

openIDL - Architecture Definition Workspace

- Contributors
- Process
- Deliverables:
- Scenarios
 - Stat Report
 - Subscribe to Report (automate initiation & consent - assumes stat report)
 - Load Data / Assert Ready for Report
 - Generate Report
 - Reconciliation (Manual day1?)
 - Data Quality
 - Reconciliation (to do Mon 9/12 w/ SusanC)
 - Deliver Report
 - Auditability/Traceability
 - Notification
 - Data Call
 - Communications for resolving conflicts, etc.
 - Load Data
 - Create Data Call
 - Like Data Call
 - Issue Data Call
 - Subscription to Data Call
 - Consent to Data Call
 - Mature Data Call
 - Abandon Data Call
 - Clone Data Call
 - Deliver Report
- Access and Roles
- Application Components
- Data Sources, Sinks and Flows
- Decisions
- Tenets
 - Data
 - Non Functional Requirements (to be moved to requirements doc)
 - Notes:

Contributors

Initials	Contributor
DH	Dale Harris - Travelers
DR	David Reale - Travelers
SC	Susan Chudwick - Travelers
JM	James Madison - The Hartford
SK	Satish Kasala - The Hartford
KS	Ken Sayers - AAIS
PA	Peter Antley - AAIS
SB	Sean Bohan - openIDL / Linux Foundation
JB	Jeff Braswell - openIDL / Linux Foundation

Process

The Architecture Definition Workspace is where we as a community come together to work through the architecture for openIDL going forward. We take our experiences, combine them with inputs from the community and apply them against the scenarios of usage we have for openIDL. Below is a table of the phases and the expected outcomes of each.

Phase	Description	Outcome
Requirements	Define the requirements for one or more possible scenario for openIDL. In this case, we are focused on the stat reporting use case.	A set of requirements. openIDL - System Requirements Table (DaleH @ Travelers)

Define Scenarios	Define the scenarios sufficiently to gather ideas about the different steps. The scenarios will change over time as we dig into the details.	A few scenarios broken down into steps.
Brainstorming	Gather ideas from all participants for all the different steps in the scenarios	Detailed notes for each of the steps in the scenario(s)
Architecture Elaboration and Illustration	Consolidate notes and start defining architecture details. Network Architecture - different kinds of nodes and how they participate Application Architecture - structure of the functional components and their responsibilities Data Architecture - data flows and formats Technical Architecture - use of technologies to support the application	Diagrams for the different architectures <ul style="list-style-type: none"> ▪ block diagrams ▪ interaction diagrams Tenets <ul style="list-style-type: none"> ▪ strongly held beliefs / constraints on the implementation
Identify Spikes	From the elaboration phase, will come questions that require answers. Sometimes, answers come through research. Often, answers must come from spikes. Spikes are short, focused deep dive implementation activities that help identify the right solution for aspects of the system. The TSC must approve the spikes.	<ul style="list-style-type: none"> ▪ spikes defined ▪ spikes approved
Execute Spikes	Execute approved work to answer the question that required the spike.	Spike results documented.
Plan Implementation	With spikes completed, the team can finalize the design of the architecture and plan the implementation.	Implementation Plan
Implement	Implement the architecture per the plan.	Running network in approved architecture

Deliverables:

Scenarios

Stat Report

Subscribe to Report (automate initiation & consent - assumes stat report)

Define jurisdictional context/req (single or multi versions of same report)

How often it runs (report generation frequency)

- PA - avoid 13 lines and 13 reports per state per carrier - come up with a way to simplify distro of reports
- KS - make sure subsections covered, sections expanded - dont see anywhere we discuss data access

Extraction Details / Metadata

- KS - want an und of what data will be accessed by EP, when a report runs what data will be accesses, field by field, EP says all that, discussed whether or not EP being code is enough, said we want something more than just code to tell us whats being accessed
- PA - work with Auto Coverage Report is complex, want fields going across rows, and know what filtering criteria is, take these fields on this line over these dates
- KS - declarative approach to EP, declarative less writing JS code and more saying "fields accessed in EP" and a section "Aggregations", early on suggested approach (DR?) some sort of a "pre-processor" (was JB) - "this is an EP, know things in the EP, requires way to gen code from it
- PA - great idea, seeing 3 tiered thing: 1 explicitly biz level (top of data call - earner prem, most recent year), 2 more like lang agnostic metadata, 3 full implementation
- DH - really talking about "smart contract" will be - really the REQUEST, details -agree we want what fields, how info aggregated, how filtered, what date param used
- KS - agg rules, parts of the implementation detail, agg rules for data access, date, param, section on param for the subscription, look at parameters, considered higher level stuff, suggests other things, place to see those things, whether they drive implementation or not - agg, access, filtering - sections of pseudocode, can get out of sync with actual implementation, who puts that in there (REG or implementor of the EP?)
- DH - could be both
- PA - having success w/ Auto Coverage report due to DH doing all the mid-level data stuff, able to reproduce it, whoever is doing EP has to have a clear plan
- KS - basic model we have right now, for REG to put in prose and implementor to create map-reduce to make it, suggest 1 level deeper, more declarative sections like data access, maybe not field by field - data, filtering
- DH - want to have specific fields you are using, worry about "fishing expedition"
- KS - describe sections that need to be filled out as part of data call, specific (field names, actual aggs, pseudocode) - data access is mis

Outputs / Aggregation Rules

- DH - from HDS? sep from final report?
- KS should be same logically
- DH - output from HDS is agg data, will be anonymized?
- KS - not anon until combined in analytics node - what outputs? Data Points, how different from Aggregations - Agg: sum of the coverages by zipcode, what about outputs is different?
- DH - shouldn't be, outputs are aggregations
- KS - other point, consider when we combine things is there an expectation there is a functionality beyond that point, ex: the ND thing, does something on Analytics Node, compares against registered VINs, not just aggs from carriers, some possibility the report itself does work itself (compares to other data, etc.), find a way to describe that as well
- DH - what are you doing with the data when in the Analytics Node (AN)?
- KS - outputs on the EP - aggregations, the reduce - what do we want to know about the AGG itself? in code, do we need to come up with standard lang or just prose?
- DH - should not be computer code, (not tech people need to be able to read and und)
- KS - start with prose (human readable) and aspirational goal for some structure
- JB - whats requested to begin with, subset of EP
- KS - how accurately it can be expressed
- DH - then final report - or sep section,

Analytics Node Function (what are you gonna do with the data after combination?)

- KS - formatting final report
- DH - anonymizing data, deleting data once report created
- KS - req from customer to verify or keep evidence of that (deletion), prove data was removed, any core report logic (other than formatting) - anything like collab correlation data, has to be described here, ND example: comparison against registered VINs
- DH - running against 3rd party data, would want to know "what specifically are you matching on and expected results - what do you hope to get out of that 3rd party data" -
- KS - part of what we showed where we correlate with X data, some you want to do on the carrier node before you agg, uses data you don't want to share, might want this section about correlated data on both inside carrier node and outside node
- DH - on Extraction
- KS - talked agg and outputs, didn't talk about correlated data or third party data
- DH - think it would be more efficient for one party doing it rather than each doing their own
- KS - ex: address, dont want to share it, if we are all ok if it isn't exposed w/in Carrier node, every carrier would do it individually BEFORE aggregation
- DH - would want is within EP, already set up will go against 3rd party data, already set
- KS - hinting, this is not just code, some sort of API call avail, all carriers agree API is call-able, hornets nest, could be complicated, requires another component avail to EP
- JB - specify logically, some may have an API, logical not phys requirements
- KS - dont need to do it for stat reporting, not a near-term goal, good to know its avail -
- KS - has to make report available, put it somewhere to be accessed
- DH - reconciliation / qual check at this point before avail to be accessed (reasonability test)
- KS - manual thing, part of process that says someone gets eyeballs on before automatically released, chance for REG to look at report and say "ok", some Release by the requestor?
- PA - seems like want to have stat agent review before Reg sees it (esp with Stat Reports) - have to be able to have stat reporter access analytics node, thoughts?
- KS - logically making the report in some status avail based on permissions, requestor or stat agent can see pre-public release, these are request specific: same person doesn't have access across board only for that specific request, permissions based on that particular data call/stat report
- DH - before AN, part of EP, part of ask, do we want a mock up of what report will look like? so REG can say "this is the level of info I am looking for"
- KS - dry run of report, tells you what would leave your node but doesn't format
- DH - thinking ask rather than EP, "what is this report going to look like", not sure if REG will mock it up or intermediary, dont want them to receive and say "not what I wanted", format detail (what is in which columns vs rows) - what info are they actually looking for, simple, "written prem, by zipcode, cars color=red" - wants to know, as carrier, what is being asked for (format-wise), so EP gets into "how am I going to do that"
- KS - mock up of report or description ([Sean W. Bohan](#) - ask George and Eric to weigh in, join next Mon-Tues calls)
- DH - descrip fine, complicated= parts a, b, c, Peter's stat reports more complicated than "give me written prem by zip", complexity of request and output expected
- KS - what report am I getting?
- PA - sent Andy and Padma for next Monday
- SB - maybe put mock up on REG to put into the data call (not just why they want the data but what they expect from the report)
- KS - talked about carrier being able to see the results before shared, as a dry run, diff section, rolling in mind, does reg or requestor want similar funct at some point, some sort of dry-run capability for the requestor?
- JB - "heres what I want" and get data is problematic, peter's efforts, data I would like and what - role of the intermediary (who is actually creating data) - translates biz request
- KS - is there. aneed for REG to get more than consent/like but some sense report will give them what they need,
- DH - example: want by zip, don't write a lot of "red cars", do they want zeros or noting, or zips where they have red cars (gross ex)
- KS - is it sufficient to say we have made data calls quick enough, you dont get what you want you do it again
- JB - purpose of data call is to see whats there (sorry, we don't have red car + zip)
- KS - formulated question either dont get or get what you want
- JB - clarity of ask, goes back to how to define what you are asking for,
- KS - w/o debugger, have to figure out all code in advance before running, some support of what is avail to REQUESTOR as opposed to wide open report, do whatever you want
- JB - elastic search query form, what CAN be executed, focused topic we need to spend time on anyways

Roles and Permissions

- KS - identified the REG and/or stat agent, role of report reviewer can review report before published (diff than report approver) - merge into others - describing can be done by the REG but might also be done by the implementor, exact fields being returned - creating thing called data call or stat report, more detailed than when impelmentors gets in there, will find they need more, can update EP and configuration
- DH - collab between implementor and REG

UI/Interface

Extraction Pattern

Aggregation Rules

Messaging

Participation Criteria

Two Phase Consent

Data Path (from TRV to X to Y - where is the data going and for what purpose)

Development Process (extraction/code)

Testing

Auditability of data

Identify Report

- Who?
 - originally everyone participating in call from Regulators to Carriers to Intermediary (Participants)
- What is it (metadata)
 - Naming it
 - identifier
 - Requestor
 - type of input
 - generation source
 - line of business
 - what output should look like
 - explicit math for aggregation
 - Purpose of data (what being used for)
 - similar to what is captured on a data call
 - DR - stab at making a vers of this, idea of what it should be (ref reqs), see how it looks, whats missing, etc. - find gaps as opposed to trying to be complete **here** - for todays putpose some metadata along lines of reqs, would we do first req/draft of what it would look like, anything missing? (feels like reqs lite)
 - KS - info req section in reqs table, first iteration/sol will highlight gaps
 - SK - any existing samples of data calls/reqs? metadata assoc w/ request, match up, covered in the list?
 - PA and KS to discuss what will be shared, integrating w/ other depts, large list of data calls from other systems, working with ops teams to bring it together, high level looking to make big improvements on metadata and reqs
 - SK - date thinking couple (date of req, deadline data, expiration date)
 - KS - for a report these are the fields we fill in: (a la data dictionary definitions), what data call was intended to capture but inc all of details Dale pointed out, there is bridging vs pointing back to reqs, layout for report - THIS IS WHAT WE ARE TRYING TO DO/WHAT THIS REPORT IS
- Identify Stat Reporter

Identify who is subscribing

- Defining participants and role
 - Data Providers (Carriers)
 - Report Requestors (DOI)
 - Implementors (AAIS & etc.)
 - Stat Reporter (not necessarily same as implementor, general approved or cert stat reporter))
- producer of the data and the receiver of the data (source and sync/target)
- Carriers providing data, DOI creates request
- DH - Who are the participants? Carrier, Requestor, Intermediary (AAIS? other stat agents? those building extraction patterns and formatting report), implementor of report

