



THE LINUX FOUNDATION

NORTH AMERICA



From Active to Archive: Shepherding Repositories Through Their Sunset

CHA^{COSS}



DIGITAL SERVICE AT CMS



Agenda

1. Who Are We?
 - a. What is CMS?
 - b. What is CHAOSS?
2. What is the Repository Lifecycle?
3. How do we Sunset Repositories?
 - a. CHAOSS Guide
 - b. Case study: Private Sector
 - c. Case study: Public Sector
4. Future
5. Conclusions

Who Are We?



THE LINUX FOUNDATION



NORTH AMERICA

<https://github.com/DSACMS/decks/blob/main/ossna2026.pdf>



Presenters



Dawn Foster

Open Source Strategy Consultant
CHAOSS



Remy DeCausemaker

Open Source Lead
CMS OSPO



Dinne Kopelevich

Software Engineer
CMS OSPO



Natalia Luzuriaga

Software Engineer
CMS OSPO



Sachin Panayil

Software Engineer
CMS OSPO

What is CMS?



THE LINUX FOUNDATION



NORTH AMERICA

<https://github.com/DSACMS/decks/blob/main/ossna2026.pdf>



What are the Centers for Medicare and Medicaid Services? (CMS)

“CMS is the United States federal agency that provides health coverage to more than 160 million people through **Medicare**, **Medicaid**, the **Children's Health Insurance Program**, and the **Health Insurance Marketplace**. CMS works in partnership with the entire health care community to improve quality, equity and outcomes in the health care system.”



Who We Serve: The American People

67.8M

Medicare Beneficiaries
(2023)

82.3M

Medicaid Beneficiaries
(2023)

21.1M

Healthcare.gov
(2023)

\$1.76T

CMS Budget - 22% of the federal budget
(FY 2024)

\$1.05T

Total Medicare Payments
(FY 2024)

\$846B

Total Medicaid Payments
(FY 2024)

6,613

CMS Employees
(FY 2024)

1.5M

Health Care Providers
(2024)

22%

National Health Care
Spending is Medicare
(2024)

What does an Open Source Program Office (OSPO) do?

An OSPO is responsible for defining and implementing **programs, policies, and projects** that guide open source efforts.

30% of Fortune 100 companies have implemented an OSPO, with 84% of OSPOs developing or managing GenAI infrastructure.

CMS is the **first** United States Federal Agency to implement this private-sector styled approach to an Open Source Strategy.



Save us Money



Save us Time



Reduce Duplicate Work



Reduce Duplicate Costs



Reduce Security Risk



Reduce Continuity Risk



Accountability for Contract Performance



Engine for Talent

What is CHAOSS?



OPEN SOURCE SUMMIT

THE LINUX FOUNDATION



Embedded Linux
Conference

NORTH AMERICA

<https://github.com/DSACMS/decks/blob/main/ossna2026.pdf>



CHAOSS: Community Health Analytics for Open Source Software

Open Source, Linux Foundation Project

- **Working Groups:** OSPOs, Science / Research, Data Science, UN SDGs, Funding Impact, AI Alignment, DEI, Education, Package Metadata
- **Chapters:** Africa, Asia
- **Software:** GrimoireLab and CollectOSS / 8Knot

