2023-06-01 TSC Meeting Notes

Date

01 Jun 2023

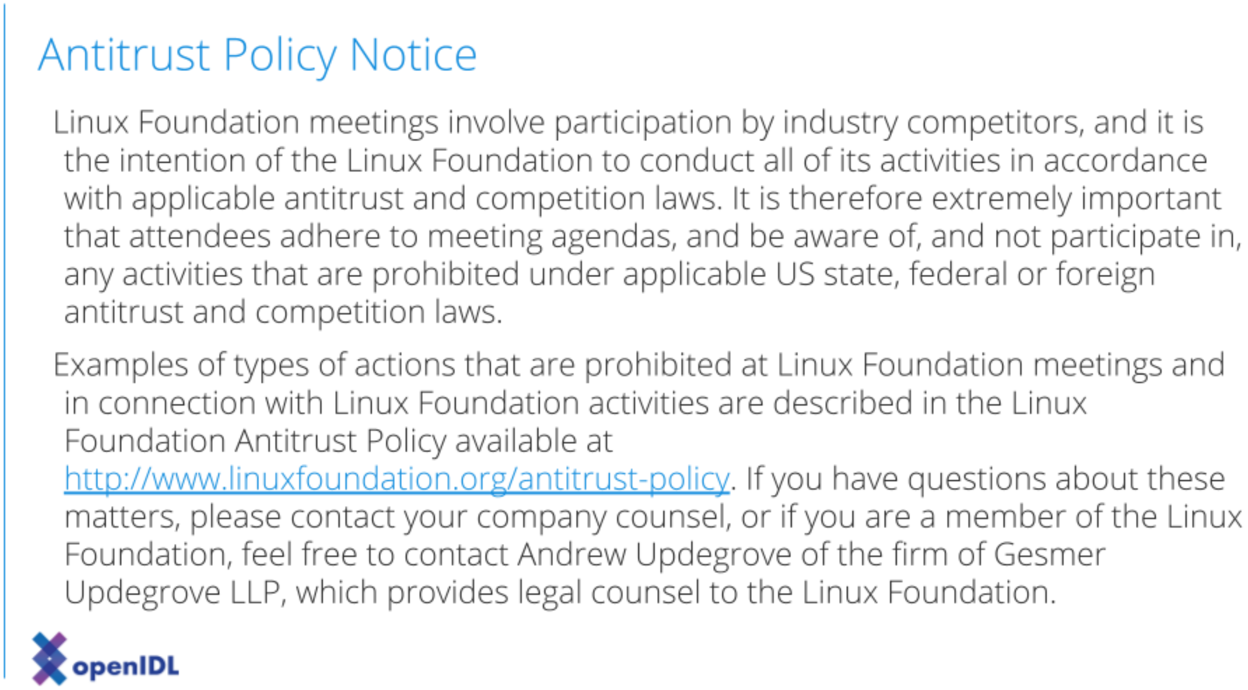
ZOOM Meeting Information:

Thursday, June 1, 2023, at 9am PT/12pm ET

Join Zoom Meeting

https://zoom.us/j/7904999331

Meeting ID: 790 499 9331



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.

Attendees:

- Sean Bohan (openIDL)
- Ken Sayers (AAIS)
- Peter Antley (AAIS)
- Brian Mills (AAIS)
- Tsvetan Georgiev (Senofi)
- James Madison (Hartford)
- Jeff Braswell (openIDL)
- Mohan S (Hartford)

TSC Voting Members Attendance:

- Ken Sayers, TSC Chair (AAIS)
- James Madison (Hartford)

Meeting Agenda:

- Opening
  1. Call to Order
  2. Anti-Trust, Review of TSC Meeting format and Participation by TSC Chair
- TSC Activity Desk

    1. Architecture Working Group (SeanB)
    2. RRDMWG Update (PeterA)
        a. AAIS Stat Reporting using openIDL internal Project (PeterA)
    3. ND Uninsured Motorist POC (KenS)
    4. MS Hurricane Zeta POC (KenS)
    5. NodeBuilder Workshops (JeffB/SeanB)
    6. Infrastructure WG (SeanB)
- Discussion:
    - Strategy to add new members and new application development projects to the openIDL network
- AOB

FOLLOW UP:

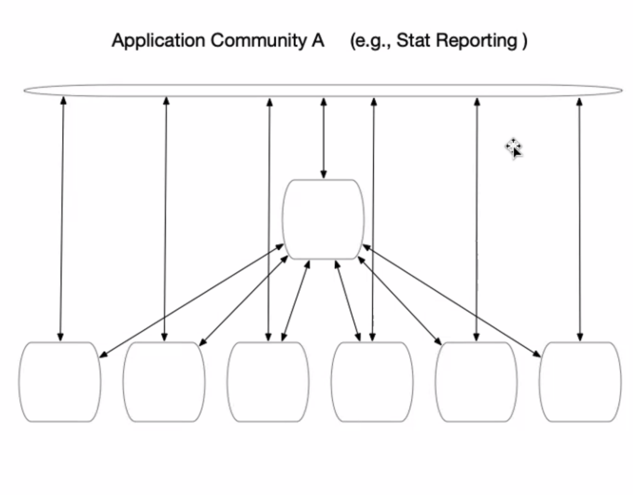
-

Recording/Meeting Minutes

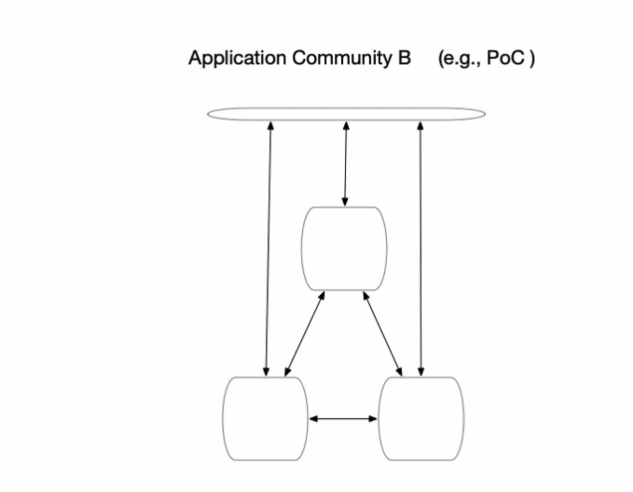
Discussion Items:

- AWG
    - architectural decisions discussion
    - OLGA decisions into a state of acceptance
- RRDMWG
    - developing app called OLGA to load data
    - translating existing stat plans to ref tables
    - finishing residential property farm/ag lines this month
- ND
    - Meeting with ND in June, recapping
    - carriers and ND DOI
    - feedback from carriers and discussing what they want to do going forward
    - head of DOI (commissioner) will be next NAIC president, having him as ally will help openIDL
    - sway on getting NAIC to consider openIDL instead of against it
    - June 13 meeting in ND
    - Remote option possible - reach out to LoriD (AAIS)
- MS Hurricane Zeta POC
    - Working through logistical hurdles w/ LF, Chainyard, Senofi, CrisisTrack
- NodeBuilder
    - Office Hours - if folks want to arrange a time
    - CONTAINERS
        - Lot of hooks into AWS, extreme lockdown environment, **fan of containers** - some things carriers cannot do (Cognito for example)
        - Strategic - what if we put as much in containers as possible to remove dependency on any cloud?
        - containers universal, clouds are not
        - "if these things are in place run containers"
        - we should minimize infrastructure influence, put as much into a container
        - Cognito example, less container than pluggable auth solution
        - complexity of making everything pluggable is a lot
        - cloud agnostic is a great idea but higher complexity, too much to get something working
        - precise and not too complex with cloud agnostic architecture
        - part that has to be cloud specific, done one way, hosted solution and carrier trusting vendor, THEN sensitive data part
        - if we want to restart conversation, needed to restate whats come before, where we are, biggest thing has been complexity of cloud agnostic
        - From directional, steering, margin you go either way, cool to go with cloud specific, pain to abstract, over time hit carriers with lockdown they have
        - as soon as you hit all that lock down cloud, not as cool as marketing
        - constraints
        - notionally - if we can somehow not get coupled, not easier upfront but long term...
        - Both points right on - can't go 100% platform as a service for everything
        - can modularize/containerize cores
        - fringe elements? things particular for that platform or company
        - standardize as much to make it work vs perfect
        - as much that can deploy across clouds
        - addressing things not containerized and not specific
        - how they can be made more compatible
        - What kind of containers are we talking about? All Docker images they run are prebuilt docker images, validated by IT, docker doesn't inherently have same portability
        - Similar process, docker images in internal repo, screened by security, procedural - once you clear annoying hurdle you can put into docker image
        - documenting how internal walls they hit
        - Do you run it in your cloud or NOT in your clooud
        - IT can be split
        - part run in YOUR clooud touches data, part you dont run is network, setup, comms between orgs, data being accessible

