2023-05-01 Architecture WG Meeting Notes

Date

01 May 2023

ZOOM Meeting Information:

Monday, May 1, 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 Bohan (openIDL)
- Mason Wagoner (AAIS)
- Jeff Braswell (openIDL)
- Peter Antley (AAIS)
- Joseph Nibert (AAIS)
- Ken Sayers (AAIS)
- Yanko Zhelyazkov (Senofi)
- Ash Naik (AAIS)
- David Reale (Travelers)
- Faheem Zakaria (Hanover)
- Nathan Southern (openIDL)
- Tsvetan Georgiev (Senofi)
- Brian Mills (AAIS)
- Adnan Choudhury (Chainyard)
- Dale Harris (Travelers)
- Satish Kasala (Hartford)

GMT20230501-1...1920x1080.mp4

Agenda:

- MS Hurricane Zeta POC Architecture Discussion (KenS) (No change)
- Update on openIDL Testnet (Jeff Braswell)
- IWG update (YankoZ)
- OIDL-0004 Decision (Kens)
    - OIDL-0004 - openIDL Setup and Management Documentation
- Update on RRDMWG and internal Stat Reporting with openIDL (Peter Antley)
    - OLGA: Sitemap + Work Flow(s)
- AOB:
- Future Topics:

Notes:

testnet

- Scheduling workshop
- conversations with Selective Insurance and Hartford
- Touch base with TRV
- kick off in the next weekish
- infra as code before you get to operator

IWG

- Publishing docs to ReadTheDocs
- start in wiki, move to RTD
- consistent with Fabric

OIDL-0004 Decision

- Decision

DMWG

- working on personal auto data standard
- fairly mature draft
- doc is interesting
- large section for each line of business
- personal auto fairly mature
- need to come up with where to put it
- so people can look it up
- short term: needs to be in PDF, on meeting notes
- need how to document and how to make the standards available to people
- old doc PDF from 1990s
- do we do PDF standard or whole website?
- how do we onboard someone new
- generic rules that apply outside of the domain
- section on personal auto
- Sean to reach out to Anoncreds community for how to host data standard
- reference: https://developers.google.com/transit/gtfs/reference
- OLGA application

High' Level

Key Functionality

1. Login
   a. Username& Password
   b. Password Reset
2. Upload FIle
   a. Create Job based on Master File
   b. Chunk Master File
   c. Load Data to {lob}_raw
3. Validation Event
   a. Select series of rows
   b. Validate selected rows
   c. Log errors in error table
4. FIle Submission
   a. Move Data RAW->STG
   b. Archive Source File
   c. Destroy Chunks
   d. Clear Error(s) from table

Requirements

1. Read List of All files in Olga
2. Access all files not yet loaded/Discarded
3. Read List of submitted files
4. Select an existing file, advance to /${filepage}
5. Trigger Validation event on selected file
6. Discard selected file/files.
7. Upload New file
8. Logout

Upload WorkFlow

1. Select a file from local storage
2. upload a file (30 million records)
3. Recieve the file
   a. recieve file in landing bucket
   b. establish a job associated with master file name.
      i. chunk the file into 10,000 records chunks. Link each chunk with master job.
      ii. Write chunk file meta data to sqs
         1. process each chunk load data from s3 to database
         2. utilze check sum to ensure all data is loaded
   c. Display Job status based on aggregate status of all chunks

1. File is Recieved by API Gateway
2. Lambda::Receiver
   a. Recieve file from API
   b. Places file in S3::Upload
3. Lambda::Chunk
   a. Triggered on 'put' event on S3::upload
   b. Attach EFS
   c. Move file from s3->EFS
   d. Create Job in Postgres::Job
   e. Read Through file
   f. Create Chunk on Configureable interval
   g. Write Chunk metadata to SQS:Chunks
   h. Add record to Chunk Table. Postgres::Chunk
4. Lambda::Load
   a. Read from SQS::Chunks
   b. Open Chunk File
   c. Convert ASCII characters to Numeric
   d. Generate Insert Statements
   e. Bulk Load to Postgres::ETL Schema:: openidl_etl_{comp}
   f. Limit Max Concurrency to not overwhel JDBC Driver

Types of Failure
1. Invalid File
   a. Reject File -> Start Over
2. Invalid Record (line)

Time	Item	Who	Notes

Documentation:

Notes: (Notes taken live in Requirements document)

Recording: