



OPEN SOURCE  
LEADERSHIP SUMMIT

# How to Manage Open Source at Scale in a Global Enterprise?

Peter Giese, SAP Open Source Program Office

14. March 2019



Gardener



CLA Assistant



UI5 Web Components



OpenUI5



Vulas




InfraBox



Kyma

## Linux Foundation

In addition to involvement with the Linux Foundation projects listed here, SAP is a Silver member of the Linux Foundation itself

[Visit Site](#) 

## Apache Software Foundation

SAP is involved with the Apache Olingo project, and with Apache Hadoop through Altiscale

[Visit Site](#) 

## OpenJDK

SAP is a contributor to the OpenJDK project

[Visit Site](#) 

## Cloud Native Computing Foundation

Platinum member; SAP is represented on the Board

[Visit Site](#) 

## Eclipse Foundation

Founding member and a strategic development member of the Eclipse Foundation.

[Visit Site](#) 

## Open Stack Foundation

SAP is a Corporate Sponsor of the Open Stack Foundation

[Visit Site](#) 

## Cloud Foundry Foundation.

Founding member and a Platinum member; SAP is represented on the Board of Directors

[Visit Site](#) 

## Open API

Silver member.

[Visit Site](#) 

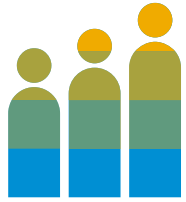
## ToDo Group

General member.

[Visit Site](#) 

# SAP: The World's Largest Provider of Enterprise Application Software

---



**96,000+**  
Employees



**180+**  
Countries



**18,300+**  
Partners



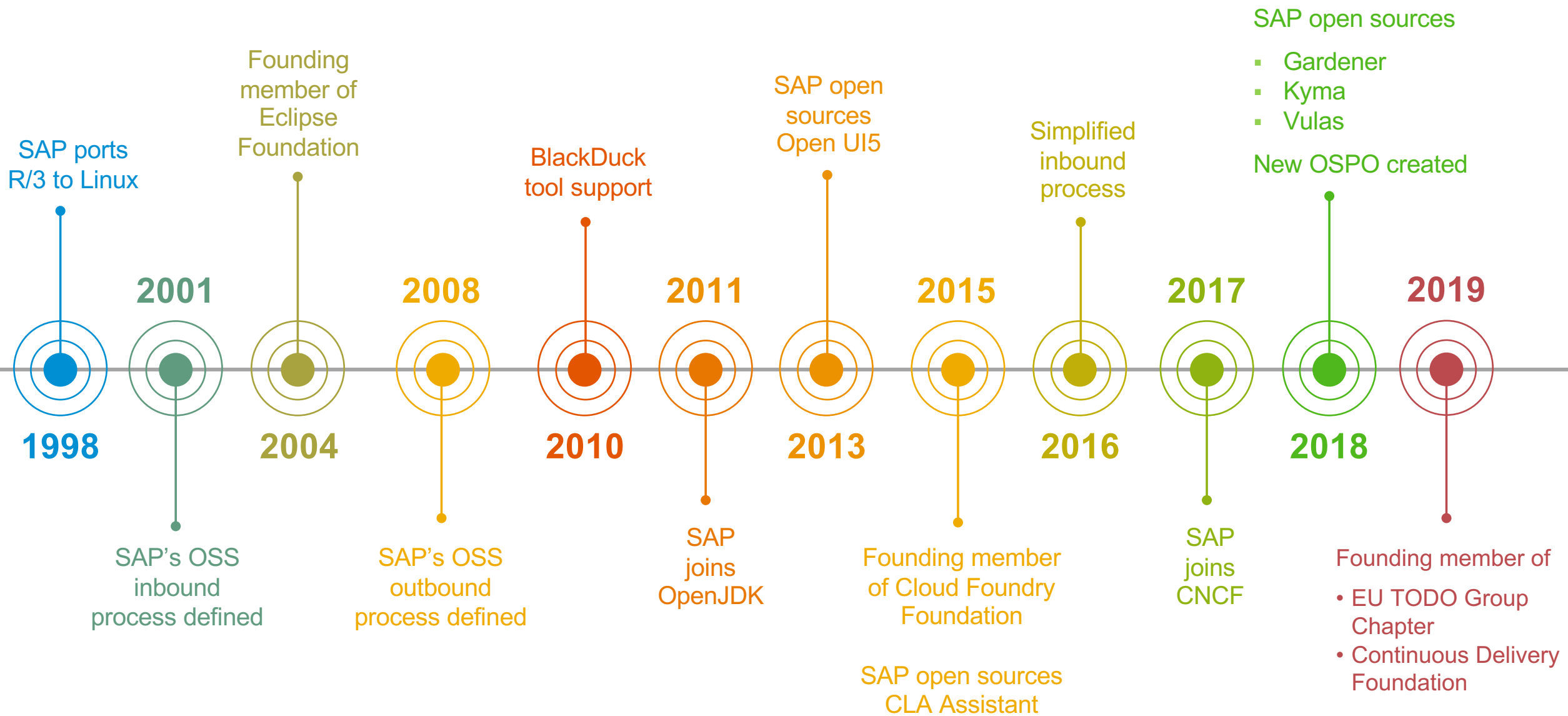
**77%**  
of the world's  
transaction revenue



