

2023-02-16 Infrastructure Working Group Agenda and Meeting Notes

February 16, 2023

16 Feb 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 Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.



Attendees:

- Sean Bohan (openIDL)
- Nathan Southern (openIDL)
- Yanko Zhelyazkov (Senofi, Co-Chair)
- Peter Antley (AAIS)
- Joseph Nibert (AAIS)
- Faheem Zakaria (Hanover)
- Jeff Braswell (openIDL)
- Aashish shrestha (Chainyard)
- Tsvetan Georgiev (Senofi)
- Allen Thompson (Hanover)

Agenda Items:

1. Antitrust Policy
2. Status Updates
3. IWG Near Term Agenda
 - a. GitHub
 - i. Policies
 - ii. Prerequisite for the Merge
 - b. Maintainer Policies
 - i. Maintainer Role
 1. Technical Leadership at a code level
 2. Help onboard new contributors/maintainers
 3. Review/Comment/Approve PRs
 4. Escalate to AWG or TSC
 - ii. Maintainer Selection
 1. 2 Maintainers per Infrastructure Partner
 2. 1 Maintainer per Member (Carrier)
 3. 2 Maintainer openIDL Org
 4. 2 Maintainer AAIS

- iii. [Fabric Maintainer Policy as an example](#)
- c. Merge (ND POC, Testnet, etc.)
 - i. Inventory (outline changes/improvements, what can use, feature/bug fix)
 - ii. Model: Network vs. Application Communities
- d. Developer Certificate of Origin (all of openIDL and new contributions)
- e. License Compliance (LF has automated tools)
- f. Security
- g. Documentation
 - i. Currently ReadTheDocs
- h. Bug Tracking/Issues
 - i. GitHub Issues, Jira?
- i. CI/CD
- j. Release Management

Minutes:

Maintainers

- Yanko
- Dont have to come up with exact numbers
- Currently have 3-4 orgs - openIDL, AAIS, Chainyard, Carriers
- how do we assign particular role
- in terms of approving changes to codebase, question - who should approve to get a PR in main codebase
- carriers?
- do we need IPs responsible for technical execution of change - well tested, standards, practices
- approval process should be more complicated than technically adept
- understand business perspective on changes
- would feature be used, significant, mainstream feature, plug and play
- Ken
- Quorum for commits - no red tape, no surprises either
- in the beginning, make sure all have a chance to look at it before approved, controversial then escalate
- Aashish
- initially take a look, when we have enough people to contribute, set up processing votes
- Insurance agencies as stakeholder? wouldn't rule it out, but don't tend to have big IT departments, shouldn't close that off
- Yanko
- how do we approve changes, not so much pull requests
- need to set a good process in terms of reviewing issues and tickets
- make sure that changes are approved before we reach
- Jeff
- distinction - changes and updates in master
- Commits to master branch
- KS
- trust a couple reviewers to make judgement
- Jeff - what will be evolving baseline, if it impacts chaincode for network, would want to include reviewers from other areas
- Aashish - testnet ref implementation for future projects?
- Jeff
- using baseline for ND, revolutions made, goal making the ND project to be part of the mainnet
- some things done in testnet to make configs simpler, didn't impact function
- testnet is not baseline, but we need to get to the baseline
- diff repos in github, who has the rights to update master in repos?
- get to point for what becomes the master
- Aashish -
- merge is important, what is the process of what goes into master
- JB
- merge is fundamental, diff sections working on diff things, diff app level things
- straightforward
- diff things with chaincode - fundamental
- merge - inventory of where things are, one-time exercise very important
- Yanko
- did take the ND POC branch as base for testnet
- can discuss on changes next time we meet
- focus on how we can document those changes
- goal for this meeting - how to get there
- how we can inventory or document those changes before we make pull requests
- not using old way fabric was deployed, figure out if we merge network changes to point where we can intro new fabric operator or wait
- JB -
- idea not to force anything on current ND work, eye towards that app reside on main openIDL, use newer tools
- testnet mockup
- KS
- some of the basic ways we set up nodes and orgs is different using operator?
- YZ -
- not incompatible, from infra as code pov - still in place, setup Kub clusters, AWS resources, secrets etc
- what is changing is how we deploy HLF network itself - cert auth, peers, config, new channels, config channels
- 2 things after that - have the ansible playbooks available (all fab specific ops) good for automation/mass updates
- there is the FABRIC OPERATOR CONSOLE - UI to do ops on fab network (deploy chaincode, channels, etc.)
- could merge networks on operator, may have dedicated networks for diff projects, another thing is deployment of chaincode itself, using Kub (dedicated plugin in fab to deploy chaincode as bots rather than let peer control them)
- change discussing - BAF - not touching that

- Aashish
- on testnet, team is more geared towards IBP was working?
- YZ - yes
- now open source thats the approach
- KS - not need BAF in the future?
- YZ - dated, doesn't exist (it is now HL Bevel)
- KS - tear out and use operator instead?
- YZ - security related issues which can be fixed
- Aashish - dependent on operator and operator console?
- TG - go hand in hand with Fabric, rather than maintain in openIDL would rather use whats in community already
- Aashish - would these decisions be part of what goes into master?
- Yanko - yes
- Operator Consolve vs BAF, vs Bevel
- Yz - started working on this, community didnt exist, weren't communicating well enoguh in advance
- resolve by being more transparent, what features, where in play
- propose - use github to create issues and get those approved ahead of time, knowing this might be a feature to change things
- same with bug fixes, minor improvements, issue and bringing it up with this group make it easier for all to understand
- part of meeting should be addressing/lookiung over new issues
- JB - suggestion - maintainers org around local work and significant things
- get started, first and second
- set up maintainer roles, get process started locally
- some number of reviewers for a pull request, etc.
- brought to attention of another group of reviewers
- per repo?
- skillsets, bug fixes
- local work needs to be done and the bigger issues for larger group
- KS - branch rules?
- control branch elements
- JB
- since we made copies of full branches, need reorg
- KS - follow "start safe and expand"
- YZ
- one of the next times we meet, go through reasons why we chose operator, created few docs with current issues, outline why we chose operator



Action items:

-
-
-
-