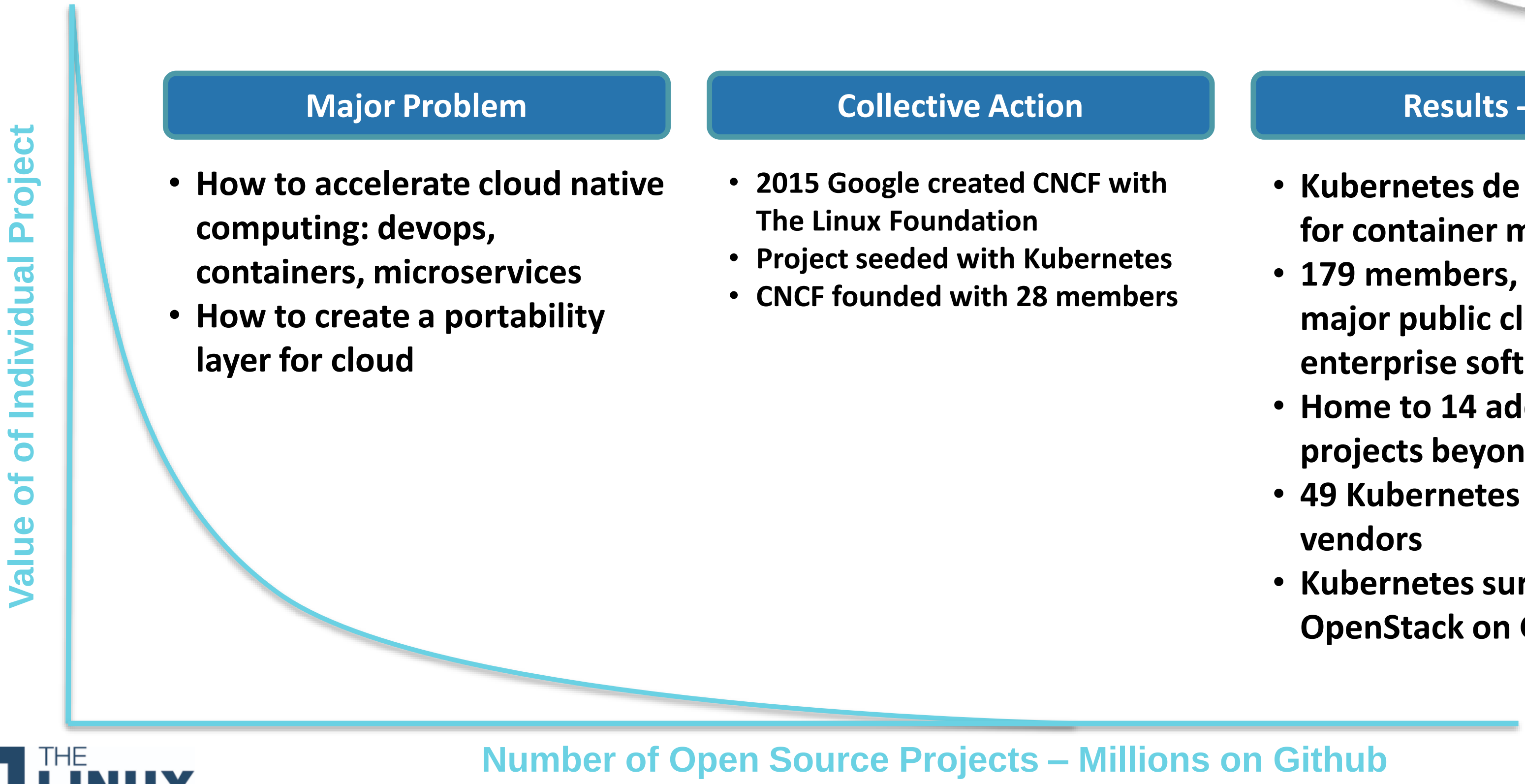


How do we make important projects sustainabl

Successful Projects depend on members, developers, standards and infrastructure to develop products that the market will adopt.



WHEN THIS CYCLE WORKS, IT WORKS WELL



Major Problem

- How to accelerate cloud native computing: devops, containers, microservices
- How to create a portability layer for cloud

Collective Action

- 2015 Google created CNCF with The Linux Foundation
- Project seeded with Kubernetes
- CNCF founded with 28 members

