# ClearlyDefined Roadmap

## SUMMARY

On February 1st, 2026, the Open Source Initiative (OSI) and the AboutCode Foundation (AboutCode) signed an agreement to delegate the operations of the OSI's ClearlyDefined to AboutCode to ensure the long-term sustainability and continued development, maintenance, and availability of ClearlyDefined's code and data.[1]

AboutCode is an international non-profit organization that develops, maintains, and sustains open and interoperable tools, curated data, and practical standards for managing and securing software supply chains. AboutCode leadership includes founding members of ClearlyDefined and SPDX, core contributors to CycloneDX, and creators of Package-URL (PURL), ScanCode, and OSS Review Toolkit (ORT). With recognized expertise in open source code origin, license, and vulnerability management, AboutCode is uniquely positioned to drive the long-term success of ClearlyDefined.

This benefits the ClearlyDefined community and the larger open source ecosystem. OSI secures the project's long term sustainability. AboutCode furthers its commitment to quality open metadata about origin, license, security and project health by supporting the expanded access to ClearlyDefined's data, evolving its architecture, and enhancing its features to deliver more accurate data faster and more efficiently. This will also lower the operational and hosting costs for sponsors.

This document is a proposed, draft roadmap for ClearlyDefined to address ClearlyDefined's technical debt, reduce sponsors' infrastructure costs, and expand ClearlyDefined's functionality, community, and sponsors. This roadmap presents the path to improve ClearlyDefined data quality and availability, user experience, and community adoption to solidify the project's long-term sustainability.

## CONTEXT

ClearlyDefined is a centralized and curated database for open source software origin and licenses, and a crucial community resource for software supply chain management, SBOM enrichment, and open source compliance.

ClearlyDefined was first established at OSI by Microsoft as a collaboration among 12 organizations, using contributed infrastructure from AWS, Google, and Azure and funding from Bloomberg and Microsoft. The original vision was to start with open source licenses, and then expand to other use cases, including vulnerability and project health.

Several original organizations left the project due to various reasons, leaving Microsoft (and GitHub) as the largest code and data curation contributing organization, main financier, and sole infrastructure provider, with ongoing and important contributions from SAP, Kusari, the Eclipse Foundation, and Bloomberg.

## PROBLEM

Started in late 2017, the ClearlyDefined tech stack has grown old and accumulated technical debt, making it harder to attract community contributions. Its architecture limits data accessibility, and requires significant computing and hosting costs. For reference, the ClearlyDefined data files consist of about 70TB of JSON definitions and harvest documents (e.g., ScanCode scans); access is limited to a throttled API, limited by the blob-based storage architecture.

"Data harvesting" is the name used for ClearlyDefined's crawler operations that download and scan open source packages with ScanCode and other tools. This is the most compute-intensive and costly part of operating ClearlyDefined. Microsoft (and GitHub) have funded all the core infrastructure, the bulk of curations, DevOps resources, and have provided ongoing funding for the project and can no longer provide ongoing funding to the same historical level.

[1] https://opensource.org/blog/ensuring-the-long-term-sustainability-of-clearlydefined-osi-and-aboutcode-sign-mou

# ClearlyDefined Roadmap

AboutCode

## SOLUTION

The proposed solution is a three year plan with specific yearly milestones and deliverables to achieve the targeted outcome of lowering the overall cost of operations, while increasing the reach, efficiency and accessibility of ClearlyDefined data and code.

The key points are to:

- Combine ClearlyDefined and AboutCode operations and backends when possible to enable scan data reuse and avoid duplicated, wasted compute resources across ecosystems, creating mutually beneficial synergies
- Refactor ClearlyDefined backend and frontend to reduce the technical debts and improve data quality
- Enroll new sponsors to share the cloud compute costs
- Unlock data access for increased community adoption
- Enable new use cases around security and project health

By sharing funding and resources across its members and community of adopters, we can provide a more sustainable, cost-effective future for ClearlyDefined.

## THEMES

The three-year plan focuses work on key improvements each year:

- 2026: Reduce technical debt and operating costs, and improve the harvesting and curation queue and UI to facilitate community contributions
- 2027: Migrate to a federated data architecture to further reduce hosting costs and unlock data access. Initiate work on new use cases, vulnerabilities and project health and risks
- 2028: Deploy security and project health use cases

There is also substantial, ongoing work to maintain the code, data, and infrastructure and support the community included as part of this roadmap.