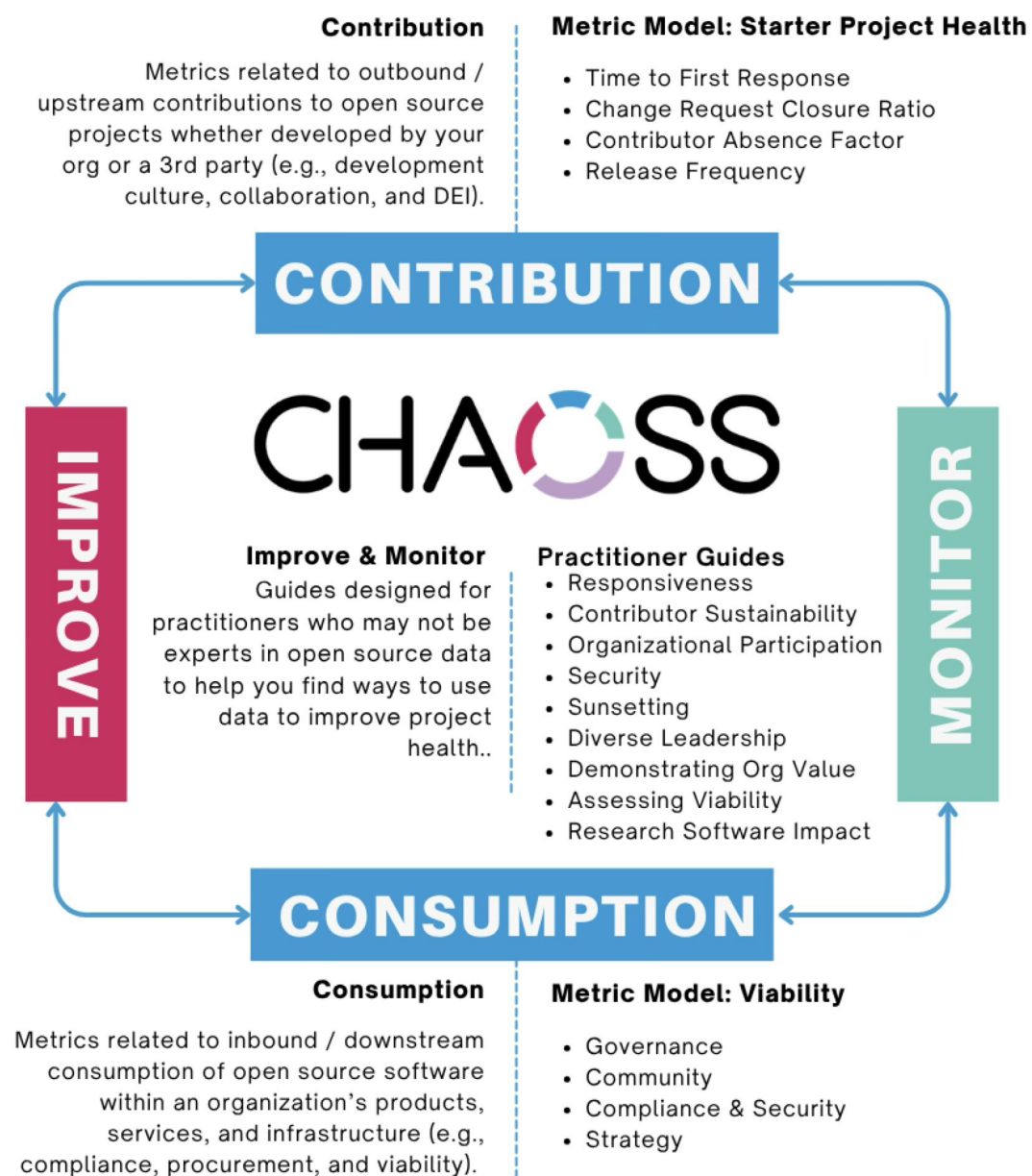
<https://chaoss.community/kb-getting-started>



CHAOSS Practitioner Guides

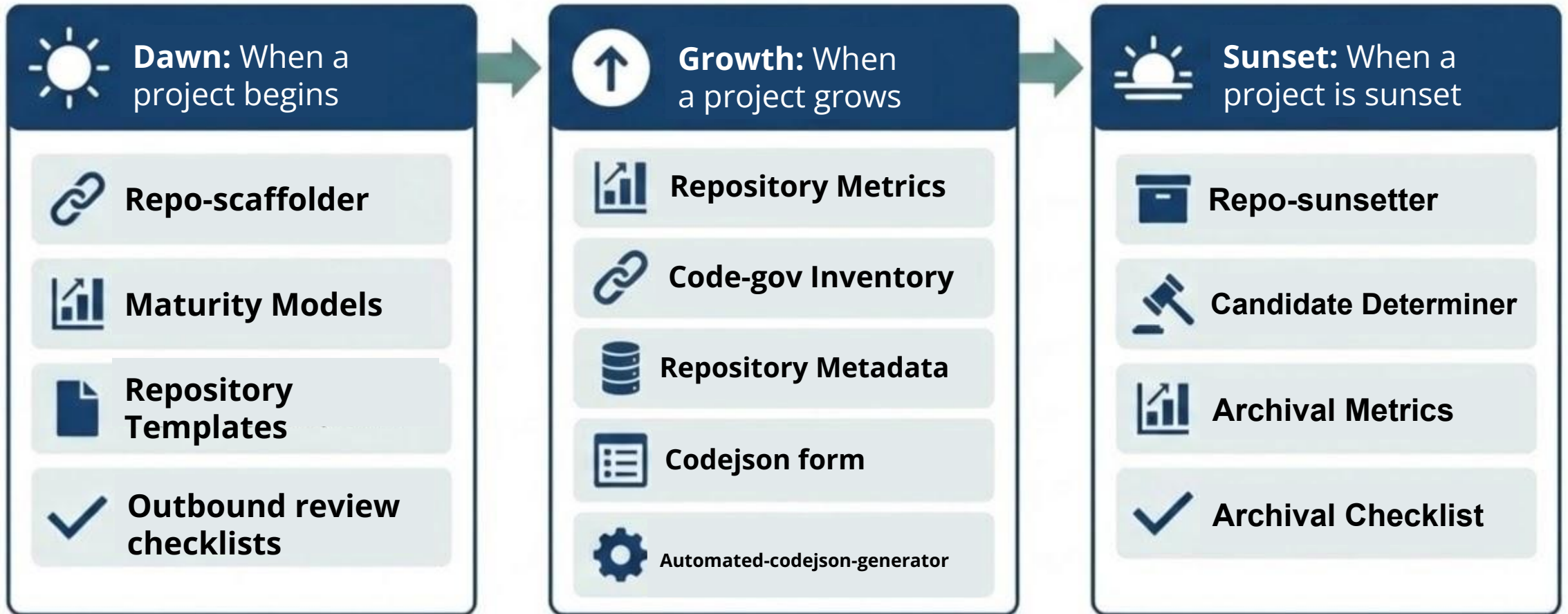
Practitioner Guides to improve your work in open source

Guides are open source (MIT licensed)



What is The Repository Lifecycle?

