

2023-03-06 Architecture WG Meeting Notes

Date

06 Mar 2023

ZOOM Meeting Information:

Monday, March 6, 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 Updegrave of the firm of Gesmer Updegrave LLP, which provides legal counsel to the Linux Foundation.



Attendees:

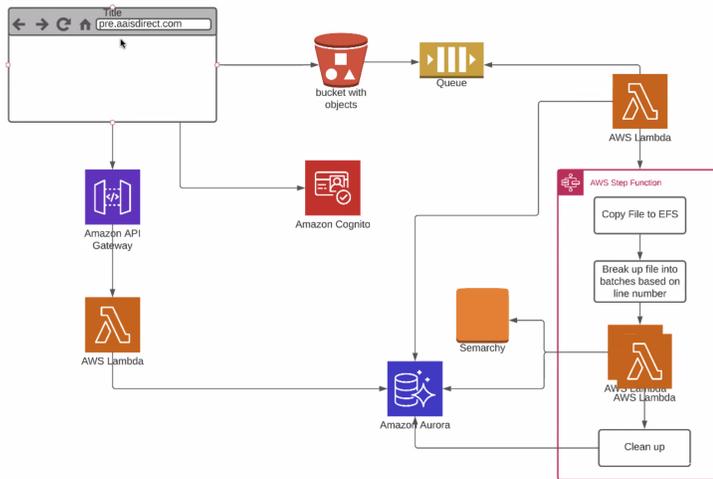
- Mason Wagoner (AAIS)
- David Reale (Travelers)
- Jeff Braswell (openIDL)
- Nathan Southern (openIDL)
- Yanko Zhelyzakov (Senofi)
- Ken Sayers (AAIS)
- Satish Kasala (Hartford)
- Dale Harris (Travelers)
- Ash Naik (AAIS)

Agenda:

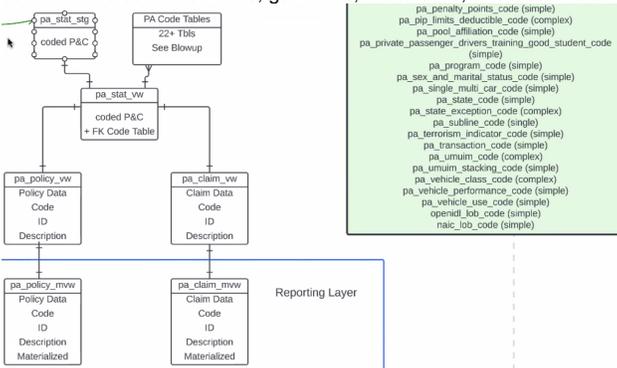
- Update on ND POC
- Update on openIDL Testnet (Jeff Braswell)
- Update on internal Stat Reporting with openIDL (Peter Antley)
- Update on Infrastructure Working Group (Sean Bohan)
- MS Hurricane Zeta POC Architecture Discussion (KenS)
- ETLs (PeterA)
- AOB:
- Future Topics:

Notes:

- ND POC
 - Carriers received reports of VINs not registered, chance to see where they gave VINs as insured but not registered in ND
 - Started loading Feb Data as of last Weds (5 so far)
- openIDL Testnet
 - Test data from 2 carriers, working on new vers of testnet using Fabric Operator
 - Code Merge and strategies for single codebase (for POCs and members), working towards testnet and mainnet
- Internal Stat Reporting w/ openIDL
 - working on production-izing auto data
 - working on the decode process and utilizing db for decoding
- Hurricane Zeta POC
 - waiting on carrier for internal analysis (AWS region followup)
- GT2
 - fun of inherited cloud formation templates
 - Data Layer for new ETL plan
 - high level of GT2
 - Recycle as much code as possible and add features required
 - GT2 to be renamed



-
- upload file, COgnito controls auth, app moves text file to bucket, names of files added to queue
- lambda kick off, step funct kicks off, set up EFL (elastic file storage), break up into batches, run edit packages and load to table and any record with error write error to error table, returns to user
- users highlight error records against SDMA
- gap in GT2 - does not allow inline editing
- using aurora (not required) but lets them stream more data through, using EFS + Aurroa, high throughput data
- series of lambdas to load to same table
- combo of policy and claim records, sparse, bunch of nulls, wide and sp
- load data into combined table, get errors, make edits, after records edited, approve submission an dmove into HDS



-
- stg - coded raw stat records, move dinto columns
- personal auto stat view - connects codes that come into code tables
- state code can be shared across all lines
- after foreign keys, breaking stuff apart - policy view and claim view
- descriptions - human readable tables
- rows and codes and joins

- lot of IO
- painful to make those joins over and over
- using materialized views with policy and claim view, all policy data, codes, descriptions - ready to materialize
- reporting layer - policy-claim, materialized view each line
- talk in depth on GT2 next week
- based on a ref implementation for ETL - all expected to run in carrier cloud
- all in the HDS
- really discussing these tables, part that matters (minimum) thing that needs to happen - need to be able to write extracts on these tables
- carrier could skip ref
- when we get to the bottom tables, will we have diff sets of codes for states, etc.? thats where standard data model comes into play?
- EPs should expect - not state name but abbreviation, etc.
- AR, ID, etc and primary key for each too
- wide table
- easy to run reports against, still at the same per transaction level
- each row easy to read for any reporting tool you want to connect to
- coverage codes dont change often
- FlyWay discussion (PR from Joseph for demo)



Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording: