A Query-Based Feature Store at OLX

OLX Brasil

OLX Brasil

Feature Stores for ML

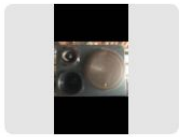
Biggest
C2C
Marketplace in
Brasil

Feature
Stores
for ML

Estou procurando por....

Imóveis | Autos e peças | Para a sua casa | Eletrônicos e celulares | Vagas de emprego | Serviços | Músicas e hobbies | Esportes e lazer | Moda e beleza | Agro e indústria | Todas as Categorias

Anúncios recentes



Caixa de som automotiva

R$ 170

Projetor Benq 4.000 mil lumens + Telão 02 x 02...

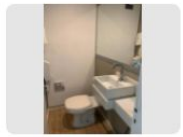
R$ 3.000

Compressor de ar honda fit 2005

R$ 800

Vendo Pia de banheiro com coluna bege 69 rea...

R$ 69



Sala Comercial Centro da Barra

R$ 220.000

Câmera de ré automotiva ?

R$ 70

Divisor para Antena Digital e Tv a Cabo 1/ 2...

R$ 14

PS4 1tb

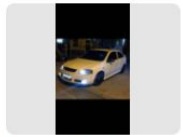
R$ 1.350

Ótima casa com 2 quartos, na rua Cristóvã...

R$ 160.000

Colar ajustável de ouro 18k/750.

R$ 840



Astra 2006/2006

R$ 22.000

Hb20 1.6 automatico 2015

R$ 48.000

6M

Daily Active Users

500K

New Ads Per Day

5M

Chat Messages Per Day

Problems

How to minimize scam impact?

How to minimize scam impact?

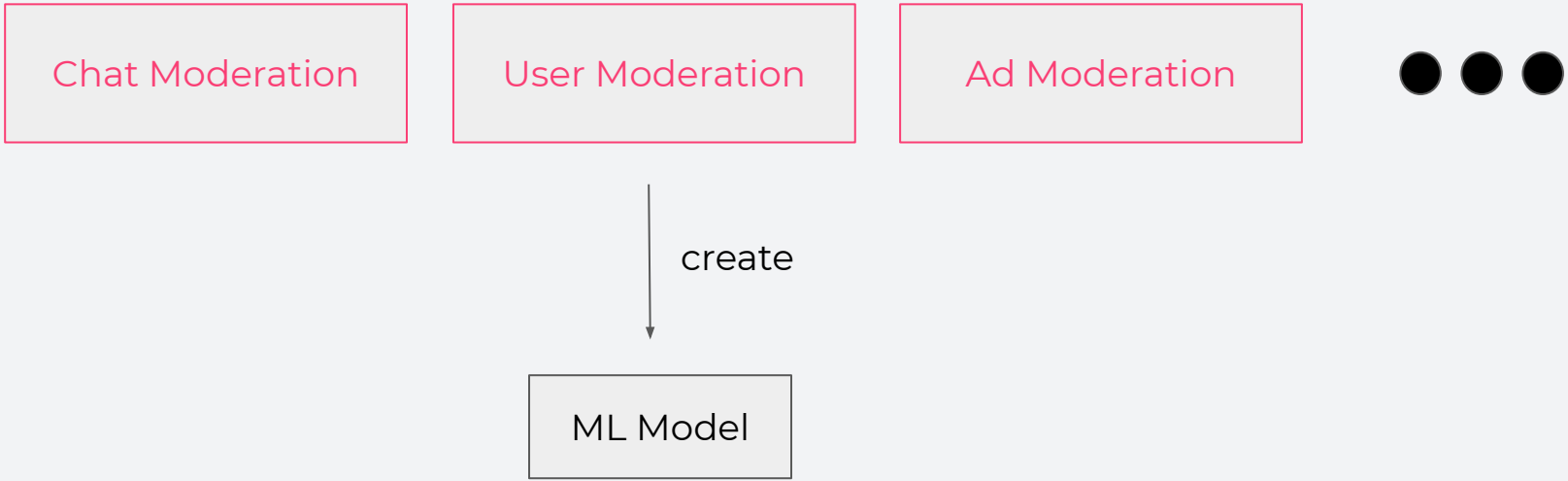
We use Machine Learning!!

