# Title: Deal insights - January 2019

## Team:

Engineering: Sarthak Dev

Data Science: Sai Charan, Jagadeesh

Leads: Swami, Ajeet, Srivatsan

PM: Aditi Balaji

## Problem Statement:

To TAG deals as "Trending UP" or "Slipping Down" and also give [Causal](https://en.wikipedia.org/wiki/Causal_inference) reasoning for why the deal is "Trending UP" or "Slipping Down". Additionally, we must also identify the context in which we should show/not show the TAG and Signals. Thus deal insights is a layer on TOP of the predictive deal scoring feature (the earlier scope).

## Roadmap:

* Get the stuff working **ASAP**

# Data Science:

**STAGE 1: Calculating the time delta (in days) for observing a significant change for each account**

* For V1, the time delta will be calculated for each account based on their historic time to deal closure from lead creation
* This time delta will be used to measure the differences in the sales activities (during the lifetime of a deal)

**STAGE 2: Identifying Interesting Deals for each account**

* Deals where there is a ***significant movement in deal scores*** in the time period delta-t are labelled as interesting deals

**STAGE 3: Identifying the signals for Deal Score changes for each deal**

* Upon identifying the interesting deals we will use the activities happened in the delta-t and ***identify key signals which are impacting deal score movements***
* If needed, we will build a second layer model to get weight estimates
* We will focus on keeping things simple and ship a minimum viable experience ready while keeping the code/architecture simple for extension

**Stage 4: Context on when to show/not-show the signals**

|  |
| --- |
| Possible Signals |
| Average Response Time of the Lead |
| # Reach-outs by Lead |
| App events and Website Events by Lead/Contact |
| Email Metrics: # SENT, # RESPONSES, # OPENED, # CLICKED, # BOUNCED |
| Phone metrics: # Incoming Calls, # Outgoing Calls, Duration of Incoming & Outgoing Calls |
| # Appointments Created/Modified |
| # Notes Created/Modified |
| # Tasks created/Modified |
| # Chat Created |

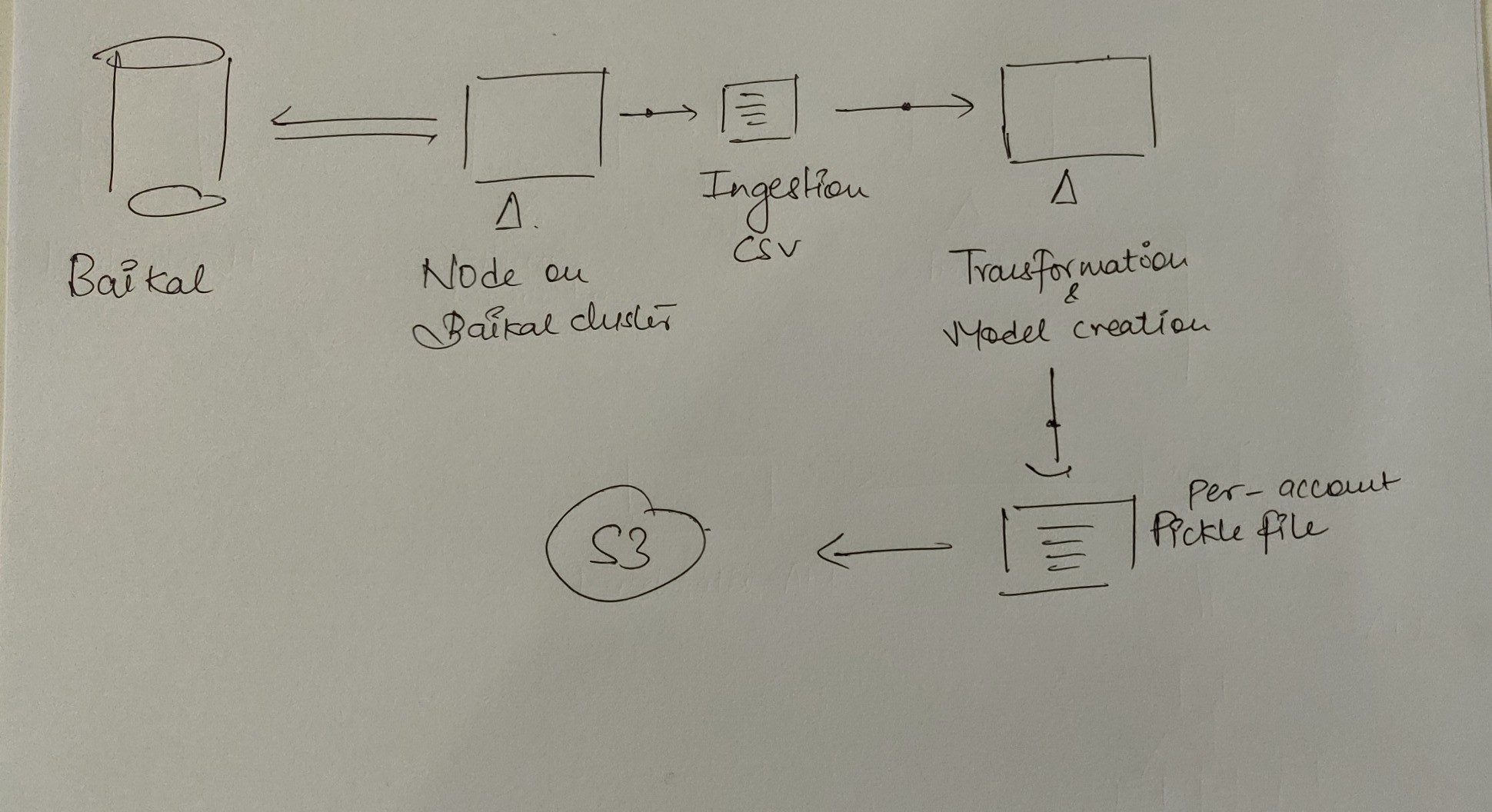
## Engineering:

The project will have two primary functions:

* Training
* Prediction

### Training

For training the model, we will use **Baikal** as the data source, since it has a *complete replica of the Freshsales database*. Being a data source at rest, it will give us the flexibility to make heavy Hive/Impala queries. As a result, the codebase will also be placed inside a new machine inside the Baikal cluster (only machines inside the cluster have access to Baikal data).



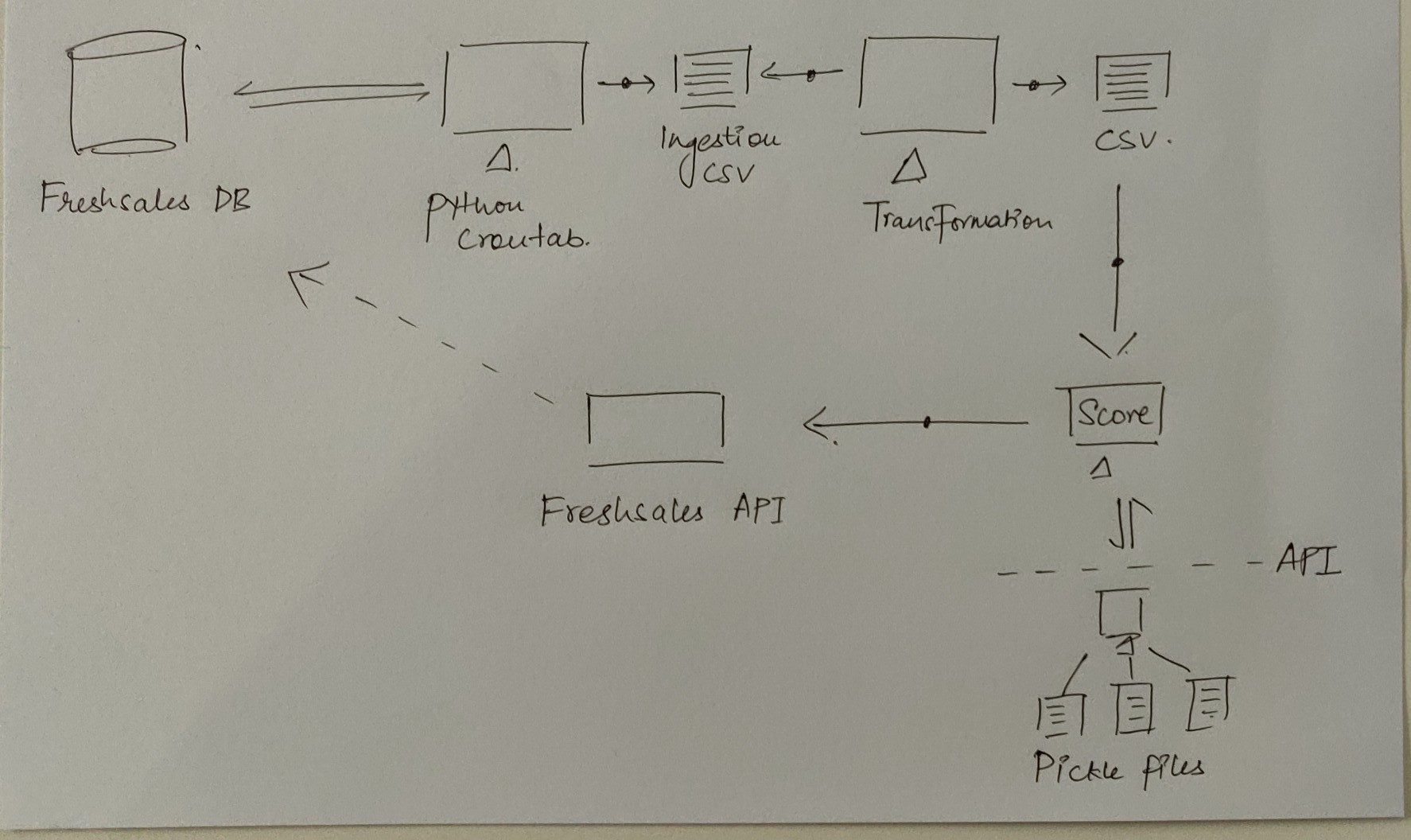
The code workflow will be designed to generate a CSV at every stage (ingestion, transformation, model training/creation), to have full visibility of data retrospectively. Every stage will also have a due monitoring process for health-checks (dots on the diagram). Barring the transformation and training code, the rest of the code will be written in Python.

The models will be stored as pickle files. On creation, these files will be uploaded to our S3 bucket.

*Note:* This section will be amended with the model training architecture upon further clarity there.

### Prediction

The prediction will run once every 30 minutes. The data source, due to the immediacy, will be the **Freshsales RDS**. The ingestion code will keep a log of the *last\_retrieved* timestamp.



The codebase will be placed inside the Freshsales stack itself. The flow will be similar to the training process, where every stage will generate a CSV, and have a set of monitoring health checks (dots on the diagram). For getting the score, the running cronjob will make a HTTP request to a lightweight microservice which will return with prediction results.

This microservice will be responsible only for storing the latest model pickle files and predicting results through them.