Open Enterprise Agent

Maintainer and Contributor Call

Tuesday 12-Dec-2023

Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

All Hyperledger community members must adhere to the Hyperledger Code of Conduct

Agenda

1. Introduction
2. Community update
3. Mediation
4. Architecture Decision Records
5. Security
6. Q&A

Community update

- Last week we officially announced the Open Enterprise Agent to the community
- Today is our last maintainer and contributor call of 2023. Will resume on the 9th of Jan 2024
- Cardano Identity Wallet launched
- Community and Ecosystem Identity Education

Mediation

- DIDComm V2 Mediator will be moved to OEA Lab in Q1 2024 (including interoperability test-suite)
- Now supports protocols HTTP(S) and WS(S)(websockets)
- WS(websockets)
  - Message is delivered over websocket connection
  - Pickup protocol with live delivery mode enabled
  - No Polling required by clients (e.g. web wallet or mobile wallet) using mediator
- Alpha environment (https://sit-mediator.atalaprism.io/)
- Feedback: please raise issues or reach out on discord

Architecture Decision Records

- Open: 0
- Draft: 6
- Accepted that needs to be superseded: 1
- Missing: *?*

Suggested Actions:

1. Confirm what active development should have ADRs - work on opening these
2. Rebrand and update text in ADR templates/site
3. Ensure current state is correct
4. Work on process for keeping them up to date

Security - Principles

Principles

- **It should be hard to build insecure solutions that create harm for the people using or subjected to them**
  - *Practice a secure software development life cycle (SSDLC) so that our systems are secure by design*
  - *Be part of making sure the ecosystem is secure [feedback on protocols and other services]*
  - *Ensure our systems for building are protected from abuse (so that the code stays secure)*
- **Ensure we can accept contributions with a low barrier to entry**
  - *Use automation and tooling to help contributors, both as they are developing and within build pipelines*

Security - Potential Controls

- Create a Security Policy
  - Document how to report vulnerabilities
  - Document what audits will take place and how we publish the findings
    - **Question**: If audits are organisation sponsored - how do we manage interactions? How do we communicate results? How do we set up scope?
- Automation Security
  - Run the full pipeline only when a maintainer comments/reviews (Limit normal pipeline)
- Application Security
  - Code
    - Use Static Application Security Testing (SAST) for Quality and Security
    - Adopt Secure Software development lifecycle (SSDLC)
    - Continue to have external code audits
    - Undergo Dynamic Application Security Testing (DAST)
  - Features
    - Driven by community based on value - for example the existing integration to Keycloak and Vault
- Ecosystem Security
  - Make sure we feedback issues and improvements that we see from integrating/implementing and/or when we do performance and interop testing

Security - Next Steps

Step 1 - Write a security policy and add to repo (SECURITY.md)

- Reporting Vulnerabilities
  - **Decision Required - Manual Process (e.g. Discord/Email) or Github Private Security Advisory (PSA)**
  - If manual - document email/communication channel and advice on how to report + template
  - If PSA - set up github, set up template and run through table-top with maintainers
- Security Notification Process
  - How we publish information the vulnerability, it's remediation etc - usually a specific high-urgency channel for security related topics
- Security Principles
  - Include our principles from the previous slide last in the document

Security - Next Steps

Step 2 - Secure our automation pipeline

- Limit what the default build pipeline does [if not a maintainer]
- Set up a full pipeline to trigger if maintainer approves

Security - Next Steps

Step 3 - Get contributors thinking about security

- Simple checklist in PR or release notes
- Buddy up with a maintainer if there is a security implication

Security - Next Steps

Step 4 - Attach Snyk to Repository

- Touch base with Community Architects to sense check and approve
- A set of maintainers take ownership of testing features of Snyk
  - Review IDE tools and build integrations
    - (repeat for SAST, Software Composition Analysis (SCA), Container Scanning, Infrastructure as Code Scanning)
  - ***All configured in way to provide auxiliary information, not to block***
- Report back after [*Insert appropriate time period here*] to maintainers and propose full adoption plan

Security - Future

- Continue to iterate on best practices for a Secure Software Development Life Cycle
- Implement additional controls as and when we feel they are necessary

Q&A

Any questions?