**78%**  
of the  
world's food

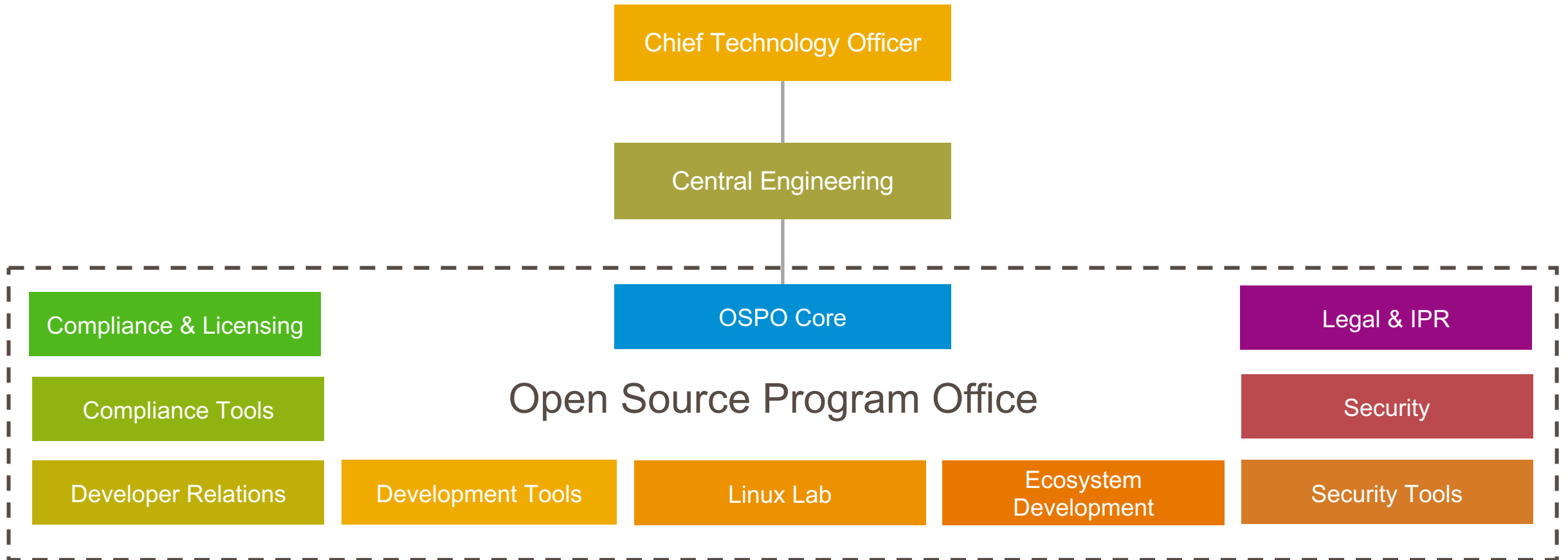


**82%**  
of the world's  
medical devices



# Open Source Program Office (OSPO)

---

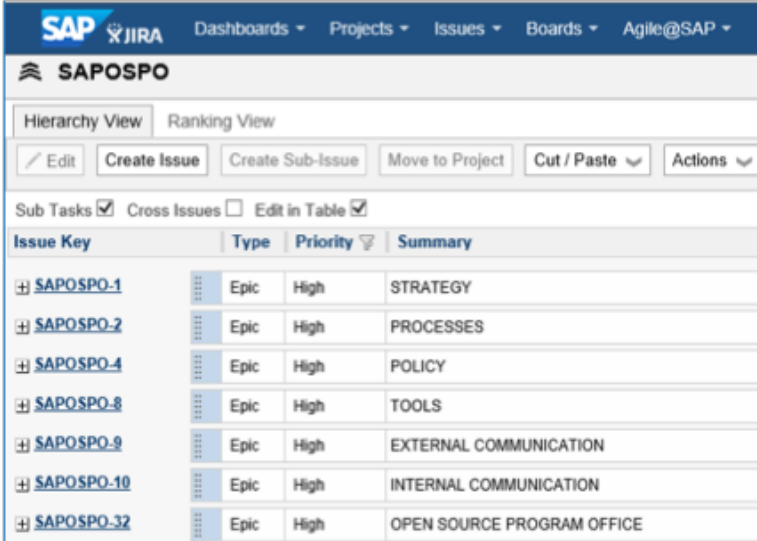


# OSPO Working Mode – Scrum

One joint OSPO product backlog in Jira

Divided into seven epics

Each epic handled by a cross-functional scrum team



The screenshot shows the SAP Jira interface for the SAPOSPO project. It displays a backlog with seven epic items, each with a key, type, priority, and summary. The interface includes navigation tabs for Hierarchy View and Ranking View, and buttons for Edit, Create Issue, Create Sub-Issue, Move to Project, Cut / Paste, and Actions. Checkboxes for Sub Tasks, Cross Issues, and Edit in Table are also visible.