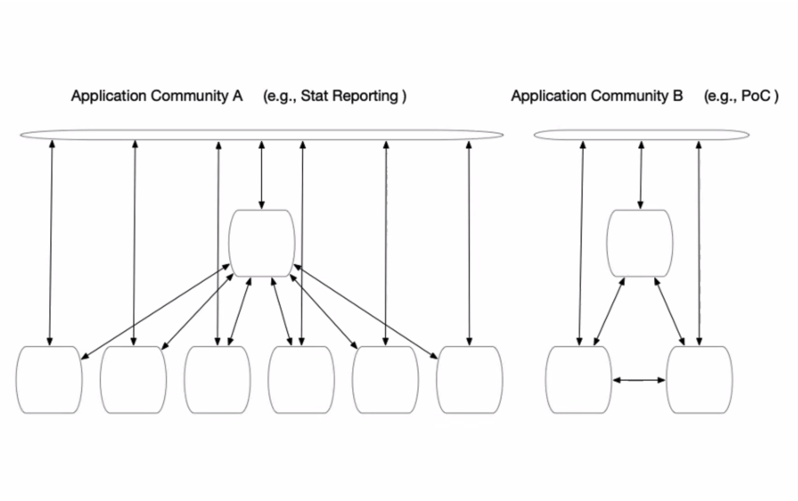
        - be more bare metal oriented if hosted, the stuff you run is more agnostic
        - On HDS side, where data is in there space, some things ported for diff environments
        - things in workshop related to network side of things, something to strive for in general
        - containers generic - Kub or Docker containers, in future, use simpler containers is a possibility, both suported on most cloud platforms
        - "things you can't do" - very interesting
- Infrastructure WG
- Strategy to add new members and new application development projects to the openIDL network



    -
    - process to add new app families and products to openIDL network
    - other factors as we flesh out specifics
    - illustrate - in a HLF network, round boxes rep nodes (peers, node environments that contain peers, K8 clusters
    - top bar conceptualizes default channel, message bus, coord activites across community
    - arrows - participation across the channel
    - sep channels for each participant to submit data privately
    - existing config for network
    - common channel, private channel for privacy



    -
    - POC, even if sep network, parties talk to each other, each in own conversations and have some "all" in some fashion
    - create peers with channel connections and member policies
    - beauty of fabric - config permissioned subsets and in what channels


Application Community A (e.g., Stat Reporting)
Application Community B (e.g., PoC)