Results - 2018

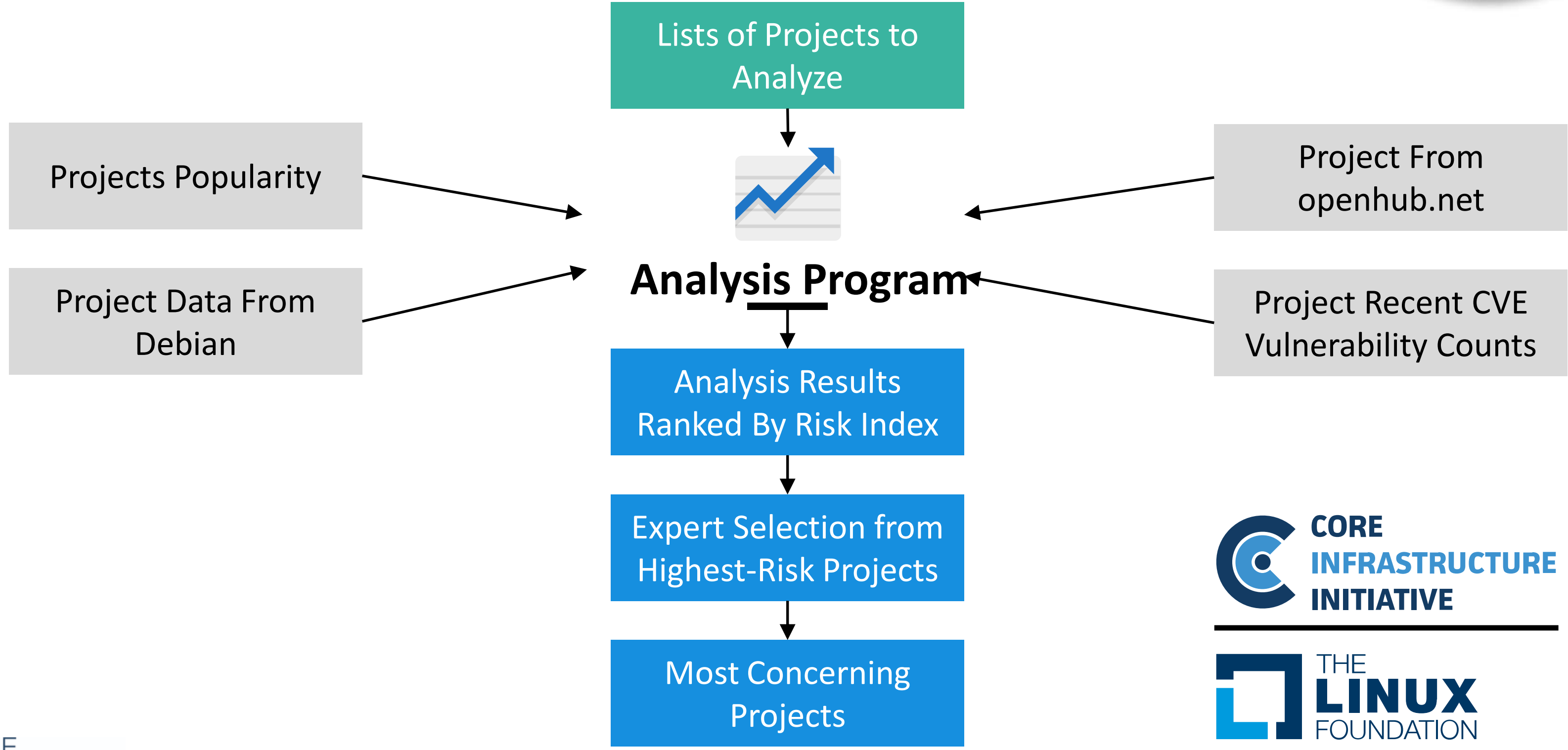
- Kubernetes de facto standard for container management
- 179 members, including all major public clouds and enterprise software vendors
- Home to 14 additional projects beyond Kubernetes
- 49 Kubernetes certified vendors
- Kubernetes surpasses OpenStack on Google trends

SOMETIMES THE SYSTEM DOESN’T WORK

Questions to ask

- What is the most important and security critical shared software in the world?
- Who is creating and maintaining that software?
- Why are the creating and maintaining that software?
- Is it secure, reliable, and healthy?

