2023-03-16 Infrastructure Working Group Agenda and Meeting Notes

March 16, 2023

16 Mar 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.

openIDL

When:
Thursday March 16, 2023
9am to 10am PDT/12:00PM-1PM EST
ZOOM Please Register in advance for this meeting:
https://zoom.us/meeting/register/tJwpcO2spjkpE9a1HXBeyBxz7TM_Dvo8Ne8j

Attendees:

- Sean Bohan (openIDL)
- Nathan Southern (openIDL)
- Aashish (Chainyard)
- Yanko Zhelyazkov (Senofi)
- Tsvetan Georgiev (Senofi)
- Faheem Zakaria (Hanover)
- Peter Antley (AAIS)
- Ken Sayers (AAIS)
- Surya Lanka (Chainyard)
- Adnan Choudhury (Chainyard)

Agenda Items:

- Antitrust
- Merge and the divergence of the two existing code bases that we are having now
- HLF Maintainers Guidelines: https://github.com/hyperledger/toc/blob/gh-pages/guidelines/MAINTAINERS-guidelines.md
- HLF Repo Structure: https://github.com/hyperledger/toc/blob/gh-pages/guidelines/repository-structure.md
- Next week's AWG topic: openIDL - Architecture Decisions
- Project Governance: https://hyperledger-fabric.readthedocs.io/en/release-2.5/CONTRIBUTING.html#project-governance
- Next topics
- AOB

Minutes:

- Fabric Operator last week
- thinking about the merge - makes sense in overall picture
- BAF vs Operator
- two codebases are very diverged, doesn't make sense to put the changes into a single repo
- idea: need to keep the old codebase, move forward, start adapting Operator and the new way of doing networks
- proposal: safer to have a sep repo, clean slate of codebase, dedicate new repo to the Operator and the way things are set up right now in the Testnet fork of it
- clean slate - removing things left in the repos and not used anymore, start on a better footing, continue building on top of that
- anyone starting to review Operator will be focused on that repo where everything is right now, merging it all together will be convoluted and confusing, hard to describe
- understand issues
- CY - clean repo means easier, no legacy of BAF
- Senofi - future of BAF, ND POC, some changes not merged, haven't felt compelled to that
- reasonable approach to leave ND POC as the last project using that framework
- openIDL: sounds fine for a clean copy, objective for ND POC to use the current frameworks - can we find a way to migrate ND
- AAIS - prefer to set up a diff network as "PRODUCTION", doesn't think we want to use BAF or Mongo, trying to figure out timing of next level of ND, production to something else with the time in between and funds to do it - tear down and set up with new architecture (if not -how connect old to new)
- Senofi - looking at taking chaincode or app changes might be there, bring that in, not sure how much customized it is or how much it is diff from what we have in the main branch
- those pieces we need to be careful of if we move ND to new
- AAIS - some diff things configured differently - had to beef up machines for # of rows ND had, tweaked PDC config to be bigger and broke up data as it goes across
- things around chaincode needed to be tweaked, project by itself to identify
- Senofi - invested effort in upgrading testnet to operator, some experience building a new network on the sidelines of an existing one, def bring that over
- CY - just talking blockchain network and not application?
- Senofi - some changes there, not major, changes in how we consume secrets and configs - for app components too? would change too?
- deployment not that much different, some configs, slightly different than ND
- CY - in terms of Infrastructure - app pieces, chaincode, not effect how you deploy the infra, sizing is a diff exercise
- need to figure out what components built pull into the main branch
- AAIS - step up to configurable alternate UI, ND wont work for generic case anymore, app layer is use case specific, what mech to put in place to make it possible to deploy node
- maybe a node participates in multiple use cases? possible
- CY - chunking, those things can be pulled it
- AAIS - need to see if they break, what things need to chunk, dont want to config PDC to be 8mb
- CY - thinking of setting hard limit on couch db
- AAIS - could be an issue for ND
- need to go out and analyze changes, report on what it is, define the project, multiple levels, network config, IAC, application code
- Senofi - help up und how to migrate network, having extension or improvements documented so we can better und how we do migrations
- one of the prereqs is to get CY up to speed with new way of building network, providing docs and
- NEXT IWG - present new deployment, after that work on what changes need to be merged or not
- openIDL - org a call to talk Operator, get ahead of the game on that, suggest knowl transfer
- CY - infra only or app and infra?
- Senofi - network and app? important: config changes, diverging from the main branch
- Senofi - diff template files for config, edit param for config, not in the template and merge code and deploy it won't work
- those configs are important - ND not only case where implementation might be different
- may have those dedicated custom figs to be done, in deploy, clone main repo, do changes on top due to new params
- crucial to und what may be breaking, good review of the new framework
- Capture architectural decisions
- ADR template from github
- arch decisions in markdown
- start with HOW
- put it in the Repo or the Wiki
- break it into dif repos - hard to keep good coherent list
- first order of business, is this a reasonable format?
- move on next week to James' list of ideas
- bring your opinion on arch decisions
- maybe working on the postgres one as an example
- following week james and peter can start working on arch decisions of hds
- use of Fabric needs to be justified - documented way to say "this is why"
- capture surrounding concerns for a decision - good to document, thought through
- template has
- Senofi:
- current etl effort - code is intertwined with network setup
- idea: remove etl from network layer
- new name for new repos
- doesn't tell us right now what they are/do
- name it in a more specific way
- move out all that is not the core network into its own repo
- we und better what exactly each repo achieved
- looking at pros and cons if we do that, can discuss
- major prob - set up network, set up a lot of other components to wait on (SSL Cert, etc.) doesn't make sense when just setting up core network
- ETL is not for every carrier/tenant - doesn't make sense to create those code components
- ETL data modeling standards will apply to source nodes, another major subsystem
- not get too fine - grained approach but good if we sep layers/repos