Issue Key	Type	Priority	Summary
SAPOSPO-1	Epic	High	STRATEGY
SAPOSPO-2	Epic	High	PROCESSES
SAPOSPO-4	Epic	High	POLICY
SAPOSPO-8	Epic	High	TOOLS
SAPOSPO-9	Epic	High	EXTERNAL COMMUNICATION
SAPOSPO-10	Epic	High	INTERNAL COMMUNICATION
SAPOSPO-32	Epic	High	OPEN SOURCE PROGRAM OFFICE

Policy

Processes

Tools

Strategy

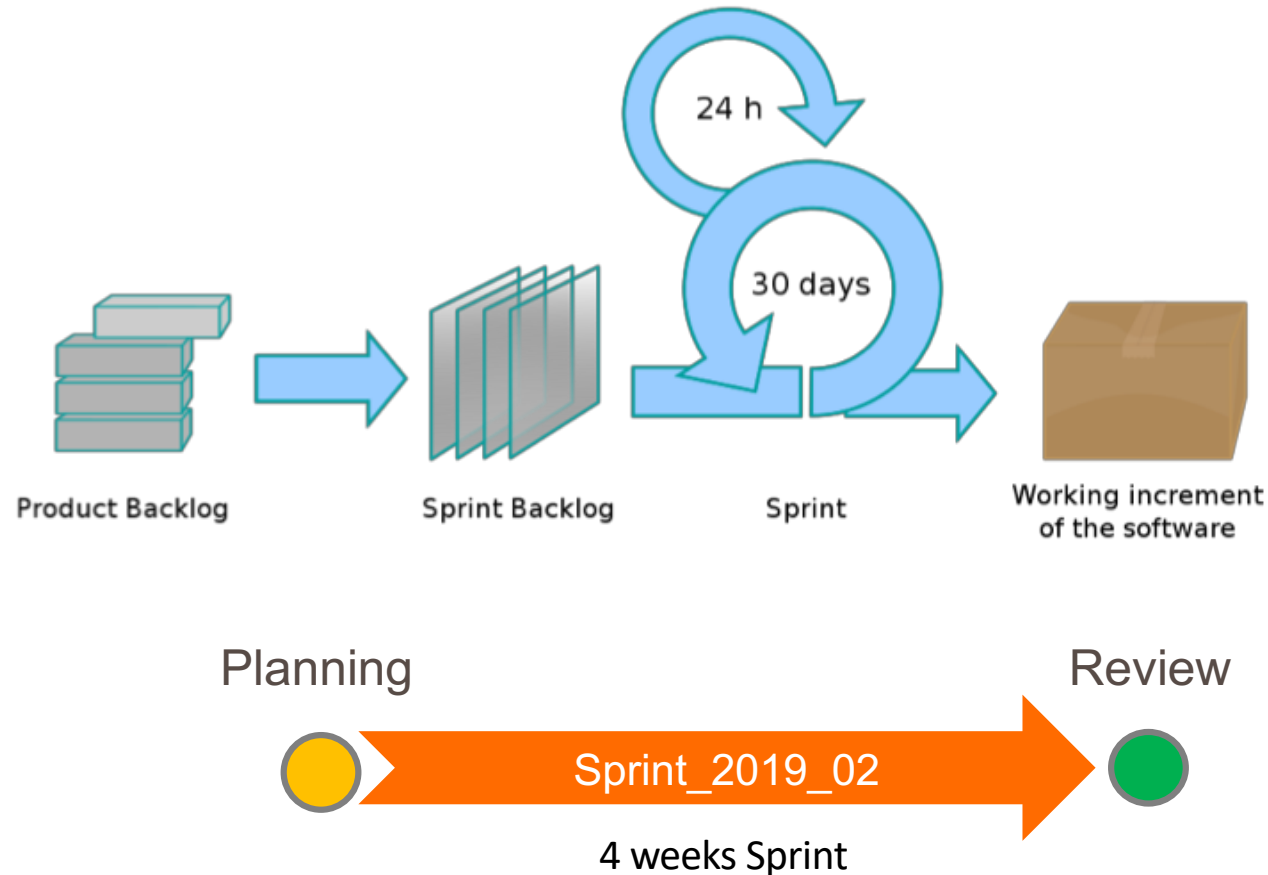
External Communication

Internal Communication

OSPO

# OSPO Working Mode – Scrum

Source: [https://commons.wikimedia.org/wiki/File:Scrum\\_process.svg](https://commons.wikimedia.org/wiki/File:Scrum_process.svg)



After nine sprints:

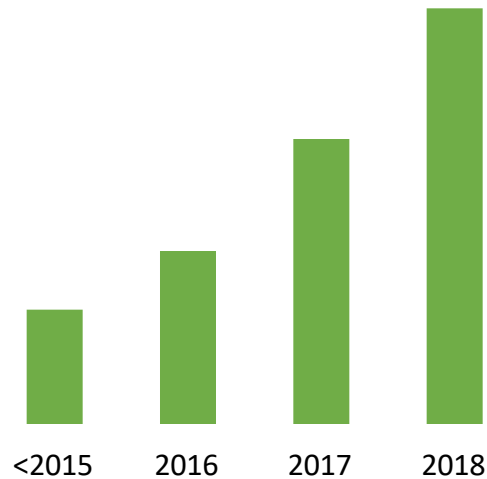
150+ sprint backlog  
items completed

400+ product backlog  
items created



# Exponential Growth of Open Source in Enterprise Application Software

## SAP Inbound Open Source



### Customer Expectations:

- SAP product standards
- Compliance



## SAP Outbound Open Source



1000+ Contributors on Github

### Customer & Community Expectations:

- SAP product standards



CII Badge Program

# Core Infrastructure Initiative (CII) Badge Program

---



- CII Badge is an open source **secure development** maturity model
- CII Badge criteria **codify best-practices** used by open source projects
- **CII Gold Badge** best matches SAP's Product Standards

Some projects cannot fulfil the [[contributors\\_unassociated](#)] continuity criterion that “the project MUST have at least two unassociated significant contributors”

→ **CII Silver Badge**

# CLL Gold & Silver Badge Criteria – Open Source Tools from SAP

---

[[dco](#)] → CLA Assistant



[[vulnerabilities fixed 60 days](#)] → Vulas



[[no leaked credentials](#)] → SCCS; not open sourced (yet)



## CII Badge Gold [\[dco\]](#):



The project SHOULD have a legal mechanism where all **developers** of non-trivial amounts of project software **assert that they are legally authorized to make these contributions**.

The most common and easily-implemented approach for doing this is by using a [Developer Certificate of Origin \(DCO\)](#), where **users add "signed-off-by" in their commits** and the project links to the DCO website.

However, this MAY be implemented as a **Contributor License Agreement (CLA)**, or other legal mechanism.

# CLA Assistant

---



CLA Assistant ([github.com/cla-assistant](https://github.com/cla-assistant)) helps to handle the legal side of contributions to open source projects and to streamline the contribution workflow.

It is also provided as free hosted offering ([cla-assistant.io](https://cla-assistant.io)) which allows contributors to sign a CLA from within a pull request by authenticating themselves with their GitHub account.



# CLA Assistant Demo

# CLA Assistant - Adoption

---



**4.300+**  
**300+**  
**52.000+**  
**43.000+**

linked GitHub repositories  
linked GitHub organizations  
CLA signatures  
users



CII Badge Gold [[vulnerabilities fixed 60 days](#)]:



There MUST be no unpatched vulnerabilities of medium or high severity that have been publicly known for more than 60 days.



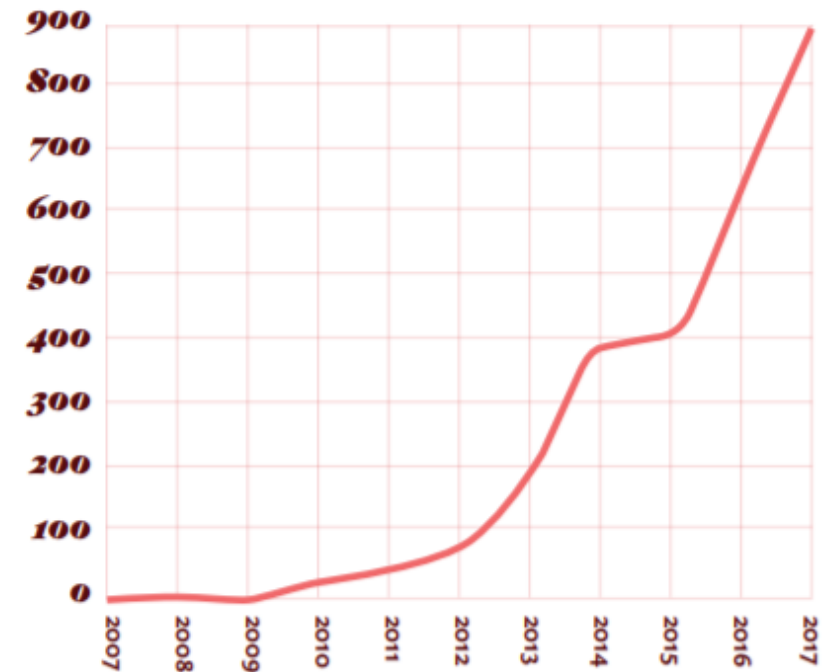
# Vulnerable Open Source Components

80% to 90% of software products on the market include OSS components

Using components with known vulnerabilities:

- Included in [OWASP Top 10](#) (2013-2017): A9
- Root cause of major data breaches
  - Mossack Fonseca (*Panama Papers*) breach
  - *Equifax* breach

## Open Source Vulnerabilities Published by Year

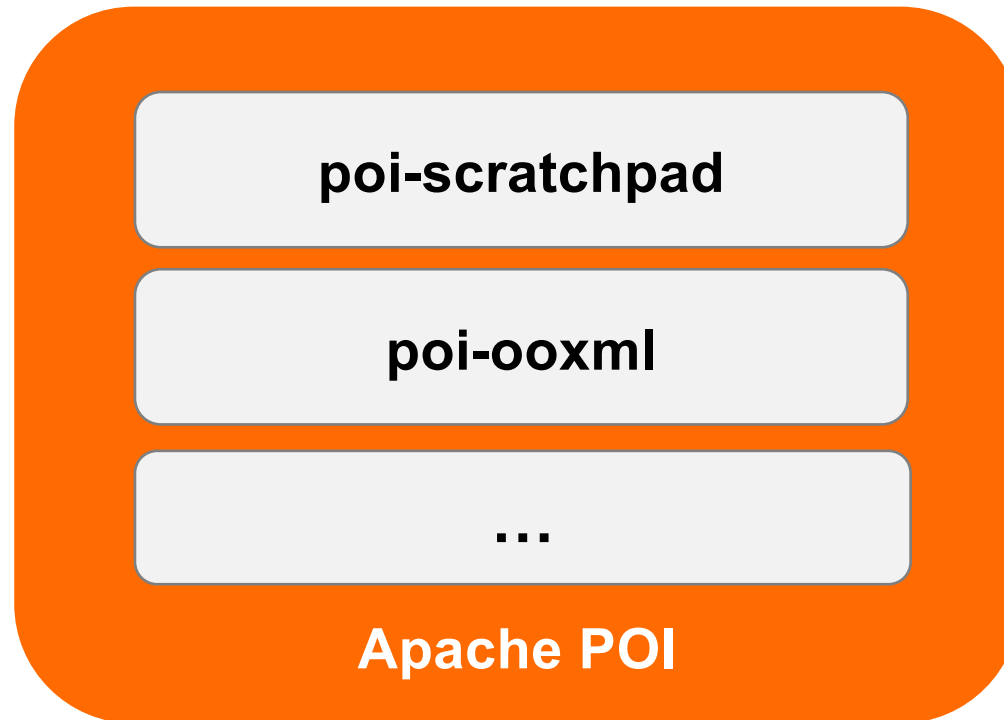


(“The State of Open Source Security”, Snyk, 2017)

# Impact Analysis is Difficult

---

Vulnerabilities are assigned to entire projects (e.g., Apache POI, Tomcat), but sub-components (e.g. Jar archives) are used separately



# Existing approaches (based on meta-data)

---

- Most tools “somehow” map finer-grained OSS components (e.g., JAR archives) to vulnerabilities using the project metadata
- **Actual code often ignored**

## Limitations:

- False-positives (e.g. multi-module projects)
- False-negatives (e.g. re-bundling)
- Focus only on detection (no app-specific analysis)

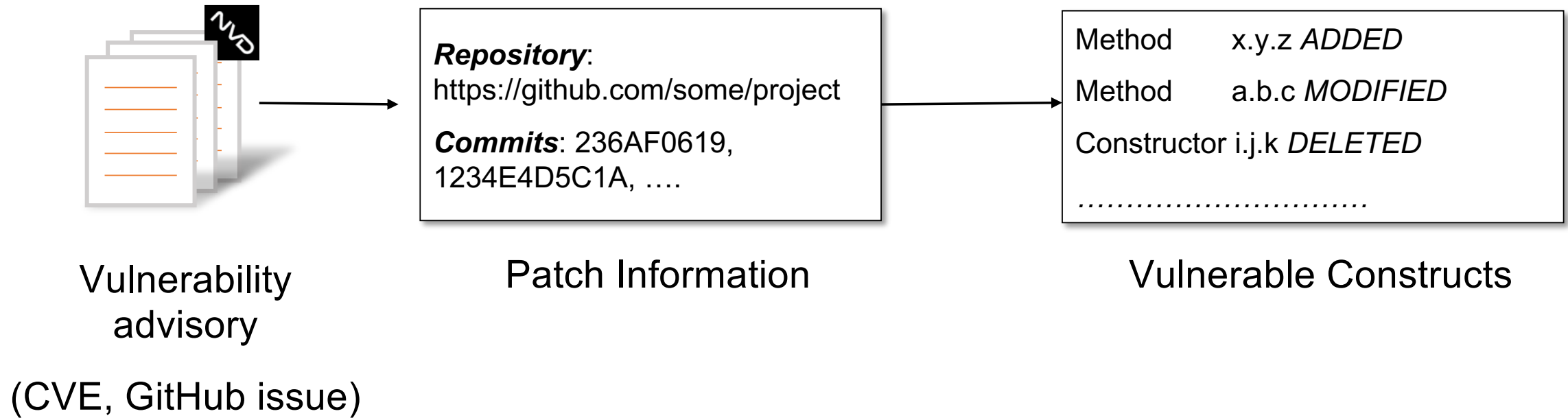
# Impact Analysis is Difficult

---

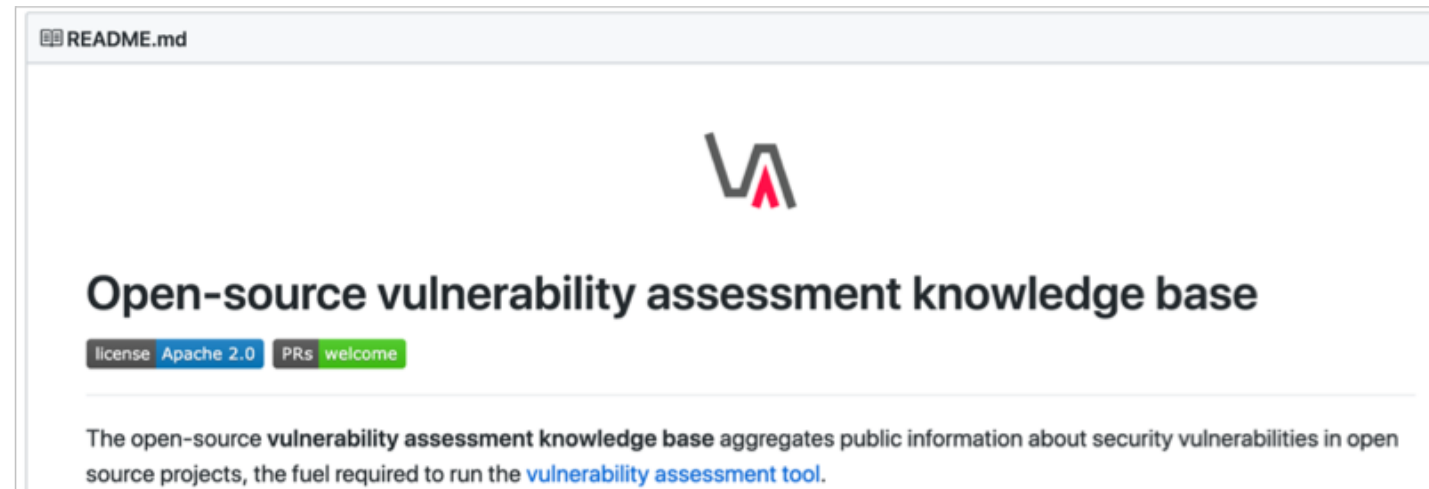
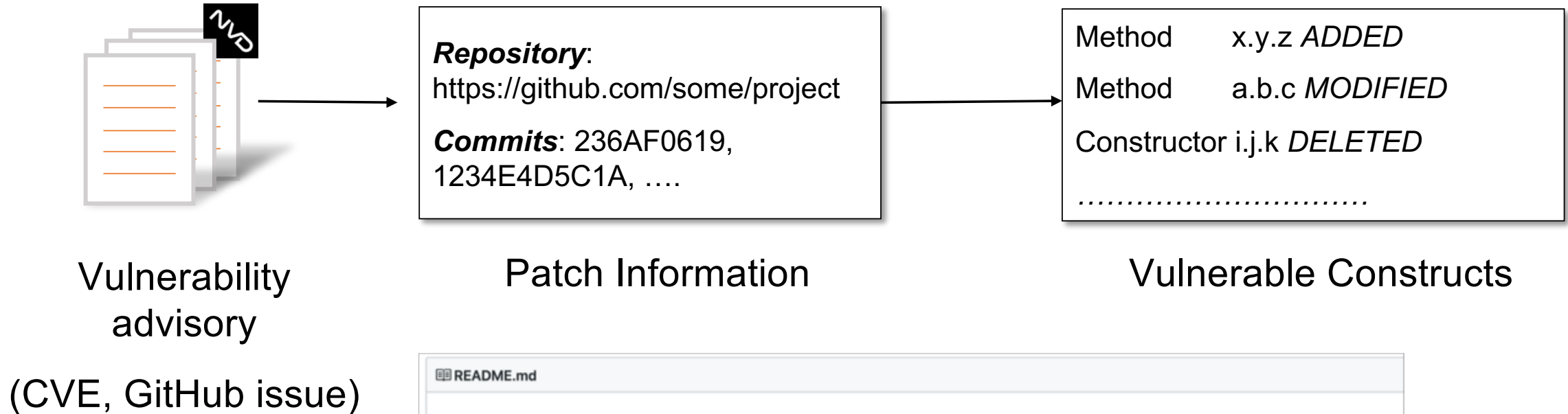
Vulnerability descriptions (in natural language) often not useful