Connecting Subscriber and Report

- Carriers and DOIs, want to capture that Carrier is data provider for a specific report and DOI is specific receiver for a report
- not data itself, more metadata about report, who getting specifically
- who get from / give to
- Notion of give-take between implementors and carriers and DOI about the intent
- Section about ability to communicate and improve to come to consensus it is the report we want
- Communicate about = user interface, carrier gets a chance to say "this one" and the ability to comment on report before implemented, and then implementation and then feedback to agree to
- Stat Reporting or data calls too? apply to both but focused on Stat Reporting and can bridge later date
- Reqs for stat reporting in handbook

Parameters of Subscription

- Specific to each report (loss dates, premium dates - other variables?)
- Some general to all reports
- Line of Business, Dates, Jurisdictions,
- Differences in report by state? Something Stat Reporting folks can answer
- Territory, Coverage? Diff reports same time period, grouping not a filter

Editing Subscription

- create/read/update/delete subscriptions
- self-service or governed thing?
- right now, sign up thru stat reporter for reports a Carrier wants run
- AAIS does it for them or on their own - something to be done
- Part of governance of openIDL (members, credentialing,)
- Audit log - auditability of subscriptions - managing subscriptions as part of openIDL - AAIS thing, funct of openIDL
- openIDL not a stat reporter - is there. a specific designation? AAIS is stat reporter working thru openIDL, if others join, they could be doing stat reporting on openIDL, there will be a "Stat Reporter" as intermediary,
- defines a seat in openIDL network (how to say "AAIS is doing X")
- DH - Trv joins openIDL, selects which stat agent they would do stat reporting through - could be report by report but guess all-or-nothing
- PA - not all or nothing as AAIS doesnt do all line (work with AAIS, then Verisk, ISO - can't be complete)
- DH - don't do MassCar and Texas w/ AAIS
- KS - identifying report, id stat reporter, per report detail (each report stat reported via AAIS), stat reporter per report or by line of business - per report connection covers all cases

Ending Subscription

- Delete
- Give subscription an end data (effective expiration on the subscription itself)
- lead time where AAIS or Carriers want to know if they are continuing or moving to new stat agent in openIDL
- Autorenewal

Load Data / Assert Ready for Report

080122

- ?? Facilitate semi-auto inquiries, metadata management scheme
- ?? Day 1 - PDF uploaded somewhere

080222

- KS - Homework, turn the above into arch statements or drawings/tenets, not in the requirements, feel little like requirements still, how do we add progress outside meetings?
- PA - like about reqs - key of what genre a req comes from and a unique ID - can we get a unique ID for these elemtns and a table, what refs what reqs, do homework
- KS - components or arch elements as oppsed to reqs - talking solutioning, trying to take reqs and apply to scenarios, break out into a set of arch statements for each component (LD1 assert up to a date on the data, LD2), then consolidate - AAIS team to org this doc into that format (due next Mon 8/8)
- SK - is the reqs based on discussions, done, next step to jump into solution design and arch?
- PA - jumping in makes sense, int in 2 things: interactions of network and HDS, hard to think of how data load happens w/o knowing target
- SK - deliberated reqs, organized, next step not to re-deliberate reqs but to solidify the arch or at least start on it NOT reclassifying this into another set of reqs
- KS - avoid that, these are functional areas sys needs to support, not get to details of tech for a while, all the ideas that need to hold true, made progress in open ended way
- JB - top down/bottom up - some sense going back to phases of the sys we started with, keep in mind arch we are dealing with network, not centralized data center, keep in mind org funct around aspects of that network, reflect some of the initial thinking arch needs to be supported, what are the elements for producers, processors, receivers of data
- KS - need to be tolerant of chaos, in between meetings remove chaos and refine, brainstormer, raw material
- PA - outlined our big boxes?
- KS - Data formats? Stat plan

Define Format

- What is the data? Glossary or definition? What is being loaded (stat report well-defined)
- Assumption - stat plan transactional data, metadata is handled by spec docs as yet to be written
- Data existing in HDS, what schema says, there to fulfill stat report, this is just data thats there, period and quant/qual of data designed to do stat report, for this purpose just a database
- Minimal data catalog - whats the latest, define whats there (not stat report per se), whats in there is determined thru the funct described (time period, #, etc.) - diff between schema for a db and querying it, format for what could be in there
- Minimal form of data catalog - info about whats in the data
- Schema is set but might evolve - "type of data loaded" - could say "not making assertions this data is good for a specific data call but to the best of our ability it is good to X date"
- KS - must be able to develop report from extracted data

Load Function

- Deeper in process of data you have getting into openIDL, details of managing

- Process, raw data in carrier DB, turned into some "load candidate", proposed to be loaded into system, needs to go thru edit package
- DH - before HDS?
- KS - from your raw data to accepted HDS data (load function) and will inc other pieces like edit package
- DH - internal loading to the carrier
- KS - carrier resp for turning data into intake format (stat plan)
- DR - req for "heres what data should look like to be ingested" -
- data model - stat plan day 1, day 2... data model
- KS - process of taking it in, do work to make more workable in the middle, dont commit to saying "what you put in front end is exactly what ends up in HDS" - right now not putting it exactly, turning it into at least a diff syntax and never will be 1:1, semantically close,
- DH - more sense for decoding
- KS - load funct part of openIDL, carrier entry point, what carrier putting into load func is stat plan, THEN run thru edit package, review/edit (a la SDMA), "go" and then pushed thru HDS - carrier not doing transform, carrier loading thru UI (SDMA), may even be SDMA (repurposed) to load HDS at end of day
- DH - HDS w/in carrier node?
- KS - adapter package - need to support 1 keeping data in carrier world and dont want everyone to write their own edit package and load process, agree on somethign that runs in your world that is lightweight edit package
- DR - simplify, essentially a data model, how does it lie in HDS, may or may not be a different input data model that is whats loaded, once in HDS and "loaded" should conform and have any edit packages already run on it, all running on carrier side, dont want it going out and back - caveat, edit packages are shallow tests, not looking at rollup or reconciliations, "is it in the format intended?"
- KS - row by row edits, not across rows, had to have x w/o errors, etc. - syntactical and internal, "if you pick this loss record cant have a premium"
- DR - sanity checks and housekeeping
- after edit, push to HDS (tbd format, close to stat plan day 1)
- PA - extensibility, adding more to end of stat plan in the future

Transform

- whatever we need, might do some small decoding, def turn in from flat text to TBD (database model in HDS)
- normalization? some light transformation in the beginning
- assumes not collapsing records, like stat plan same level of granularity every record input is record in HDS (time being)? 1:1
- decoding has reference data to lookup

Edit Package

- Big (all of SDMA)
- when we discuss loading data is it already edited and run thru FDMA rulebase and good to go or raw untested data
- ASSUMING thru the edit
- Can tell how goods the data and through when
- pointer to SDMA functionality:
- PA - SDMA - business level rules, large manual process for reconciliation BEFORE turning in reports (today), business and schema testing (does data match rules and schema? cross field edits)
- KS - cross field edits - loss records, diff coverages, do have a publishable set of 1000s of rules if used SDMA will just work, just plug SDMA in - can and has been pulled out, proved it could be done, rules could be run as an ETL process - havent done, back and forth and fixing of records not part of it, run the rules as ETL process

Data Attestation

- do we have an automated way to attest to data?
- cannot attest completeness
- Provide data attestation function. Carrier attests to data for a particular date. Attestation parameters? Data attested, time frame (last data of complete transactional data), level of data (must define for attestation: like stat reporting day 0, 1, 2)
- different attestation for claims and premium data
- Must have data formats / levels defined for attestation
- on extraction - check last attested date. If last attested date meets requirement of data call.
- attesting to the quality of the data (meets 5% error constraint for data from x to y dates)

Raw Notes

- Have it or don't by time period
- Assumption - run report, everyone is always up to date with data, loading thru stat plan, data has been fixed in edit process, ask for 2021 data its there
- Automated query cant tell if data is there, may have transax that haven't processed, dont know complete until someone says complete
- Never in position to say "complete" due to late transax
- If someone queries data on Dec 31, midday. not complete - transax occur that day but get loaded Jan 3 - never a time where it is "COMPLETE"
- Time complete = when requested - 2 ways - 1 whenever Trav writes data, "data is good as of X date" metadata attached, Trav writes business rules for that date, OR business logic on extract "as long as date is one day earlier" = data valid as of transax written
- Manual insertion - might not put more data in there, assume complete as of this date
- Making req on Dec 31, may not have Dec data in there (might be Nov as of Dec 31)
- Request itself - I have to have data up to this date - every query will have diff param, data it wants, cant say "I have data for all purposes as of this date"
- 2 dates: 12/31 load date and the effective date of information (thru Nov 30)
- Point - could use metadata about insertion OR the actual data, could use one, both or either
- Data bi-temporal, need both dates, could do both or either, could say if Trv wrote data on Jan 3, assumption all thru 12/31 is good
- May not be valid, mistake in a load, errors back and fixing it - need to assert MANUYALLY the data is complete as of a cert time
- 3-4 days to load a months data, at the end of the job, some assertion as to when data is complete
- most likely as this gets implemented it will be a job that does the loading, not someone attesting to data as of this date -where manual attestation becomes less valuable overe time
- as loads written (biz rule, etc.) If we load on X date it is valid - X weeks, business rule, not manual attestation - maybe using last transax date is just as good - if Dec 31 is last tranx date, not valid yet - if Dec 31 is last transax date then Jan 1

- Data for last year - build into system you cant have that for a month
- Start with MANUAL attestation and move towards automated
- Data thru edit and used for SR, data trailing by 2 years
- doesn't need to be trailing
- submission deadline to get data in within 2 years then reconciliation, these reports are trailing - uncomfortable with tis constraint
- our ? is the data good, are we running up to this end date, not so much about initial transax than claims process
- May have report that wants 2021 data in 2023 bug 2021 data updated in 2022
- Attestation is rolling, constantly changing, edit package and sdma is not reconciliatioj it is business logic - doesnt have to be trailing
- As loading data, whats the last date loaded, attestation date
- sticky - go back x years a report might want, not sure you can attest to
- decoupling attestation from a given report (data current as of x date),
- everything up to the date my attestation is up to date in the system
- "Data is good through x date" not attesting to period
- Monkey Wrench: Policy data, our data is good as of Mar 2022 all 2021 data is up to date BUT Loss (incurred and paid) could go 10 years into future
- some should be Biz Logic built into extract pattern - saying in HDS< good to what we know as of this date, not saying complete but "good to what we know" - if we want to dome somethign with EP, "I will only use data greater than X months old as policy evolves
- Loss exposure - all losses resolved, 10 years ahead of date of assertion, as of this date go back 10 years
- decouple this from any specific data call or stat report - on the report writer
- 2 assertion dates - one for policy vs one for claim
- not saying good complete data, saying accurate to best of knowledge at date x
- only thing changing is loss side
- saying data is accurate to this point in time, as of this date we dont have any claim transax on this policy as of this date
- adding "comfort level" to extraction? - when you req data you will not req for policies in last 5 years - but if i am eric, wants to und market, cares about attestation I can give in March

Exception Handling in LOADING

- Account for exception processing
 - What is an exception?
 - PA - loss & premium records, putting stat plan in JSON, older data didn't ask for VIN, some data fields optional
 - KS - exceptions can be expected, capturing & managing situations to be dealt with, not "happy path", need to have error codes and remediation steps, documentation for what they all mean and what to do about them (SDMA has internal to edit package) - things like "cant get it in edit package b/c file not correct", etc. - standard way of notifying exceptions throughout system, consistent, exception received and what to do about it
 - PA - ETL stuff, exceptions based on S&S topics, whats the generalize way to handle? or specific except cases?
 - KS - arch needs way to report and document and address/remediate exceptions (consistent, notifying, dealing)
 - PA - options:
 - messaging format,
 - db keeping log of all messages
 - hybrid approach of both
 - KS - immediate feedback and non-sequential (messaging or notification feedback)
 - JB - data loading transfer of data or into HDS?
 - KS - data loading starts with intake file in current statplan format, ends when data in HDS
 - JB - lot of exceptions local to this process loading data, reported to anyone or resolved or level of implementation of who is reporting data,
 - KS - some user interface, allows you to load a file and provide feedback, but a lot is asynchronous, no feedback from UI
 - JB - gen approach to be shared across
 - KS - consistent way to handle across system (sync/asynch, UI vs notification)
 - PA - 2 lambda funct loaded in, 2 S&S topics (1 topic per lambda), seems like nice granular feedback, as we get more lambdas throughout node would be unweildy, master topic to subscribe to resources
 - KS - too deep for now
 - PA - one general exception thread or thing to subscribe to, get large amount of exceptions as opposed to making the QA team to ind subscribe to each resource (some kind of groupings?) - lot of components throwing exceptions and dont want to sub to each component
 - KS - do we want to audit exceptions? Likes/Unlikes, Consents, etc. - are there exceptions we want that to be captured on ledger or somewhere to be audited later?
 - PA - consent to data call and dont have data required that should be recorded/captured/to chain, etc. (consented to participate and no data)
 - KS - funct in exception handling, getting close to NFR (disaster recovery, continuity, reacting to scalability, etc.) need to get there at some point - digging deeper, specific exceptions will have different decisions

