2023-03-27 Architecture WG Meeting Notes

Date

27 Mar 2023

ZOOM Meeting Information:

Monday, March 27, 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

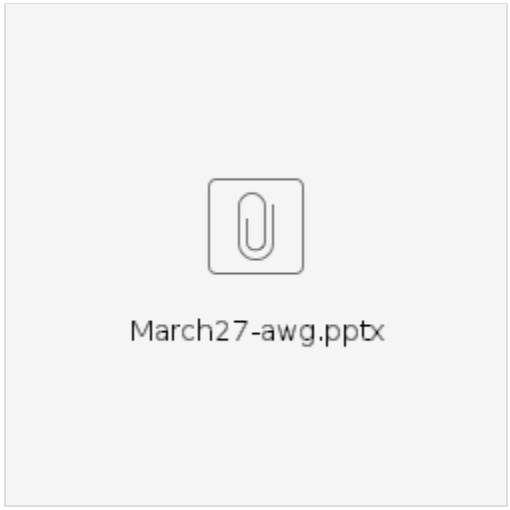
Attendees:

- Sean W. Bohan (openIDL)
- Allen Thompson (Hanover)
- Mason Wagoner (AAIS)
- Peter Antley (AAIS)
- Justin Cimino (AAIS)
- Brian Mills (AAIS)
- Jeff Braswell (openIDL)
- David Reale (Travelers)
- Joseph Nibert (AAIS)
- Ken Sayers (AAIS)
- Yanko Zhelyazkov (Senofi)
- James Madison (Hartford)
- Brian Hoffman (Travelers)
- Dale Harris (Travelers)
- Satish Kasala (Hartford)
- Tsvetan Georgiev (Senofi)
- Aashish Shrestha (Chainyard)
- Faheem Zakaria (Hanover)

Agenda:

- JamesM Discussion: coding & operating standards: queries, null, etc.
- 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)
- MS Hurricane Zeta POC Architecture Discussion (KenS)
- Architecture Decision Capture Process (KenS) - Discussion
- AOB:
- Future Topics:

Notes:

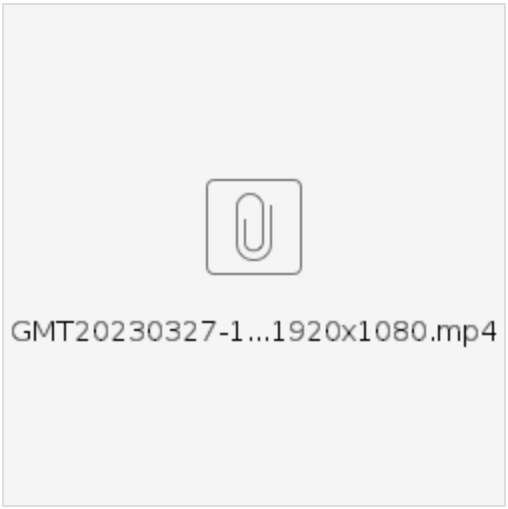
- PA
- 

    - ingestion format conditional
    - https://github.com/openidl-org/openidl-main/tree/awg/openidl-statplans/pa
    - AWG branch, stat plan as a PDF
    - stat plan, going forward tyalking about reg reporting, stat stuff might be a buzzword
    - recycling auto stat plan, openIDL reg reporting standard for personal auto
    - personal and commercial auto but for this personal
    - big ?: decoding and how we go about it
    - link 1 - stat plan PDF
    - stat record broken out by column
    - join on all codes and ref tables
    - personal auto stat view - not claim or prem specific yet
    - SDMA and GT2 - stor all in one table
    - Source of truth is stat plan instead of sql
- James A
    - modeling group
    - 5-6 questions, arch in nature
    - 1. lookup tables in code, as opposed to more maintainable
    - location of record for these codes, in the JSON and those are the official locations of codes and values
    - comfortable with json as place to maintain it
- KS
    - interaction points of business people?
- JM
    - maintain lookup tables in excel, can send to anyone and they can navigate - intermediate tech and business
    - helps
    - in json and modify
    - lookup tables change at same freq as code
    - able to release independently
    - BA/QA type roles - maintaining it
- PA
    - utilizing flyway to manage db
- JN
    - biggest concern is lookup tables - for insert of data only
    - if we know version 001, ABS table will only have 3 records, if we want to add more things, vers 00.2 -would know by looking at schema history - truth kept on DB and not on some obscure file
- KS
    - two things - schema and data
    - for ref tables, data is much less changeable, doesnt change as often, more often than schema
    - wouldn't use flyway for transactional data (states, lookups, never transax)
    - is Flyway meant to be applied to data as much as schemas? more schema management than data migration
    - lost with schema vs data
    - new code for personal auto? needs to go to every carrier

