Analytics Engine Squad Notes

Directory

- To Address Next Meeting
  - Meeting Notes
    - 2020-08-21
    - 2020-08-14
    - 2020-08-07
    - 2020-07-31: First Squad Call
    - 2020-07-30: Focused discussion on PIH's ETL work
    - 2020-07-16: Identify considerations for squad success

To Address Next Meeting

  i. DUC: project - pepfar etl for generating MER indicators on FHIR?
  ii. Discussion on the road-map; request for feedback
      - Schema selection choices and flexibility
      - Schema evaluation (e.g. OMOP, SQL; is no-SQL approach possible?)
  iii. Work tracking process (Bugs, tickets) while working towards prototype?
  iv. How we consolidate our code, code review process & how we expect this to change over time; expectations for merge/release frequency
  v. Address nuances contributors are working with
     1. Requirements of external agencies
        a. Example at Google. Decision: Best if everything we develop lives under OMRS.
  vi. Time offering
      a. Rules of thumb we’ll follow for decision making
         i. For reference: Decision Making Playbook
         ii. Who has decision making authority
         iii. What to do when lack of consensus is blocking forward movement

Meeting Notes

2020-08-21

Attendees: allan kimaina, Bashir Sadjad, Burke Mamlin, Ian Bacher, Jennifer Antilla, Juliet Wamalwa, Kenneth Ochieng, Piotr Mankowski, Tracy, Grace Potma

- Dev. updates
  - Batch implementation pushed to same GH repo. Demo’d recently to FHIR squad. https://github.com/GoogleCloudPlatform/openmrs-fhir-analytics
  - Piotr working on dockerized version of streaming app to synchronized between an OMRS and HAPI FHIR instance
  - POC needed in 6 weeks - so Piotr working on getting atomfeed solution cleaner/simpler
- Review of MVP goals
- OMOP/schema feedback
• Presentation from Alan:

- Proposal: Use FHIR as intermediate
- Discussion: How to approach building FHIR schema to prepare for other use cases that would otherwise be forced to create their own pipeline/engine if they don’t use FHIR
- Need to support other formats as well
- Discussion on Talk: To schema or not to schema... deciding on a foundation for our analytics engine
- Proposals for issue tracking, code location, and code review (TL;DR: GitHub)
- Agreed on using GitHub for issue tracking in this squad

• Bashir on Vacation next 2 weeks

2020-08-14

Attendees: Antony, Burke, Allan, Bashir, Christina, Daniel, Debbie Munson, Jayasanka, Jen, Juliet, Tracy

• Welcome!
  - Antony (Palladium), Debbie (PIH)
• Announcements
  - Awesome job on Squad Showcase!
  - Did you notice we’re now Analytics Engine Squad? For example, our zoom call shortlink is https://om.rs/zoomaes
  - Renamed #etl Slack channel to #analytics-engine to reflect this
• Dev. updates
  - Bashir: Code for monitoring changes, transforming data later into a FHIR resource. Opensourced now https://github.com/GoogleCloudPlatform/openmrs-fhir-analytics
    - Location: Google Cloud Platform for now while still getting started. Apache License. Can go to OMRS if that’s a better place in future.
    - Please check it out!
  - Allan: Evaluating w/ Cornelius (Ampath FHIR expert) - went through data tables. Found 70% of Ampath data represented w/ FHIR; what about other 30%? Looking to expand that.
    - To use FHIR as intermediate storage, we definitely need a custom schema that covers that other 30%.
    - Challenge: How to limit this so people don’t depend too much on custom schema. This wouldn’t realize the benefit of the entire pipeline.
    - Some implementers not comfortable with FHIR - this would enable them to use custom schema. But, adds scope: Requires us to support two schemas.
• Continue discussion on data warehouse schema:
  - Still need to discuss schema. E.g. OMOP.
    - API to access schema? At Google, history seemed to be SQL was trusted, i.e., SQL as an API for accessing data. The context is BigTable to SQL based APIs migration.
    - At the end of the day, the query complexities are still there, even if we pick a non-SQL API.
    - For end-users: Agreed that consumers need to have tooling at the end that allows them to get the data out - even without knowledge of SQL.
    - For under the hood - Schema Flexibility: Concern around committing to relational database approach if we don’t have to
      - Nice because people can bring whatever they want
      - Hard because it will be harder to build things on top
    - E.g. Palladium: does development, other players consume data. Need frameworks that empower those users to come up with their own reports. So many funding agencies that all have their own reports (e.g. CDC, USAID, DOD), have different report requests at least every week! Need to be readily consumable for ad-hoc reporting. Where everyone understands queries for what’s in the system.
    - Being able to perform ad-hoc reports could be an argument for SQL
    - Not clear that OMOP has wider adoption than FHIR. Well adopted in clinical research community.
• Keep researching/asking? Or just start with FHIR?
  - Using FHIR for data analytics tools is already happening. Unclear: Are people widely comfortable with OMOP? Easier to work with. Primarily designed for de-identified data.
  - We always have the option for FHIROMOP in future as there are open source tools for that transformation.
OMRS quite involved helping set up OHDSI community
FHIR background is to support both data structure for SHR and for analytics
• Where do we go from here?
  • TODO: Allan to review OMOP
  • TODO: Grace to set up meeting w/ Squad & James/Carl re. ADX/OMOP advice
  • TODO: Burke to start Talk conversation around initial schema (w/ link to Analytics on FHIR)

• Concrete MVP goals:
  • Single machine pipeline, while easily scalable (note on MySQL bottleneck)
  • Horizontally scalable warehouse
  • Metrics API with a minimum set of indicators
    • Implement the 10 common indicators; show people how quickly they can be calculated. Idea is this could be a separate module that uses the warehouse.
    • Deploy as docker image; they get an end point they can query to issue queries to underlying data. But has to be scaleable. OMRS itself not horizontally scalable. MySQL would be the bottleneck for scalability in the data pipelines.

2020-08-07
• Recording:
  • Attendees
    • Bashir Sadjad Burke Mamlin
  • Updates
    • TODO - Grace set up call w/ James K & squad tech leads
    • TODO - Grace consolidate visuals of ETL demos from implementers
    • Deidentification of randomized data set ongoing by Allan at AMPATH
    • Bashir experimenting with converting data to FHIR resources and pushing to a FHIR warehouse; includes bulk-update. To make open source code available to all next week.
      • This code can be used to extract OMRS data as FHIR resources. Generic tool, not just for analytics, even though original use case was to push to FHIR warehouses.
    • Review of hot-spots for run times
  • Proof of Concept
    • Need to i.d. the intermediate schema is that they're okay with (e.g. some for/against FHIR) - Is there any other analytics schema out there that makes it easy to come up with the concepts around it?
    • OK to use SQL as main interface? (commitment that end product would be used)
    • Need to i.d. intermediate schema, assuming it's not FHIR.
      • Just build and get feedback from prototype?
      • Do we need separate mechanisms for standardized indicators?
        • Need to get to a data warehouse where we can get to those specific features; maybe phase 2 we get to specific querying
        • TODO: Tracy to post and s
    • Other standardized schema examples: OMOP, I2D2
      • OMOP use case suggestion: Different way to transform the data into a standardized format, to have common data for analytics. Another way of trying to do data analytics / warehousing at scale. Advantages is that because it's been standardized as a model, a lot of tools being built against it.
      • Existing libraries for converting FHIR to OMOP
    • Building on REST API FHIR store loses some of the advantages of storing in SQL
    • Need to support not-cloud-based: We can provide tools (e.g. Spark), but responsibility of implementers to manage their own cluster if they want scalability beyond single node.
      • Would be nice to provide these sites w/ different versions of Docker component files for such cases.
    • "Engine"
    • Deidentification

2020-07-31: First Squad Call
• Recording: https://iu.mediaspace.kaltura.com/media/Analytics+Squad/1_id8660id
• Attendees
  • Burke Mamlin, Allan kimaina, Bashir Sadjad, Christina White, Daniel Kayiwa, Dimitri R, Ian Bacher, @Jayasanka, Jen, JJ, Mike, Steven, Tracy
• Introductions
  • Technical Leads Allan & Bashir
  • Mekom: Keen to include analytic platform in work for clients
  • Jayasanka: Working on DHIS2 reporting module, love for analytics/viz
  • Jayasanka: Working on DHIS2 reporting module, love for analytics/viz
  • AMPATH: Have been supporting hacky ETL solution; excited to have more robust solution
  • PIH: Often less than ideal approaches to ETL & analytics, excited for something that helps guide things forward
  • KenyaEMR: Steve - historically has seen reporting solutions have all been suboptimal – e.g. having a ministry report running all night
  • ITECH: Christina: Hoping to implement in Haiti. Tracy: secondary use of data at large scale; researcher of predictive analytics & clinical decision support
• Our prototype goals
  1. Clarify purpose as prototyping (it’s unknown what final tool/product will look like)
    a. Specific indicators
    b. Scalability (run on 1+ nodes)
    c. Goal is to create something useable soon (i.e., get to a working prototype people can actually use)
    d. Lessons learned from OMRS19:
1. Common need for flattening encounters (e.g., people trying to analyze How many patients did we see where X happened?)
   1. Creating a table for every form - an idea from prototype approach that suggests how it will work to automatically produce e.g. a CSV for each form

2. Lessons learned from BIRT & others:
   1. Our data model is constantly changing; the questions you want to answer are always new (nature of medical info - which test is done changes year to year)
   2. Need tooling that’s at production level to not have to define our whole schema up front. Use standards & tooling conventions wherever possible.

3. Need to echo standards so we don’t have to build multiple aspects - tools coming to existence built on standards (e.g. expect to see more of this w/ FHIR). Want tooling we can support in 1-2 years.
   1. FHIR as (1) first intermediary step/data store w/ initial transforms, +/- (2) build your own extension as needed w/ transforms you need
   2. There will no doubt be efforts outside of OpenMRS to transform FHIR to OHDSI’s common data model
   3. ITECH: working on prototyping way to pipeline data into a side-by-side HAPI FHIR store for q. OMRS instance - hope is to use some kind of ETL
   4. Bashir has been experimenting with extracting into FHIR schema
   5. Will need to discuss schema more!

4. Leveraging recent community EIP work
   1. Debezium chosen as way to broadcast elements, create an outbound from OMRS to get data from
   2. Now need tool to help route messages (and well dev/d designed, saves ++ dev time) - Apache Camel well-suited, but can’t drop Camel on EIP side
   3. OMRS should be able to spout out FHIR messages out of the box. Until it can, need to identify the layer that does this and who’s getting it done. Ideally FHIR subscription model is what OMRS should be able to do. So “if you need X data, subscribe to OMRS and FHIR model can generate for you”. Mekom willing to support full-time developing ability for OMRS to support doing this.
      1. ETL tech layer should expect to receive FHIR data - not the job of the ETL pipeline. Extract FHIR and transform as needed from there.
      2. Need reliable event generating system. That’s what the pipeline pulls from. (Right now in OMRS this is basically MySQL)

- How often we meet
  - Weekly
  - 1. First priorities:
    - a. Requirements & what’s getting built - ground in prototypes groups have already made
    - b. One prototype every squad call? (20% of time on prototypes every week)
    - c. Send doodle (or Slack poll) for ?Wednesday, next presenters
  - 3. Sprint structure

- Where we collaborate digitally
  - 1. We use Talk for discussions and decision-making (documenting decisions)
  - 2. We use Slack #etl to chat (and assume those discussions will never be seen again)

- Data environment
  - 1. Centralized server with unified sample dataset that we can use to do development and analysis
  - 2. Cloud vs On Premise

- Any other role clarification

2020-07-30: Focused discussion on PIH’s ETL work

Attendees: allan kimaina Bashir Sadjad Ian Bacher Grace Potma Mike Seaton Mark Goodrich

- PIH’s ETL infrastructure:
  - https://github.com/PIH/petl
  - Chose Pentaho because concern about need for other devs who understand more cutting edge tech if they chose a different option.
  - This tool executes pipelines: Pentaho separates them. Analysts using PowerBI to create reports; don’t write in SQL. PowerBI couldn’t directly suck data from MySQL. Approach was crashing machines of end users; and it became clear that PowerBI was going to be what end users were using to review analytics. PIH added a MySQL to MS SQL-server bridge since PowerBI can consume data from SQL-server directly.
  - Is the goal to calculate some fairly standard metrics or do they vary from site to site? They vary and there data analysts who are not SWEs but know how to use data, e.g., through SQL. They may have custom needs.
  - Another source of variability is minor differences between different OpenMRS implementations.

- FHIR:
  - Requiring FHIR would be a big concern because not useable/friendly enough for end-user Analysts.
No way the users of their analytics tools would be able to write a query like this:

```
DECLARE target_code STRING DEFAULT "3141-9"; # This should be replaced with UCL code in a real scenario.
DECLARE target_system STRING DEFAULT "http://fhi6.org";
DROP FUNCTION IF EXISTS fhir_test_analytics.timestamp_str_to_date;
CREATE FUNCTION fhir_test_analytics.timestamp_str_to_date(timestamp_str TIMESTAMP AS DATE(CAST(timestamp_str AS TIMESTAMP)));
```

But we need some middle schema anyways, can that middle schema be FHIR? Potentially, yes. It also depends on the adoption of things like SQL-on-FHIR.

There is an agreement that if we want the middle layer to be flexible enough to do various analysis, its data model and queries will be inherently complex.

- The current main pain points:
  - **How to get data out of OMRS**: currently it is through cron jobs that recreate data warehouse tables from scratch. Incremental support is definitely useful.
  - **Dealing with OMRS data model** is not easy.
  - Is the current PowerBI system work on a single machine or a cluster? How many patients and observations are we talking about?
    - Yes, PowerBI runs on a single machine.
    - One of PIH's bigger installations has ~400K patients and ~40M observations.
  - People want to be able to drill down to actual data, so just providing the target metric is not enough.