***CVE-2012-5633:*** “The *URIMappingInterceptor* in Apache CXF before 2.5.8, 2.6.x before 2.6.5, and 2.7.x before 2.7.2, when using the *WSS4JInInterceptor*, bypasses WS-Security processing, which allows remote attackers to obtain access to SOAP services via an HTTP GET request.”

# Vulas – Vulnerable Constructs



# Vulnerable Constructs



<https://github.com/SAP/vulnerability-assessment-kb>



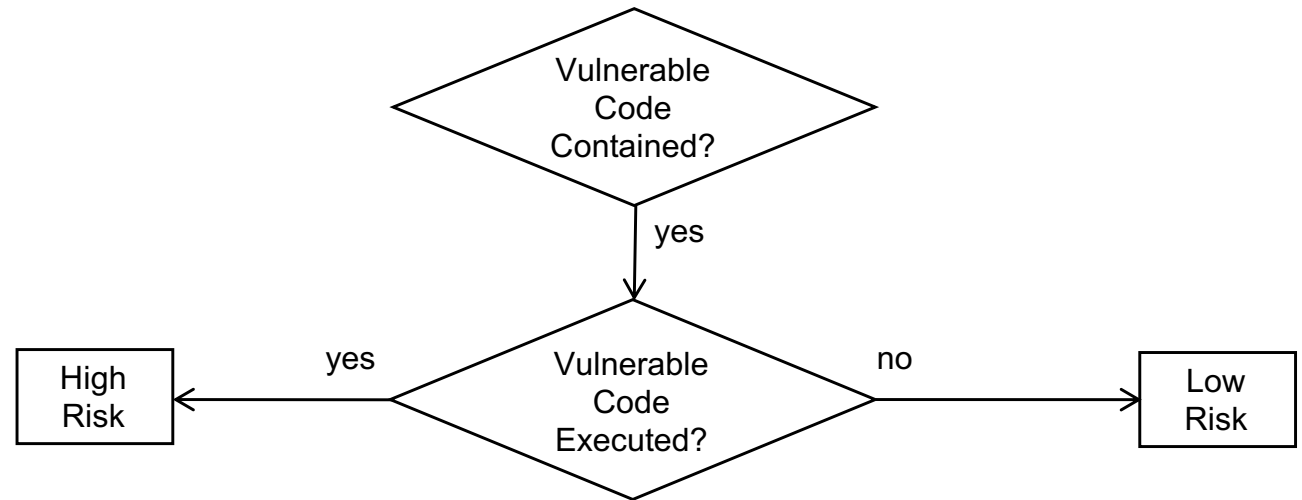
- From vulnerability to **vulnerable constructs** (actual code)
- Code-centric **detection** of known vulnerabilities
- Static and Dynamic **assessment** of vulnerable code
- Metrics to support selection of non-vulnerable libraries (**mitigation**)

Supported languages: Java, Python

# Vulas Approach

**Assumption:** If an application contains and executes vulnerable constructs, then there is a significant risk that the vulnerability can be exploited in the application context.

