

Enarx project CCC yearly review 2021

General Information

1.1. Name of Project

- Enarx

1.2. Project Description (what it does, why it is valuable, origin and history)

- Enarx aims to make it simple to deploy workloads to a variety of different TEEs in the cloud, on your premises or elsewhere, and to allow you to have confidence that your application workload is as secure as possible.
- The problem we're trying to address is that there are many sensitive workloads that you shouldn't entrust to a public cloud to run, or may even have concerns about running on on-premises systems. TEEs (see below) provide a great opportunity to help secure these workloads, but they're not easy to use. Enarx aims to make it simple to deploy workloads to a variety of different TEE technologies in the cloud, on your premises or elsewhere, and to ensure that your application workload is as secure as possible.
- Enarx came out of some conversations between Nathaniel McCallum and Mike Bursell (the project's founders) and led to them presenting the idea to a number of people at Red Hat, who agreed to provide some initial resources to see if it was plausible. When the Consortium was announced, we realised very quickly that it was very well aligned. Enarx was accepted as one of the initial projects in the Consortium.
- Enarx is proceeding well in terms of code, with end-to-end execution achieved on 2021-01-11 on SEV. SGX execution is believed to be close.

1.3. How does this project align with the Consortium's Mission Statement

- Enarx aims to be hardware-vendor neutral, providing a run-time environment which works across platforms. We hope to make it easy for people to deploy their own applications or applications created by 3rd party vendors, supporting the “protecting data in use” part of the charter which is core to the Consortium. Being conceived as an open source project, we welcome input and collaboration from all parties. Minimal Trusted Compute Base and high auditability are high on our list of design principles. The code is very modular, meaning that there may well be opportunities for other projects to re-use components for their needs.
- Since its acceptance into the CCC, Enarx has continued growing in maturity and continued to make its code, documentation, build processes and project

management open to all. We have released a number of Rust crates, including a ground-up CBOR encoding/decoding module named ciborium (<https://crates.io/crates/ciborium>).

1.4. Project website URL

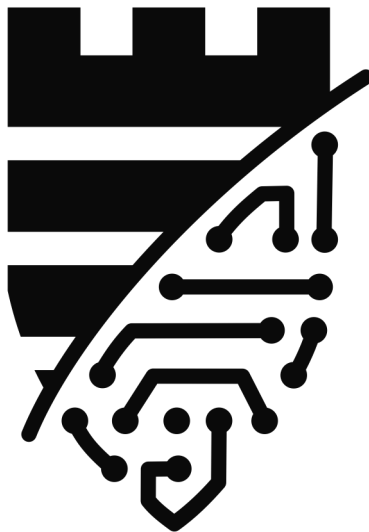
- <https://enarx.dev> (we initially used the .io domain, but changed to .dev in mid-2020 once concerns about the use of the domain surfaced in the media)

1.5. Social media accounts

- Twitter: @enarxproject
- LinkedIn: <https://www.linkedin.com/company/enarx>
- YouTube: <https://www.youtube.com/channel/UCteWrNEMJfpbBUEKaQdNoBA>
- RocketChat: <https://chat.enarx.dev/>

Legal Information

2.1. Project Logo URL or attachment (Vector Graphic: SVG, EPS)



2.2. Project license

We recommend an OSI-approved license, so if the license is not one on the list, explain why.

- Apache 2.0
- All dependencies are checked at each pull request for compatibility with Apache 2.0
- Enarx enforces DCO (Developer Certificate of Origin) for all pull requests

2.3. Existing financial sponsorship

- The Enarx team thanks all those who have contributed to the project from its inception.
- Red Hat was the initial sponsor of the project, but has decided to cease providing engineering resources from 2021-02-01. Other options are being investigated.
- Equinix provide CI/CD hardware
- RocketChat provide a chat server instance

2.4. Trademark status

- No trademark

2.5. Proposed Technical Charter

- This is under legal review (progress was interrupted by illness), with involvement from officers of the Consortium.