- striving to use Fabric as common infrastructure
- once set up made easier
- each has some cloud space
- not floating in a vaccuum, nodes and peers exist in configs and infra
- how do we bring together and manage this - ultimately, need ability to add new POCs and apps to existing networks so we dont need to tear down/rebuilt
- proposing, these communities canbe added to openIDL as a general network, some time of management channel, upgrade of infra for chaincode
- cert things at the foundational level, for all participatngs, common channels within communities, grow openIDL network, have things started up, foundational set of network principals at the outset - some in dev and others in production so we can onboard an dmake progress developing apps in sep networks
- think thru and leverage to make the actual config of membership and participation and channels thru fabric operator
- imitial concept of test, dev, mainnets
- still may have testnet off to the side, reload a new version of HLF, K8, tear down restart
- testnet to add new things to, Peter and stat reporitng POC, these devs need to be done in a way to be supported after POC, more steady-state use and be part of the network
- assigning new node IDs, chronologically, whenmembers are added, who is joining network and members will know who is requesting to join, using operator, all know who is asking and what role
- gov and specification - communicating this, resources, not a centrally provided servicew
- in a decentralized network, require each member and membership canjoin the network
- throw out as a goal to work with current projects in development
- using operator
- flesh out - what are the specific things, diff companies and cloud platforms
- concept - arch stratgy for evergreen growing network, added over time
- Technical issue - mixing dev and production networks
- mult apps running on a node - don't want diff nodes for diff apps
- there is no exclusivity - connections added for an existing node
- tecnical complexity of two diff UIs, chaincode, etc.
- choincode used for messaging consisten, diff channels using same chaincode
- we have seen chaincode tweaked for ND that didn't need to be done for stat reporting
- chaincode largely crud but diff entities pushed across, new APIs, some diff set of chaincode avail for diff apps
- UI, APIs, etc.
- chaincode tweaked for apps
- there is a desire to minimize the diffs that exist, things need to be done, ways done commonly for messageing
- analyzing where those things go for messageing
- diff chaincode for diff communities for diff purposes
- for production vs dev - not everything on one network all th etime
- way fgor POCs to be put into the community for openIDL monitoring and support
- not to build a hjuge web of apps, way for openIDL to support this
- built consistently for monitoring and governance, way to est commonality in the initial setup

- avoid incompatible networks that require teardown and rebuild to be supported as part of openIDL
- Some degree of reuse
- The biggest challenge - hardcoded name of the common default channel
- prob - two sep networks, combingin them together, combining data from 2 channels with same name
- one of the things easy to solve - remove hardcoded name, eveyr POC define unique name of default channel, any time merge into openIDL network
- NAMESPACES!
- chaincode itself, could have diffs in chaincode, HLF designed to support mult chaincodes not necwessarily linked to each other, one peer serves diff apps
- that require different and unrelated chaincodes
- could deploy as many chaincodes as we like, consume into the same apps
- changes in date req and UI
- variations on the theme - look at thigns done
- needs of the app
- und where there are variations
- Related - setting up nodes, how we get provisioning work supported , board level discussion
- when we get a new use case, able to respond quickly and setup quickly, AAIS doesn't bear all the cost all the time
- area to leverage work being done consistenly with infra partners, node workshop is to spread the knowledge
- agree - point to make this easier to do and not harder
- NodeBuilder workshops are showing the tech stack is partially incompatible with diff companies for internal standards of tech
- direction to address that
- need to discuss, hartford is getting same feeling gotten from travelers, cannot run in their cloud the way it is
- Nodebuilder show how to build, wont be able to put them into their clouds w/o changes
- related to services tightly coupled, not bgi challenge to decoupld
- ID management cognito or something else, go open source w'/ cloud native solution deploy container, things possible
- mainly the things on AWS or other cloud, may do w/ Infra as Code
- no one solution
- some efforts, implementation on the carrier side to bring containers into their environ
- see openIDL - ref implementation, doesn't mean cant be adjusted to fit reqs of particular carrier
- how nodes set up gave rise to node as a service
- connection as proxy relay in protocol
- how to address we anticipate making possible for carriers to participate
- capture these as architectural decisions
- specifially introduced - not have default channel be fixed channel - diff communitis "use case channel"
- AOB

Time	Item	Who	Notes

Goals:

Action Items: