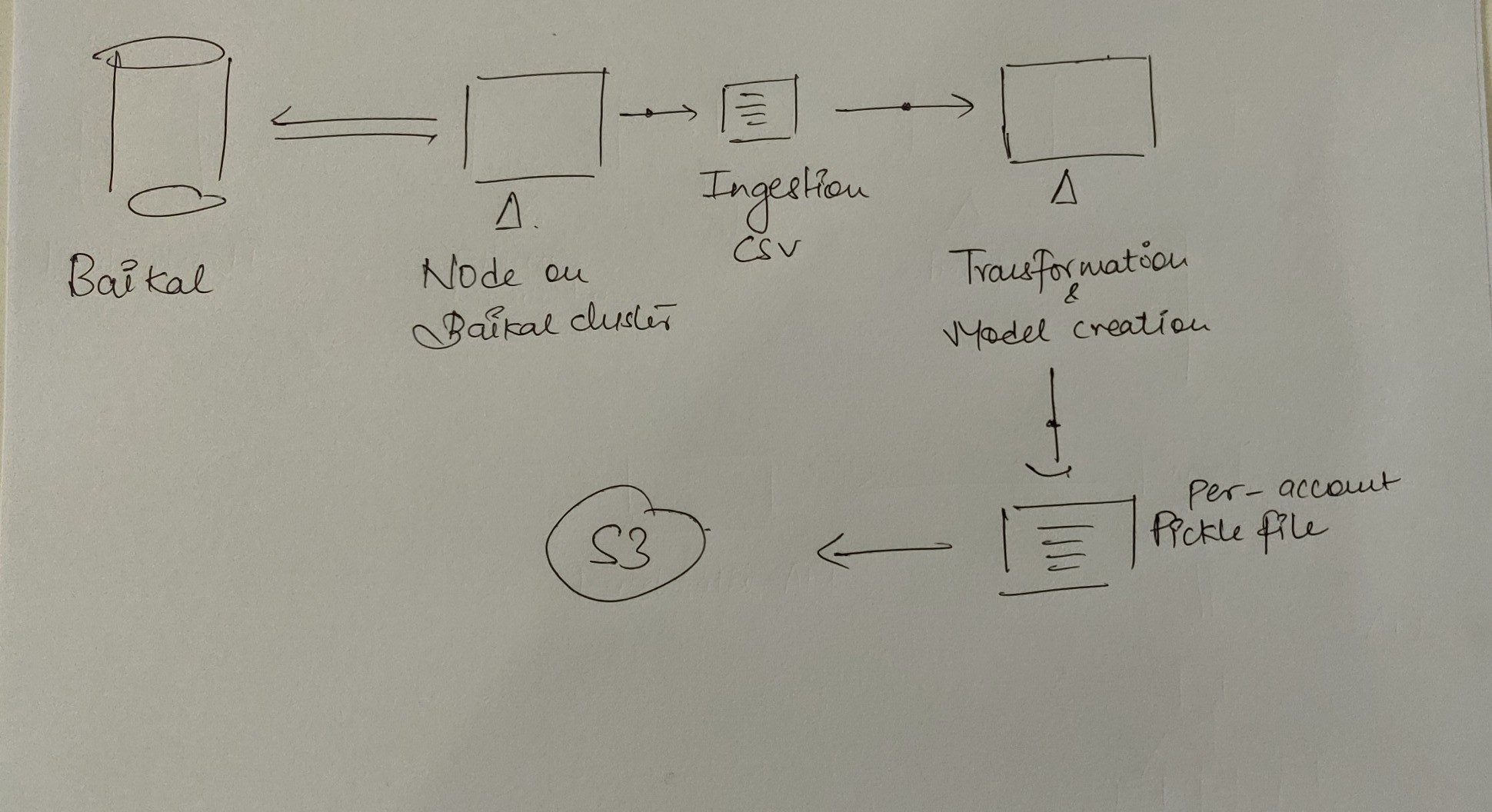
# Title: Deal insights

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).