The deliverables are organized around these objectives:

### 1) Create better license and package scans, faster and more efficiently

We plan to improve the tooling and processes to eliminate redundant and wasteful rescanning of the same packages multiple times for license and other software metadata. This includes:

- Enable reusing scans across ClearlyDefined harvests and AboutCode data collections
- Design a priority mechanism to scan packages based on actual usage
- Improve the performance of ScanCode for faster and more cost efficient scans
- Improve ScanCode license detection based from the analysis of ClearlyDefined curated data
- Adopt federated data design with data mirrored across multiple hosts to improve service availability and reduce costs, reusing existing scans

### 2) Reduce technical debt

The goal is to improve the code, data structure, and infrastructure maintainability and reduce the technical debt to improve the user experience for community adoption. This includes:

- Update dependencies to latest versions
- Refactor storage of scans and attachments
- Switch to the open source DocumentDB from the proprietary MongoDB

- Maintain and improve the ClearlyDefined code base, CI/CD, and infrastructure
- Update project documentation to reflect new and ongoing development
- Facilitate outreach and onboarding for new contributors and sponsors

### 3) Shared scans for shared costs

Distributing the workload of scans with the wider ClearlyDefined community will reduce the resource costs on existing providers. This includes:

- Expand the pool of organizations hosting scans to share the scanning costs
- Design new prioritized scan queue to scan packages in an order that aligns with business needs
- Integrate distributed scanning with federated data access with on-premises, private data usage

This also includes direct collaboration with open source foundations, and large projects such as Linux distributions to help them scan and share scans of their upstream and downstream packages.

### 4) Integrate PURL and SBOM for easier interoperability

PURL is an official Ecma standard (ECMA-427) for package identification in the software supply chain, and undergoing ISO standardization. The standard was created and is maintained by the AboutCode team.

ClearlyDefined uses another approach for package identification called "coordinates" that predates PURL. Most ClearlyDefined adopters use a PURL converter to ClearlyDefined coordinates, to enable easier integration in software supply chains automation workflows, such as SBOM and vulnerability management.

To position ClearlyDefined as the provider of correct data for software supply chain management, it is critical to implement PURLs in ClearlyDefined and expose ClearlyDefined's package metadata in SBOM formats (SPDX and CycloneDX) to directly support workflow automation and SBOM enrichment. This includes:

- Create new API endpoints to access ClearlyDefined data keyed by PURL
- Create a new SBOM API endpoint to expose reference package data in SBOM formats
- Adopt PURL as a core identifier in the database and file store

### 5) Improve curation processes and user experience

There are significant UX issues in the current data curation UI, and this discourages community data curation contributions. It is also difficult to make batch contributions, or exchange curations with other tools and projects including AboutCode, ORT, and OSSelot. This includes:

- Develop and deploy a new and improved UI for efficient data curation
- Make it easier for organization to contribute their curations back in bulk
- Improve the data models for curation storage
- Create a new specification for multi-stakeholder data curations exchange

### 6) Support for more use cases

The original vision for ClearlyDefined was not limited to package origin and license metadata. We will expand the use cases supported by ClearlyDefined data and tools, including:

- Security with vulnerability data
- Data for project health and lifecycle events and analytics
- Usage to find the most used open source components and drive scan priorities

## PLANNING

The planning items to consider are:

- Development personnel costs to maintain and upgrade the code,
- DevOps personnel costs to operate the services, and
- Outreach personnel costs to promote the project and manage the community.

The work for this roadmap is organized in discrete, focused projects. Each project is further broken down in tasks and deliverables provided in the Tasks and Deliverables section. This will directly support key ClearlyDefined and AboutCode maintainers and core contributors for DevOps, development, data management, project management, documentation and community outreach tasks.