Metrics/Reporting (process)

- KS - Within Data Loading, Feedback (status, presence) on what is loaded, # of records, total premiums
- DH - by line of business, record counts, premiums, paid losses, incurred losses, significant metrics and claim count
- KS - SDMA good example, same as SDMA provides (DH says yes)
- DH - maintaining 5% tolerance ? then we need to know if within tolerance AND where they are relative to tolerance
- KS - tolerance here is a load thing - amount puot into system (5%), but you can load pieces, certain amounts, going back to report based tolerance, cant determine **here** - data loaded for last month or quarter - some time period of loaded data calculating tolerance for AND could be multiple loads
- DH - requirement could be "must be in tolerance" and combination of loads must be within tolerance
- KS - want to go back and check tolerance across a time period
- DH - or a counter that looks at aggregate tolerance over periods of time (calendar year? quarter? month?) - could be continuous
- KS - and know when to do that counter - editing before gets to HDS, clarify? should not get error records in HDS
- DH - you might, if within tolerance? theoretically you wouldn't but practically you will - always have bad zip codes or edits, but minimal in scope and a resource constraint to fix every single record
- KS - if i had a bad zip, why in HDS being queried? might be valid in other ways but is that how we want to work?
- DH - do you delete whole record?
- KS - not based on zip CAN use but based on zip CANT? pretty complicated - what does SDMA do? not passing thru error records to data lake

- PA - to follow up with Andy, gut feeling still passing, not eliminating rows
- DH - deleting rows = out of balance
- DR - how do we know wrong but not know right zip code?
- PA - situation where zipcode featured in wrong state (NC in CT) trigger an error event
- KS - are there some errors we let through b/c a report WONT break? Ify
- DR - some automated loading tooling that would prevent from happening, pre-edit package, Carriers may need to inc downstream, if we know it is wrong, it might be ok - HDS is not the one use case (know stat reporting ok), might not be used or below threshold, but if there is a different use case cant make that assumption errors are ok - goal accurate to best of knowledge, if obvs wrong attempt to fix it
- DH - logistical nightmare, errors cant fix b/c not getting data, valid limit but only on a few records wont pass edit - maybe on HDS a flag "this record did/did not pass edit"
- KS - IBM thought about this, initially built braindead vers of transform, loading errors into record, put into HDS, and could have records that were self-descriptive of what errors and then up to Extract Pattern to decide
- DR - if something fails edit package, missing or wrong, could make all missing, all null, if null pattern doesn't have to know - if a record is accurate to one limit field and set to null, wont pass edit package and parser will decide "cant use" w/o metadata assoc with it - cleaner than playing game of "data is wrong but lets say whats wrong so others can use it" - too complicated
- KS - "null" is valuable in some cases, error vs value situation
- DR - trying to put logic on data call extract vs db, isn't always an error and data call should decide if it needs it, shouldnt matter to data call, why matter if not there - put logic onto extract pattern than HDS - wrong or missing
- JB - conflicts b/w "state and zipcode" - needs some resolution to fix something (inconsistency)
- KS - hybrid approach - load errors into records and ignore if they dont matter, could bloat things
- DR - build funct on bloat and bloat becomes feature
- PA pass or not-pass?
- KS - loss of fidelity, all use cases for reading data not known, one is fine other is bad, need extract pattern, if null not use record
- PA - minimum amt of edit package everything should pass?
- KS - no, doesn't sound like it -
- DR - didn't pass edit package could work in cert cases, but zip code is perfect ex: which is right? know use cases in stat report, up to 5% iffy, would work but doesnt scale with other use cases, flags help at record level only, a use case could say "ill accept b/ it is 5%" - need to know what error is
- KS - we can have arch bloat and as we learn if this bloat has value we can remove or not - shrink arch to assume null, disciplined to revisit this (record-level errors),
- DR - record level error, limit to that, boolean, simple, just do record leveling and no further is reasonable approach (stat reporing on day 1, allows others to build more funct on openDL and allows carriers to build knowing it is there)
- Dh - some errors more imp than others, zipcode not important for some reports, nature of error, dont know higher level of error coding, if you need to get too geo specific, this record wont work but looking at general, then perhaps it might - rather than not using record at all it makes it more complicated
- DR - use cases, for other data calls maybe its ok for what that datq call is doing, 90% of records saying "no issues" thats a call for data call to make, put complexity on writers, maybe moot but on day1, let people know some aren't quite perfect - there will be more fine-grained for the carriers, will see more clearly results of the edit package - keep to simple flag
- KS - simple approach now doesn't obviate the use of the more fine-grain later, if we find it is imp to know the error is the zipcode and x doesn't want to use it we can go back to data and add errors later
- DR - carrier decision, hold it as data req for HDS, later req could be added (optional or mandatory) "you could add metadata about exceptions" and that might obviate need for remediation - DH aware - this is rep of downstream system of record
- DH - systemic issues we fix, these are one-off
- DR - hard, one-off issues = one-off corrections (data entry error) and not directly impacting admin of policy or claims, hard sell to always clean up, dont want HDS out of sync with system of record and dont want SoR spammed with error cleanup and affectig ops
- KS - threshold? everything goes through even with error can enter HDS?
- DR - ideal world depends on data users (whats acceptable) but we could say at minimum whats most imp
- DH - functiioon where run through edit package and where we can fix some errors (what they can) to get within tolerance, run thru edit package to pass edits
- KS - not sys of record, result of fixing stuff in source systems and getting them out, have to get into and then back o ut of source systems
- DH - we make edit changes in SDMA today, not going back to source system
- KS - out of sync between SDMA and source system, hds doesn't match
- DH - edit correction, then yes not match source systems
- DR - get in trouble, modifying HDS for one deliverable, might be ok, but harder to use HDS as source of truth, not accurate compared to source systems, if you need to do it that way tread HDS as source of truth and push corrections downstream - HDS accurate rep of downstream systems, but if most corrections/edits not in source of truth makes sense to push edit funct into HDS and there can be edits on data call or extract pattern - could execute extractpattern ourselves to test - if EP exists, references edit package, doesn't mean running EP in parallel as needed to correct things, correcting NOT at the source but in this method, simplifies architecture, if HDS is dumb source of truth, complexity goes to EP, say EP is stat report and says "6% problems, out of tolerance", if we know and can run that see out of tolerance and fix some things in ephemeral data store to move INTO tolerance, point to alt locations in EP
- KS - dont those locations needed by other EPs? loss report that needs those fixes?
- JB - some errors fix at the source, biz decision for the carrier
- DR - fixed today might be source of truth next week, dont want to keep track of error fixes you did, some might have been fixed in changes to the policy, dont need to fix downstream anymore, make tooling correct to see tolerance and adjustments thru UI, whatever gives flexibility and business decision to see why OR really 1-offs and no systemic fix or fix wont happen in a year - vis to see problems and made decision to change - simplifies getting right, HDS reps source of truth and working on marquee use case (stat reporting), slightly more complex with interim edited data store (div-ing vs full replica)
- KS - HDS in sync with systems and overlay of edits visible to EPs, but not durable not permanent - exists as a temp thing
- JB - diff patterns, diff things, replicating corrections?
- DR - complexity of EP to solve
- JB - record of what errors were in HDS could choose to deal or not, having a copy of HDS knows records with errors, maybe/maybe not addressed
- DR - sometimes you need to make changes to get to threshold, just recording doesn't help, changes in pance now HDS doesn't rep core systems (state maangement prob of ANOTHER source of truth)
- JB - if errors that significant must fix?
- DH - might be system issue that needs to get fixed
- KS - do a record level (fix-record), discuss later, better to have record have a fixed version, here's what we got and the fixes to it, simple to see, diff way to do what dr was describing
- DR - could do "foreign key" to fixed record, db arch instead of flag

- DH fix record ONCE
- DR could get lots and lots of those, based on EP, 13 columns of foreign keys, could spiral, some state management, source of truth changed and now core record changed to do we need to link edit package...
- JB if small p% has errors,
- DR - depends in ephemeral or not, tooling to fix errors automatically, read data, modify, fix, kill
- KS - ephemerality - how ephemeral or length of time ephemeral
- DR - time prob - 2 months doesnt work, 2 min 2 days maybe
- KS - i have fixes to rec, will fix downstream systems, do we need a re-run?
- DR - seperation of concerns is important, logic data, look for specific field, linked table - now messing with core system to deal with corner cases
- KS - not corner case, know records will be messed up and fixed at HDS, things exist
- DR - how many times will we do edits outside Sys of Rec for data calls
- DH - dont, no edit package on data calls
- DR - never on data calls?
- KS - on the load
- DH - look for reasonability, dont have edit package to make sure interrelationship of edits is sound
- KS - this system, edit package on all data you load
- DR - go two ways, proliferation of edit records, pointing to diff versions of records, could hold like "ONE only ONE edit record, option of querying that or not depending on use case, dep on database design could be problematic
- KS - could design away with "views", if have a view

Data Catalog (meta data about whats in the db - some notion of whats available currently)

History Requirements

- rollout period - not keeping 20 years of data

Schema Migration/Evolution

Create Report Request (Configuration)

Looks a lot like "Identify Report"

Define jurisdictional context/req (single or multi versions of same report)

How often it runs

Data Accessed

Outputs

Roles and Permissions

UI/Interface

Extraction Pattern

Aggregation Rules

Messaging

Participation Criteria

Two Phase Consent

Data Path (from TRV to X to Y - where is the data going and for what purpose)

Development Process (extraction/code)

Testing

Auditability of data

Generate Report

Rule Base for each report

Extract Data (will involve aggregation)

Transmit Data (from HDS to analytics node)

Combine Data (various sources)

Consolidate Data (at the report level)

Traceability

Format the output

Validate against participation criteria (vs report config)

Exception Processing

Messaging

Generate Report

Auditability/Traceability

Reconciliation (Manual day1?)

- SC - Reconciling statistical data (in HDS) to the financials, financials of company would be NAIC reported financials, final check that stats matched financials and if they didn't: why
- KS - time to get it right
- SC - company TRV size takes a while due to volume, try to do it on the 1/4, time depending on when it is due, did come up with Regs, today a lot of stat agents dont ask for reconciliation until end of following year, question: how soon could you get an auto report and one said "June" and she said "does that inc reconciliation?" stat agent said "we have 'edits' that give them comfort data is reasonable, but ult how do you know if you have everything if you cant reconcile back to something"
- KS - reconcile data to x numbers, ask openIDL to recon against these #s or give the data
- SC - down to the role of openIDL and who is taking responsibility, today Stat Agents take resp the data is valid, the role of stat agent is to add level of review/authenticity
- KS - stat agents resp for reconciliation
- SC - NAIC info is public, openIDL or stat agent could get independently
- KS - when and can it enter HDS un-reconciled? it has to be, attestation is some report that requires it to reconcile before run report on that data - if we do stat reporting, stat report wont do recon, will have to be reconciled before, 50 stat reports wont all reconcile, "assume it has been reconciled" or it couldn't be run
- PA - trying to have data in a correct form available, traling by a 1/4, best of carriers ability its avail, traling by 1/4
- KS - less than 5% errors and reconciled by financials
- PA - do we have to do reconciliation? VERisk said they could do some reporting faster than a year (didn't need yellpw book)
- DH - you need to do reconciliation as part of stat handbook
- SC - brought up reconciliation, verisk said "have other ways to show data is accurate", could report \$10MM of auto but you have \$15MM, under 5% and reasonable - not sure what ways verisk has
- JB - differences between financial report vs data from stat reports, reconciliation try to explain that difference - explanation to what difference are - recon a explanation of differences (small) - info in financial report is high level,
- PA - stuff coded from one value to a diff value, sometimes can be very large, not always small reconciliations
- SC - mult reasons for errors (missing data, bad input, fat fingers) then reconciliation there are reasons you would differ, NY has "free trade zone credit", dont report statistically, one entity it is 1/2 of auto premium in NY, big but a reason for it
- PA - how do we see reconciliation happening in a mature state, what is AAIS' role in it, right now very inv w/ carriers, helping to bring together, set up in a mature state so it is "self-service", AAIS not too involved, highlights where there is a problem or do we need stat agent heavily involved
- KS - dont make that decision yet, reconciliation process supported by openIDL, get data from NAIC, comms process for carriers to explain why, process that lets carriers attest to data being reconciled
- PA - if carrier had table w/ yellowbook info for 1/4s and line, could have extract pattern comparing HDS to stat records to yellow book - where do we want to keep yellowbook info
- KS - public data - maybe access thru API, may need a spike to see whats the best approach
- PA - as of now, aside from yellowbook, haven't needed to go out of network to answer question, can load yellowbook data in, keep inside network, some kind of external source
- KS - now? where?
- PA - NAIC
- KS - phys? API, file?
- PA - 10 page CSV, loaded quarterly, all 2k carriers on it, all lines, metrics for said lines
- KS - worth digging into, find best approach (align to reconcile data, when avail)
- PA - will run down this week
- SC - financial data reported quarterly to NAIC
- PA - reached out to NAIC, this year have quarterly data