The Repository Lifecycle



Repo-scaffolder: <https://dsacms.github.io/repo-scaffolder>

Maturity Models: <https://dsacms.github.io/repo-scaffolder/#maturity-model-framework>

Metrics: <https://dsacms.github.io/metrics>

Code-gov: <https://dsacms.github.io/code-gov>

Repository Metadata: <https://github.com/DSACMS/gov-codejson>

Codejson-generator form: <https://dsacms.github.io/codejson-generator>

Automated-codejson-generator: <https://github.com/DSACMS/automated-codejson-generator/>

Repo-sunsetter: <https://github.com/DSACMS/repo-sunsetter>

The Repository Lifecycle: Dawn

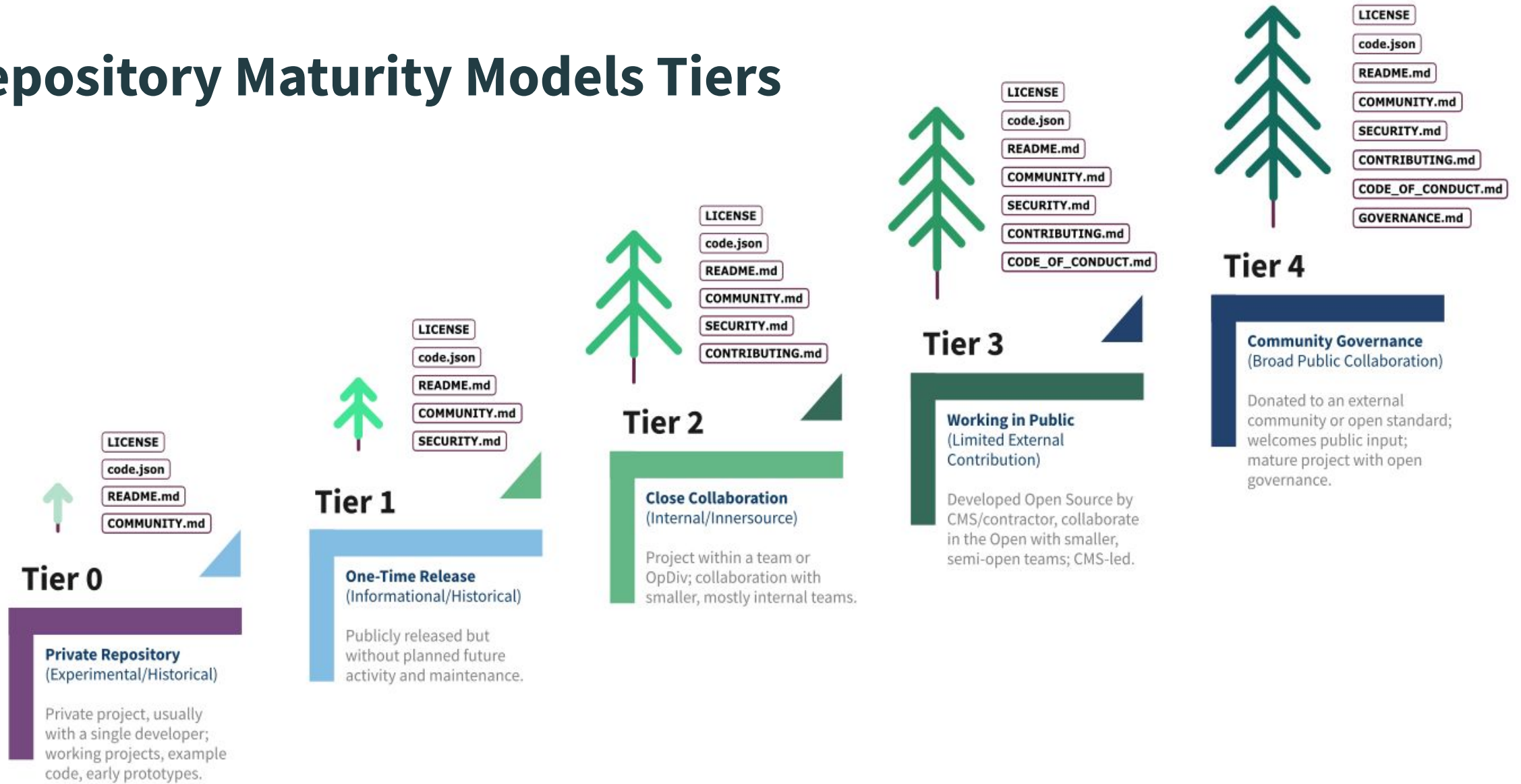
**Dawn: When a project begins**

-  **Repo-scaffolder**
-  **Maturity Models**
-  **Repository Templates**
-  **Outbound review checklists**

Q: How do we know where we are in the repository life cycle?

