

5/24/2022 HDS Task Force Meeting

Date

24 May 2022

Antitrust Policy

Antitrust Policy Notice

Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrust-policy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.



Attendees

- Sean Bohan (openIDL)
- Megan (AAIS)
- Peter Antley (AAIS)
- Joan Zerkovich (AAIS)
- Truman Esmond (AAIS)
- David Reale (Travelers)
- Brian Hoffman (Travelers)
- James Madison (Hartford)
- Ken Sayers (AAIS)
- Jeff Braswell (openIDL)
- Nathan Southern (openIDL)
- Rajesh Sanjeevi (Chainyard)
- Satish Kasala (The Hartford)

Agenda:

1. Discuss HDS needs, requirements (see notes below)

Time	Item	Who	Notes

Action items

Notes:

- Data Model (Peter to send to Sean for wiki)
- Continue discussions
- Bring all up to speed
- History of this meeting
- More tech than business but more specific than implementation
- Taking the stat plan, as they have it, all the elements, data dictionary
- How to get into a queryable db
- Form HDS in such a way to satisfy reqs

- Business reqs, various schools of thought on warehousing
- Claims and claim events
- Conversation - how much overlap between homeowner claims, personal claims
- How generic to make the claims
- Break out a personal auto claim
- David - decision for highly normalized data model
- What is driving this decision
- James - good business meetings for monster list of elements
- Don't bother biz with normalization
- Break it down to some chunks
- How big do you chunk it up
- As soon as you chunk you talk about relating it
- Denormalize to some reasonable extent
- Excessive norm is an anti-pattern in his works
- Between ridiculously flat and wide and brutally normalized
- How do you glob entities together and relate them
- Not to say "relational db" discussion but we will have thousands of elements, strategic - answer large # of queries
- Inevitable 000s of elements
- Lot of stuff
- Chunk it up
- As we do this, tricky, represent the biz and optimize tech
- What are the entities and relationships
- First level of glowing
- David - sure
- It doesn't, existing data, doing this is complicated when loading
- More normalizing more difficult to take existing form of data and move it in
- What biz req is necessitating this?
- What ad hoc query is asking us to go this route
- Taking my flat file to this is not a trivial exercise, looks messy, needs business reason
- Not high performance (fast and efficient) system
- Maybe we can get away with less norm, harder to ad hoc query but easier to maintain
- Trade-off analysis
- James
- How norm do you want to get
- Not so much performance as semantics of the data
- Even if query is allowed to run over night
- How long can extraction pattern run?
- Will change how you struct model
- At this level, more semantic, some human is going to need to look at this model and know how to ask questions
- Thinking - aspire to answer a lot of questions from this model
- If you over normalize = insane, too flat just as confusing
- Balance point needed
- Need some biz reqs - what are those boundaries, run a pattern in hours v minutes v seconds
- 1000 flat and wide is tough, 100 or 400 could be ok
- How extensible
- More norm harder to adapt
- No good sense of all the data calls COULD be
- Not saying not where we end up, really don't see the need YET or at least hasn't seen discussion or biz req
- Not easy to take a flat file and turn it into this
- As carrier - happy to shift complex to data seekers to write harder queries vs making slick HDS super easy to query could lead to challenges on our end
- Where do we want to put most of the burden
- Ken - extractions something that regulator and more biz people need to understand
- Easier for them to grasp extraction
- Another force, makes sense to me, transactional loading of data, where you piece together policy or claim, look across to formulate
- Basic putting together policy and claim (over time)
- Abstraction so many are used to in this industry
- Satisfy
- Who writes extraction pattern?
- Ken - put onto the chain, is looked at by the folks who consent to extraction
- Und by looking at it or ask technical person to look at it
- Less complicate, more effective at consent
- Regulators say "this is the data I want", who is that technical person?
- AAIS? Carriers?
- Make sure we know who (entity, resource) is so we can formulate complexity
- Peter
- Eric as an example could write an extraction pattern
- GeorgeB - maybe not
- Brian - some easy boolean on top of this?
- Ken - high bar, lovely to get there, pure sequel obtainable
- Brian - we don't want Regs to be super technical, they know what they are asking for, technically what they would need to do in some way
- Wouldn't want them to be technical
- Joan - currently writing system that needs technical help, eventually anyone (biz, writer, etc.) can build query for the system
- Sure a lot will dev tools to make it easier and easier
- Defining HDS, gearing for policy and claims data, solid design, so tech people today can write extraction patterns - AAIS will be one but we want anyone to be able to do this
- We will develop tools to make it easier and easier
- Wouldn't focus on who is the person to write the question - today it is tech and over time less technical
- David - these reqs to level-set

