

openIDL - Target Architecture

High Level Requirements of the System



Notes from the team

KS

- support stat reporting
- support ad-hoc data calls
- verify data availability
- Data stays private to the carrier
- Only results of extractions leave the carrier
- common extraction request across all nodes
- common data model for extraction across all nodes
- Any one extraction uses the same model for all data owners
 - JM - Agreed, but per level of the published model
- trust extractions - we are executing code after all
- Correlated data can be accessed as part of the extraction
- All updates to the system are well managed
- Support multiple "footprints"
- physical db schema maintenance is minimized
- Technical choices for implementation can vary from carrier to carrier for those items that reside in the carriers perimeter
- Passes audit by All members of TSC

JM

- Security model has white hacks as part of regression testing
- Done when everything is in a comprehensive regression test base and all tests pass
- Each major box has "push button" install process
- Reference tables all pre-populate as part of HDS install
- DDL in the db to build out the model in each major technology
- Test records self install to HDS and test base runs.
- Capacity and DR specifications are published and tested

TE

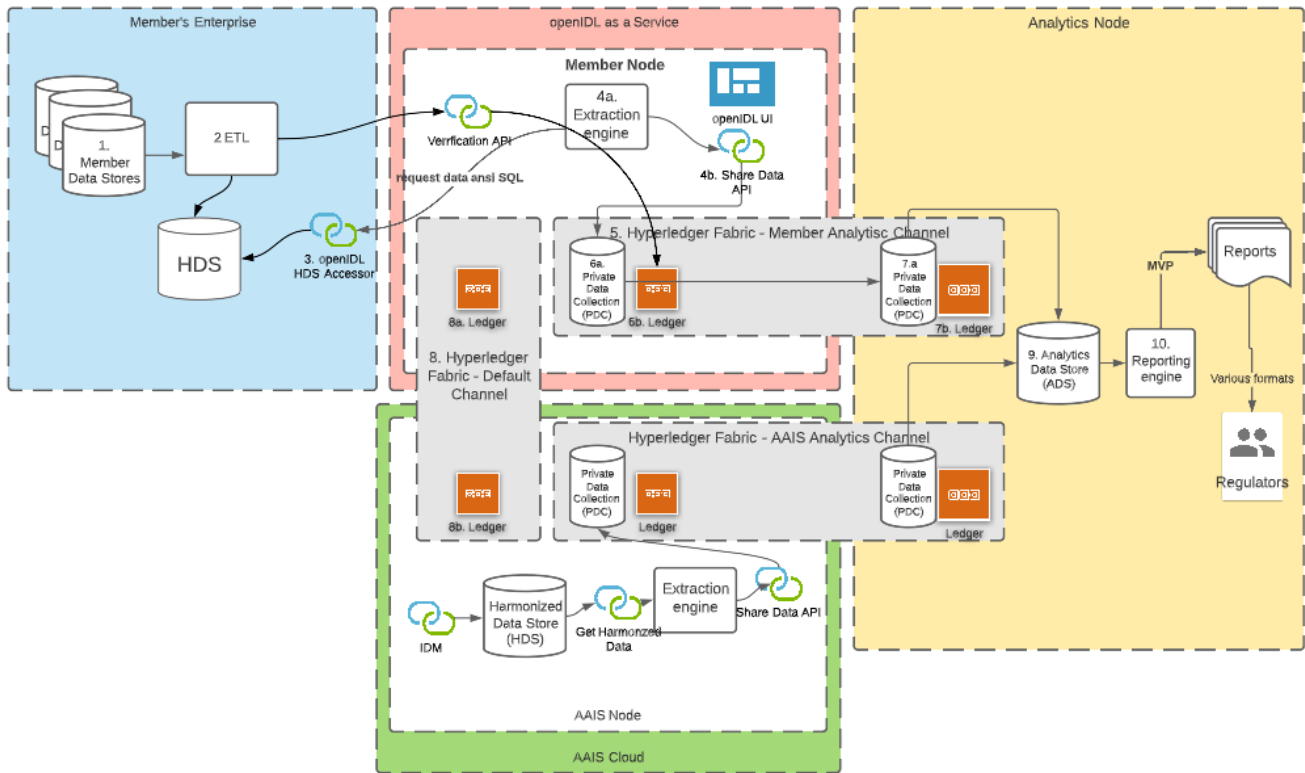
- Insurer needs a standard for regular Policy-level "experience recording" assertion
- Insurer's data moves from batch/chunk integrity at load, to Per-Policy integrity over time and at time of inquiry
- Analytics node is the "box" when we talk "openIDL in a box" as it determines the value of the information as a result (who gets the analysis and why) - we need different types/sizes as well as Orgs/Roles (total it's all AAIS)
- Analytics Node host ("information seeker") or Seeker's Agent (e.g. NAIC, PCI, etc. on behalf of >1 Seekers) for Org/Roles/purpose - creating Extraaction Patterns, etc. (today is AAIS or whoever deploys the network - ND, MS, etc.)
- "AAIS" cloud/node(s) need to become "(Stat) Agent" orgs/nodes (>1) acting on behalf of >1 Data Owners

SB

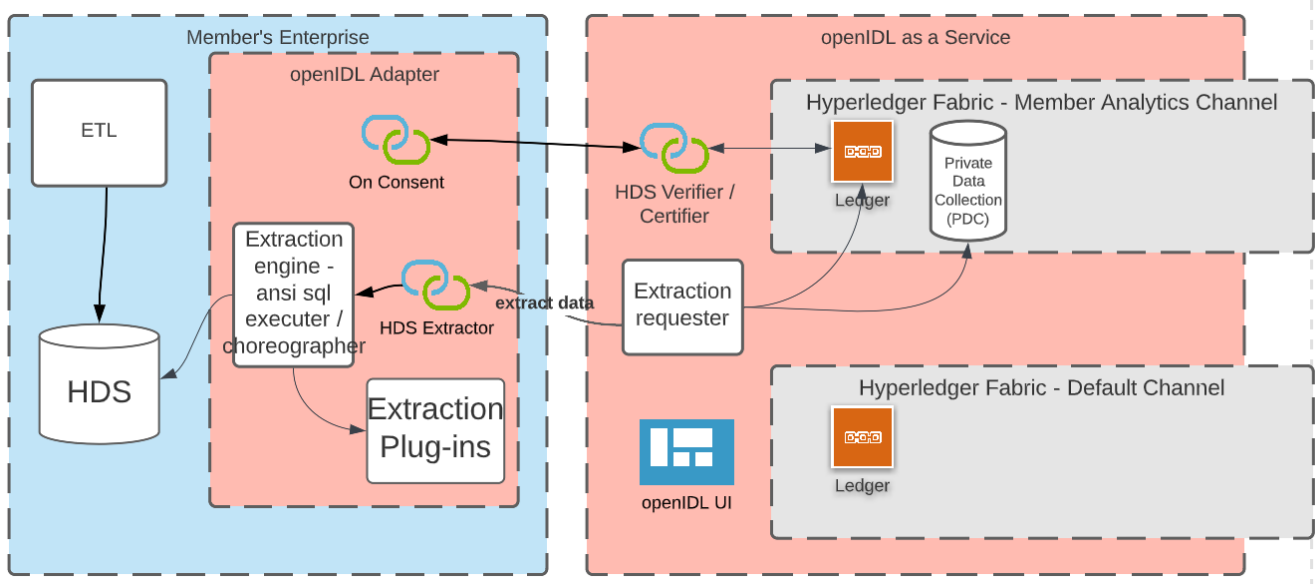
- There is a test-net and a main-net
- Governance Framework required for why (principles) AND how (mechanics) things get done
- States and prospective members can Pilot/POC via testnet?
- openIDL is running at least 2 nodes (CA and analytics) to operate NOC-like service
- Is there "one node architecture" for all or would a state have a different kind of node (please say no)
- Applications (stat reporting, etc)----openIDL Network----openIDL TestNet

Target Application Architecture

openIDL as a Service - High Level Application Architecture



Details of the Member



Components

- ETL
 - Member owned and operated software to process data and load into the HDS through the HDS loader api
- HDS
 - Member owned database that meets the expectations of the HDS extractor
- openIDL Adapter
 - Hosts openidl components needed to participate in openIDL
 - HDS Loader
 - Used by ETL to push data into HDS
 - Makes sure loaded data is registered with the network
 - HDS Extractor
 - Called when extraction ins run
 - Processes standard query to return data
 - Extraction Engine
 - Executes Queries
 - Runs scripts when needed
 - Extraction Plugins
 - Additional worker components for doing special math or accessing apis to correlate private dataa with external data
- openIDL as a Service
 - Hosted solution managing components required to connect to the blockchain and establish identity in the community
 - Hosts the UI for the data calls and statistical reporting
 - HDS Verifier / Certifier
 - makes sure the state of the member data is verified and shared on the network
 - Extraction Requester
 - sends extraction request to member
 - processes results and places into
 - HLF Analytics Channel
 - communicate with analytics node to share extracted data
 - HLF Default Channel
 - communicate with network to participate in stat reporting and data calls

Flows

- Carrier loads data
 - Carrier notifies that the data has been loaded
 - Must define a "package" of data that is now available
 - Who is notified? default channel or analytics channel? I think it must be the analytics channel.
 - Carrier responsible that the data has been verified
 - Some kind of verification utility to show data is verified?
 - Standards define what data "SLA" must meet
 - Does the extraction check verification?
 - Each "standard" has versions/levels that identify items
 - Each extraction declares what standard/level is required
 - separate discussion around validating data on the way in - can we normalize the rules for validation
- Carrier Consents
 - Consent is registered
 - No data moves at this time
- Data Call Comes Due
 - All nodes are notified of data call due
 - Nodes run extraction pattern
 - Result is placed into PDC
 - Data is replicated to the Analytics Node
 - Analytics node is notified that data call is due
 - Analytics node is given list of consents
 - Analytics node looks for data for each consentor
 - All data is combined into a single data store
 - Report processing commences
 - External data requests must be controlled by carrier and have a chance to consent or not

Discussion about the adapter

Flows

- Carrier loads data
 - Carrier notifies that the data has been loaded
 - Must define a "package" of data that is now available
 - Who is notified? default channel or analytics channel? I think it must be the analytics channel.
 - Carrier responsible that the data has been verified
 - Some kind of verification utility to show data is verified?
 - Standards define what data "SLA" must meet
 - Does the extraction check verification?
 - Each "standard" has versions/levels that identify items
 - Each extraction declares what standard/level is required
 - separate discussion around validating data on the way in - can we normalize the rules for validation
- Carrier Consents
 - Consent is registered
 - No data moves at this time
- Data Call Comes Due
 - All nodes are notified of data call due
 - Nodes run extraction pattern
 - Result is placed into PDC
 - Data is replicated to the Analytics Node
 - Analytics node is notified that data call is due
 - Analytics node is given list of consents
 - Analytics node looks for data for each consenter
 - All data is combined into a single data store
 - Report processing commences
 - External data requests must be controlled by carrier and have a chance to consent or not

Target Network Architecture

Target Data Architecture

See [Technical Considerations](#)

Target Technical Architecture

Digging into the integration between the hosted node and the carrier.

Feedback on Current Architecture and Implementation

See [this site](#) for feedback from Travelers based on deployment experience with the current architecture.