Core Infrastructure Initiative Census Project



Lists of Projects to Analyze

Projects Popularity

Project Data From Debian

Project From openhub.net

Project Recent CVE Vulnerability Counts

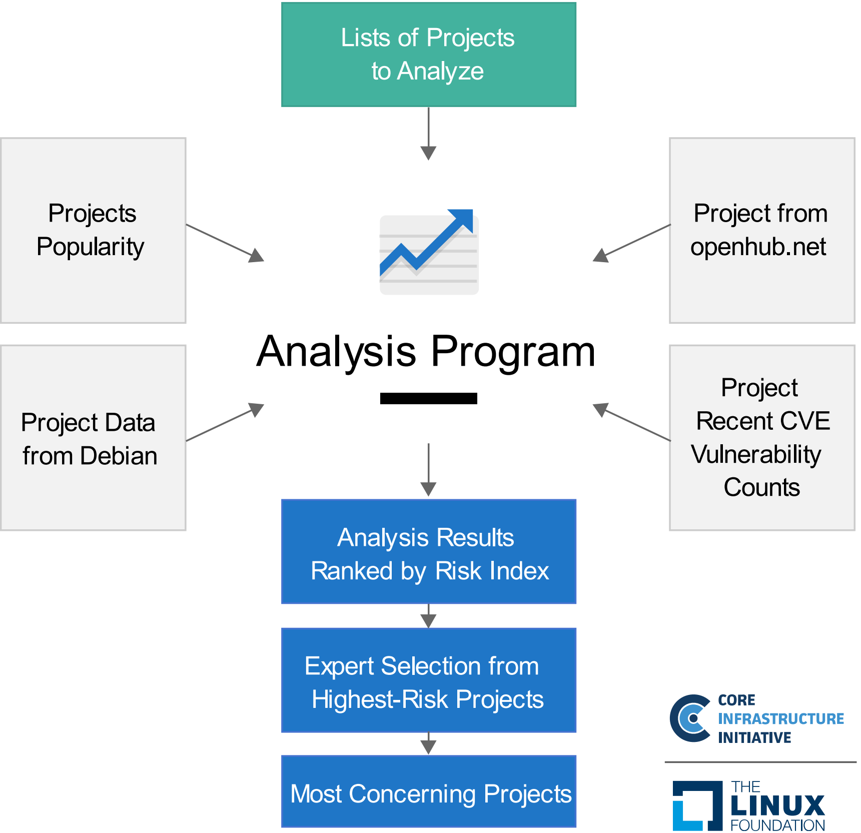
Analysis Program

Analysis Results Ranked By Risk Index

Expert Selection from Highest-Risk Projects

Most Concerning Projects

Core Infrastructure Initiative Census Project


Lists of Projects to Analyze
Projects Popularity
Project from openhub.net
Analysis Program
Project Data from Debian
Project Recent CVE Vulnerability Counts
Analysis Results Ranked by Risk Index
Expert Selection from Highest-Risk Projects
Most Concerning Projects

CORE INFRASTRUCTURE INITIATIVE

THE LINUX FOUNDATION

Current Algorithm

- Project has website (1 if no)
- Written in C or C++ (2 if yes)
- CVE vulnerability reports: 3 points if 4+ , 2 points for 2-3, 1 point for 1.
- 12 month contributor count: 5 points for 0 contributors, 4 points for 1-3 contributors, 2 points if the number is unknown.
- Top 10% most popular Debian package: 1 if yes
- Exposure values: 2 points if directly exposed to the network (as server or client), 1 point if it is often used to process data provided by a network, and 1 point if it could be used for local privilege escalation.
- Application data only: *Subtract* 3 points if the Debian database reports that it is “Application Data” or “Standalone Data” (not an application)

Tremendous Systemic Risks to the Internet Still Unaddressed

Binary Package Name	Source Package Name (If Different)	CII 2016 Census Risk Score
ftp	netkit-ftp	11
netcat-traditional	netcat	11
tcpd	tcp-wrappers	11
whois		11
at		10
libwrap0	tcp-wrappers	10
traceroute		10
xauth		10
bzip2		9
hostname		9
libacl1	acl	9
libaudit0	audit	9
libbz2-1.0	bzip2	9
libept1.4.12	libept	9
libreadline6	readline6	9
libtasn1-3		9
linux-base		9
telnet	netkit-telnet	9

Source: CII 2016 Census

The Big Risk:

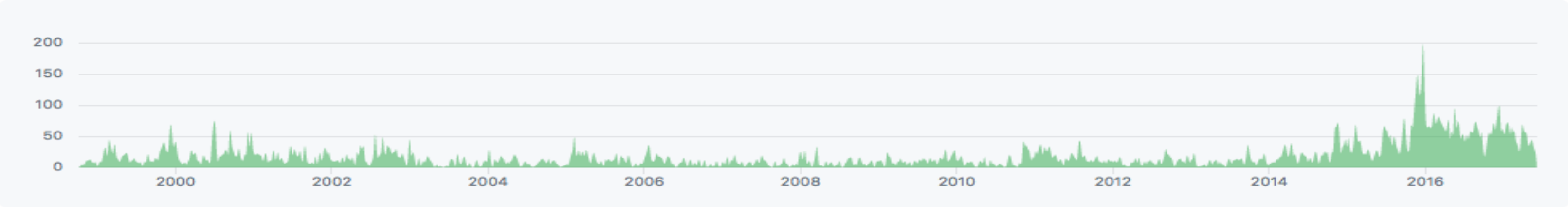
Commonly used open source code and libraries are among the most at risk to cyber attacks or other potential threats that could bring down the global Internet.