Below is the budget summary for estimated FTE days of the projects in this roadmap. The Tasks and Deliverables section of this roadmap covers each project in more detail.

| Project | Description | Timeline | Estimated FTE days |
|---|---|---|---|
| **clearlydefined-scan-once** | Build tooling and processes to reuse existing scans and avoid rescanning to reduce resource costs | 2026-2028 | 300 |
| **clearlydefined-3.0** | Refactor ClearlyDefined to reduce technical debt and accelerate future development | 2026-2028 | 470 |
| **clearlydefined-distributed-harvest** | Distribute scan workloads to share resource costs across organizations hosting harvesters | 2026-2028 | 220 |
| **clearlydefined-purl** | Create a new PURL-based API to improve data interoperability with other tools | 2026-2028 | 160 |
| **scancode-plus-plus** | Improve ScanCode performance and quality for more efficient ClearlyDefined operations and curation | 2026-2028 | 250 |
| **clearlydefined-curate-next** | Build a new data curation UI, models, and exchange standard for more community contributions | 2026-2027 | 180 |
| **clearlydefined-new-use-cases** | Expand use cases, including security and project health, for more comprehensive data and wider community adoption | 2027-2028 | 330 |
| | **Yearly total for 2026** | | **652** |
| | **Yearly total for 2027** | | **728** |
| | **Yearly total for 2028** | | **555** |
| | **Total** | | **1,935** |

## CONTACT

For more information the roadmap for ClearlyDefined, please send an email with the subject "ClearlyDefined roadmap" to:

Philippe Ombredanne, Lead Maintainer of AboutCode Foundation – pombredanne@aboutcode.org

Adam Herzog, Community Manager of AboutCode – adam@aboutcode.org

# ClearlyDefined Roadmap

## TASKS AND DELIVERABLES

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-scan-once** | **2026-2028** | **Build tooling and processes to reuse existing scans and avoid rescanning to reduce resource costs** | | |
| | 2026 | Harmonize ScanCode options between ClearlyDefined, ScanCode.io, and PurlDB | Updated code and data storage in each project. From that point on, each project can directly consume scans produced by the other projects | 40 |
| | 2026 | Create JavaScript TypeScript library for PURL federated hash ID computation | Released and documented library | 10 |
| | 2026 | Reuse AboutCode federated data for harvest metadata | Updated harvest code to access AboutCode federated data and fetch existing package metadata for a PURL | 30 |
| | 2026 | Reuse AboutCode federated data for ScanCode scans | Updated harvest code to access AboutCode federated data and fetch existing ScanCode scans for a PURL | 40 |
| | 2027 | Push new harvested scans to AboutCode data federation | Updated harvest code to push scans to AboutCode data federations | 30 |
| | 2027 | Push other harvests to AboutCode data federation | Updated harvest code to push other tools output to AboutCode data federations, in particular raw upstream registry API call results | 30 |
| | 2027 | Store, raw unmodified other harvested data | Updated harvest code to always keep raw metadata from API calls, such as npmjs JSON or Maven pom.xml | 30 |
| | 2026 | Update historical CD data - Batch 1 | Code to update and backfill existing data. Run and monitor updates | 15 |
| | 2027 | Update historical CD data - Batch 2 | Code to update and backfill existing data. Run and monitor updates | 15 |
| | 2028 | Update historical CD data - Batch 3 | Code to update and backfill existing data. Run and monitor updates | 15 |
| | 2026 | Deploy existing data to data federation - Batch 1 | Code to batch push to data federation. Run and monitor push | 15 |
| | 2027 | Deploy existing data to data federation - Batch 2 | Code to batch push to data federation. Run and monitor push | 15 |
| | 2028 | Deploy existing data to data federation - Batch 3 | Code to batch push to data federation. Run and monitor push | 15 |

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-3.0** | **2026-2028** | **Refactor ClearlyDefined to reduce technical debt and accelerate future development** | | |

# ClearlyDefined Roadmap

AboutCode

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| | 2026 | Switch to DocumentDB for local deployment | Updated code, tested and documented, replacing the proprietary MongoDB with the open source DocumentDB, tested in CI. | 30 |
| | 2026 | Update infrastructure and stack - Batch 1 | Updated code using the current versions of libraries. Validated dependencies ensure removal of outdated and unmaintained libraries. Established and documented process for vulnerability management | 15 |
| | 2027 | Update infrastructure and stack - Batch 2 | Updated code using the current versions of libraries. Validated dependencies ensure removal of outdated and unmaintained libraries. Established and documented process for vulnerability management | 15 |
| | 2027 | Update and complete documentation - Batch 1 | Updated and improved documentation including schemas for various API and data formats, guides, and tutorials for usage and installation | 15 |
| | 2028 | Update and complete documentation - Batch 2 | Updated and improved documentation including schemas for various API and data formats, guides, and tutorials for usage and installation | 15 |
| | 2026 | Keep license texts and notices from analyzed projects | Updated and documented code to collect, store, and use the accurate license texts and notices from ScanCode scans | 30 |
| | 2026 | Create SBOM API endpoint - CycloneDX | Updated and documented API endpoint for CycloneDX SBOMs for a package version | 20 |
| | 2027 | Create SBOM API endpoint - SPDX | Updated and documented API endpoint for SPDX SBOMs for a package version | 30 |
| | 2026 | Ongoing maintenance of code, data, and infrastructure | Updated dependencies, CI/CD, data files organization, data grooming, and infrastructure DevOps | 50 |
| | 2027 | Ongoing maintenance of code, data, and infrastructure | Updated dependencies, CI/CD, data files organization, data grooming, and infrastructure DevOps | 50 |
| | 2028 | Ongoing maintenance of code, data, and infrastructure | Updated dependencies, CI/CD, data files organization, data grooming, and infrastructure DevOps | 50 |
| | 2026 | Ongoing community management, and new adopter onboarding | Updated project documentation. Online presence and in-person events. Outreach for integration and adoption in other FOSS projects. Enrollment of new contributors and corporate sponsors. | 50 |
| | 2027 | Ongoing community management, and new adopter onboarding | Updated project documentation. Online presence and in-person events. Outreach for integration and adoption in other FOSS projects. Enrollment of new contributors and corporate sponsors. | 50 |
| | 2028 | Ongoing community management and new adopter onboarding | Updated project documentation. Online presence and in-person events. Outreach for integration and adoption in other FOSS projects. Enrollment of new contributors and corporate sponsors. | 50 |

# ClearlyDefined Roadmap

AboutCode

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-distributed-harvest** | **2026-2028** | **Distribute scan workloads to share resource costs across organizations hosting harvesters** | | |
| | 2026 | Enable self-interested distributed harvesting | Container image and model where users can host a harvester on their own infrastructure. Code to ensure local harvest is used as a priority for own components harvested locally, and share the scans back to ClearlyDefined | 50 |
| | 2026 | Create harvester/scan guide | Documentation to host a community scanner/harvester for deployment and usage | 20 |
| | 2027 | Improve Harvester onboarding guide | Updated documentation for harvesters onboarding | 30 |
| | 2026 | Outreach to enroll new harvesters | 2 new harvesting organizations onboarded and participating in the shared network, over 12 to 24 months | 20 |
| | 2027 | Outreach to enroll new harvesters | 5 new harvesting organizations onboarded and participating in the shared network, over 12 to 24 months | 50 |
| | 2028 | Outreach to enroll new harvesters | 5 new harvesting organizations onboarded and participating in the shared network, over 12 to 24 months | 50 |

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-purl** | **2026-2028** | **Create a new PURL-based API to improve data interoperability with other tools** | | |
| | 2026 | Create reference JavaScript/TypeScript library to convert PURL to CD Coordinates, bidirectional | Released and documented library | 20 |
| | 2026 | Create API endpoint in CD to access definition and harvests keyed by PURL | Code with new API endpoint deployed and documented for definitions | 30 |
| | 2027 | Design and details plan for code and data migration to use PURL internally | Design planning document to switch to use PURL all the way. To be implemented in 2027 | 30 |
| | 2027 | Create and deploy extended PURL-based API endpoints in CD beyond definition and harvests | New PURL-based API endpoint deployed and documented for definitions | 30 |
| | 2027 | Migrate code to use PURL throughout (provisional) | Code migrated to PURL - Batch 1 - Deployed in production | 30 |
| | 2028 | Migrate code to use PURL throughout (provisional) | Code migrated to PURL - Batch 2 - Deployed in production | 20 |

