2023-02-13 Architecture WG Meeting Notes

Date

13 Feb 2023

ZOOM Meeting Information:

Monday, February 13, 2023 at 11:30am PT/2:30pm 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.

openIDL

Attendees:

- Jeff Braswell (openIDL)
- Sean Bohan (openIDL)
- Nathan Southern (openIDL)
- Tsvetan Georgiev (Senofi)
- Ken Sayers (AAIS)
- Mason Wagoner (AAIS)
- Dale Harris (Travelers)
- Yanko Zhelyazkov (Senofi)
- Peter Antley (AAIS)
- Faheem Zakaria (Hanover)
- Ash Naik (AAIS)
- Brian Hoffman (Travelers)

Agenda:

- Update on ND POC (KenS)
- Update on openIDL Testnet (Jeff Braswell)
- Update on internal Stat Reporting with openIDL (Peter Antley)
- Update on Infrastructure Working Group (Sean Bohan)
- ETL Discussion Continues (PeterA)
- MS Hurricane Zeta POC Architecture Discussion (KenS)

- AOB:
- Future Topics:

Notes:

ND Uninsured Motorist

- All data from all carriers and DOI
- Doing data call and DOI will work together on report
- Report in the next day or so
- 9 Groups, 11 Companies + ND DOI + ND DOT

Update on openIDL Testnet (Jeff Braswell)

- Progress on Testnet continues
- Conversations with Hartford and Travelers for test data
- Opportunity to use new tools like Fabric Operator to make setting up more efficient
- Moving stat reporting farther down the road, solve issues for carriers, security issues topics to be address

Update on internal Stat Reporting with openIDL (Peter Antley)

- Going well, focused on ETL
- Mason continues to work on code to decode stat records, urrently on Mobile Home Owners
- PA efforts on ETL
- Near future: interested in getting internal stat reporting, get data into node, next 6 months
- Big changes: looking to bring up node in near future, best way to do that - waiting on Operator to be functional

Update on Infrastructure Working Group (Sean Bohan)

ETL Discussion Continues (PeterA)

- Mechanism to log in, mechanism to submit data
- Proposed discussion flow:
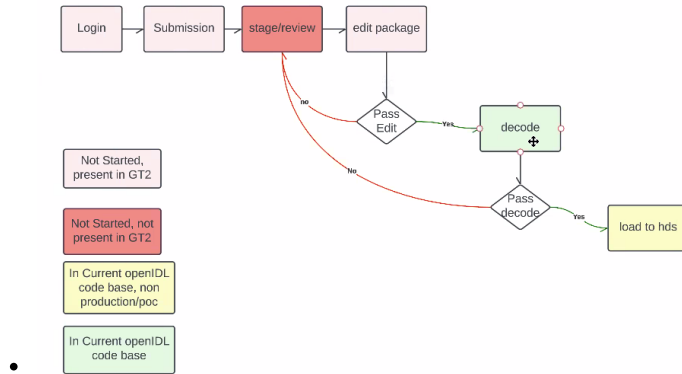
openIDL ETL Workflow
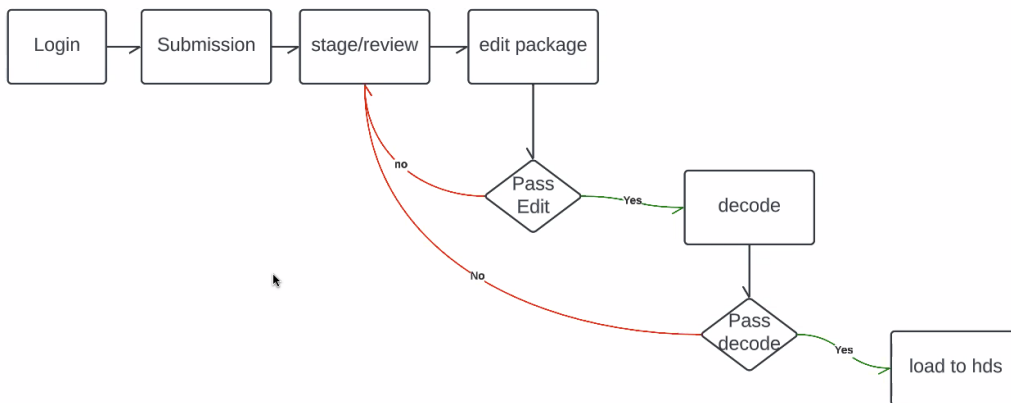
- Carriers submit stat data via SDMA
- Run edit packages against those coded messages
- after edit packages with SDMA, get go/no go decision (pass / fail)
- Submit if passed, review if failed
- Next stage is DECODING if passed
    - (change 01 to Alabama)
- error checking within decodes
- if decode works, edits work, move-load into HDS
- Last week discussion

