

2023-08-28 Architecture WG Meeting Notes

Date

28 Aug 2023

ZOOM Meeting Information:

Monday, Aug. 28, 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:

- Sean Bohan (openIDL)
- Jeff Braswell (openIDL)
- Yanko Zhelyazkov (Senofi)
- Nathan Southern (openIDL)
- Allen Thompson (Hanover)
- Ash Naik (AAIS)
- Peter Antley (AAIS)
- Dale Harris (Travelers)
- Tsvetan Georgiev (Senofi)
- Faheem Zakaria (Hanover)

Agenda:

A. Status Updates

- RRDMWG - this week's meeting (PeterA)
- Infrastructure Working Group - this week, Thurs 12 at noon ET
- Next AWG Call - Monday Sept 11, 2023 (NO CALL NEXT MON DUE TO LABOR DAY)
- Next Technical Steering Committee Call - Thurs Sept 7, at 12 noon ET

B. HDS and carrier side:

- How will carriers load the HDS
- How will carriers review data call requests
- How will extraction patterns be executed
- Possible containerization of the extraction processor (?)

Notes:

- PA
 - OLGA status
 - currently building OLGA
 - good solution for all at AWS
 - need some porting to run azure
 - not sure where with their timelines
 - OLGA is good for taking a large set of multi-line stat data, ingesting it rapid way
 - then, doing large amt of validations quickly, allowing users to update single and bulk records
 - for Cat reporting, a lot more sparsely populated
 - way less attributes for stat record
 - not sure if loading the cat reporting may be more trad load, need to come up with plan/timeline for that
 - whoever is operating that node has PostgresDBm load stat records into DB, has scripts to do that
 - maybe first iteration could be manual
 - every carriers has HDS
 - stat record is positional record
 - AAIS has one script, runs mapping, maps stat record to json, another script makes insert tables and inserts all into DB
 - doesnt take long, some operator runs on local machine
 - possible to extract data for CAT POCs, put into format to help loading
 - basically stages of data processing, relatively doable
 - D or F? reasonable way to do POCs with managed data loading
 - F - great way to get folks on network, still need a true node somewhere
 - node as provided for connection, talking about just loading data into hds
 - in case of azure, native, try to have native environs for HDS to manage data, alt - prep data for trusted provider
 - part of discussion, architecturally, you will have HDS
 - immediate POC discussion might be spin off
 - good etl tools, might be easier to put directly in file db
 - outputting format not a big issues, getting it together, and aligning it with ea column in schema - is it agg, derived, inc cert subset - getting THEN formatting THEN calculating properly is the challenge

Line of Insurance	Accident Date	Date Code	County Code	Institution Code	Loss Amount	Claim Count	Cause of Loss	Accident Date
1	51	100	100	100	100	100	100	100
2	52	100	100	100	100	100	100	100
3	53	100	100	100	100	100	100	100
4	54	100	100	100	100	100	100	100
5	55	100	100	100	100	100	100	100
6	56	100	100	100	100	100	100	100
7	57	100	100	100	100	100	100	100
8	58	100	100	100	100	100	100	100
9	59	100	100	100	100	100	100	100
10	60	100	100	100	100	100	100	100
11	61	100	100	100	100	100	100	100
12	62	100	100	100	100	100	100	100
13	63	100	100	100	100	100	100	100
14	64	100	100	100	100	100	100	100
15	65	100	100	100	100	100	100	100
16	66	100	100	100	100	100	100	100
17	67	100	100	100	100	100	100	100
18	68	100	100	100	100	100	100	100
19	69	100	100	100	100	100	100	100
20	70	100	100	100	100	100	100	100
21	71	100	100	100	100	100	100	100
22	72	100	100	100	100	100	100	100
23	73	100	100	100	100	100	100	100
24	74	100	100	100	100	100	100	100
25	75	100	100	100	100	100	100	100
26	76	100	100	100	100	100	100	100
27	77	100	100	100	100	100	100	100
28	78	100	100	100	100	100	100	100
29	79	100	100	100	100	100	100	100
30	80	100	100	100	100	100	100	100
31	81	100	100	100	100	100	100	100
32	82	100	100	100	100	100	100	100
33	83	100	100	100	100	100	100	100
34	84	100	100	100	100	100	100	100
35	85	100	100	100	100	100	100	100
36	86	100	100	100	100	100	100	100
37	87	100	100	100	100	100	100	100
38	88	100	100	100	100	100	100	100
39	89	100	100	100	100	100	100	100
40	90	100	100	100	100	100	100	100
41	91	100	100	100	100	100	100	100
42	92	100	100	100	100	100	100	100
43	93	100	100	100	100	100	100	100
44	94	100	100	100	100	100	100	100
45	95	100	100	100	100	100	100	100
46	96	100	100	100	100	100	100	100
47	97	100	100	100	100	100	100	100
48	98	100	100	100	100	100	100	100
49	99	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100

- same table used for all, same columns
- attributes we discussed
- Evolution of the data model itself, not just policy and claim event records
 - struct of the db, what qualifies
 - combination of data modeling and assets.
 - SDMA and legacy systems, keel two tables together
 - during load process, all together
 - filter based on transax code
 - 2 tables in DB
 - ingested, call in same table
- How data call requests would be reviewed
 - when you consent to data call it is executed
 - provision, requested, can consent to data call but executd at a later date
 - Roadmap?
 - Right now - consent = execution, open item for deferred execution
 - part of UI to review data calls, ability with node-as-a-service, have UI interrogate request, download, examine and try locally
 - running sql to local table before you commit and direct to response
 - how to host UI, interact with db
 - orchestration and control to be housed closer to db
 - done at the carrier side near HDS

- Extraction
 - description of what was requested goes with the executable
 - when it does execute, run against EP schema
 - executing SQL, can't modify HDS
 - EP Schema with Peter's POC
 - EP Schema temporary
 - ep permission to create temp tables, after EP run that would be wiped clean
 - expecting on EP running at a time in the machine
 - detail, phys workflow
- Containerizing
 - R&D topic
 - need some more insight into the diff EPs and data calls
 - wont be able to address, how we have more than 1 pattern
 - stat reporting was so complicated - may be worth having it as an app container to do the report
 - talking about stat reporting lately
 - some agg fields

Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording:



GMT20230828-1...1920x1080.mp4