Data Quality

Extraction error detection & handling

- Everything needs to be edit-able
- Fixes don't happen in current month (monthly correcting and then moving on)
- Latency of error correction could be a year
- need to make sure we have facility to capture corrections made while NOT bastardizing HDS

- internal or architectural? DR is aware
- SC - Errors:
 - missing information (on record provided)
 - current environment vs future
 - today - flat file from upstream, flat file submitted with missing limit, info passed to AAIS, flagged by AAIS, returned to carrier (can see instantly by state), these 2 states need fix made, go into SDMA to make fix then submit, AAIS approves, loaded by AAIS
 - it had already gone thru the edit
 - DH - load into SDMA, not approved yet, Susan makes corrections, goes thru edit again once Susan made corrections (see right away if fix worked), if in tolerance it is "approved" by AAIS
 - PA - doing upload to SDMA, staging area, AAIS not running load until it is approved (edit package engaged)
 - SC - loading it to AAIS system, told to fix errors, fixes, then "officially submitting" and AAIS "approves"
 - PA - can't go to HDS until "approved"
 - DH - where within process is edit package? where is facility to correct the errors, if HDS is supposed to be matching to source systems, then we shouldn't be making changes to HDS for other purposes beyond StatReporting - decision in ArchWG - how handle error corrections and fidelity of HDS
 - PA - direction, making update, go about making corrections of data already inside HDS, first example - data before HDS, different Error type
 - JB - case Dale mentioned, HDS is out of sync with source system, SS has error, needs time to fix, copies of DB with errors to be corrected - would suggest errors corrected get corrected in HDS but a log to inform source system of corrections as made - instead of lots of copies of collected data
 - JM - crossing boundary - doesn't care what carriers do - where do we stop caring - only thing, HDS has to be right, up to the carrier how they get it right
 - JB -yes but instead of making fix and a copy of DB it seems it should be fixed in HDS
 - SC - internal issue, AAIS needs to edit data, thats their job , if they say "2 errors" and they get fixed she says "done" and pushes to HDS - conflict with source system is something SHE deals with
 - JB transferred to AAIS for edit checks,
 - PA - held before data lake until adter corrected
 - JM - cant occur until content is in it
 - PA - edit pre ETL
 - JB - do it 2x, if you correct HDS need to run edit in that environ
 - PA - how do we have chick-egg issue
 - JM - policy vs implementation ? - HDS is great cutoff point, everthing inside, up to the carrier to get it right in HDS - BUT Edits tell you whats right - carrier accountable up to HDS, if accountable on the carrier side and can verify before HDS, do the edits, send to HDS - what if I say "right, but edit stuff can't run iutnil other side - alrwyd loaded to HDS - now what do I do? - accountability? where run edits is key question
 - PA - edit package run today, run on etl on load, no knowl on load - 2nd part AAIS does reconciliation after, sometimes errors arise
 - error type 1 - pre HDS , edit package fails on load - but what if loaded in HDS what is the recon process and what the process for that
 - JB - financial types of reconciliation
 - PA - yellowbook #s, compare #s submitted vs financial #s and due to granularity things come out wrong, financial reconciliation before stat reporting
 - JB - 1x year vs monthly
 - JM - reconciling financials? where?
 - SC - public info
 - PA - reach out to team with gap analysis, grey areas in codeing vs what they have, validate where /why numbers are off
 - SC - those arent errors, do reconcile, out of process doesnt become errors, differences and reasons why page 14 doesn't match - but NOT errors
 - PA - validity AAIS gets turning in reports on carriers - not only passed edit package but biz data matches fin data and a reason if it doesn't - why states listen to AAIS, how are we ensuring we are doing stuff correctly
 - JB - diff record exception
 - JM - annual value add - edits? HDS needs two stage?
 - think its right but flag then run edits and get "ok/not ok" - question - who runs the eidts? in principle edits run on anyone centralized db
 - JB - copy of edits made avail to all
 - DH - one body resp for edits, not every single carrier
 - JM - you put data in HDS, centralized code runs on all dbs, puot into HDS in some manner "this is not fully approved/edited" and decision: edit in place or is it a 2-stage thing?
 - SC - even if every carrier ran edit package themselfess, ult AAIS HAS TO RUN EDIT PACKAGE - resp lies with statistical reporting partner
 - PA - extract patterns to un T/F that a package was run - do test on clean or dirty data
 - JM - edits form of extractPattern, is it sufficient if it checks all the data
 - PA - regulator!
 - JM - need feedback - run edits, if answer wrong, accountability to get it right
 - phys load or set flags
 - PA - should be running edits before load,
 - JM - WHERE? edits have to be consistent lang, thing needs to be well-defined structure
 - PA - rules engine, java, repackage rules engine as step in process going thru load (pass/no pass)
 - JM - engine has to run against well define struct - b/c our data runs against well defines struct, now you are in HDS? put it into well def struct to run the rule that is the post-edit vers of that structure
 - PA - messaging format of HDS - stat plan, objects, run edit package against that
 - JM - stat loading and knowl, if run edits against that, once passes - put it somewhere else or flag it - 2 concepts pre and post - saying to all carriers it needs to be PRE data but it has to have a shape - HDS? JM perceives when you demand "struct in diff way" and sees it as HDS
 - PA - diff pipleline but sees why it is outside of HDS
 - JB - data standard for saying how data will be considered, keep in mind dist arch, AAIS can't run anything on db at carrier - raw, wont be sent to AAIS
 - PA - collections of stat records, running rules against them, if HDS is stat plan JSONified, run EPs, passed valdiation and legit extract

- JM - HDS is JSONified stat stuff, edits, things all can see are ALL HDS in his mind - if prescribing shape b/c edits won't work, first place carriers have to do that
- PA - pipeline A before HDS, where prescribed the data hits first
- JM - widget shape here, then ep - prescribing shape, set of edits then HDS - pipeline A is a prescribed shape, do whatever it takes to get it right, once edit passed drop into HDS
- DH - wants to have DavidR weight in
- PA - Pipeline A (infra before HDS), need to pull rules engine, before how much do we want to control creation? JM talking about HDS being a larger thing, where does the balloon around openIDL begin? PipelineA is infra, carrier does all before? will still design load up to plugin
- JB - think of pipeline A as data format
- PA - wont process and give feedback
- JB - need data format to be standard to run rules against, gives flexibility to reconstruct design with same format (transit from flat file to whatever).
- PA - docker image with initial process? where is the official inbound point of openIDL community vs carrier
- JM - one step at a time - HDS in the dark (far right), run extract patterns on - before HDS has to pass edits - edits need to be centrally maintained, id DRules expecting something - pipeline A - already in that shape - sayig to carriers, prescribe format of HDS, to be right prescribe the edits, has to hit a prescribe shape here - carrier can do whatever to get into that form, that form is prescribed, java thing, json, all prescriptive, no flexibility
- PA HDS, can write queries against, layering other things not HDS
- JM - centralized group do edits, carriers get it into that shape, must be part of standard of stuff to be prescribed
- PA - meat of Drules, lot of it is testing stat plan, start ingesting as json, checking positionality
- JM - thou shalt not load HDS until edits passed, edits managed, approved format, carrier must get data into shape - reload until passed and THEN move to HDS
- PA - can we have a bucket, fire lambdas against it, won't move to secondary bucket until passes
- DH - suppose use HDS for other things, communicating with reinsurers, something outside of stat reporting, now that HDS not necessarily reflects source systems
- JB - source consistent, take time to get corrected, logically - more correct versus HDS
- PA - HDS more right than source system
- JB - fixed at HDS but not at source
- JM - policy, carrier accountability, edit finds something wrong, iterates on changes, if it takes 6 months to get back to source, for next 6 months other reports don't reconcile - accountability in governance statement "if you find an error you are accountable to reconcile"
- JB - consolidated data in HDS for other purposes, if corrections were in HDS the right place to do it
- JM - better that doesn't line up is wrong
- JB - log for where / when changes done
- JM - carrier accountability - more right data - where is accountability to carrier? whatever it takes upstream - tell us changes you made requirement - log that says "to get this loaded here are 7 edits" - accountability to make it transparent
- PA - meta on each row with last update date and what changed
- BH - if systems don't reconcile - BAD - what else are we doing with it? problem to be solved, may be a log, sounds painful
- SC - reality - keep a log today (she does of every change made) - most cases data SC didn't get on her file (stat file) - is it really diff from source system? she didn't get it on her file due to mapping upstream -know zip code is wrong or vin is wrong don't change things in her file or tell source system theres too many (agents inputting) - ok if under 5%
- JM - practical question - do edits - syntactically and semantically: find alpha, don't know if someone mistyped VIN, but no idea T/F in real world - HOW RIGOROUS DO EDITS NEED TO BE? - even if edits flag error? can we accept it?
- SC - happens all the time, might get edit "limit on policy is \$1MM and you got something else - not an error"
- JM - 2 levels of edits? showstopped (dead) and one we accept
- SC - wont ignore fact error was received, will go and looks "did I have the right limit" - edits help und if there is a problem - is it internal edits ?
- JM - what is the purpose of an edit? don't edit more than you have to - what is the purpose in this context - all sorts of mech for internal correction - don't edit more than you need to without purpose - some things you have to fix, principle: only put in edits b/c hardcore reason to do it (not just clean data"
- JB - work to be done - application and analysis and insight, not policy-level corrections
- JM - do edits have levels? severity of error (which means will it be addressed)
- JB - sanity check errors vs record format errors - can and will catch but WHERE in process
- DH - gut check for AAIS as stat agent on how rigorous they need to be
- JM - levels - showstopping and scary and "oughta check"
- JB - accuracy in general (THRESHOLD)
- JM - confidence scores from address cleansers -
 - showstoppers (break system)
 - competency score (".7 good enough? yaaay")
- JB - data quality scores, pick battles
- SC - basic: does every field get a val - current and future, if not ABCD - if that field is filled? if so whats in there, nebulous - stat agents bear resp of "data is reasonable", know it is not garbage, how much has to be "good" - what does "good" mean (every field filled w/ reasonable value"
- JM - mTable that does this - argument - for every field "type, table, range, = score"
- SC, come across something, didn't meet the threshold, kick it back?
- JB - levels determine response
- JM - governance ? - value, string, etc. - don't measure if you aren't gonna govern it - if you are gonna put a rule in there, must have governance polity - arch has to provide for edit layer and series of thresholds to get a score and governance policies by score
- JM - pass/fail and scoring
- PA - extra metadata for user queries
- KS - "close out the quarter", might go back and add to it - close out means can't change later, cant put in records that apply in that 1/4 later if you "close out" - do we need some way of sensing we are opening up a 1/4 again and need to re-assert it is ok?
- SC - have had situations where we discovered issue and "need to fix year" for a line or situation, b/c today timeline is so stretched out - takes too long - go back and adjust the year b/c the reports hadn't been issued - recent sit in MASS where they wanted to change format of something, had to refile and had to insure when refilled the dollar hadn't changed at all b/c they closed out the quarter already - nice to say "over/done", this case money wasnt part of the problem, but if discovered issue with \$, must be some threshold, why would you go ahead with an annual report KNOWING there is missing \$ - can update #s quarterly for up to 2 years, as necessary, REGS want quick /soon data, how long keep something open - "close it out" - can't just say "ill make sure under 5% at end of year" at the very least 1/4 has to be finished as best of your knowledge