- arch layers/repos to split codebase
- AAIS
- what layers would split them into?
- Senofi - what we have right now
  - first layer network - infra as code, kub clusters, fabric network
  - second - app layer - openIDL main repo - stay the same
  - third - ETL layer - not related to all network or all apps, data and states on the sidelines
- AAIS - common app stuff - another repo?
- Senofi - app wise we need to be careful, should group apps that work together or live together
- adding apps for sep purpose go to sep repo
- for time being, apps right now, seem to work in synch environment, same purpose, dont see any misplaced in current repo
- adding diff type of app, not related to current ones, does make sense
- openIDL - centered around data model, data model stat reporitng, diff lines of business but would be family of apps, some other data interface required is a sep use case
- should we have a repo for the data model?
- AAIS - one folder called openiDL HDS-DDL - almost all of his models, business integrations, all sql, javascript, ref codes all DDL saved in DDL directory (needs cleanup)
- openIDL - put data standards in data model?
- AAIS - hope to get personal auto finished soon, written up in april, have an auto release before june
- Senofi - data models in the same repo make sense?
- AAIS - loves a good tree 🙂 breaking it by line, pretty elegant and working for now, working on
- https://github.com/openidl-org/openidl-main/tree/etl/openidl-hds-ddl/ref/pa
- Senofi - figure out if openIDL main is the best place for it?
- AAIS - likely data model and EPs connected to the scenarios of usage
- ND - couple EPs and data model is different and report processor on the analytics node - diff scenarios, diff combos of EP/UI/Reporting
- openIDL - tend to live together, interact at primary components
- AAIS - cert fields in ND were removed because not relevant, things like "transaction month", processor to handle reports needs to understand transaction patterns, no truly generic report processor or UI, intermingled with the data model
- Senofi - dp seeing as part of ETL, currently how thats processed by external tools like a lambda -
- AAIS - processed 2x: getting into HDS and extraction processing is bespoke to app/process/use case
- all will create diff reports and process differently
- AAIS - ETL more likely to be consistent across use cases, will change with variance in use case
- openIDL - good disctinction between app and ETL
- AAIS - gut feeling, coming up with a new use case, would have new domain model for that use case layered on top of diff pieces
- mult repos, new domain model
- every report would have its domain model and eventually etl input-output logic
- apps should be relatively stable and not affected in a major way
- Senofi
- fields for transactions relevant to chaincode and not data model/etl
- fine separation between transactions and data that lives on the network (specific to data model and how processed)
- app wise - not related
- network, infrastructure, chaincode - once changed, change application
- chaincode should live with apps
- concern: make sure data model doesn't impacts apps
- AAIS
- not sure he agrees - agnostic to the use case
- excited to get out of opinion and do code cleanup
- ND is very diff than stat reporting
- exercise selection of how we partition stuff
- content of payloads will change, EP data call request, more info for humans, universal, middleware for the network

GMT20230316-1...1792x1120.mp4

Action items:

- [ ] Sean to schedule deployment presentation and get feedback from all
- [ ]