A: Maturity Models!

Repository Maturity Models Tiers



Repository Maturity Models





What is my project's maturity model tier?

Answer the series of questions below to determine the maturity model tier of your project:

- Does your project have more than one contributor?
- Do you plan on shipping more code or releases in the future?
- Do you plan on accepting innersource contributions to the repository? (e.g. internal staff and contractors only)
- Do you plan on accepting outside contributions to the repository? (e.g. pull requests and issues from the public)
- Do you plan on having other individuals/teams outside the agency co-planning the development roadmap with you?

The Repository Lifecycle: Growth

↑ Growth: When a project grows

-  **Repository Metrics**
-  **Code-gov Inventory**
-  **Repository Metadata**
-  **Codejson-generator form**
-  **Automated-codejson-generator**

Q: Why do repository metrics matter?

A: They help guide us down the path of growth, or the path of archival.

Repository Metrics: Reading the Signs

During repository growth, we monitor **development activity metrics** as a leading indicator of project health over a \$THRESHOLD_TIME (e.g. weeks, months, years)



Time Since Last PR Opened



Time Since Last PR Merged/Closed



Time Since Last Issue Opened



Time Since Last Issue Closed



Time Since Last Formal Release

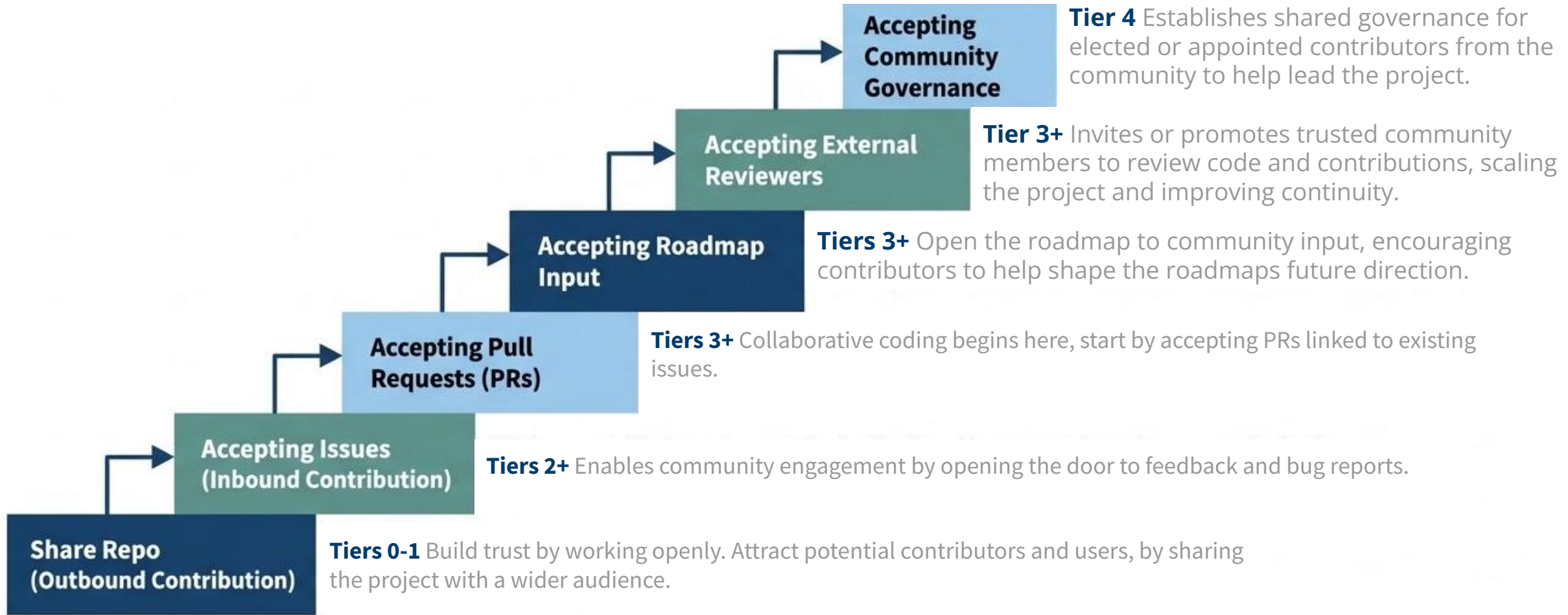


Time Since Last Direct Commit

Q: How do active projects navigate the path of growth?

A: Follow the steps on our Contribution Ladder!