- DH - does AAIS have to close out quarters
- PA - getting better for the future, like what TRV doing smaller slices, update by quarters for all
- DH - do we really need to close a quarter?
- SC - maybe not "close" but maintain data integrity
- DH - is there metadata that needs to be est that says "ok, data thru Jan is within Tolerance", accumulates over course of time
- KS - across all data, individual state?
- SC 5% has to be BY STATE BY LINE
- KS - if you don't load anything over 5%, dont allow, cant be over - closing out, interesting, discussing "attestation" - attest as of date: date range is good, edit package says "quality data". attestation range of data "as of X date, data for Qx is good, use for reporting now)
- DH - some time in the future, find something wrong with Qx, "as of today I can say last month's data is correct
- JB -change month or period, re-run that check
- KS - update data to be in sync with source systems (HDS Not source of truth) - any time problem with upstream or ETL, requirement: closing out a quarter by attesting "as of x date, quarter 3 has been loaded and any change to that must be re-attested" - simple as "up to this date"
- DH - other than ETL issues DR described beofre, something funky happended between source and HDS, diff than what SUsan is describing with "true errors" - when fixing those errors will be a new set of transactions, new load with corrected info, as done will be run thru edit package and maintain 5% tolerance
- KS - if a transax is changing data that would come. from a report that would have gotten diff data need to reattest
- PA - Regs want us to strive to make the data better, not a req to repro report when it was generated
- KS - this req: I changed data and reattesting it is ok, changing the data just saying, not saying reproduce - CLOSE OUT: loaded data, ready for reports to run, now changed data, needs to be auditable, data that was there and attested to, changed and been closed out
- DH - go back as a req - do we need to close anything out? dont see purpose to having it "close", policy this year will have claims for next 10 years. I can't close 2021, can close data for 2021, not sure what "closing" means
- JB - not the same as closing a financial report - this is a data qual check to make sure threshold still valid for a time period, re-attesting - can still add data
- DH - making a glitch correction vs fixing data, SC's example: not changing data but adding new records to fix whats out there, transactions thru edit, within tolerance
- KS - data thats ready by a report in that time period will get different data - does it matter - close or re-attest
- JB - get rid of idea of "closing"
- SC - be careful, "closing" is semantics, at some point to produce timely reports needs to have deadline, today report monthly to AAIS, report monthly to other stat agents that req monthly, needs to be there 45 days afte rmonth ends OR 45 days after end of 1/4 (AAIS), regardless of when sent needs to be in and under 5% by May 15, to produce reports, have to be timelines, dont wait until end of year
- DH - small carriers who only load 1x a year
- PA - due to old contracts, moving them to openIDL on a diff cadence
- SC - good example, if only report annually and now report Feb15 for prior year, runs thru edit package find errors - is it in and under by Feb15? clean by? longer you go the longer you push out when you do reporting
- PA - diff in the future, spring 2021 lots were turning in stuff late, no repeat of that
- KS - assumption: nothing in HDS above 5%
- DH - needs to be architected, is there a precursor to HDS where info loaded, read into edit package, correction then HDS or is HDS a landing point and a secondary DB for stat reporting that has the correct info - how do we put that plumbing together
- PA DIAGRAM
- DH - many errors dealing with are omissions, coming from plumbing, ult source into data files used to create stat files, where info has not been provided that should be, while stat file may not rep "truth" the corrections should rep TRUTH
- KS -attesting data loaded in HDS is TRV ability to tell the truth, wont match source sys for reasons, but attesting it is the data you can puit into HDS, for stat reporting
- DH attesting to " good for stat reporting"
- Everyting in HDS is usable for stat reporting
- DH - outside of HDS do we need metadata that says "as of aug20, info in the hds, the last load was in tolerance and sequence of loads into HDS are within tolerance" - do we need to inc control mechs (policies, premiums and losses)
- KS - opinion regarding claims vs policies, cant use for loss data up to this date, certain years old before used for loss reports
- DH - "accident year" wont close for sev years, have info, "incurred losses" is what they THINK it will be may change over time
- KS - attesting that data in HDS is good up to this date
- 08222022
 - DR - Can't start making changes to HDS directly, gets out of sync with source system, can end up not matching sourcr systems then State Management problem, hairy, load new data, what edits already made? (not better than used to do) - doesn't think you can edit directly in place, HDS in his mind still design tenet one: faithful rep of back end systems... Dale made clear need to have facility to make changes, cant do on fly and takes time and needs to be done - solution something with foreign Key to a CORRECTED table or a federal other store of view, updated or changed as needed and as processes improved goal would be thing is short lived, alive for corrections and next extract - HDS can't be anything but rep of systems of record
 - KS - Edit package not based on completed report
 - JB - if there were errors that came from source systems, had exceptions (fatal nature) and couldn't accept data and had to edit source system, takes time to correct something in source systems, easier to extract
 - DH - clarify - errors they had runnign thru SDMA, most instances (not a lot) had 486 instances to correct stuff/169 of those were "liability limits missing" - feed was not providing the appropiat liability limint, doesnt mean source didn't have it/ correct just NOT being fed to them
 - KS - ETL is wrong or source system is wrong, have to keep what was fed from the source system and when making vhnage has to be sep place understood that this was changed, can go back and find changes made and fix them
 - DR - situation where, limit wasnt there, in source system BUT in HDS, whatever reason, new record in and fixed (ETL is fixed) and something is there, how to handle mismatch? which to trust? One in HDS is prob right? FORces making decisions as to what to do when reload HDS, code a lot of judgement in or precode decisions in how to update.
 - KS - keep it simple, see if patterns, automate where can, track what changed to (dont lose previous) and deal with it when refreshed
 - DR - obvious prob is bloat, shadow versions of everyting
 - KS - 480/10MM not bloat
 - DR - 2 assumptions: A. Not a lot of changes, B architected to take adv of fact not a lot of changes, make it in a way to not hurt you and way to automate processes, so bloat becomes ephemeral,
 - JB - do something like that, HDS has correct info so queries are correct and audited record. keep track of what did change
 - DR - HDS can never be edited in place, cant be something to keep track of something that diverges from downstream systems - only SoR is Downstream SoR, cant maintain business logic of having to decide what to update, preferable: the edits will be referenced,
 - JB - complicated Extract Pattern, looking for exceptions

- KS - do views to accomplish that, want to make exception hard, not easy path, make whatever it is, keep both in mind, when you have few then satellite table rather than core table and then deal with view idea, as long as keep consistent pattern not bad
- JB - run reports against HDS,
- KS - extraction has to see corrected data otherwise why make corrections at all,
- DR - too challenging to write "HDS is faithful rep of Core system" but an edit needs to be made, pull that data, easier someone does EP that does nothing but that table with edits applied (convenience function: first thing build corrected table, build EP, run extraction)
- DR - Ephemeral Bloat
- PA - 2 weeks ago on with jamesM, ETL pipelines, edit packages - scoring errors - talking about two metadata columns on load, simple pass/fail - ex: 5% by line by state, wrong zip in wrong state, maybe have metadata columns - 2 flags
- DR - like "flag" if wrong, didn't pass, flag it. Confidence score is imprecise,
- KS - use case specific
- JB - on collection not single record, accumulate across records, additive processing of scores, confidence score of total
- DR - 400 out of 10MM, make any changes? They did, more errors but within tolerance - in some locations, some lines, all add up to 400
- DH - not to zip code, not vin number
- DR - leans to "don't fix on load fix on extraction", spend time pushing downstream to sys of Record as they become problems
- PA - first flag - dead in the water, real vs nonreal error, rate zip codes diff, some pass/fail
- DH - pass/fail depends on case, ipcode may be bad, state ok, if doing extract for state of AL, not looking at zipcode, zipcode bad doesn't affect ability to pull AL data
- DR - confidence score too tough, too specific, "here's an edited row IF YOU NEED IT", logic of EP could make decision to pull or not
- PA - doing load, zipcode wrong but state code correct, will I gen edited row to omit zipcode
- DR - edited row based on why edit, it's the fix, omission, correction, leave up to extract pattern and say "something wrong with row, here is the correction"
- DH - may not fix it - Error / Fix Error / Not Fixed
- KS - can't have fidelity the specific knowledge of knowing exactly what was wrong w/o going crazy, need to see biggest/scariest
- DR will happen as we build, flag up, fix = omit zip, if under the state/line then great - we will be using to improve downstream processes
- KS - if we can track what fixes need to be made all the time, across carriers, can work on fixes
- DR - up to each company to decide -
- KS - want to know what activity is, what you do with it is your choice - good req: track changes and report them
- JB - exception for data qual checking, more than "there is an error here" - want to share what is wrong
- KS - is it true, fixed record or deltas
- PA - need an error log table, what keys failed
- DR - wouldn't that come out of it anyway? right now tell you what to fix?
- KS - are we keeping it in the right place?
- PA - 2 sep systems: data lake vs SDMA, 2 diff systems, not carrying error in record
- DH - dead simple: error and here's the fix, inferred what was fixed by what changed, if error flagged and new row with correct, great if not then assumption says "good enough as is", can see what works if that simple, if we see spending too much time fixing rows, someone will fix downstream, solution too much gets way to complicated
- PA - can we work thru what it will be like to work thru these errors, more in terms of SQL than Mongo - select all rows with a Y in the error col, re-run edit package to get those? what col or errors?
- DR - judgement still on, whether needs to be corrected, if there is a change needs to be made, will gen new linkage, 400 transactions point to 400 corrected transactions, if no pointer, then ignore, if there is use the corrected record, judgement based on team w/in carrier so it meets requirements - in theory could spin up new table with corrected columns, make a materialized table of corrected rows, do extract against that - super simple don't worry about what's wrong
- PA - wondering not be able to say "grab all records with no errors"
- DR - could grab all rows, if any has "Y" for error, go retrieve alt row with that foreign key and overwrite. - convenience funct, make a new table on day one, make extracting from that table and drop it
- PA - corrected table, updates daily
- JB - source isn't single table but a set, have complications with replica of tables
- DR - convenience funct, save complication on extract writers part
- PA - single table design
- JB - not forever, will want simple relational scheme in the future
- PA - hesitant, single table with stat records
- DR - see where this takes us, where we need to go, feels like bar for Regs isn't high, just wants to get data more frequently, 80% gets us easy, try something, simple approach, Dale ability to edit as needed to implement requirements, simplicity, no state management and hits biz reqs
- PA - benefits to get more normalized
- JB - if just doing POC for stat report but if you are doing other lines of business,...
- PA - other lines have key, stable design
- JB - at some point single table looks like long cobol row, need to keep track of errors, more than just stat report
- PA - haven't heard? that couldn't be solved w/ single table design from the stat record
- KS - think there will be use cases, challenge single table approach, be ready to make change, not stuck w/ one model, can do it when we need
- JB - from arch POV, describing how to do exceptions, error handling, approach isn't general solution
- DR - only thing in horizon, dirt simple approach to solve problem reticent to complicate it yet,
- KS - deal when we get there, overthinking now a distraction, KISS, flat model, eyes and ears open
- DR - have to make it simple for them AND reliable, holding HDS must be reflect of SoR, makes it easier for carriers, flat makes it easy too (single table) - flag error points to corrected entry,
- KS - some way for EP to see errors, materialized view or something else, to make fixed val available
- DR - edits made via dashboard (TBD design)
- PA - java app running SDMA, ? as to how to implement dashboard into HDS
- DR - assumes some, b/c taking HDS and outside dependency of edit table, not auto generated, get some manual assent to the data call or stat report
- PA - like specificity funct on dash for what records to use?
- DR - since someone could be manually editing, has to be a
- PA - if load is not compliant, too many exceptions to be used for particular dataset it SHOULDN'T be moved into HDS
- DR - would be moved to HDS, as HDS will show wherever there
- PA - get under tolerance pre-load
- DH - will HDS rep source of record or corrected file
- KS - extractable part of HDS, need to say "this record or set of records is logical HDS for extraction after edit package, initial stuff"