A little love goes a long way

Back Row: Geoff Thorpe, Steve Marquess, Matt Caswell, Tim Hudson, Kurt Roeckx, Lutz Jänicke, Mark Cox, Richard Levitte, Emilia Käsper
Front Row: Rich Salz, Andy Polyakov

2014 - OpenSSL was maintained by two people and moribund
2016 – Recorded more activity than in the entire previous history of the project, including:

- Three new releases
- 3889 commits
- 481 GitHub users
- Thousands of forks.
- 1052 pull requests closed
- 47 CVEs reported and handled



How to create secure code?

We must secure the most critical open source software projects that power the world’s infrastructure, and to promote a culture of secure coding.

100 Projects Granted CII Best Practice Badge

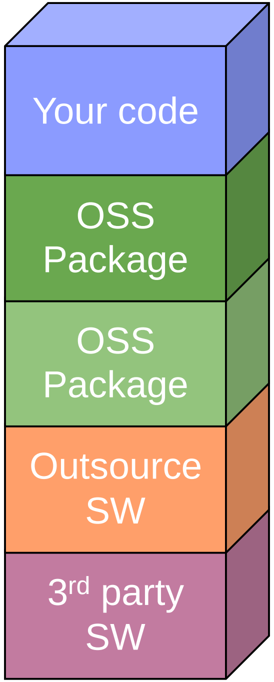
- Initiative launched in May 2016 to raise awareness of development processes and governance steps for better security outcomes
- The badge makes it easier for users of open source projects to see which projects take security seriously, it isn't a "rubber stamp" process
- 1,000 projects registered for the badge

Education

- One of the largest causes of security vulnerabilities is developers being unaware of security best practices
- We need courses for open source developers for Security and Auditing
- Organizations like SAFECode provide curriculum and training but we need more

We need to be able to pass information about software bill of materials across the tech value chain in a simple and reliable way. You can't fix bugs for code you don't event know you have.

Software Tracking: The Challenge



Your code

OSS Package

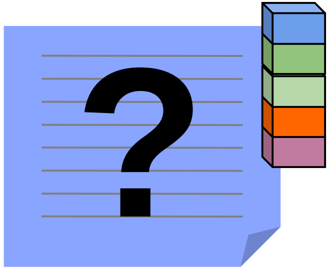
OSS Package

Outsource SW

3rd party SW

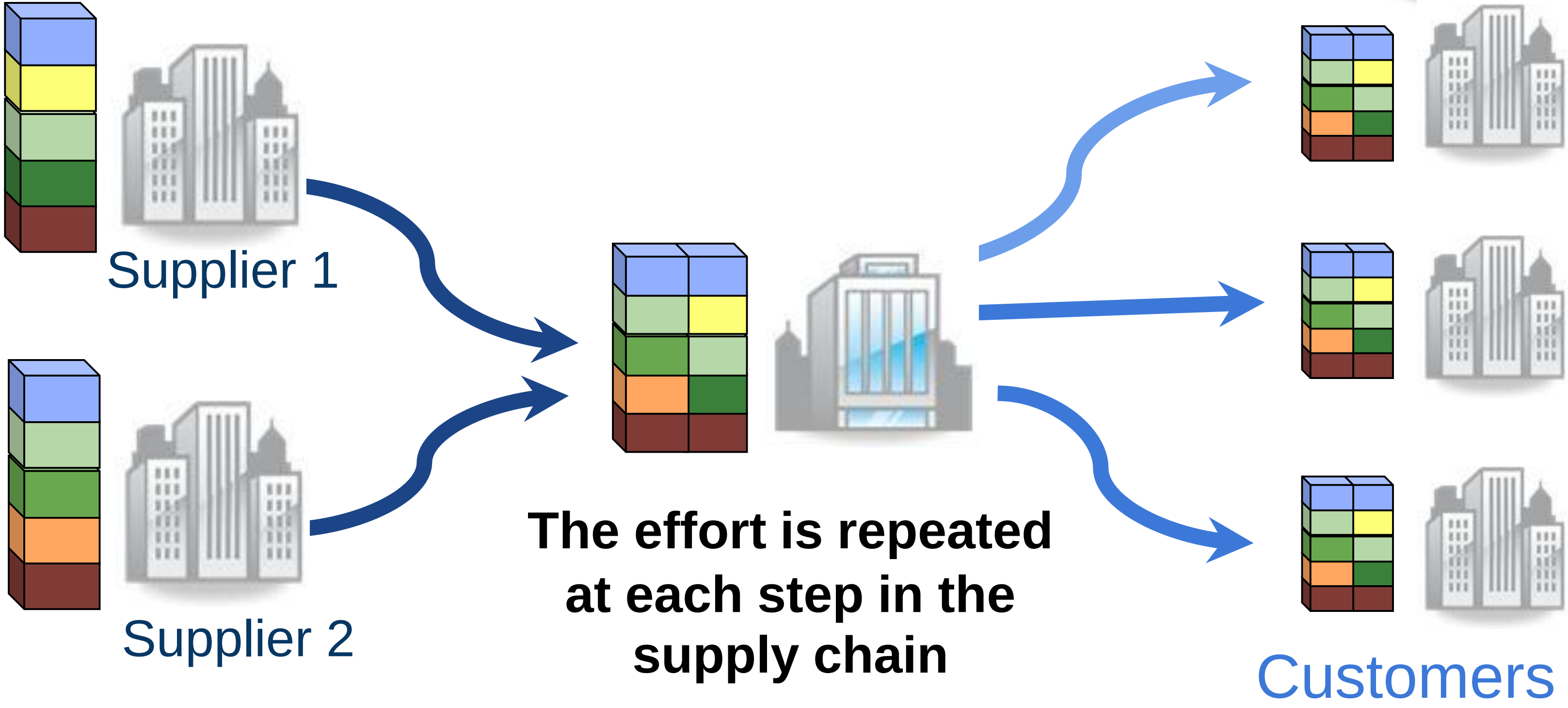
Companies combine Open Source Software with other software