Path of Growth: The Contribution Ladder



Maturity Model Ladder Guide: <https://dsacms.github.io/ospo-guide/outbound/maturity-model-ladder-guide/>

Contributor Ladder Guide: <https://dsacms.github.io/ospo-guide/outbound/contributor-ladder/>

CHAOSS Guide on Sunsetting Repositories

CHAOSS Practitioner Guide on Sunsetting Repositories

Sunset projects to indicate that they are not maintained and will not have security updates

Be responsible: do more than just click the archive button

<https://chaoss.community/practitioner-guide-sunset>

Practitioner Guide: Getting Started with Sunsetting an Open Source Project

Primary metrics:

- [Change Requests](#)
- [Issues New](#)
- [Technical Forks](#)

If you haven't already read the [Insight Guide: Introduction - Things to Think about When Interpreting Metrics](#), please pause now and read that guide.

Many open source projects, even widely used ones, become abandoned for a variety of reasons (e.g., evolving interests, family situations, employment changes), but abandonment can be done in a responsible way by proactively sunsetting the project (Miller et al. 2025). Sunsetting is an important consideration for corporate environments where it can be easy to lose track of projects that were created by employees who later walked away from the project and left if abandoned. You don't want abandoned open source projects with security vulnerabilities sitting in your organization's source code repositories where someone might trust that project simply because they trust your organization. Finding inactive projects and responsibly sunsetting them is a good business decision and something that many open source teams / Open Source Program Offices

Private Sector Example

Origin story for the guide:
VMware's OSPO, Git in
Order, and an ongoing
process for sunsetting
repositories



VMware Archive

Verified

186 followers

Palo Alto, CA

<https://vmware.com>

Popular repositories

kubeless

Public archive

Kubernetes Native Serverless Framework

Go 6.9k 742

octant

Public archive

Highly extensible platform for developers to better understand the complexity of Kubernetes clusters.

Go 6.3k 488

What is the Path of Archival?

The Repository Lifecycle: Sunset



Q: How to Determine Dormant projects?

A: Use Archival-Identifier!

Repository Metrics: Reading the Signs

Measuring repository reuse via forks serves as an indicator for project health by assessing usage and activity of its downstream community



Number of Forks

How many forks have been created?



Popularity

Stars of forks > Stars of repository



Activity within Forks

Have technical commits been pushed to forks within \$THRESHOLD_TIME?

Repository Metrics: Reading the Signs

Other indicators of project health:

OpenSSF Criticality Score

Calculates the influence and importance of a project. Score between 0 (least critical) to 1 (most critical) on metrics like **org_count** and **dependents_count**, in addition to activity

$$C_{project} = \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \frac{\log(1 + S_i)}{\log(1 + \max(S_i, T_i))}$$



Repository Contents

Is the repository empty or only contains README?



github.com/github/empty-repos

Use the empty-repos GitHub Action created by the GitHub OSPO!

Archival Candidate Identifier - Simple

2 simple status determinations to classify repositories under:

Status	Description	Open PRs	Merged/ Closed PRs	Push to repo	Push to forks	Open issues	Closed issues	Releases	Criticality Score	Is repo empty/ README only?
Active	This project is under active development and has an active user base	✓	✓	✓	✓	✓	✓	✓	✓ Above \$THRESHOLD_SCORE	Has content
Dormant	Project upstream and downstreams are inactive	✗	✗	✗	✗	✗	✗	✗	✗ Below \$THRESHOLD_SCORE	Empty OR Readme-only

release v1.0.0

A GitHub Action to identify repositories that are candidates for archival based on development activity, usage, criticality score, and repository contents.

archival-identifier

archival-identifier is a GitHub Action that ties it all together, looking into project health indicators to identify archival candidates

- Development Activity Metrics
- OpenSSF Criticality Score
- Reuse Metrics via Forks
- Repository Contents
- Status Determination

Results: 2025-11-01 to 2026-05-16

Repository	Open Issues	Closed Issues	Open PRs	Merged PRs	Closed PRs	Releases	Commits	Criticality Score	Forks	Active Forks	Is Empty/README-Only	Status
.github	0	0	0	1	0	0	2	0.29899	2	0	Has Content	Active
dsacms.github.io	0	0	0	2	0	0	4	0.27279	0	0	Has Content	Active
dedupliFHIR	0	0	0	0	0	0	0	0.31751	3	0	Has Content	Dormant
opportunities	0	0	0	0	0	0	0	0.2709	0	0	Has Content	Dormant
metrics	0	0	0	0	0	0	4	0.39515	5	0	Has Content	Active
oss-community-runbook	0	0	0	0	0	0	0	0.26152	0	0	Has Content	Dormant
cms-gource	0	0	0	0	0	0	0	0.27377	1	0	Has Content	Dormant
repo-scaffolder	0	0	0	0	0	0	0	0.40312	17	0	Has Content	Dormant
form-demo	0	0	0	0	0	0	0	0.26029	0	0	Has Content	Dormant
decks	0	0	0	0	0	0	0	0.27734	2	0	Has Content	Dormant
ospo-guide	0	0	0	41	6	0	12	0.31013	4	1	Has Content	Active
iv-cbv-payroll	0	0	5	55	7	0	21	0.45115	14	1	Has Content	Active
repodive-tools	0	0	0	0	0	0	0	0.25951	0	0	Has Content	Dormant
reverse-scorecard-generation	0	0	0	0	0	0	0	0.23655	0	0	Has Content	Dormant