- JB - simplify initial HDS agrees with source system is STAGING HDS and make correction and push to Production HDS
- DH - have 50MM records and correct 5000s
- JB source vs corrected, staging and queryable
- DR - copies imply you have a staging and fixed db and fixed is fit for purpose for cert things, might not correct everything (resource intensive) - it is a GOOD ENOUGH db, reps best of carriers ability compared to core system, can be referenced into another row/table somewhere - if you make a copy, make logical divisions, right back where we started - "correct enough"
- KS - EXTRACTABLE - implement is tbd, original stuff, edits made, extraction pattern
- DH when do the battles begin?
- PA - initial load of data today, needs to be some way for loader to get stats like SDMA does now, - UI, do load, get stats back
- KS get data from carrier, fix data, when EP happens fixes are visible
- PA - doing now, carrier is fixing stuff, give carrier tool to fix before extraction happens
- DR - 2 types of fixes, might be able to affect in source sys of ETL and in data store - SDMA tools, the error checking tool highlights those, still the same DB still there, not loaded in staging
- PA - as a data engineer, def think before do first submission you want to run checks before, putting data in and THEN running data checks
- DR - only place to make changes is in this db, no where else to do it
- PA - need an SDMA person, when doing a load and stuff goes wrong are you correcting in UI or dataset? Doing correct in UI
- DR when they fix upstream fixing the flat file, sometimes impact source systems, in this case load into the <not HDS> and then say "oh errors" and then decide to fix
- JB - if existing db with records, loading batches, other criteria to check (all, just what loaded, timeframe), seems natural for simplicity
- DR - worried, other DB/datastore that is corrected that gets loaded in
- PA - not accepting flat file until it is good enough
- DR - doesn't want to work on flat file, would rather take flat file via API at that point, if fixing all in flat file,
- PA - scenario - state column doesn't populate, load 1MM records, fix flat file rather than fix errors - some preprocessing will keep error count lower
- DH - can build an edit package they'd have anyway, qual checks to prevent egregious errors
- DR - building shadow system, to build diagnostic, additional flow - test environment (pre-load), lot of times, changes needed in this case are going to occur in this db, if can make downstream he will but many times he can't, can run diagnostics starts feeling like in-between
- PA - w/o preprocessing to check keys are there, end up w/ extraordinary # of errors
- DR - initial problems, but closer to steady state ops of this, prob wont have a lot of probs cropping up alot - if you do have probs on load, decide where to fix em, otherwise maintaining dev env in parallel
- DH - maybe a solution is, as we run edit package we have an ability to delete those records that were part of this batch load to fix outside of this load, if you had to
- DR - dont want it to be stateful, reload when fix probs, make it simple, could have vers in parallel, probs would fix probs, load into test again, load into proper one, geared towards ETL errors
- PA - weird not to do pre-processing check
- DR- issue is scale, obvs build checks before anything pushed in QA process for operational changes but in production load...
- PA - data not code, saying "before we run load we have x checks run, are valid keys here
- DR - not loading entire data set, could be some checks occur before load
- DH - high level QA process, critical errors vs
- DR - not realtime system, doesnt need to be perfect when it lands, doesn't see need for perfect when hits HDS
- KS - removes some of the bloat of records - NEEDS OT BE A WAY TO AVOID UNNECESSARY ERRORS (PRE-PROCESS< whatever)
- DR - capture in QA vs operational load, not thinking of operational process,
- PA - every load as from a 3rd party, seems like ingesting 3rd party data will check
- DR - not internal for
- JB - not just loading anything w/ stuff that is incorrect, some kind of prechecking
- DR cant commit to prechecking - need to be addressed, cant agree to it yet

Reconciliation (to do Mon 9/12 w/ SusanC)

Reconciliation (make sure report is correct based on request - reasonability check on the report - NOT financial reconciliation)

- PA - from AAIS Andy can share
- KS - business perspective then fill in gaps
- SC - only think of the financial reconciliation? Does Andy (AAIS) do another? Reconciling statistical to financial by annual statement line and state basis
- Andy - company, line state, matching what get from NAIC, what AAIS does on their side
- KS - mix? req to do all of stat reporting through AAIS thru particular line/state combo?
- PA - 2 part:
 - 1 lot of ambiguity, first look at stuff and give reasons for why #s vary, adjustment things
 - 2. doing recon, doing company/line, expect with data from them they get line/state combo, do not partially report
- SC - correct, stat agent gen reports ALL of data except where it is statutorially required - MASS says all commercial / auto must for through CAR, AAIS would not have biz from TRV for MASS, when reconciling would not have lines for state of MASS - no data from TX goes to AAIS, Commercial data to ISO, Personal goes to TICO, surety to SFAA - from that perspective, nothing missing, b/c going to another stat agent, as PA alluded to, always going to be differences - why diff if are diff - ex: homeowners is very close w/o lot of differences
- KS - what causes Diffs?
- SC - so many: journal entries (made to books, journal entry made to books and records), you have big cust that renews at end of year, all the data hasn't gone thru rate-quote-issue-system, SC has nothing, but b/c you know its 12/31, rev has to be booked to that cal year, NAIC financials have \$1MM more than stat records - smaller insurers, subsidiary reports to other stat agent, small # of biz, that stat agent doesn't ask TRV to provide anything, Page 14 against what you receive statistically: very off, small %, 1% or 10k (under that) - always historically done reconciliation on their own - 2020, andy could pull page 14 and statistical and SC could provide file, w/ diff by line and by state - state of KY, premium tax booked to premium (statistically they dont report premium tax), so page 14 in KY would have prem included - in NY, Free trade zone, not stat reported BUT IS on the financials - international business, coverage of someone abroad but by US company - # of known differences, can identify those and provide this so that AAIS can be comfortable in the reconciliations - waiting, look at it, have questions, wouldn't be phys possible (would gen too many)
- KS - proactively provide reasons why numbers wont match

- AM - not all members do that today
- SC not all need it - smaller and diffs aren't there
- AM - other challenges - TRV stat agent for everying, use AAIS for some and ISO for others, harder to reconcile to NAIC, trying to balance against property annual statements - do stat reports by AAIS stat plan line of business but all reconciliation is byt NAIC stat plan line of business
- KS - why do we have our own?
- AM - do get annual state/line business historically w/ exception of auto, believe are programming to do reconciliation - typically homeowners doesn't matter, sometimes little chunk booked under earthquake, lets look at homeowners, need to look at little things as well
- SC - endorsement? homeowners or dwelling, go to diff line that the base - can get messy (esp fire allied?)
- AM - historically struggled with reconciliation, feed from NAIC is annual, when TRV sends data and AAIS gets feed, can't reconcile - trying to get faster, sooner, get quarterly data but 1/4 is broken out by company and line or company and state but not company-line-state - struggle in the past:

<SEAN CHECK RECORDING>

- DH - accrued accounting vs cash accounting
- JB - policy 2 years ago, no claims or adjustments, how know years worth of premium due?
- DH - recorded premium a year ago
- AM - field of month covered
- SC - reconciling: premium stat records to prem financial records / not reconciling in stat reporting - only recording paying claim if travelers recognized loss
- JB - policy in effect w/ natural premium, couldn't reco collecting if payments intermittent, etc - policies with commitments is there
- SC - if policy cancels, change comes through, offset records, etc.
- JB - if not actually in transaction record
- KS - when do we do, against what data do we do recon
- PA - when first then how?
- PA - when doing right now, is that we get all fin info from NAIC, final form avail 1 year after (
- AM - typically year after, may/june 2022 we get final of
- SC - when publish? 2021 -
- AM - april and may get feeds, may/june is final for prev year, to reconcile 2021 data can't recon until May/June 2022 today - first 1/4 2021, due no later than end of may 2021 but can't reconcile until at least a year later
- KS - if recon is a part of our process, how speed it up? shorter than a year?
- DH - wait till end of calendar year for data to be avail - publishing now 2022 data, but books dont close until Mar 2023, when you can do the reconciliation
- JB - pub quarterly to NAIC?
- SC pubbed quarterly, but can reconcile from own systems but what gets pubbed by NAIC is by company/line and by company/state but not company/line/state (per 1/4)
- JB - ask NAIC if they can provide
- giggles
- SC not pubbed that way
- AM - have asked - hard to say realistically which was more valuable
- KS - is after a year pubbed that way?
- SC - is pubbed that way so they have to get it published, get data in files to stat agents takes another month or two
- KS -when can we do Recon
- SC - cannot start until May/June of the following year
- AM - Q1 its been 15 months,
- DH - what Eric was saying, less concerned about recon, just wants speed of data quicker
- JB - Eric cares when reconciled, wants more info on ongoing basis re: changes in policies/coverage
- SC - interesing when AM said "having to accept the data" - int b/c kind of seperate, stat records are in format, meeting edits and checks, from that perspective these are valid records, but cannot validate its reconciled for almost a year - imp to make distinction ,if you had a customer for a couple years, get stat records, going back to last years recon, if you have a small issue and something missing due to volume but 1/4 homeowners was big change - something not right?
- AM - refer to rules engine, only thing rules engine does "does ea ind record represent itself" - internally making sure before a year, ability to say "member gave us \$1mm in premium by 1/2 this quarter - should question it if it drtastically drops or increases
- SC - high level checks can be done - industry isn't AAIS - it is INDUSTRY
- KS - sev months ago, AAIS is by law supposed to do Reconciliation (in the handbook) - WHY?
- PA - annual reports supposed to be reconcileds
- KS - what requires Reconciliation to occur? stat reports, reg reports? data calls?
- JB - once a year check on prev year sanity, their sys of record, draw conclusions,
- SC - believes idea is you have reconciled before - andy gets 2021 data 3 months ago, reconciliation should be done before annual reports go to states in march - state of CO, could come by and want to know data, will reconcile to own state/lines - data call, doing recon on the data call and sometimes when they get data - not same as process to reconcile all Stat data, recon back to financials only real control any state of bureau has
- KS - expect recon for stat reporting due to timeing, if you try to recon against data call, 2 years ago say "yes been reconciled" but 2 months...
- DH - dont do any recon on data calls, do internally but REGs aren't coming back
- SC - state of CO in Feb asks for 7 pieces of data, they get it, make sure internally matches, big discrepancy theyd come back
- DH - in some calls would have to provide Page 14 and 15
- KS - talking about reqs for openIDL - allow to provide page 14 to respond to data call for a line of business?
- DH - no its anonymized, wont' know TRV, Hartford, Hanover
- PA - depending on whats going on, situation where state earned Prem, could potentially und that someone was wrongf
- KS - state want AAIS on their node, each one of the results is first reconciled and then returned, if not some error or exception handling
- JB -only certain cases
- KS - support all cases, deal with those too
- JB -report to REG outside of openIDL, or require addl data
- PA - main focus for this group, RR the annual reconciled, goal using this sys that current infras
- KS - have to have reconciliation for part ofd the process for stat reporting, can discuss for data calls - will need funct for some kind of recon, just to do stat reporting
- JB - stat report yes, not for data call
- AM - NAIC, CSV files,
- KS - NAIC provides data, AAIS subscribes to it - sounds like diff process for data calls vs stat reporting
- PA - both coming from same HDS ideally, both reconciling and validation
- JB - dont use "reconcile", cant happen on every data call

- PA - every year reconcile, every year find bugs
- JB - within tolerance you dont, not adhoc data call
- PA - find out something failed ETL and only have x of earned prem for auto
- KS - dont have financial records yet
- JB - heuristic, "this doesn't look right" but a sanity check
- KS - only biz people like susan know its a bad #
- JB - threshold, something changes by 50% vs 10% questionable
- SC - need to put in reasonability checks, go back and ask
- KS - premium capture when change premium recording by certain amount requires explanations (hit or miss)
- PA - once a year reconciling auto records, same data source making calls from
- KS - exception: when the data calls are happening before
- PA - every year shoring up most recent year
- KS - not going after data call reconciliation right now
- PA - data loaded best of ability trailed by 1/4, a year later "we feel reasonable we got all the data in there"
- KS - some doesn't work, do have to tell thats good (REG can do it, etc.) - tabling that
- DH when providing pg 14 or 15, doing calls now for EOY 2021 data, reconciled to financials,
- KS - would be covered?
- DH - stat reporting is 2 years lagging
- DH - for data calls today, some are for EOY 2021 data, could reconcile right now as financials published for 2021 - OTHER calls are "what do you have today", we can't provide proof of reconciliation b/c dont have reconciled financials
- <SEAN CHECKS RECORDING>
- PA - missing premium, less data than where supposed to be
- KS - fatfingered extra zero, data misalignment, has to be dealt with, exception path for data to let TRV know their data doesn't match NAIC data and then deal with it
- DH - SC? DOes that happen really? DOing it internally but dont know what reported back to AAIS
- SC - given to AAIS like ISO, both - built in excel, does have page 14 and transmittal, using either and not the same with AAIS should match, b/c of volume instead of providing x lines of business x 50 states you have mult explanations w/in any one of those - give excel with col for global, col for journal entries, col for NY, col for KY - for any state-line you can see instead of words
- JB - reconciliation matrix
- KS - for TRV, doing it in such a way not too slow a process
- SC - AAIS doing it at the same time, here's pg 14, stats, diffs
- AM - SC is saving AAIS time from AAIS having to reach out "why are there these diffs?" - her excel sheet is proactive
- SC - we have subsidiary reporting to diff stat agent, dong give anything, do review, they come back "why are these 7 different"
- KS - raises exception, above threshold, provide what they provide AAIS: Line x state accounting of difference
- SC - making AM comfortable w/ differences, know why, here's how - yes they dont match perfectly but you are comfortable w/ reasons they dont match
- KS - put rationale somewhere for audit purposes?
- AM yes - need to see where they are - some companies send 2 files: premium and losses - ask "where are they" get them, recheck -some scenarios rechecking - proactive or "we forgot to send you this"
- KS - at point in time where NAIC data is avail, grab data, reconcile, transactions should match record to record
- AM - not record count - earned prem, written prem, paid losses, etc.
- KS - by line by state throw differences against threshold
- AM - more than 1% of \$10k it is questioned, if someone not reconciling you can't inc data to the states at recording time
- KS - what visibility does NAIC have to which companies have for stat reporting?
- PA - trusts AAIS - AAIS turns in list of reporters and non reporters, on AAIS to
- AM - misinterpreted "reporters and non reporters"
 - reporters: companies included in reports sent
 - non reporters - added to DB, if no match put on the list, non-report list lists companies not included in the report - companies they did not get data from OR didn't reconcile, low quality data, etc. BY LINE AND STATE
 - ind company could be under both reporters and non reporters
- KS - by NAIC line - yes

9/13/2022

- PA - yesterday going over financial reconciliation, when and why important, typically what you expect to see due to financials, carrier has spreadsheet to report adjustments acknowledge "whys", talked about the timeline on that, AAIS gets public financial data from NAIC shows up 5 months from close of YEAR
- SC - idea of annual reconciliation cant be started until end year financials, dont get till may timeframe, talked ongoing reporting and making reporting more timely, talked about checks and balances (prior year/ 1/4) to feel comfortable - aais or other stat agent verifying validity of records, barring financial info the record counts and \$ amounts were reasonable
- PA - for SC, not in terms of data AAIS gets from NAIC, when carrier does internal review how long for them to get idea of what financials should be?
- SC - 30-45 days before can get a file (safe to say)
- PA - level of precision? rough and then get updated six months later (ANSWER: thats it) - 45 days after business, Carrier gets financial to compare it to, already have business records - once they have financial and biz records do they have what they need to do reconciliation?
- SC - by end of april, sent 1st 1/4 stat records to AAIS
- PA - by time sent already dont internal check, buit NOT compared to financials
- SC - getting stat records from systems, focused on validity of ind records, getting to AAIS< fixing errors over tolerance - perfect world, 60 days of 1/4 ending, differences need to be reconciled but most part could see overall w/in or very close (not worried they have missed something) - diff between MISSING and TWEAKING
- KS - Hartford process?
- JM - Mike Nurse and Kevin - know it cold
- KS - similar as Regulators, hoping to get diff perspectives on this (Carriers, Regs, AAIS), good to get that if possible on
- SC - goal to build reconciliation into openIDL?
- KS - goal is to see if we HAVE TO, sounds like have to for reporting
- DH - AAIS does, openIDL doesn't seem to
- SC - AAIS does March for year (if year of reporting is year 0 AAIS by march of year 2), but also record and dollar amts per quarter or per year - basic checks and balances numbers are reasonable
- PA - could see a dashboard lets you see that, take human interaction , lots of reasons stuff can change

- KS - if stat agent is going to build reconciliation via openIDL, needs to be some way to get data from openIDL to do reconciliation, dont hold data in openIDL like AAIS does now, need method to reach in+ get it+ process
- PA - will want to get something, make it easier to put it into openIDL
- KS - agrees
- PA - otherwise give up more granular data MORE regularly, if we can have mutl tables (stat records, financial summary data_ never leaves - have internal diff to compare if HDS is in synch with financials, and then an EP to return T/F if in bounds or not) - will have NAIC #s by time to make the report but if situation where as soon as financials are available and can evaluate if they are correct
- KS - pipeline - clean syntax, reconcile, declare ready for extract - def have to have reconciliation one way or another as part of openIDL process
- PA - auto coverage report, limited to auto stat plan, do stuff like General Liability, any of those use Multiple stat plans? weird caveats to reconciliation? mult stat plans to single report?
- SC - homeowner policy with personal liability endorsement
- RS - opposite
- SC - personal liability endorsement on homeowner polciy reported under homeowner stat plan but prem \$ not booked to 040 (where homeowners gets booked) - homeowner module might have \$1100 but 040 has \$1000 and 171 would have \$100 (040 and 171 are ASLs)
- KS - that record would have annual statement line, dont capture right now? we do capture homeowners, what would ASL be on that record
- SC - homeowners go to 040 but liability endorsement AAIS wants LE included under HOMEOWNERS Stat plan but when comparing to 040 only \$1000 in 040 because rest to 171
- KS - clarify to record level, specific record endorsing, have ASL for liability
- SC - we report ASL for homeowner
- AM - stat record as it is booked
- KS - dont need to account for it, use direct reconiliation instead of our codes - do we know true across board?
- AM - not sure did it that way in past, only other line in issue is auto until recently when added statement to auto stat plan, AUTO easy to und but anything going to "all other" easy to ID, aligned with coverage codes
- AM - not required field yet, not everyone provides yet, TRV giving it
- PA - reserve sections?
- AM - took place of reserve field
- KS - req for openIDL?
- PA - def want (discussed mult times)
- PA - right now carrier using reserve section, rollout plan for all carriers? (Mike Puchner and Padma)
- KS - haven't discussed re: reconciliation?
- PA - looking at SQL queries, ask better question about
- KS - felt almost done yesterday, get perspective from others (Hartford and Hanover and REGs) - hijack Friday meeting for that purpose
- PA - add to agenda items - may have to come back with another question, all making sense now
- KS - talked about 2 types:
 1. reconcile whole year - NAIC #s against prior data could be done as part of process or ahead of process (does that count for true reconciliation?)
 2. REASONABILITY CHECK
- SC - asking carriers to do it on 1/4 basis?
- AM - not in place of AAIS doing it - still doing it, still matches, THEY are explaining why differences (fat finger, missing file), still want to know if they are off more than what they are telling us they should be off
- KS - not in lieu of
- SC - incumbent on AAIS to do reconcilkiation, carriers can support

Financial Reconciliation (Oracle? Source of truth to tie against those #s?)

Statistical Reconciliation

Auditability/Traceability

Deliver Report

- PA - operator/operations - walk thru ELowe's perspective and discuss others - Eric reps VA, will need one report per season per line, some lines more than one report, just talking annual now - get report to Eric, get it from every carrier writing in VA, make it both as seamless for EL to get carriers and as seamless as possible for carriers to acceppt all this - today have to have one EP per report (50 EPs/50 reports - ea EP slightly different) - how do we group things together to minimimize the number of requests/consents (auto coverage report)
 - DH - Stat Reporting? Likely to say yes to all 48 states, for TRV all-in, doesn't matter, prob isn't going to be challenge could do by state by line and an option to "select-all"
 - PA - stat agent (in this case AAIS) be keeping track of all reports and then have Regs from ea state subscribe to which reports they want done and then Carriers can subscribe to ones they want to fill
 - DH - will choose stat agent, not only YES to all states all lines but also which stat agent doing it (no state with 2 stat agents) - a stat agent per state/line - only one w/ 2 stat agents is TX (only recognizes ISO/Verisk) and MASS (MASS CAR)
 - PA - Stat agent needs to keep track of where reports are, keep list of subscribable events
1. Stat agent keeps index of reports by line by state
 2. Regulator - chooses which reports they want to receive - standard stat report (otherwise a data call)
 3. Carriers - will LIKE by line item (stat agent-state-line of business)
 4. Stat Agent will attach an Extraction Pattern (EP)
 5. Carrier will consent
 6. Generate Report
 7. Submit Report to Regulators and Participating Carriers
- PA - 3, 4, 5 happening today - ? for Ken - why doing the EP after the LIKE? if we attach EP before they LIKE it can do Like & Consent at same time (KS - like confirms the carrier is on board with the essence of the data call described in the text. Don't want to develop EP for data call no one likes)

- DH - agree for stat reporting, for data call it is different - stat reporting obligated to report, no choice, for data call may not want to participate in data call and share data with openIDL, may do it themselves and go direct to regulator
- JB - collected by openIDL on behalf of REG,
- DH - may not go thru openIDL, even though Reg may ask thru openIDL, collected by openIDL for those who have consented
- JB - rationale, efficiency, less expensive, concerns about sharing data based on %, if it is for reg
- DH - anything Carrier provide they get copy back (to regulator and their portion of it)
- JB - don't want to be larger of certain %
- DH - if participating, get copy of whole report back, their % and other carriers, value of having openIDL do it, see results, right now they know their piece, dont see what Reg sees

Make report available (S3 bucket public/private? start private)

- PA - submitting report, gen report and put into S3 bucket, downloadable link, part of the data call will have URL of report, carrier goes to URL and downloads document - security is on the S3? anyone download from link? Password, Cognito?
- AC - no security, public link, right now avail indefinitely
- PA - EP and URL is good, email / messaging to let Carrier & Reg know report was avail, with message can tie in password, report public is a security hole

Deliver to participants (carriers)

- KS - anyone participating in a data call should be able to see report associated w/ that data call? Only participants?
- DH - didn't pay, can't play (carriers that do not participate do not get to see end result, carriers that participate DO get to see result)
- BH - market share perspective (in reqs already)
- KS - can't see things/ details before cleared % requirement - should be able to see what YOU (as a carrier) would contribute, but now whole (or other carriers) until the report is complete

Deliver to Regulators (requestors)

- KS - requestor needs to be able to see if they have access to data call, and see one or more reports related to that data call - if you can get to data call you can get to the report
- DH - participants and REGs, credential who can see it?
- KS - can't get into UI w/o credentials, report viewer?
- DH - for carrier and requestor, dont want low level analyst sending report to newspaper
- KS - levels of access to access a report, true of creating and a lot of functions (regulator has login and can see but not create data calls) - finer-grained permissions
- DH - once submit a data call to a state, no control over it
- KS - requirement not to be able to download it, method to keep them from abusing, sharing file
- DH - NOT editable
- Watermarking reports? maybe not a requirement
- DH - only the requesting body can get the report
- KS - how does it relate to carriers being able to get report?
- DH - if EL does a data request and gets results doesn't mean GeorgeB can get results - needs to do his own data call
- KS - scope of the data call
- DH - recipients limited to state requesting data without further approval - dealing with data calls too - it would be nice if REG as making request would identify if the request is adhoc or recurring?