Software Bill of Materials (BOM)



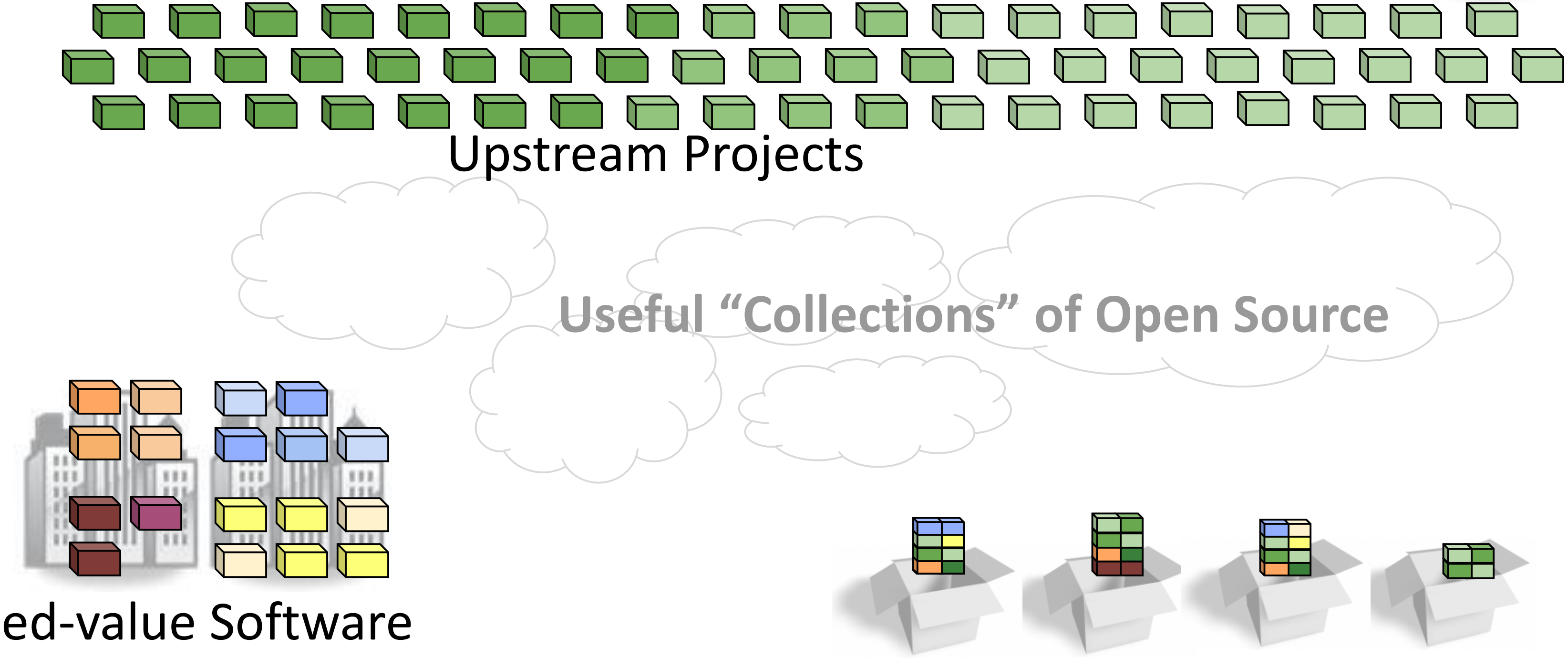
Creating an accurate bill of materials and notices requires effort & research

Software BOM: The Challenge



The effort is repeated at each step in the supply chain

“Open Source”-scape



Software Package Data eXchange

Open Standard:

- A standard format for communicating the licenses and copyrights and identity associated with software packages