- Static reachability analysis
- Dynamic analysis
- Combination of static & dynamic analysis







# Vulnerability Assessment Tool Demo

# Vulas – Adoption @ SAP

---

- 1+ Mio. scans of 800+ applications
- Recommended @ SAP after comparison with existing open / commercial tools

## Enterprise-ready

- Usable in CI/CD pipelines, but also for legacy software
- Aggregated reports and audit of findings → could be exported to SPDX
- Support of CERT: Which of our apps are impacted by vulnerability X?
- Non-disclosed (internal) vulnerabilities can also be added to knowledge base

## Client-side tools

- Plugins for Maven and Gradle (Java) and setup tools (Python)
- Command Line Interface (CLI) for everything else

# Vulas is Open Source

---

## **Establish Collaboration to Reduce risks coming from usage of vulnerable OSS**

- Open source foundations/projects: Upstream fixes, contributions to vulnerability knowledge base
- Enterprises: Productivity features (reporting, usability, etc.)
- Universities: New languages, new analysis techniques

## **Links**

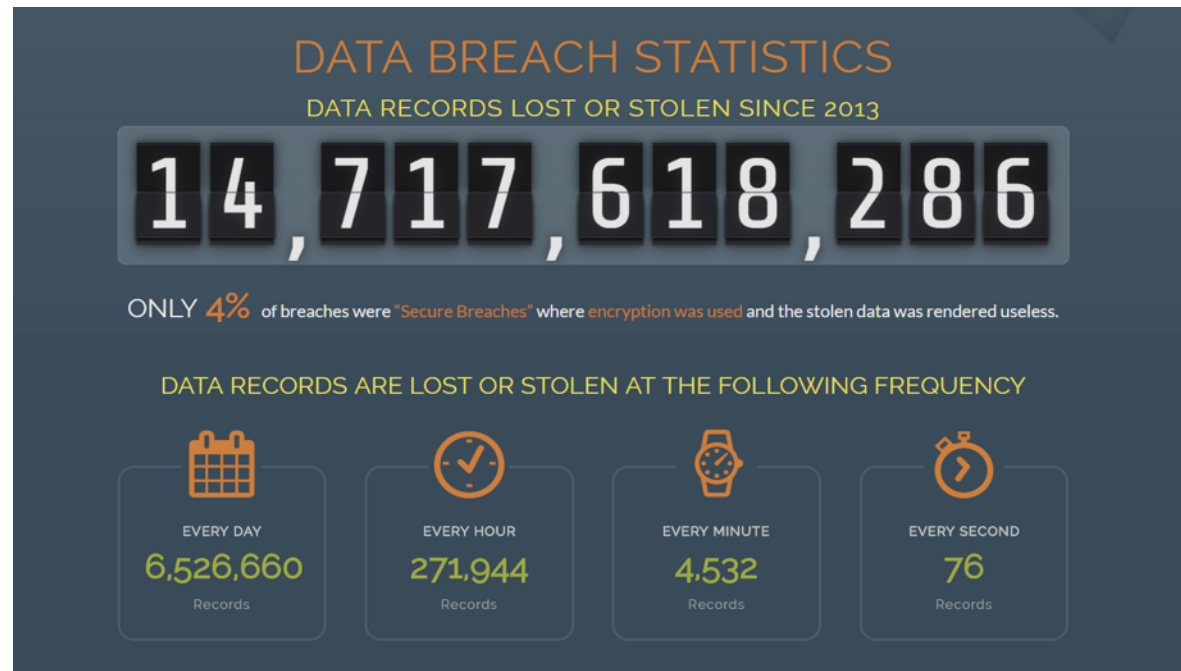
- GitHub repo: <https://github.com/sap/vulnerability-assessment-tool/>
- Documentation: <https://sap.github.io/vulnerability-assessment-tool/>
- Newsletter: [vulas-news-request@listserv.sap.com](mailto:vulas-news-request@listserv.sap.com) (“subscribe” in the body)
- Knowledge base with public vulnerabilities: <https://github.com/SAP/vulnerability-assessment-kb>

# SCCS – SAP Credential Code Scanner

CII Badge Gold [[no leaked credentials](#)]:



The public repositories MUST NOT leak a valid private credential (e.g., a working password or private key) that is intended to limit public access.



Source: <https://breachlevelindex.com/>

# SCCS – SAP Credential Code Scanner

---

**SCCS** is a standalone static source code scanner that looks for credentials and sensitive data using:

- Regular expressions
- Inference rules
- **CNN** deep learning for passwords

SCCS looks for

- Passwords, encryption keys, hashed data, tokens (cloud APIs), signatures, e-mail addresses, users / user IDs, internal domains, IP addresses

# Dashboard

Repositories

31

Scans

0

Findings

0

## Scan New Repo

Repo

Scan Options: (hold ctrl for multiple)

c\_cred.yml

default\_config.yml

model 0

model 1

model 2

Fresh Scan:

On

Submit

# Run Better Together With Open Source

---



[https://commons.wikimedia.org/wiki/File:Collaboration\\_logo\\_V2.svg](https://commons.wikimedia.org/wiki/File:Collaboration_logo_V2.svg)

CLA Assistant → cf. Github issue list

Vulas → Support for additional languages like JavaScript & Go  
→ collaboration on vulnerability assessment knowledge base

SCCS → collaboration on credential code scanners



OPEN SOURCE  
LEADERSHIP SUMMIT

# Thank you!