Receipt

- SB - Receipt that the Regulator has received/downloaded report - given to Reg, Carrier and Stat Agent
- DH - maybe PULL w/in UI
- KS - automatic? follow link, you know you received it
- DH - notification report is avail, you go in and pull the report, as pulling report UI could then update to reflect report was taken
- KS - assumption you are not following link from notification
- DH - is the report so big it can't be emailed or delivered
- PA - question, requirement, we dont want to make a file b/c we dont want it to be shared, KS said "not be able to download it"
- KS - struggle w/ how we can not have a file
- JB - have a file, encrypt it
- DH - want file protected so not editable
- KS - once downloaded impossible to prevent sharing
- DH - some form of report
- KS - know who downloaded
- DH - if info is anonymized, are you telling Regs who participated?
- KS - option the data call could have, who participated, when does it make sense for Reg to know who participated
- DH - ok if carriers are listed on a report IF boundary/threshold for data (%) met - how much of market are REGs going to have included in any data request
- KS - how would they know?
- DH - asking provider "how much of the market is this representing?" - might ask Stat Agent (AAIS in example) as intermediary
- KS - NAIC? is that the NAIC #, the total market
- DH - NAIC page 14
- SB - what are the elements of the receipt?
- DH - Data Call Name (whatever its called), who received it, when received (time/Date), a receipt for every individual accessing (downloading?) a report - indicate WHERE they are from (VA DOI ELowe = who and where) - include organization
- PA - if resp for reports want to know who provided data for report and who downloaded, when filed - can stand up to rigorous audit, like to know who turned it down (DALE doesnt want) - doing auto coverage report for TRV in State X, when they say "we dont want to do this report (whatever reason)" needs to say that was travelers not participating
- KS - scenario for data calls, not req to do reporting for TRV (might not use openIDL) for stat reporting, req to do the reports, contracted to do stat reporting for carrier, can tell if they werent on the list, know they COULD have and can tell if they are on the list-
- DH - Stat Report SHOULD HAVE, Data Calls COULD HAVE

- SB - doesn't want receipt with Hartford, Hanover, and NOT_Travelers
- DH - carrier's business how they respond
- KS - how public is the list of contribs to a data call or stat report, just list? other contribs see other contribs - if EL asks for a data call he can see participant but can other participants see who else - Can Carrier1 see who else participated?
- DH - public info once hits Insurance Dept, gets to point of aggregated and anonymized, could be public, but a lot of data calls under confidentiality agreements
- PA - turn in report, list of #s responsible for, turn in list of participants and non-participants
- DH - maybe not participate over tolerance? why sign up with AAIS for stat reporting and not use for a given state
- PA - some companies work with AAIS do to Stat Reporting, have legacy systems, may not write all the lines (dont write coverage, or have team and do analysis themselves)
- KS - targeting carriers? how does a regulator say "I want TRV to respond TODAY"
- DH - maybe not do it over openIDL, can make data call to TRV, obligated to respond w/in their level of authority
- KS - in the system?
- DH - envision, similar to sign up for AAIS or stat agent, say "these lines are open to openIDL doing our data calls"
- KS - that notification, new data call comes in
- DH - could envision a REG request a data call to only one Carrier, wouldn't want to entertain going thru openIDL if JUST TRV, data already there, can pull it themselves, who creates EP, formats report, all that stuff, maybe own
- KS - over time used to doing things thru openIDL, skillset to build report and continue
- DH - just their data, may want more eyes looking at it, verifying whats being reported, resp for the info going out, just them, nuances in results you. make sure management is aware of, aggregated and anony, that concern doesn't exist - build this more for where it is a wide spread call, not 1:1 carrier:regulator - future build 1:1 into system but for now they have well-defined processes

Exception Handling in Reporting

- PA - what kinds of exceptions
 - report generation fail
 - Combiner logic failed
 - way combining stuff, got logic, EP, put it all together - combining logic for each report will be unique, EP implemented and attached to data call but haven't talked about combiner logic and attached to EP
 - Potential where each report has slightly different combiner logic
 - define EP, not all the code in a report, then there is formatting report after combined, diff from report to report
 - potential there is bespoke report logic for a particular data call or stat report - could vary and could be same
 - how much do you put into the data call - will have learnings
 - take a look at combiner logic
 - report generation failed
 - Diff then combiner logic
 - retry report gen
 - didnt meet % threshold (company wont participate)
 - data in PDC does not match expected format (something went wrong with EP)
 - data in PDC does not pass edits
 - possible
 - tolerances - specific record may not pass edit but w/in tolerance, how to handle?
 - ex - NC state w/ SC zipcode, under 5%, include/not include?
 - "for a record they do not have a limit", on a couple of records,
 - if doing report, missing stuff, then omit record (acceptable solution?)
 - visible in the EP, when you do EP, aggregate and ignore records would be visible in that code, hard to see, not in text

Auditability/Traceability

- DH - entire comms module needs to be auditable
- JB - requests and transactions on-chain, how do we inc email notifications
- KS - audit a report was received, what do we do with the system that isn't auditable
- JB - benefit of common channel, audit trail for interactions there
- KS - utilize and put things like receipts next to data call (JSON object in ledger,), receipts, contributors, etc
- PA - can't inc raw data (no raw data)
- DH - timestamps of when pulled w/in each carriers node
- KS - consent timestamp
- JB when deliver data itself, private data collection is hashed and thats a record (on chain) - inc into a consistent scheme - some application that uses it, info on chain
- KS - all updates to data call itself are auditable, on blockchain a given - who has access to whats in the audit trail/traceability? unearth some things not sharing - hanover could see who consented even if they didn't -
- PA - hash the stuff on chain and only give keys to consented
- KS - if a company needs some audit report, managed by admin of the network, should not make audit info avail by default
- KS - attributing things orgs touch to them, part of audit trail, consent: who, when, destination, history of data call, who updated data call, when EP run by consent

Notification

- KS - notification report completed
- DH - notification of data call (new)
- PA - external to UI or outside of UI
- JB - within network, preferred way, channel comms request
- KS - inbox expectation?
- DH - push or pull
- JB - default channel, comm of request, responses, likes - same mechanism
- KS - what it does now - find data call, def not robust UI, push-pull?
- DH - push preferred

- KS - subscription model, subscribe to notifications
- JB - application looks for those events and pushes notification
- KS - pull worth considering, inbox items you should consider, if Dale gets a push "you have report" , has to find emails he needs to respond to, otherwise logs into system, notifications in Inbox
- DH - what has been sent and disposition of those items in the UI, perhaps a delete option, acknowledgement (instead of inbox filled forever)
- KS - notification management (delete /archive)

Data Call

Communications for resolving conflicts, etc.

Load Data

Create Data Call

Like Data Call

Issue Data Call

Subscription to Data Call

Consent to Data Call

Mature Data Call

Abandon Data Call

Clone Data Call

Deliver Report

Access and Roles

Permissioned Access

- GAP - final reports are currently public (possibly indefinitely)
- no report data accessible to the public
- All entities on the network are known and have credentials (Carriers, Regulators, Stat Agents, openIDL)
- MFA required
- Credentials should be recoverable
- Credentials need to be renewed every 12 months from issuance
- Identity + Authentication schema tbd
- SOC2 compliance ([Sean W. Bohan](#) to do some digging)
- w/in carriers - finite set of people with access to reports, credentialed w/in carrier (NEED SESSION ON ROLES)

Roles

- General
 - an org can authorize members of an org to have a role (role administration)
 - openIDL does not validate identity of each, someone in each org resp for providing/rescinding access
 - Entity resp for nominating person, openIDL can fulfill (Day1)
 - credentials and levels of access, not all need to be credentialed through openIDL, but w/in network (est node w/in network -openIDL, app users w/in carrier node?)
 - access the node and levels above (UIs, etc.), sep from AWS accounts at application level, more study on levels of credentialing needed
 - app standpoint in other things, a service user or app user making connections to db, service-user credentials - app users self service
 - Every carrier has own membership model, CA from openIDL, diff than infrastructure running node,
 - Fabric perspective, everyone who can access node, from pers of deploying chaincode, every user should have Cert from CA (openIDL), PKI, how manage users like logins/etc. diff story, integrate with user management system (email, username, password, 2fa, allow user to login), private keys managed by org, not on device or PC of end user, ususally done when talking about auth end user transaction on network
 - Using UI, to issue request, interacts with chaincode - user who initiates would have to log in, key is managed w/in THEIR org, when user logs in, provides U/P, auth is used internally to check proper priv key, PKI management implmentation, system checks cert, fetches info from cert, level/role, can define what user is capable of, so Fabric can reco user
 - Cert managed by org, in the case of openIDL issuing cert, not each org doing own certs
 - Who manages? openIDL or Ifra Partner can manage on behalf of carrier, could be delegated as a service, manages on behalf of carriers, membership service provider model - any org has to have their own CA and their own internal management of certs
 - Organization: each org that has a node on the network, every node has to have an org and 1:1, one node one org, one org mult nodes
 - openIDL could manage CA, but means all users all nodes, sets of signed certs
 - each person has own Certs, could say "users do transax w/ same cert", every user needs own cert to trace who did what
 - Certs necessary part of infra, how they get issued and managed - topic for discussion, openIDL would have role, theyd have role in membership of priv network, make it easier for members to join

- Implementation detail rather than biz req
- Every user needs own cert - every org? each individual interacts should have own cert, not same as how to admin people who access local resources or AWS, diff management of access
- Not going to have one cert per org and then track w/in cert, need mult certs per org (diff roles and capabilities)
- which is better in terms of security, each user having own cert is best
- network management function rather than put it on each org, multi-tenant node (own topic)
- Carrier
 - up to carrier
 - permission to enter openIDL (not carrier AWS)
 - Roles:
 - Infrastructure (AWS infra within carrier account)
 - Deploy resources
 - Monitor resources
 - Application Users (w/in carrier node)
 - Load Data
 - Correct Data
 - LIKE a data call
 - Request makes sense
 - Also potential of multi-level consent
 - Business decision THEN commission EP (not sending EP with request)
 - Messaging
 - Test Extraction Pattern
 - Administrator
 - central source w/in carrier
 - carriers would say "here is who we want to have certs, roles and authority limits" and openIDL would coordinate for each
- Stat Agent
 - Application-level
 - AAIS (example) attaching extraction pattern (EP) to data call
 - Statistical Reporting
 - Stat agent in charge of the total reports that need to be done
 - stat agent: "If you are doing these lines in these states, these are the reports you are obligated to take part"
 - needs discussion - right now, REG is starting data call, 40 states would have to each create own data call, AAIS attaches EP individually for each line and each state
 - lot of repetition in this process
 - DH: auth of the handbook? "we are doing stat reporting as a stat agent" w/o having to have REG ask for the reports?
 - PA - way things work now, come march all inv in reports, current system REG asks for report, flipped
 - KS - goal of stat agent is resp for managing and completing stat reports, REG is resp for making data call, stat reporting resp for managing the stat reports, sys needs to work differently than it does now
 - JB - community of carriers reco capability for request of stat reports in future, other parties, done with the consent and agreement of the community of carriers, requests considered legit - stat reports happen, going to occur, not like adhoc data call, stat report provided by orgs that can do so -
 - DH - stat reports are specific and limited to whats in handbook and any other request or REQUESTOR is a data call
 - PA - NAIC Stat Handbook
 - JB - other orgs making data calls on behalf or other types of reporting, network and community recognizes - could have agency that does stat reporting
 - DH - role for stat agent for stat reporting - org that can do stat reports - do we need to recognize Stat Agents are accredited?
 - JB they are credentialed
- Service Org (analysis firm, infrastructure partner)
 - Data Calls
 - Role for the stat agent + others
 - DH - EPs would be a role for the stat agent (or Implementation Partner)
 - KS - not comfortable calling data calls a stat-agent-only responsibility
 - KS - regulators could do their own EPs (in house or contracted)
 - REGs not doing execution
 - KS - anyone could do it, inc regulator, if couldn't might have help
 - Extraction Pattern
 - Implementing an EP
 - Responding to feedback on an EP
- Regulator
 - Manage Data Calls (CRUD)
 - Collect / Review Reports
 - Messaging
 - DH - similar to a message board, request for data call is out there, when openIDL first discussed, REG asks for "I need info" and message board becomes forum for und intent, can provide feedback on what req is, how easy and readily available info will be, so REG could modify the request (part of the LIKE process)
 - CRUD EPs
- openIDL
 - Certificate Authority
 - Possible for each org to have their own CA but felt it would be simpler at beginning for openIDL to deliver certs
 - not require carriers to use existing CA
 - Coordinating services of CA, some degree of review of who applicants are, governance angle, some notion of membership and who is joining/on network
 - Network Monitoring
 - Network Administration
 - Implementation Support
- Infrastructure Partner
 - Set up nodes

- Maintain Nodes
- Report on Network Health
- KS - role at the top level, can provide support for carriers that want to participate and necessary level (onboarding, training, etc.) - they are knowledgeable and expert w/ tech stack and apps w/in openIDL, provide support where necessary - Use experience to recommend improvements to system overall, provide support
- Participate in governance
- Developer / Contributor
 - Some reports require implementation beyond EP
 - Application development running on openIDL
 - Participate in governance

Application Components

Data Sources, Sinks and Flows

Decisions

Tenets

Data

ID	Tenet
1	Data will be loaded in a timely manner as it becomes available.
2	HDS will track the most recent date that is available to query for pre and post edit package data.
3	Data owners will correct any mistakes as soon as they are made aware of the issue.
4	Data owners will follow current practices for logging policy and claim records as they do today. A new record will be created for each event. All records will be loaded in a timely manner after the creation event.
5	There will be a distinction between edited and unedited records. (Successfully gone thru edit package)
6	HDS data is attested to, some way to attest to meta data, date range up to which its good, capture some info about hds "data in HDS is good up to now for this purpose"

Non Functional Requirements (to be moved to requirements doc)

Notes:

Time	Item	Who	Notes