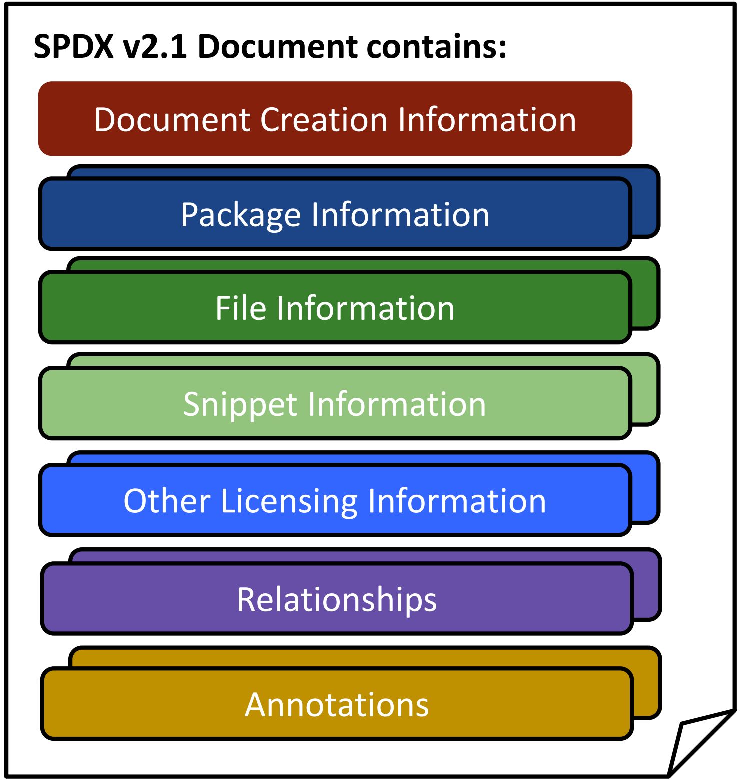
Vision:

- To help reduce redundant work in determining software BOM information and facilitate compliance

Guiding principles:

- Human and machine readable
- Focus on capturing facts; avoid interpretations

What makes up an SPDX Document?



SPDX v2.1 Document contains:

- Document Creation Information
- Package Information
- File Information
- Snippet Information
- Other Licensing Information
- Relationships
- Annotations

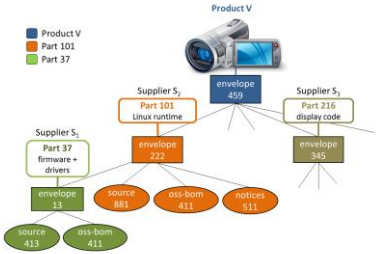
Emerging “Between Organization” Trust Models

Software Parts Ledger - utilizes Blockchain to manage open source across the supply chain. Utiliilzes Hyperledger Sawtooth Platform & SPDX based BOM to conform to OpenChain best practices.

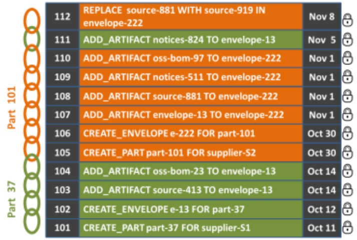
See: https://github.com/Wind-River/sparts

Accepted 2018/3 into Hyperledger Labs - https://github.com/hyperledger-labs/hyperledger-labs.github.io/blob/master/labs/SParts.md

ClearlyDefined - Announced 2018/3 - calls for participation in currating the metadata to summarize projects. See ClearlyDefined.io for more information.



Software Parts Ledger


Part 101
Part 37

112	REPLACE source-881 WITH source-919 IN envelope-222	Nov 8
111	ADD_ARTIFACT notices-824 TO envelope-13	Nov 5
110	ADD_ARTIFACT oss-bom-97 TO envelope-222	Nov 1
109	ADD_ARTIFACT notices-511 TO envelope-222	Nov 1
108	ADD_ARTIFACT source-881 TO envelope-222	Nov 1
107	ADD_ARTIFACT envelope-13 TO envelope-222	Nov 1
106	CREATE_ENVELOPE e-222 FOR part-101	Oct 30
105	CREATE_PART part-101 FOR supplier-S2	Oct 30
104	ADD_ARTIFACT oss-bom-23 TO envelope-13	Oct 14
103	ADD_ARTIFACT source-413 TO envelope-13	Oct 14
102	CREATE_ENVELOPE e-13 FOR part-37	Oct 12
101	CREATE_PART part-37 FOR supplier-S1	Oct 11