- Backs into other requirements
- Flat-wide-dumb might be ok
- Highly normalized, much quicker
- Without those requirements not making informed decisions
- Joan
- First used case (Reg Reporting) takes a while to run, whereas ND is realtime queries
- Challenge - get solid design for policy/claims
- Once data is there, other queries you can see coming thru
- Policy/claims in a hurricane - can ask regs question and get quick response
- Understand design of DB, how to query quickly, get info, shift gears later
- Requirements - well designed db for policy and claims data
- Can have a design for large complex ? From a regulator
- Design so we can und where data elements are and can respond to a quicker query
- David - well designed db isn't a requirement
- Aspiration
- Joan - well designed = easy to understand, easy to und where the data goes, etc.
- Requirements are - db needs to serve regulatory queries, ok if they take while to run (min to hours to deliver), other req is und the data in the HDS so can also create smaller more responsive reports as needed
- Well-designed is "understandable" - Brians point, relatable, und by regulators, not too complex
- David - say "relatable" - diagram would not be intelligible to a regulator
- Not DB design thing, a tooling requirement so they can write queries
- Effective tools for NoSQL, etc.
- + good documentation of data
- Critical requirement
- Doesn't define how we should build the db
- What does are the performance requirements not "well designed"
- Req - what does this thing have to do
- Making a DB that is good at large data sets AND easy to use might not be successful
- Maybe side db for small queries vs ad hoc queries
- Discussing
- Jeff - tempting to look at this as a relational DB
- Higher level - semantic logical relationship of classes of info
- Diff lines of business
- Patterns going forward that will support that
- Logical relationships in components that make up data
- Can't combine down to a simpler physical representation
- David - no requirements
- Loading file way it is in one big table would work
- To say it wouldn't, need to requirements to say why
- Blindly undertaking efforts without requirements, why I don't think we are getting anywhere
- Jeff - take flat file, existing spreadsheet
- Reasonable to say, requirement that data model include diff lines of business in the future, reduce for simplicity if necessary
- Not paint ourselves into a corner
- Spend time on relationships
- Home, etc
- Effort to und the relationships is worth the effort
- David - not a technical exercise
- Relationships could be a good exercise
- This says "HDS" - we are deciding what HDS looks like, not theoretical modeling exercise
- Starting to look at tech and functional requirements of HDS
- If not doesn't belong at AWG
- James - this is the business logical model
- Need to get big list from main WG
- Focus here, business logical model, does bore and confuse people on Friday call
- Asked Dale a lot of biz question
- Ex: can any driver on a policy... system must represent any vehicle can be driven by any driver BUT there is also req or "Primary Driver"
- Understand - if we need to write all those down, lets
- While this is the HDS meeting, series of events
- 1. Long list
- 2. Business logical model
- At least 3 layers
- Idea you have layers = easy to put views on this, now down to physical
- David
- Und
- Should be writing down those models but
- A perfectly valid solution, load what we have
- Requirements - why we are doing this and the trade-offs
- Not trivial
- Can lead to problems
- Understand what the benefits we are getting
- Accomplish the same business outcomes
- If extensibility is the reason define what it could be
- Don't see the requirements
- Pattern - don't have requirements, just efforts
- Doesn't need thousands
- At least objectives, what are we trying to achieve
- (Model on screen) this is a lot more work
- James - my mind, reqs - will report on variety of diff entries in the future
- Do you want a report on geo, objects, detail

- Comfortable with idea we don't create entity
- David - gives members the scoping clarity, do that scoping exercise to guide the effort for the next couple quarters
- Feels like making decisions, doing work, don't und the reqs
- James
- Who does em
- 1. Why take all this policy, etc. - start with model of 1 flat table, why even separate policy from claim
- Process
- 1 table with policy and claim details
- Every claim another row in the table
- Straw man to say, sep policy from claim, work that path
- David - might end up with entity relation table
- Lots of flat and wide tables and a few Join tables
- Can decide which one is better based on requirements
- Respond to all ad hoc in seconds? Highly normalized
- 10-20 min? Maybe a few join tables with flat and wide
- James - sep biz semantics from performance
- Still at biz logical model table
- Hartford - limit to personal lines of 4 drivers and 5 vehicles
- More then you get into exception system
- COBOL
- No new rows for vehicles and drivers in table - policy with 3 drivers, 3 rows - 4 cars and 3 drivers = 12 rows
- Specific assertion
- Answer questions about policies and drivers
- David
- What are the things we are expecting this thing to do
- Once we have those make decisions
- Value in relational diagram of how this data relates to each other
- Diff exercise
- Biz level discussion
- How does it look when we add HO, but need requirements to dictate that
- Swiss Army knife of all possibilities, none done that well
- Never had project w/o function reqs defined early be successful
- Jeff
- Flexibility to cover diff use cases and patterns and line of business
- Analyze
- Physical model - biz logical model
- Infer how to simplify
- Exercise -
- David
- Just biz logical model - if we built today it would be this?
- Jeff - not necessarily
- Exercise - diff biz, property and casualty
- David - still dictating some categorization or groupings that will occur - JSON, tables,
- Not a decision made yet
- Could be a flat and wide
- This is part of AWG, needs to be guided by tech requirements
- Relational stuff is more of a RRDMWG
- Jeff - this is a hybrid meeting - started as a data meeting
- Turned into HDS
- Really data and the implementation intertwined
- David - still need requirements
- Brian - do others agree we need requirements before we get deep into this
- Joan - disagree we need level of requirements being asked for
- Data model needs to be understood by carriers and regulators and respond to requests that can take a while to run however the system needs to be built to use this for more than RR
- All queries could take an hour to run is not correct
- Bring to the group - this is not an enterprise application we are building, it is an information exchange service
- More flexible than ent policy system or data warehouse
- Design - can start with what is well understood as policy/claims data today
- Experienced optimizing
- Bring forward into HDS
- Can respond to Regulatory Reporting
- Not transactional but flexible enough to respond to other use cases
- Can't debate for months, get a good design out there and change based on experience
- David - going to keep debating without requirements
- Can't build quality without
- Joan - somewhere in between
- Don't need long formal reqs document
- Data understood
- Reg reporting well understood
- Industry understands it
- (Not knowing diff models in member data warehouses)
- Can be something transferred into HDS today
- Shooting for model we can begin using, evolve over time, better and better
- Und his request for requirements
- Middle ground here to get started
- Brian - don't believe this can be successful without good set of requirements
- Can't do it without strong requirements
- Focus of POC is stat data that can be delivered