Development Activity Metrics

Criticality Score

Forks

Contents

Status

Repo: <https://github.com/DSACMS/archival-identifier>

Action: <https://github.com/marketplace/actions/archival-identifier>

FUTURE: Archival Candidate Identifier - Advanced

We plan to explore 3 more status determinations, assessing development activity data and fork data more in-depth:

Status	Description	Open PRs	Merged/ Closed PRs	Push to repo	Push to forks	Open issues	Closed issues	Releases	Criticality Score
Stable	Upstream activity may be below thresholds, but downstream activity indicates active usage or forks. Usually automated commits by bots (e.g. dependabot, github-actions[bot]) and minimal/no commits by humans	✓/✗	✓/✗	✓/✗	✓	✓/✗	✓/✗	✓/✗	TBD: Consider using the standard deviation of threshold?
Dormant Downstream	Project may be under active development, but does not have an active downstream user or developer base	✓	✓	✓	✗	✓	✓	✓/✗	✗ Below \$THRESHOLD_SCORE
Dormant Upstream	Project may have active users, but the core developers or upstream team is inactive. Stars of fork > Stars of upstream repo	✗	✗	✗	✗	✓	✗	✗	✓ Above \$THRESHOLD_SCORE

Q: How do Dormant projects navigate the path of archival?

A: Follow the guidance from Repo-sunsetter!

What is Repo-sunsetter?

Repo-sunsetter is a GitHub action that prepares a repository for archival. It performs the following steps to "sunset" a repository:

1. Creates a PR adding an **archival notice** to the README.md to inform users about the state of the repository
2. Creates a PR updating project **metadata** by marking project as archived in code.json
3. Creates an issue containing an **archival checklist** based on the repository's maturity model tier. The checklist contains various tasks reviewing the contents of the repository.

Repo: <https://github.com/DSACMS/repo-sunsetter/>

Action: <https://github.com/marketplace/actions/repo-sunsetter>

repo-sunsetter

Run tests passing release v1.0.0

A GitHub Action that prepares a repository for archival by automating documentation updates and providing an archival review checklist.

README Code of conduct Contributing License Security

DSAC Open Source Website

Warning

This project is now archived and no longer actively maintained. It has been archived to retain its contents for reference. Feel free to explore and fork the repository, but please note that updates or support will not be provided.

A static webpage containing various information about CMS open source repositories. This is originally the first iteration of metrics visualizations site.

github-actions bot opened 3 days ago

CMS.gov Repository Archival Checklist

Use this checklist to ensure all necessary tasks are completed before archiving your repository.

Repository Metadata

In code.json:

- Change status to archival - "status": "Ideation",
- Add archived to tags 6 + "status": "Archival",
- Review all fields to ensure project metadata is correct and accurate.

Documentation

- Clearly state at the top the README that the project has been deprecated and will no longer be updated.

Path of Archival: Archival Notice & Documentation

Warning

This project is now archived and no longer actively maintained. It has been archived to retain its contents for reference. Feel free to explore and fork the repository, but please note that updates or support will not be provided.

```
▼ README.md
... @@ -1,4 +1,7 @@
1 1 # DSAC Open Source Website
2 +> [!WARNING]
3 +> This project is now archived and no longer actively maintained.
   It has been archived to retain its contents for reference. Feel
   free to explore and fork the repository, but please note that
```

Documentation: Add Archive Notice to README #66

Merged natalialuzuriaga merged 1 commit into main from archive-notice-update 9 minutes ago

Conversation 0 Commits 1 Checks 3 Files changed 1

github-actions bot commented 15 minutes ago

Description

Hello, and thank you for your contributions to the Federal Open Source Community. 🙌

As part of preparing the repository for archival, this PR adds an archive notice to the README to inform users that this project is no longer actively maintained.


Changes

- Added archive notice banner to README.md
- Includes clear information about repository status

README.md and COMMUNITY.md preserve history and legacy, serving as the record of the investment of time, effort, and contributions of the community.







Clearly adding a notice at the top of the README sets expectations and directs contributions toward active repos on the path of growth.

Path of Archival: Repo Metadata

Field	Presence	Source	Type	Description	Options/Examples
name	required		str	Name of the project or software	
version	optional		str	The version for this release	
description	required		str	A one or two sentence description of the software.	
status	required		str	Development status of the project	<ul style="list-style-type: none"> - Ideation - Development - Alpha - Beta - Release Candidate - Production - Archival

```

- "status": "Ideation",
6 + "status": "Archival",
  
```

Source Policy	Metadata	Origin	Icon
M-16-21 	code.json	Federal	
publiccode.yml 	publiccode.yml	International	
SHARE IT Act	code.json	Federal	
M-25-21	code.json	Federal	

code.json is the Federal Standard, codemeta.json, is an international standard, repo-sunsetter uses both!