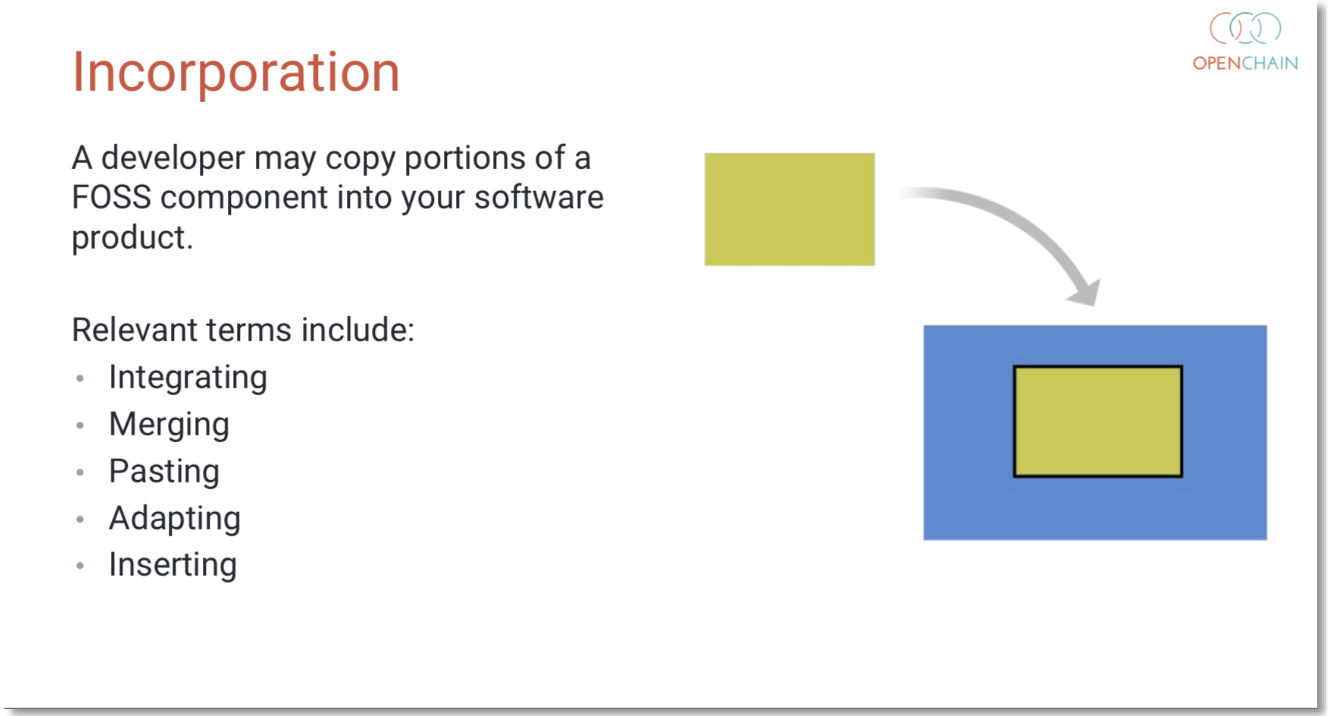
ClearlyDefined

Sharing software bill of materials is critical part of security process

- OpenChain builds trust in open source by making sharing of software BOM simpler and more consistent
- Adobe, Arm, Cisco, Harmen, Hitachi, HPE, GitHub, Qualcomm, Siemens, Toyota, Wind River and Western Digital