**2020-07-16: Identify considerations for squad success**

Attendees: Allan Kimaina Bashir Sadjad Ian Bacher Jennifer Antilla Grace Potma

a. Need for a separate non-MySQL platform?
   - i. Bashir developing things beyond MySQL already - but do we really need a non-MySQL-based solution?
   - ii. Work of squad could be just to write those scripts
   - iii. Aim: Not to be locked-in to MySQL GSoC project working on support for Postgres b/c right now other groups having to install MySQL just to run on OMRS
   - iv. Unclear: How possible is it to have a solution that runs across different SQL servers? (SQL supposed to be a standard but reality is it gets implemented differently across multiple vendors problem for OMRS)
   - v. Posed these questions in Bashir’s FHIR Analytics Google Doc
   - vi. Calculations of concepts in background end up causing headaches - this ETL approach doesn’t scale @ Ampath; KenyaEMR have similar experience
     1. Might be fine when you have small, single sites; but when you want to do data analysis on all them, you need a data warehouse where a single MySQL approach won’t work

b. Takeaway: Not going to be a single MySQL approach, because this doesn’t scale and doesn’t address the issues IN’s have already run into

   a. Final output and its usage
     - i. People used to writing SQL - need a tech that someone in M&E/Data Science can simply write SQL to generate a report

   b. Requirement: Whatever we do, ppl need to be able to run SQL on the solution
      - i. Spark? Hive? Others from Hadoop+?
      - ii. Small deployments (10’s of clinics) - should be easy for them to use in single processing environment (on device on location); shouldn’t assume everyone knows how to bring up a cluster w/ 10’s of nodes
      - iii. Output could be a MySQL DB

   b. Requirement: Whatever metric is required, there should be support to drill down further, all the way down to 1 record (easily able to go down to pt records)
      1. E.g. “Aggregate the # of pts with VL Suppression.” Ppl see # and say “this is an under/over report!”
         - i. Ability to return to MySQL

   b. Requirement: It should be easy to use the data from within OMRS (exposing the data in OMRS should be easy)
      1. Quickest way is through MySQL. But exporting data back into MySQL can become a bottleneck
      2. Module could be the reporting module, tweaked to be able to work with this.

   b. Requirement: Data in warehouse should be de-identified
      - i. Feature? Issue is compliance.
      - ii. Alternative: Access Management
a. Deployment requirements

- MVP Requirement: Runs on single node
- MVP+ Requirement: Runs on multiple nodes

1. Everything should just work regardless of how many VMs (single or 10’s) - does not require/assume cluster management. E.g. shouldn’t expose dataflow clusters to end users

- MVP Requirement: Provide simple notebook with examples on how to use, query, specific transformations

1. Don’t assume ppl know how to run spark cluster or connect data to mySQL DB

a. Others working on ETL: UW+I-TECH, Mekom, PIH, etc.
   i. These are the ppl to regularly engage with when we set up squad (even if all they do is attend weekly squad meetings to give feedback & contribute)
      1. I-TECH: Meeting w/ Jan & Tracy (to work on I-TECH’s ETL work)
      2. Mekom: DB Sync work - separate binary that syncs multiple OMRS deployments
     3. PIH: Mike & Deb & Mark re. How they’re using COVID19 as case study for ETL pipeline

b. Team/work/processes etc.
   i. Intro of Technical Leads
   ii. Our prototype goals
      1. Clarify purpose as prototyping (it’s unknown what final tool/product will look like)
   iii. How often we meet
      1. E.g. weekly vs q2 weeks, with open collaboration session other week
   iv. Where we collaborate digitally
      1. We use Talk to ____
      2. We use Slack to ____
   v. Data environment
      1. Centralized server with unified sample dataset that we can use to do development and analysis
      2. Cloud vs On Premise
   vi. Any other role clarification
   vii. Work tracking process (Bugs, tickets) while working towards prototype?
   viii. How we consolidate our code, code review process & how we expect this to change over time; expectations for merge /release frequency
   ix. Address nuances contributors are working with
      1. Requirements of external agencies - e.g. @ Google all OS forked repos must live under a Google Org (would hurt visibility)
   a. Bashir to f/u at Google.

- Decision: Best if everything we develop lives under OMRS.

   i. Time offering
      a. Rules of thumb we’ll follow for decision making
         i. For reference: Decision Making Playbook
         ii. Who has decision making authority
         iii. What to do when lack of consensus is blocking forward movement

1. Champion sites (these would be the people who’d use our outputs when we release them)

   a. Main champion site at this point = Ampath b/c they have biggest DB (& Allan’s expertise!) - able to say “with a DB this size, we can run these obs and metrics successfully” so it should work for smaller sites
   b. Elevator pitch: Solution that shortens the time and improves the quality of using OMRS data in indicator reporting, reduces unplanned technical team member overtime, and makes it easy to drill down to patient-level data to confirm the numbers are correct.

DONE: Grace post announcement on Talk w/ Doodle; Grace combine requirements into Requirements page.