Repository Metadata is required by the SHARE IT Act of 2024, and The Federal Source Code Policy (M-16-21)

Metadata is shared via Code.gov, which provides visibility and promotes reuse, reducing duplicate work, effort, and taxpayer costs.

Path of Archival: Checklists

CMS.gov Repository Archival Checklist

Use this checklist to ensure all necessary tasks are completed before archiving your repository.

Repository Metadata

In code.json:

- Change status to archival
- Add **archived** to **tags**
- Review all fields to ensure project metadata is correct and accurate.

Documentation

- Clearly state at the top the README that the project has been deprecated and will no longer be updated.
- Include suggestions for alternate projects that provide similar functionality (if applicable).
- Update all project documentation (wikis, websites, and other documentation)

Issues Management

- Review all open issues
 - Close all fixed issues
 - Comment on unresolved issues to provide context
 - Apply appropriate labels to issues
 - Close issues as `won't fix` where applicable

Security Review

The archival checklist provides maintainers a route along the Path of Archival

Checklists and ladders are human centered and human readable.

Checklists provide structure and guidance, reducing cognitive load and burden of effort.

Basic Archival Checklist (for Tier 0-1):

https://github.com/DSACMS/repo-sunsetter/blob/main/checklists/BASIC_ARCHIVAL_CHECKLIST.md

Comprehensive Archival Checklist (for Tiers 2-4):

https://github.com/DSACMS/repo-sunsetter/blob/main/checklists/COMPREHENSIVE_ARCHIVAL_CHECKLIST.md

Future



THE LINUX FOUNDATION

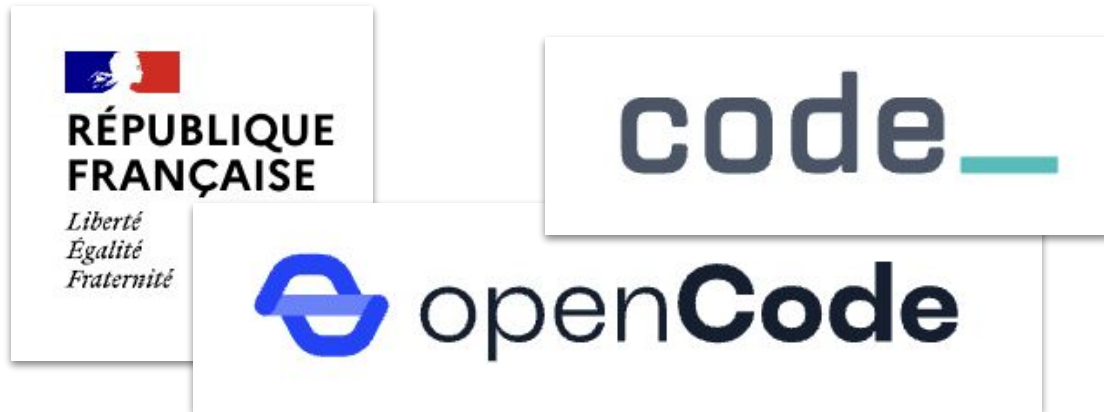


NORTH AMERICA

<https://github.com/DSACMS/decks/blob/main/ossna2026.pdf>

National OSPOS, Inventories, and Digital Sovereignty

- Germany has opencode.de
- France has code.gouv.fr
- USA has code.gov
- Other national OSPOs are growing
(e.g. Kenya, Trinidad and Tobago)



Q: How do we secure the global supply chain of software in the face of a sustainability crisis? (e.g. *log4shell, SHA1-Hulud, Mythos, copy/fail*)

A: We use open standards and open tools to harmonize the national inventory metadata.

codejson-crosswalk

The problem & the fix

Maintainers previously had to sync the same metadata across multiple files by hand. The fix is a hub-and-spoke architecture where `code.json` is the hub at the center, and each format is a spoke connected through a declarative mapping pair which all culminates in a tool called codejson-crosswalk

Under the hood

- Engine is generic so it knows nothing about any specific format
- Mappings are declarative: read field → reshape → write
- No procedural logic anywhere in the conversion path

Adding a new format

- Write a mapping pair. That's it.
- No engine changes. No orchestration code.
- Closer to filling out a translation table than writing software.

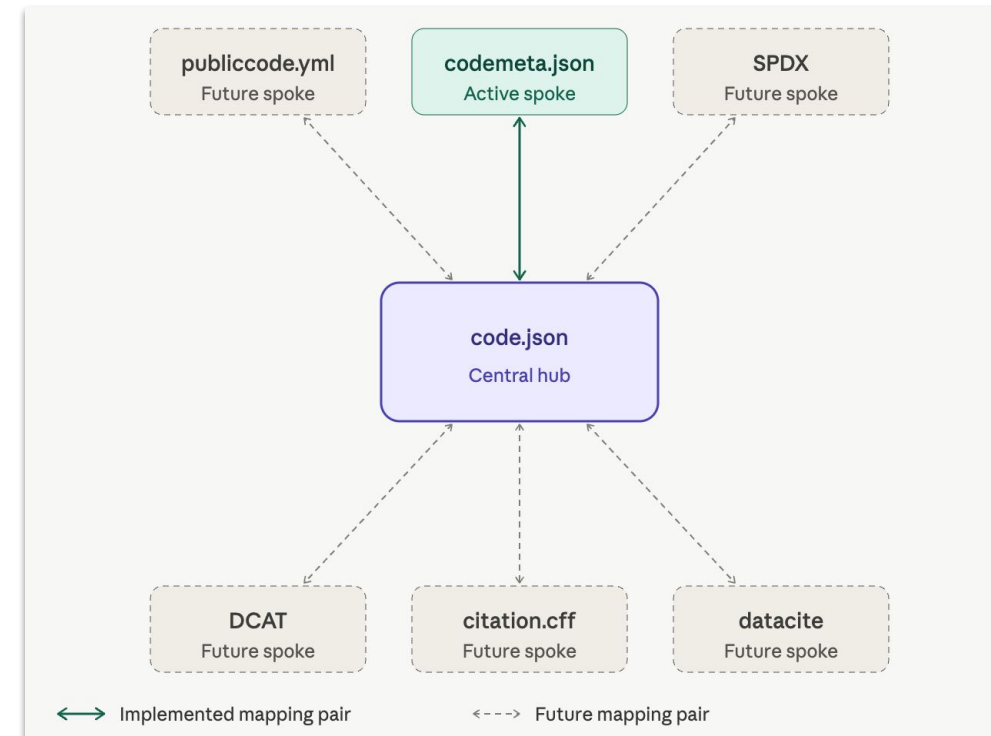
Repo: <https://github.com/DSACMS/codejson-crosswalk>

codejson-crosswalk

A TypeScript npm package for bidirectional conversion between code.json and other metadata formats

About the Project

There is no automated bridge between code.json and other software metadata formats. This package provides a single function call to convert between code.json and other formats, with code.json at the center of a hub-and-spoke architecture.



Q: How do we unite hackers on planet earth to secure the open source software supply chain globally?

A: An international grand unified permanent neutral inventory of all public sector open source code!

Software Heritage Archives

Two layers of archiving

- **Forge layer** → where development lives (GitHub, GitLab)
- **Global layer** → preservation beyond any single platform

Software Heritage = the global layer

- Partnership with UNESCO and Inria to preserve source code as part of Humanity's digital heritage
- 250M+ projects, 16B+ unique source files [Software Heritage](#)
- Just this week, 900M lines of [OIN's Linux System Table 13](#) were preserved!

codejson-crosswalk bridges the two

- [Canonical metadata](#) format: [codemeta.json](#)
- codejson-crosswalk converts [code.json](#) → [codemeta.json](#) on demand
- Federal repos could be ingested alongside every other archived project on Earth

code.json at the forge. **codemeta.json** in the archive. Same metadata. Two layers of permanence.



Software Heritage

Linux System definition
openinventionnetwork

PRESERVING THE OIN 2.0 LINUX SYSTEM

Software Heritage

Archived for long-term security and transparency across 5,100+ packages.

Preserved packages
Packages from Table 13 (for 2.0 Participants) preserved at [Software Heritage](#).

Package Name	Project URL	SWHID (archive)
389-admin	http://www.port389.org/	Software Heritage
389-admin-console	http://www.port389.org/	Software Heritage

Q: How can you get involved in these international efforts?

A: Join United Nations Open Source Week, June 22-26th!

<https://www.unopensource.org>

Conclusions



Open tools and open standards can harmonize metadata across nations into a unified global inventory.

Machine-readable, interoperable metadata is the shared language that agents, machines, and humans all depend on.



Projects are not valuable solely based upon the utility of their results or outputs, they reflect the record of our progress.

Archives provide transparency, accountability, and attribution. Archives build trust, reduce duplicate work, and reduce risk.

Archives allows historians and practitioners to more accurately and completely understand the story of open source.



Repo archival is often clouded by stigma and negativity, but project inactivity isn't a failure, it's part of the life cycle.

Reading the signs and planning before it becomes a crisis, makes sunsetting easier when the time comes.



**archival-identifier and repo-sunsetter
complete the repository lifecycle that
the CMS OSPO first launched at Open
Source Summit North America 2024.**



Thank You OSSNA!

CHA^{OS}SS



DIGITAL SERVICE AT CMS

CMS Questions or Comments?

CMS OSPO Landing Page

<https://go.cms.gov/ospo>

CMS Open Source Questions?

opensource@cms.hhs.gov

CodingItForward Summer Internship Program!

<https://codingitforward.com>

Join the [CHAOS.Community!](#)

CHAOS Practitioner Guides:

<https://chaoss.community/about-chaoss-practitioner-guides/>

CHAOS.community Slack:

<https://join.slack.com/t/chaoss-workspace>

CHAOS OSPO Metrics Working Group

[#wg-ospo](#) in CHAOS.community Slack



THE LINUX FOUNDATION



NORTH AMERICA

<https://github.com/DSACMS/decks/blob/main/ossna2026.pdf>