    - new code? does change data
- JN
    - define as a team how we handle it - dont care in DBMS versioning? then pull it out - depends on how strict
- KS
    - when do customers participarte in management of lookup data
- JM
    - some ways, ref data behaves like code
    - rows part of schema seems reasonable
    - programmer go into json to modify it?
- PA
    - depends on budget
    - could make excel and easily pull json out of excel
    - need to gen sql file
- KS
    - process involved - cost in distribution, governance, approvals, acceptance of changes not changes themselves
    - if a programmer vs someone loading an excel = trivial process cost
- PA
    - how much harder to use excel vs configs? API to access it?
- JM
    - if driivng off json files less worried
    - reduce programmer jobs due to expense
- PA
    - coverages, based on codes, categories
    - some get weird, multistate for 47 states, code gets
- JM
    - still need to be able to read json
    - versioning is intriguing
    - code that does inserts
    - released with everything else
    - how would flyway treat lookup tables and schema-ized notions
    - run into issues - modify to, make script with updates
- JN
    - how we decide to do versioning, all we have to do in the script
    - as a team discuss as well
    - peak constraints,
    - fly way is controlled version of schema, and do checks for alter/doesnt - keeps us honest
- JM
    - as long as it is re-runnable
- JB
    - 3 things:
        - data standard
        - implementation at phys level
        - verisoning and deployment of changes
    - standards often used in forms easier to manage by biz side
    - renumerations, typically used at that level
    - what will happen with sql / json
    - implementation
    - cutting out top level, def of data format and enumwerations
    - standard orgs like ISO, LEI others - will combine things into single code table for type being used for
    - becomes implemented and turned into phys usage
    - relevant to consider defs of code tables as reference for data docs and see how easy to automate/deploy/version
- KS
    - vers of the schema and the data dictionary and lookup table values
    - understand
    - standardization things like accord - top level doc
        - standard
        - vers of standard
        - fancy terms and structure to make it very official
    - JSON nice intermediate form
- JB
    - labels, structure, readable
    - normative
    - "shall conform"
    - ocumenting lookup tables seems like more work but having a form you can extraxt from is good
    - schema is a phys thing, implementation of a particular db
    - important to get buy in form the biz side, look at it, anyone can use, look at higher level docs
- PA
    - on Fri going forward, openIDL personal auto data standard as a doc
    - version, automate, check it
- JB
    - good for consistency and reliabllity
    - do want some form that someone who doesn't und the lower level can und the standard
- JM
    - right b/w JSON and statplan - gap there
    - once you get the json, somewhat readable
- JB
    - prob with XML standards, rigid, JSON has flexibility, sequence not as important
    - there are benefits to good clear semantic defs
    - not so much stat plan as it is the doc of the business side

    - starts conceptual
- KS
    - documentation of the standard
    - need the same thing for HDS which is NOT the stat plan
    - eventual data standard
- JB
    - excel example is readable
- JM
    - spreadsheets are readable and consumable
    - attempt to be in the middle
    - not sure right for this job but worth looking at
- JB
    - lead in to this (PEter's work)
- KS
    - going back to way it works, PA and JN, techs will not say "we need to add new code to this table" - will do it based on an org managing a standard saying "we need a new code", managed at standard level not code level
    - maybe excel as working docs
- PA - hard time imagining an excel could help him out
    - actual values would fit in excel but also a whole para explaining
- KS -
    - excel is a comms mechanism so those like DH and others can contribute to the standard
- JM
    - data dictionary of sorts
    - when it needs ot change, how will you do that
    - ex: ref w/ 3 rows, add 4th and 5th
    - know before x used code c and after code d
- PA
    - decode with left joins
    - stick expiration dates on "where clause"
- KS
    - must bubble up to EPs and Reports
    - run report on data 2 years old, or last year, super complicated extraction
    - morphing of the algo
    - data avail or not
- JB
    - effect other things
- KS
    - changing schema gets nasty
- JB
    - no need to change schema if updating code tables
- JM
    - if we need expiration dates on ref tables, as the years go by, put in effect and expire var rows
- JB - just because useful doesn't mean you have to implement
- PA
    - can I add Jan 1 2000
- JM
    - need to standarize on neg and pos affinity
- KS
    - where does the data come from? one time extraction, another view for this year
- JM
    - comes from biz people
- DH - 15 years ago we didn't have electric cars, now we want them as a vehicle type
- KS
    - do a report on what came through, cant ask for code on a report if it didn't exist
    - need to be respectful of changes to data
    - sensitivity to the date
- PA
    - doing decode based on date of when policy was issued
- KS
    - view has to understand the decode based on the date
- JB
    - if a code didn't exist in a hist record, b/c newer code, make exception anyway
- JM
    - effective expiration on all lookup tables
    - append case is easy case
    - when a given code needs diff meaning in the future than whan it was in the past
- PA
    - now not limited to 0-9, code 11 would need to move to the end
- JB
    - "the code is B"
- JM
    - global arch assumption, fixed width problems are over, everything is meaningful
    - codes are unique, no prob of duplicity
    - no effective expiration dates
- DH
    - add a code later on, make sure not being used in the past
    - no vehicles in 1950s showing electric cars
- KS - validation rule?
- JM

    - entry in ref table issued will never change? bold statement
- JB - include context of code was for
    - important - codes differentiate properties specify terms of policies and the business, actual models of the business itself
- JM
    - compromises too
- KS - not mutually exclusive
- JM - over-engineered
- KS - posited, using a rule to check if a code should be that, effectively an expiry date on a row, generalized for every field, dont need to write individual rules
    - use effective expiration dates
    - assumes not having value there, handled elsewhere
- JB - could have codes for entities or things that expire, expiration important
- JM
    - immutability argument, once a code never rescind it
- PA
    - more granular columns would make sense
- KS -could do by expiring all codes in those fields
    - cant expire schema
    - json key optional
    - more flexible for the future
    - processing of the historical records
- PA - expiration on every code? yes
    - default values based on stat plan?
- JB - advocate 1/1/2000, hard limit some can't go past 2186?
    - need a hard future data vs needing it null
    - tech limit on dates?
- KS
    - new item - default start and end dates
    - reco start date that gave KS a scare - 2k
    - bad idea - will we get data before 2k we need to work with? If so go 1970 or 1900



Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording: