Executive summary

There is a large growing ecosystem of free and open source software (OSS), but very few efforts to identify key projects in the ecosystem that are most in need of support. Such an effort is an important step to ensure security of the OSS ecosystem; an insecure OSS ecosystem poses a national security risk as demonstrated by the log4shell incident and described in a Plaintext report. Currently, policymakers and funders who want to support OSS do not have the right data to enable them to most efficiently target their efforts and funds.

This proposal establishes Ecosyste.ms, which presents open datasets and APIs that perform dependency mapping (i.e., identifying key software systems where open source software is critical) of the OSS ecosystem to determine which projects are most critical and most in need of support. Ecosyste.ms builds on the proposers' previous experience developing an open source search engine, but focuses on publishing data and providing APIs as infrastructure to allow others to build research and tools in this area. Ecosyste.ms provides a foundational basis for researchers to better analyze OSS and for funders to better prioritize which projects most need to be funded.

Proposal overview

This proposal fills the need for:

- *A comprehensive, structured, and open dataset* about free and open source software, its usage and authorship,
- *a set of tools and services* to resolve software dependency information quickly and to reason about its state,
- *a support structure* for those who are looking to work with and build upon these services.

This proposal addresses the market failure associated with OSS, characterized by many as "the tragedy of the commons" in which a common-pool resource (the time required of maintainers to maintain highly depended-upon software) is overused by consumers, but no one feels responsibility for addressing. This issue has been well documented, not least by Nadia Eghbal's *Roads and Bridges* study for The Ford Foundation.

The net negative of this market failure is that today, much of the digital infrastructure that we rely on is underfunded, undersupported and is increasingly called out as a significant national security risk. A first step to understanding this problem is to catalog which OSS software is most

widely used and in need of support, as identified in the Plaintext Group report *Securing Open Source Software at the Source* and the White House Open Source Software Security Summit.

Proposal specifics

Ecosyste.ms pulls data from a variety of data sources, including package managers such as pip and npm, GitHub repositories, and container ecosystems. Ecosyste.ms stores this data in such a way that is easy to reason about the usage, authorship and potential impact on the ecosystem of projects and people that depend upon it — not dissimilar from the way that Google's own PageRank enabled Google to reason about the state and value of individual pages and domains on the internet. Ecosyste.ms will provide open data releases for the research community as well as API access to its data. Although the open data and regular user access to Ecosyste.ms will be free and open, API access may be tiered and fees for higher rate limits may provide a way to keep the system financially sustainable.

Ecosyste.ms will be built atop a growing library of components that will provide some of the fundamentals necessary to build its understanding, such as parsing license information and resolving software dependency trees. Ecosyste.ms will principally comprise 1) a well-documented set of services and APIs for developers and researchers to build upon and 2) a series of openly licensed datasets, regularly released for those who wish to use data en masse in their research. In addition, the proposal provides for a period of proactive research and technical support to encourage use and support early adopters.

A comprehensive, structured, and open dataset about free and open source software, including its usage and authorship, is not widely available at the moment, and Ecosyste.ms fulfills that need.

Existing projects that come closest to implementing this approach include **Libraries.io** and **Deps.dev.** Ecosyste.ms distinguishes itself from its predecessors by:

1. a service layer-first approach by primarily focusing on providing APIs and open data releases for other people to build tools that use it
2. adding additional data sources to provide insight into OSS ecosystems that have not been analyzed before, such as operating system-level package managers.

Impact

By the end of 12 months after the start of the project, we expect to see, as a result of this proposal, substantial progress on:

- *Collecting data to inform and guide funders and policymakers* on which OSS packages are most critical and in need of support, helping them more efficiently target their money.
    - [Measured by partnerships] – establishing partnerships with at least three other organizations or institutions seeking to publish further research and/or build solutions to support free and open source software

- *More research and development of tools* seeking to understand, support and mitigate the systemic risks concerning critical open source software.
    - [Measured by integrators] – at least three websites / tools should use the Ecosyste.ms API
- *To demonstrate the economic sustainability of Ecosyste.ms service*
    - [Measured by use of freemium model] - at least 10 users should use the paid Ecosyste.ms API

The intended outputs for this project are, by the end of the 12-month period:

- To launch Ecosyste.ms as described in this proposal
- To develop a catalog of new API services to support the development of tools in this space, including (but not limited to):
    - parsing an arbitrary dependency tree and return a complete transitive dependency tree
    - a standardized interface for querying package metadata
    - a dependency discovery service to list known projects that depend on a given software package
    - A contributor metadata service to querying contributor information across various data sources
    - A CVE parsing service to detect if a CVE alert affects a codebase
    - Archive services to detect inconsistencies between package revisions and content
- To automate the publications of regular open data releases.
- To provide free access to API endpoints at a rate that is amenable to small-scale experiments and developing prototypical services, and a mechanism to increase the rate limit on a paid-for basis.
- To publish a series of exemplar experiments and proactively publish research on the state of the open source ecosystem.
- To expand the scope of data available to include containers and system package managers

Team

The team working on this project consists of Andrew Nesbitt and Ben Nickolls. Both have deep knowledge of the problem domain, having developed multiple applications and services in the space. They are especially passionate about moving Ecosyste.ms forward.

As part of Open Source Collective, Ben regularly works with funders who want to support OSS projects they depend on. Open Source Collective and Open Collective Inc. have direct interests in making Ecosyste.ms successful as it supports their business alongside others working in this space.