Technical Information

3.1. High level assessment of project synergy with existing projects under the CCC

Including how the project compliments/overlaps with existing projects, and potential ways to harmonize over time. Responses may be included both inline and/or in accompanying documentation.

- A number of the “higher-level” projects being accepted into the CCC have expressed an interest in using Enarx as lower-level infrastructure.
- Enarx shares its SEV attestation code.
- Enarx team members are working with other projects in the Consortium on stand-alone SGX attestation service.

3.2. Describe the Trusted Computing Base (TCB) of the project.

If the project supports multiple environments please describe each TCB. Also identify if dependencies of other project (both CCC or non-CCC) TCBs are taken.

- One of the key design principles for Enarx is auditability and small code size.
- Enarx is written exclusively in Rust with small amounts of x86 Assembly code.
- Rust is completely Open Source

- The TCB (from the point of view of the tenant/workload owner) for Enarx comprises the following:
 - Client agent – performs attestation of Keeps, delivers encrypted workloads to them. Must execute in an environment which is trusted by the tenant/workload owner.
 - Attestation measurement database – set of measurements of Keeps
 - TEE hardware – cryptographically signed by hardware vendor
 - TEE firmware – cryptographically signed by hardware vendor
 - Keep runtime – provided by Enarx. Includes wasmtime (Rust implementation of WebAssembly WASI – <https://github.com/bytecodealliance/wasmtime>), a micro-kernel/shim (provided by Enarx), host-side attestation code, application loading code and various other run-time components. Currently under 10Mb compiled.

3.3. Project Code of Conduct URL

We recommend a Contributor Covenant v2.0 based Code of Conduct, so if the Code of Conduct is not based on that, explain why.

- https://github.com/enarx/.github/blob/master/CODE_OF_CONDUCT.md

3.4. Source control URL

- <https://github.com/enarx> (multiple repositories)

3.5. Issue tracker URL

- Enarx uses GitHub tracking across the various project repositories.
- Planning (<https://github.com/orgs/enarx/projects/3>) and Sprint (<https://github.com/orgs/enarx/projects/2>) boards, which are project-wide, are available.

3.6. External dependencies

Including licenses, and indicate whether each is a build time or runtime dependency.

- See section 3.2 on TCB
- All Enarx code is under the Apache 2.0 license
- Rust is dual-licensed under the Apache 2.0 and MIT license: <https://www.rust-lang.org/policies/licenses>

- openssl

3.7. Standards implemented by the project, if any.

Include links to any such standards.

- HTTPS
- TLS
- WebAssembly
- Enarx supports WASI-compliant runtimes. WASI is currently undergoing standardisation: <https://github.com/WebAssembly/WASI/blob/master/Charter.md>

3.8. Release methodology and mechanics

- There have been no releases of Enarx to date.
- No methodology has yet been officially adopted for releases.

3.9. Names of initial committers, if different from those submitting proposal

- Enarx does not have a list of committers as such. Nathaniel McCallum and Mike Bursell provide steering for the project, with consent of the other contributors. Other notable contributors include Harald Hoyer, Connor Kuehl, Lily Sturmann, Mark Bestavros, Daiki Ueno, Adam Baxter and axel simon.

3.10. List of project's official communication channels

- Rocketchat: <https://chat.enarx.dev>
- GitHub
- An email list is being created for package-related announcements, etc.

3.11. Project Security Response Policy

- This is currently a draft (“proposed”) document:
<https://github.com/enarx/rfcs/tree/master/00002-vulnerability-disclosure-and-embargo-policy>

3.12. Preferred maturity level (Sandbox, Incubation, Graduation, or Emeritus)

- As Enarx has not yet made any formal releases, we believe that the Sandbox maturity level is most appropriate.

3.13. Any additional information the TAC and Board should take into consideration when reviewing your proposal

- The Enarx team would like to thank the TAC and Board for their support of the project and for their work in the broader community.
- As noted in section 2.3, Red Hat plans to stop providing engineering resources to the Enarx project from 2021-02-01: investigations are under way to try to secure funding from other sources. In the meantime, we intend to continue running the project and encouraging contributions where available.