Chat Moderation

User Moderation

Ad Moderation

Feature Stores for ML


Chat Moderation
User Moderation
Ad Moderation
create
ML Model

Chat Moderation

User Moderation

Ad Moderation

Create

ML Model

Take decision

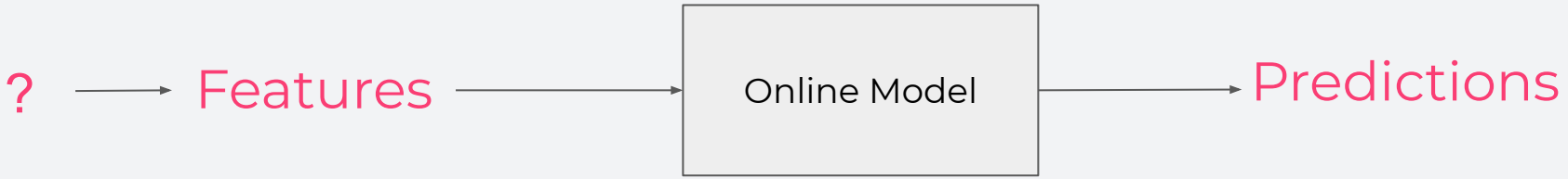
Feature Stores for ML

Chat Moderation

Chat Moderation

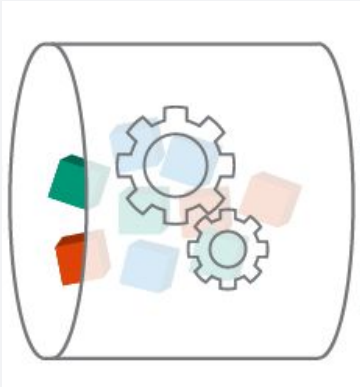
Need to ban users ASAP, in minutes, based on chat behavior