# ClearlyDefined Roadmap

AboutCode

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **scancode-plus-plus** | **2026-2028** | **Improve ScanCode performance and quality for more efficient ClearDefined operations and curation** | | |
| | 2026 | ScanCode performance: improve scan speed, detection quality and accuracy. Speed up curations | Improved scan speed with benchmarks. Plan for analysis of all curations. Add new licenses detected or reported by the community. | 50 |
| | 2027 | ScanCode performance: improve scan speed, detection quality and accuracy. Speed up curations | Improved scan speed with benchmarks. Analysis of curations to updated license and copyright detections. Add new licenses detected or reported by the community. | 70 |
| | 2027 | Port ScanCode to ARM | ScanCode toolkit ported and released for ARM on Linux, Windows and MacOS | 60 |
| | 2028 | ScanCode performance: improve scan speed, detection quality and accuracy. Speed up curations | Improved scan speed with benchmarks. Analysis of curations to updated license and copyright detections. Add new licenses detected or reported by the community. | 70 |

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-curate-next** | **2026-2027** | **Build a new data curation UI, models, and exchange standard for more community contributions** | | |
| | 2026 | Design new schema for exchanging multi-stakeholder curations | Release and documented JSON schema for data curations of FOSS package code metadata (origin, classification, license, and other metadata) | 30 |
| | 2026 | Make curation data format spec an Ecma standard | Create and anchor a new Ecma TC54 task group for a curation data exchange standard. Create spec. Validate and review with the community. | 22 |
| | 2027 | Make curation data format spec an Ecma standard | Create and anchor a new Ecma TC54 task group for a curation data exchange standard. Create spec. Validate and review with the community. Present standard to Ecma plenary. | 23 |
| | 2026 | Design and implement process and model to support for multi-stakeholder curations backend | Release documented backend and updated internal to support multi-stakeholder data curations | 20 |
| | 2027 | Design and implement new process and model to support for multi-stakeholder curations backend | Release documented backend and updated internal to support multi-stakeholder data curations | 20 |
| | 2026 | Design and implement UI/UX for batch ingestion of trusted curations | Released and documented new app for batch data curation. Deployed app for operational usage | 20 |
| | 2027 | Design and implement UI/UX for batch ingestion of trusted curations | Released and documented new app for batch data curation. Deployed app for operational usage | 20 |
| | 2026 | Design and implement new UI/UX for multi-stakeholder data curations | Released and documented UI/UX app for data curation (such as origin, or classification, or license or other metadata). Deployed app for operational usage | 25 |

# ClearlyDefined Roadmap

AboutCode

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-curate-next** | **2026-2027** | **Build a new data curation UI, models, and exchange standard for more community contributions** | | |
| | 2027 | Design and implement new UI/UX for multi-stakeholder data curations | Released and documented UI/UX app for data curation (such as origin, or classification, or license or other metadata). Deployed app for operational usage | 25 |

| Project | Timeline | Task | Deliverable | FTE days |
|---|---|---|---|---|
| **clearlydefined-new-use-cases** | **2027-2028** | **Expand use cases, including security and project health, for more comprehensive data and wider community adoption** | | |
| | 2027 | Design and implement ClearlySecured, to integrate known security vulnerability data | Initial design and prototype for looking up package vulnerabilities by PURL | 20 |
| | 2028 | Design and implement ClearlySecured, to integrate known security vulnerability data | Deployed new data and apps integrating security vulnerabilities with ClearlyDefined data | 60 |
| | 2027 | Design and implement ClearlyMaintained, to collect project health and lifecycle events to assess project health and maintenance | Initial design and prototype | 20 |
| | 2028 | Design and implement ClearlyMaintained, to collect project health and lifecycle events to assess project health and maintenance | Deployed new data and apps integrating project health and lifecycle events with ClearlyDefined data | 100 |
| | 2027 | Design and implement ClearlyUsed, to find and focus on most used open source | Initial design and prototype | 20 |
| | 2028 | Design and implement ClearlyUsed, to find and focus on most used open source | Deployed new data and apps integrating usage metrics with ClearlyDefined data | 60 |
| | 2028 | Create VEX API endpoint | Updated and documented API endpoint for CycloneDX and CSAF VEX for a package version | 50 |