Andrew holds a BEng in Robotics and Automated Systems from Uni. Plymouth. He has since spent 15 years working in industry as a software developer, most recently at Protocol Labs

developing distributed version control and package management software on IPFS. Andrew will act as head of engineering on the project.

Ben holds a BSc in Computer Science from Uni. Liverpool and a MSc. in Computer Security from Uni. Birmingham. Ben is Executive Director of The Open Source Collective, a non-profit organization that is working to create a more sustainable future for open source software. Open Source Collective will act as the fiscal sponsor for the project for the duration of the grant and will support the project thereafter.

Budget

Plaintext Group will fund 8 months of active development and 12 months of hosting and support. Open Source Collective will cover the costs of a data scientist and support staff to work with integrators and researchers and/or publish exemplars.

Roadmap

The following high-level roadmap outlines the major milestones for the project, for a more detailed plan see our public roadmap.

Q1 - We assume a start date in early 2022, beginning six months of principal development

Milestones completed (end of period)

- Design, development, testing, deployment and documentation of services (part 1)
    - Dependency parsing services
    - Package, repository, and contributor metadata services
    - Archive services
- Begin migrating and synchronizing open data sources for above services

Q2 — Three months from the start date we will be half way through principal development

Milestones completed (end of period):

- Continued development of services (part 2)
    - License parsing service
    - CVE parsing, and impact service
    - Dependency resolution service
- Complete synchronizing data sources for above services
- Automate and publish open data releases

Q3 — Six months from the start date the grantee will complete principal development and begin marketing and providing support to researchers and users of the services.

Milestones completed (end of period):

- Proactive publication of at least 2 new experiments
- Expand scope of data to include one system-level package manager

- Optional work to add billing functionality to cover ongoing cost of maintaining services

Q4 — Nine months from the start date we will have completed development, focus shifts entirely to marketing and providing support.

Milestones completed (end of period):

- Publish at least one significant experiment or research paper
- Establish partnerships with at least three other organizations or institutions working in digital infrastructure, open source sustainability, or supply chain security.

Year complete

At twelve months the project is complete. Open Source Collective will become the project's guardian, covering any necessary expenses to operate the services and provide support to the community either directly or in partnership with other funders.

Additional Information

Research based on original software inter-dependency data:

- Census Program II by Linux Foundation and Harvard Laboratory for Innovation Science (LISH) – https://www.coreinfrastructure.org/programs/census-program-ii/
- Carlson B., Leach K., Marinov D., Nagappan M., Prakash A. (2019) Open Source Vulnerability Notification. In: Bordeleau F., Sillitti A., Meirelles P., Lenarduzzi V. (eds) Open Source Systems. OSS 2019. IFIP Advances in Information and Communication Technology, vol 556. Springer, Cham. https://doi.org/10.1007/978-3-030-20883-7_2
- A. Zerouali, T. Mens, G. Robles and J. M. Gonzalez-Barahona, "On the Diversity of Software Package Popularity Metrics: An Empirical Study of npm," 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2019, pp. 589-593, doi: 10.1109/SANER.2019.8667997.
- Decan, A., Mens, T. & Grosjean, P. An empirical comparison of dependency network evolution in seven software packaging ecosystems. Empir Software Eng 24, 381–416 (2019). https://doi.org/10.1007/s10664-017-9589-y
- Niemeyer, Kyle E., Arfon M. Smith, and Daniel S. Katz. "The challenge and promise of software citation for credit, identification, discovery, and reuse." Journal of Data and Information Quality (JDIQ) 7.4 (2016): 1-5.
- da Costa-Luis, Casper O. "tqdm: A fast, extensible progress meter for python and cli." Journal of Open Source Software 4.37 (2019): 1277.

Breakdown of services to be developed:

- **License Parsing Service:** Parse open source license information from source code. Existing modules that can be leveraged: https://github.com/jslicense or https://github.com/licensee/licensee
- **Dependency Resolution Service:** Resolve the full transitive dependency tree for a software package or single manifest file. Existing modules that can be leveraged: https://github.com/dependabot/dependabot-core
- **Repository Metadata Service:** Standardized interface for querying metadata across various source code repository hosts. Modules: https://github.com/octokit
- **Dependency Discovery Service:** Search for other software packages that depend on a package across a variety of registries and platforms.
- **Package Metadata Service:** Standardized interface for querying metadata across various package registries
- **Dependency Parsing Service:** Existing modules that can be leveraged: https://github.com/snyk/snyk
- **Contributor Metadata Service:** Standardized interface for querying contributor data across various repositoriesExisting modules that can be leverag https://github.com/libgit2/rugged
- **Package Building Service:** Test the reproducibility and security of packages by attempting to rebuild them from source and comparing with published artifacts

- **Archive Diffing Service:** View difference between two package artifacts programmatically to spot inconsistencies between releases. Existing modules that can be leveraged: https://diffoscope.org/
- **Archive Digest Service:** Calculate cryptographic digests of package artifacts for comparison between data sources. Existing modules that can be leveraged: https://github.com/npm/ssri
- **Archive Inspection Service:** Programmatically list files and read individual files from an zip, tar or git archive, useful for browsing source code as well as selecting relevant other parsers and services to be run against the archive. Existing modules that can be leveraged: https://github.com/coderaiser/node-inly
- **CVE Parsing Service:** Detect if a cve alert affects a code base. Existing modules that can be leveraged: https://github.com/google/osv