- Doesn't have to be fast, understand ND and need to be immediate, deviated from what they settled on trying to do
- Tie it to requirements, uncomfortable thinking we can be successful
- Joan - don't think there is a problem there
- More concerned with carriers and relationships between data elements
- Once there can do ND data call
- Not instantaneous
- Auto data with a VIN #
- Use that for describing the vehicle, relating policy and claims,
- Keep in mind not anything too rigid
- WHERE IS THE SPREADSHEET AT 9:43
- If you look at data model would be used by ND or catastrophe data call
- More interested in data model, not transactional, can describe how quickly cert queries need response
- Needs to be one where Travelers or Hartford - can move data without a huge effort
- Not sure if it is relational or flat
- Need to put something out there and then start using it
- James - volunteer to take a whack at requirements
- From a modeling perspective
- Can always have a flat table rep everything
- Can solve any prob with flat and wide
- If willing to put out 12 rows you can do it
- If willing to deal with sparsity
- Non-functional requirement - can always solve with flat and wide with parent-child grain
- But brings up loading and querying challenges
- Can also solve
- Every table key and the value
- 75 = James (6th normal form)
- Can't solve all problems in 6th normal form
- Business will never give requirements that lead to a model
- David - agree with James, to make those decisions you need those requirements
- Tradeoff- flatt, norm, whatever
- Understand what we are trying to achieve
- Debates don't need to happen because ill-defined problem
- Jeff - requirements - extensible data model
- David - non functional - guiding principle, not a technical requirement he can achieve successfully
- Jeff - data model that can inc multiple lines of business with data properties
- David - must be extensible or multiple lines of business
- James - must be able to add a child to a parent
- Flat vehicles and drivers
- If you add a vehicle attribute you disrupt the policy structure
- David - those we could measure against
- Not that others are valuable NFR, user stories, guide discussions
- Concerned - what are we trying to achieve, should we do this, extra layer, or not
- Satisfy this or not and satisfy that
- Otherwise just debating things based on feel
- James - Satish and James will take a stab at requirements
- Tendency to over engineer
- James - commit to two weeks (from 5/24?)
- Is a vision question
- Forgetting execution
- Put flat model on blockchain
- Do aspire to answer all manner of arbitrary questions
- Inherently will start breaking up
- Joan -flexibility
- Focus efforts to get something done
- Today Regulatory Reporting, regulators have 1000+ ad hoc queries, take all those elements, do RR on this platform, design system to respond to that
- Timeframe - we want it with some efficiency respond to reports regulators have asked for
- Establish first design
- Meet needs of ND data call or catastrophe data call
- Regulators clear - more detailed info, reporting with more detail
- Data elements today that you don't have in stat reporting data warehouse which they will want to see reporting that type of Data
- Whatever the design is, something carriers can respond to soon
- Do a db design that is the best thing we can think of today systems
- Can't easily get data into HDS - issues there too
- Relatable aspect, fuzzy requirement, does speak to that transition
- Legacy systems, new systems
- Design people can use, make it better over time
- Peter - not trying to challenge flat file w/in proof of concept
- Watercolor discussion that has grown, AWG
- Do reports both ways
- 100% behind getting stat reports done with auto

Recording:



openIDL_Arch_H..._TF_052422.mp4