OPENCHAIN

Learn how open source software flows



OPENCHAIN

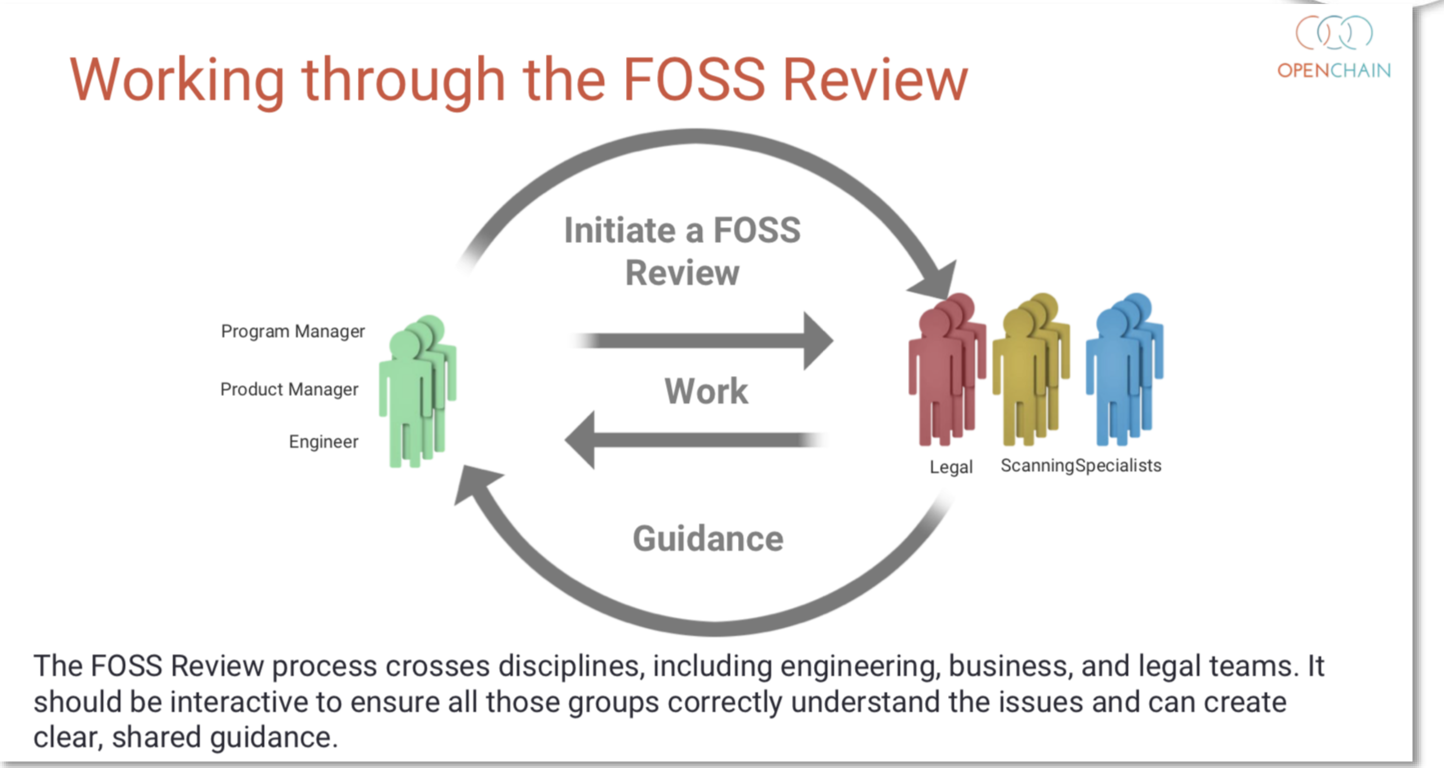
Incorporation

A developer may copy portions of a FOSS component into your software product.

Relevant terms include:

- Integrating
- Merging
- Pasting
- Adapting
- Inserting

Get a process in place



OPENCHAIN

Working through the FOSS Review

The FOSS Review process crosses disciplines, including engineering, business, and legal teams. It should be interactive to ensure all those groups correctly understand the issues and can create clear, shared guidance.

We need to invest in tools that test upstream code

Frama-C False-Positive-Free Checking

- Frama-C is a highly respected static checker
- When used with test cases and modified Unix standard functions, it is able to detect bugs without false positives
- Proposal is to modify several standard Unix functions to support false-positive-free operation on OpenSSL
- In addition, the proposal is to use the American Fuzzy Lop fuzzer to automatically generate test cases from which Frama-C can detect bugs

Fuzzing

- https://fuzzing-project.org/ is Hanno Böck’s project
  - Uses zzuf, Address Sanitizer and american fuzzy lop to find bugs in open source projects
  - Discovered numerous GnuPG bugs in Feb 2015
  - He and others have found numerous bugs in many projects: http://lcamtuf.coredump.cx/afl/#bugs
- His main activity is to convert the fuzzer output into reproducible test cases and file bugs for them
- He is also doing great work training new developers to become expert fuzzers
- CII is also reaching out to fuzzing toolkit authors

Reproducible Builds

- Debian and Fedora rely on package maintainers to compile source code from the upstream authors
- Because the resulting binaries depend on machine configuration (like timestamps and file ordering), these binaries are not reproducible
- That makes it impossible to independently verify that the binaries have not been tampered with
- Binary reproducibility should become an expected attribute of free software distros

We need to invest in audit of upstream open source code for critical shared infrastructure

Auditing

Auditing: Many critical open source projects do not have resources to audit

- Auditing finds critical bugs that won’t be found any other way
- Auditing is expensive, time consuming and only finds a subset of the bugs so it can’t be the only tool
- OpenSSL audit underway

How to get involved?

Follow up material

- See Linux Foundation-sponsored Institute for Defense Analysis (IDA report, "Open Source Software Projects Needing Security Investments”
- Some of the projects we're most concerned about (because they are ubiquitously deployed and could result in Heartbleed-style vulnerabilities) include compression libraries (bzip2, gzip, unzip, zlib) and format libraries (libjpeg, libpng, and expat)
- Unlike before Heartbleed, there is actually a group focused on these issues. Two major programs we’re undertaking with IDA:
  - CII is not only reactively looking for broken projects (i.e., fighting fires) through our Census Project
  - We are also developing the building codes (in terms of security best practices) to avoid fires in the future