-
- Stuff Peter has done and mason working on (internal stat stuff is the decoding, not the edits
- Legacy system, worked well, AAIS moving the size of the customer base in the last couple years
- EX: customer wiht a lot of records, load times were bottlenect b/c older system
- stripped out MVP of SDMA and made serverless rendition of it
- GT2 - can load data in 20 min

-
- GT2 - sinmoler UI than SDMA
- essentially - taking file, putting into S3, putting metadata about object into queue, using lambda to process all data
- way GT2 works, uses Lambda to move data from buckets to Elastic File System, shared file system, step function (orchestator)
- certain timeout in lambda, orchestrate mult lambdas together, nifty tool
- make smaller files, process in parallel
- using Aurora vs RDS
- RDS one step above linux box
- Amazon took RDS step farther, what does enterprise RDS look like?
- 2 versions big and popular -
    - first use for auruora - allows to have multi-zone replication, complex dbs in synergy aroundf the globe
    - designed to work with lambdas and do cool networking things
- Serverless might fit us well
- Use mult subnets tied into Auruora for extremely fast load

openIDL ETL Workflow w/ status

Login → Submission → stage/review → edit package

Pass Edit — no → (back to stage/review)
Pass Edit — Yes → decode

decode → Pass decode
Pass decode — No → (back to stage/review)
Pass decode — Yes → load to hds

Legend:
- Not Started, present in GT2
- Not Started, not present in GT2
- In Current openIDL code base, non production/poc
- In Current openIDL code base

SDMA vs GT2

	SDMA	GT2
Infrastructure	1. Oracle 2. TomCat 3. Java 4. Ec2	1. MySQL/ Aurora 2. API Gateway 3. SQS 4. Cognito 5. EFS 6. S3 7. Step Function/Lambda
user Interface	1. Login 2. Data Submission 3. Interactive error reporting 4. Grahical Data insights.	1. Login 2. Data Submission 3. Txt Report based error reporting
load speed	Days´	Half hour
Primary Function	1. Recieve Submission 2. Run Edits	1. Recieve Submission 2. Run Edits

- SDMA functionalty with GT2 Speed may be a goal - is Aurora off the table? use aurora?
- could run postgres with aurora
- Possible?
- Building a reference implementation, something works, not neccessarily most robust but could build it out, something as an option
- get IT folks to run in environment, as a service?
- not core component of openIDL, DIY
- Aurura - paid tool, locks into AWS, ref imolementation on AWS, lanbdas, step functions, etc. - all reasonable things to put into Ref Implementation?
- Issues?
- Aurora supports Postgres SQL
- Moe lanbdas improving, horizontal scaling makes it possible, splitting across lambdas, stripped down SDMA, loses some functionality
- UI and interaction between errors and stuff, both but mostly lambdas
- no sequential dependency on records
- dev has said Auroria allowed massive parallelization of IO, built for serverless, high horizontal scaling
- How long is too long for a load (time it takes to load data into the db)? 6 hours for 30MM records? fro you hit submit until it is all there
- most use cases, daily upload enough
- IICMBA (ND)use case, check for "is this car insured right now" very diff use case
- IICMBA competes with ND POC is trying to do
- Issues with tech stack they are targeting?
- How if a carrier isn't on AWS? Subset as a reference to be added to it? THere is a corresponding tech stack
- good azure dev spends time with PA could translate
- keeping DB/Postgres the same
- so many interdependencies

Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording:

GMT20230213-1...2560x1440.mp4