Engineering Plan

Definitions

- User event data is actions and associated metadata we track in the application. They're like verbs. Such as a user clicking on request showing, a user subscribing to the newsletter, a user signing up.
- Application data is entities that are part of the Aalto application. They're like nouns. Such as a listing, a user, a listing preference, etc.

Principles

- All business data needs to adhere to a data dictionary we define, that's understandable by the whole company, and have a centralized location in which we can audit data flow.
- It should be easy to self-serve data about our users.
- It should be easy to analyze new features using user event data and application data.

- Analytics layers designed for consumption by business users should not be re-used by engineering application logic.

Measuring Success

- Data collection is reliable and accurately reflects all of our user event data.
- User event data is stored in a data warehouse of our own control.
- Application data can be cross-referenced with user event data in Metabase.
- Data customers trained in how to use Metabase and can self serve requests with confidence.

Assumptions

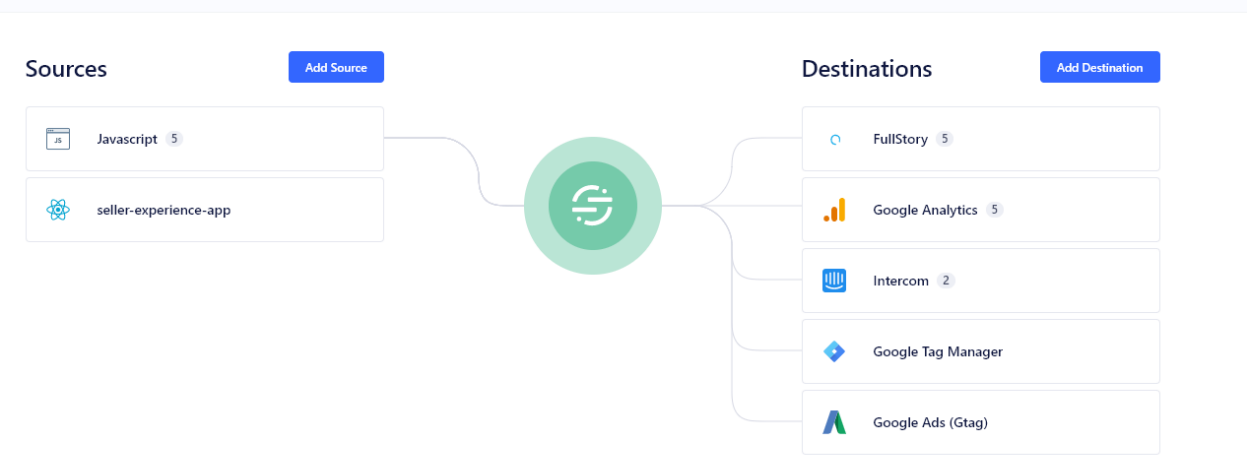
- Metabase will we the primary business intelligence tool to analyze user event and application data.
- We will not be tracking email opens from Postmark in scope for this project.
- Newsletter users will continue to exist solely in Intercom.
- CRM solution will be decided upon by business in late February

Non-goals

- Funneling application data to Intercom (such as listing preferences)

Things to consider

All of our user actions flow through Segment in order to be directed to multiple data sinks. This should be treated as our authoritative data flow; when we implement additional user level tracking, it must either be in this pipeline or an artifact created from this pipeline to be considered part of our data dictionary.



The Plan

0. Proof of Concept to Query User Event Data in Metabase

We are fairly certain that we will be going with BigQuery as a destination for User Event Data. In order to test the viability as early as possible in this project and verify our assumptions, we will be connecting BigQuery as a Segment destination prior to any kind of data cleanup, and connecting it inside of Metabase. This will allow us to test the ability to query our dataset early, present it to stakeholders, and iterate as soon as possible.

1. Audit and Clean Up Segment Destinations

Who is using analytics pipeline tools and how? Specifically who is using the destinations in Segment?

- Fullstory
- Google Analytics
- Intercom
- Facebook
- Google Tag Manager
- Google Ads

Can we get rid of any of the destinations in Segment (specifically Google Analytics, Google Tag Manager, Google Ads). If so, let's do it in this phase.

2. Investigate Data Governance for User Event Data

We want to ensure the events we're tracking are the events we've agreed we want to track. To do that we may want a data governance layer. Investigate options for this (potentially Segment Protocols may be a solution). Propose a solution (or not) that includes cost, cost/benefit analysis, and why it's a good idea.

Implementation Notes

- Include a plan for team members to update the data dictionary when new events/features are spec'd
- Provide guidance on naming conventions for user event data

3. Rebuild User Event Tracking

Data Bankruptcy

Right now we have unreliable user event data. We have that because we haven't intentionally defined what actions we want to track in the application in a canonical area. Remove all current existing event tracking and implement the new event tracking from the data dictionary. Alternative to this we can allow the existing data tracking and the new proposed events side by side and audit and merge the two. It's not worth it to do this because we already don't trust the data we have now, so let's just save time and chuck it.

The step involves doing the following...

1. Remove all existing event tracking to prevent bad data from entering the system.
2. Implement data governance layer (as decided in 2. Investigate Data Governance for User Event Data)
3. Implement new user event tracking as defined in data dictionary .
4. Test and verify analytics events are firing and properly collecting data

Implementation Notes

- We want to track all page views as a special type of user event data, exposed as rows in the same table.
- We want the same metadata for page views as for user event data (including session id, experiment grouping, etc).
- We want to piggyback off of Fullstory Session IDs for user event data (`FS.getCurrentSession()`)

4. Surface User Event and Application Data

1. Decide on the data source for analytics (should it be BigQuery?) and set it up.
2. Export user event data into a data source which can be read by metabase
3. Annotate the schema in metabase so its self serve level easy to understand.
4. Audit application data and create `analytics_x` views for each application entity that's valuable for analysis with an annotated and easy to understand schema.
5. Clean up Metabase so its easy to see the good data.
6. Create a user preferences dashboard that visualizes neighborhood areas and trends with a heatmap.

Additional Sources:

Feedback

- ~~@Craig Moxley – @Feb 2, 2021~~
- ~~@Hayley Gibson – @Feb 2, 2021~~
- ~~@Al Urim – @Feb 2, 2021~~