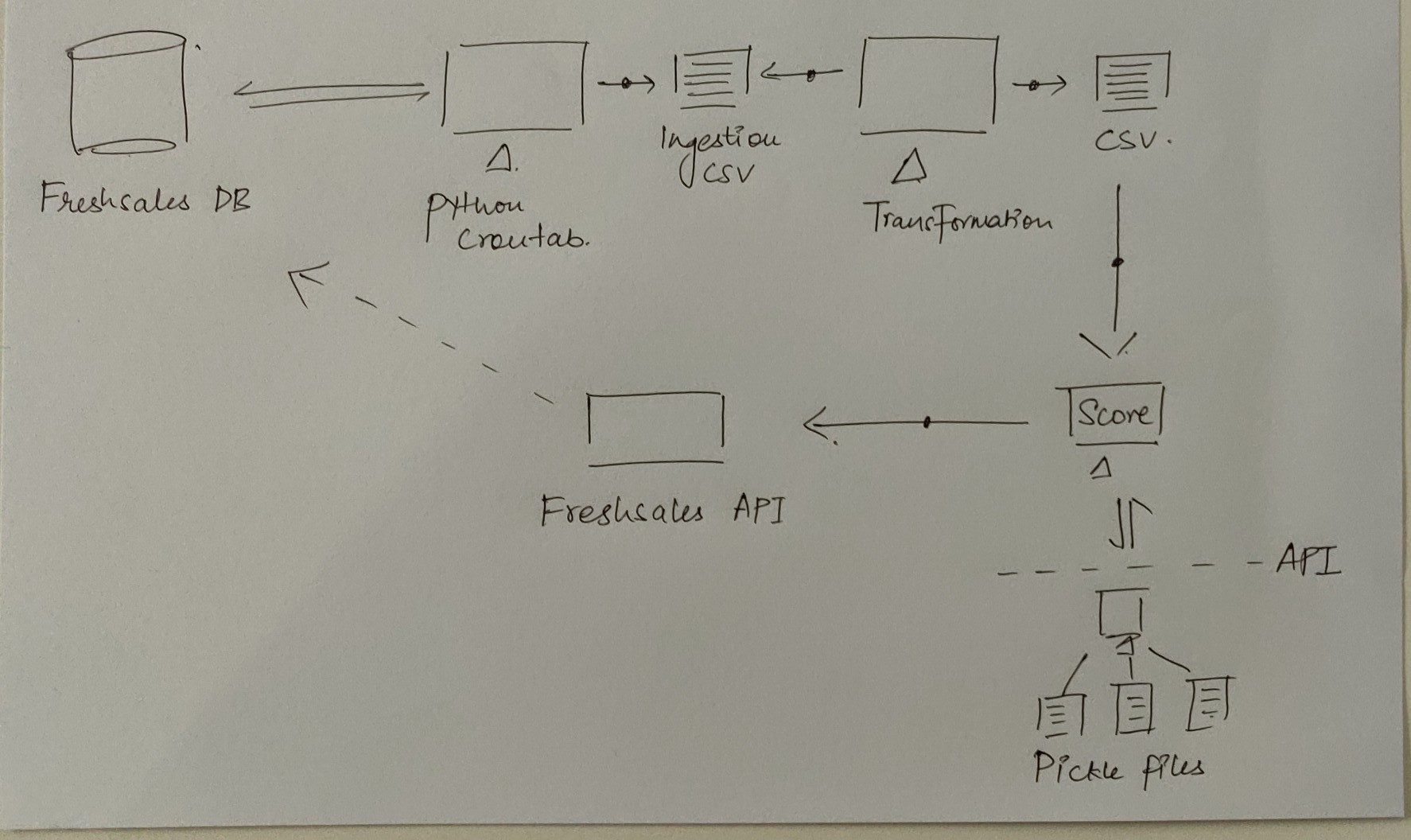
The code workflow will be designed to generate a CSV at every stage (ingestion, transformation, model training/creation), to have full visibility of data retrospectively. Every stage will also have a due monitoring process for health-checks (dots on the diagram). Barring the transformation and training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

The first version of the schema which we'll be sending as a payload back to Central

{

    'deal\_id' : 1,

    'tags': ['a', 'b'],

    'next\_best\_action': 'Follow up',

    'dominant\_signal': 'email',

    '\_meta': [

        {

            'signal': 'email',

            'score': 0.67,

            'interpret': 'This happened'

        },

        {

            'signal': 'call',

            'score': 0.67,

            'interpret': 'That happened'

        }

    ]

}

# 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  
  *(25th  percentile will be used for now)*
* This time delta will be used to measure the differences in the sales activities

**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  
  *(slipping down, trending up were the two tags discussed in the last meeting)*

**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 significantly the 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**

* We will go with the existing heuristics which was discussed in last meeting

|  |  |  |
| --- | --- | --- |
| Possible Signal | Context | Inference |
| First Response time by the lead/contact | First week, or Untill the first few responses | 1) SCORE (0-100) 2) HIGH, MED, LOW IMPACT 3) Visual FILL. 4) Explain based on historic conversions for this pattern |
| Median Response Time of the Lead | When there is more than 2 responses and as long as the lead doesn't get stale or Untill there was a response in the last 2 weeks |
| No of Reach-outs by Lead/Contact | When there is more than 2 responses and As long as the lead doesn't get stale or Untill there was a response in the last 4 weeks |
| App events and Website Events by Lead/Contact | If any event occured in last 30 days |
| Recent Activities (last 7 days) | If event occured in last 7 days |
| Email Metrics | If Email occured in last 7 days |
| Phone metrics | If Call occured in last 7 days |
| No of Appointments created/Modified | If Appointment created in last 30 days or modifiied in last 7 days |
| No of Notes created/Modified | If any notes were created in last 7 days |
| No of Tasks created/Modified | If any tasks were created/modified in last 7 days |
| Chat Created | If any chats were created in last 7 days |

* Going forward, we will adjust it as per feedback from end-users and Product Managers