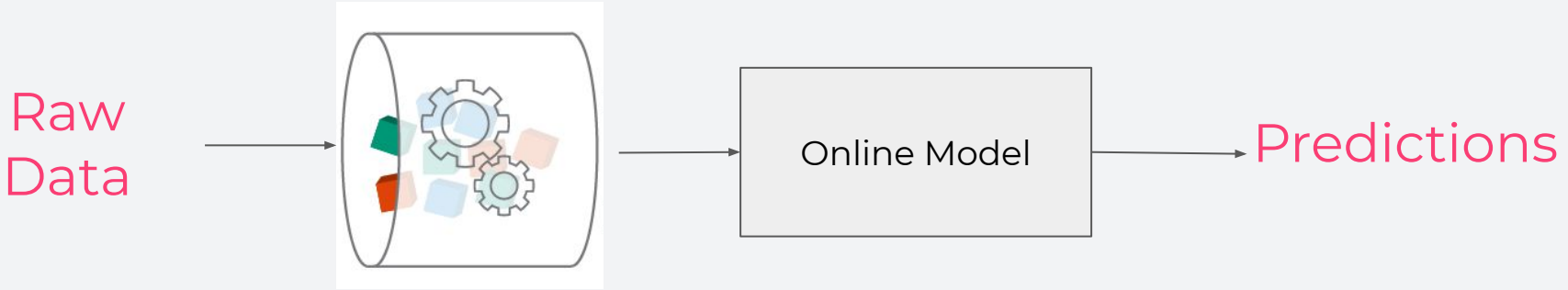

Features
Online Model
Predictions


?
Features
Online Model
Predictions

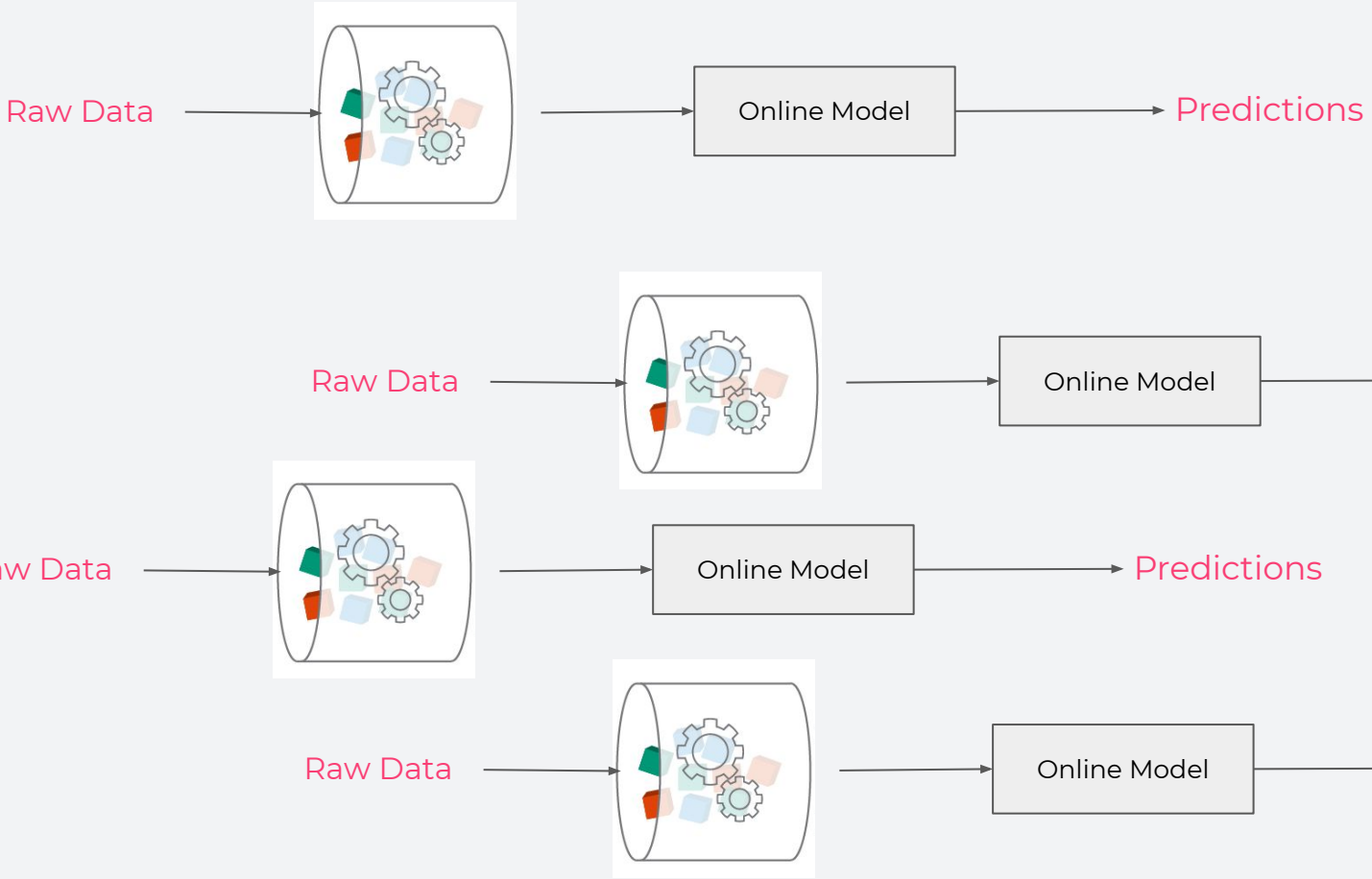
Chat Moderation

They needed a Transformation Engine to serve features in Real Time



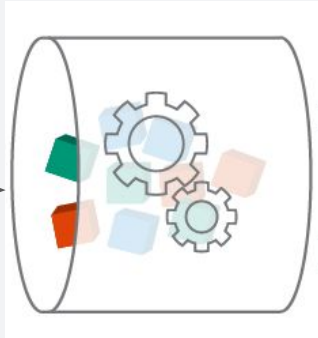

Raw Data
Online Model
Predictions

Imagine if each model in production should replicate this infra?



Feature Stores for ML

Raw Data



Online Model

Online Model

Online Model

Predictions

Predictions

Predictions

Feature Stores for ML

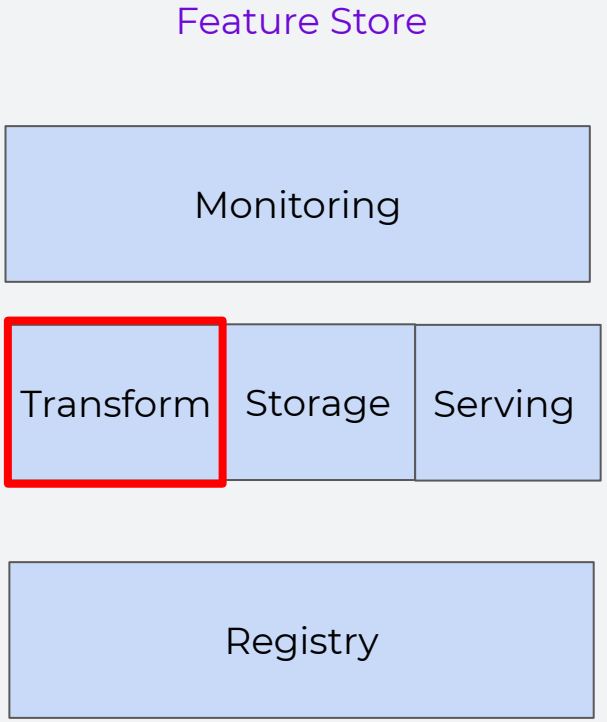
Who you gonna call?

Who you gonna call?

The MLOps Platforms Team

This would be the start of our Feature Store

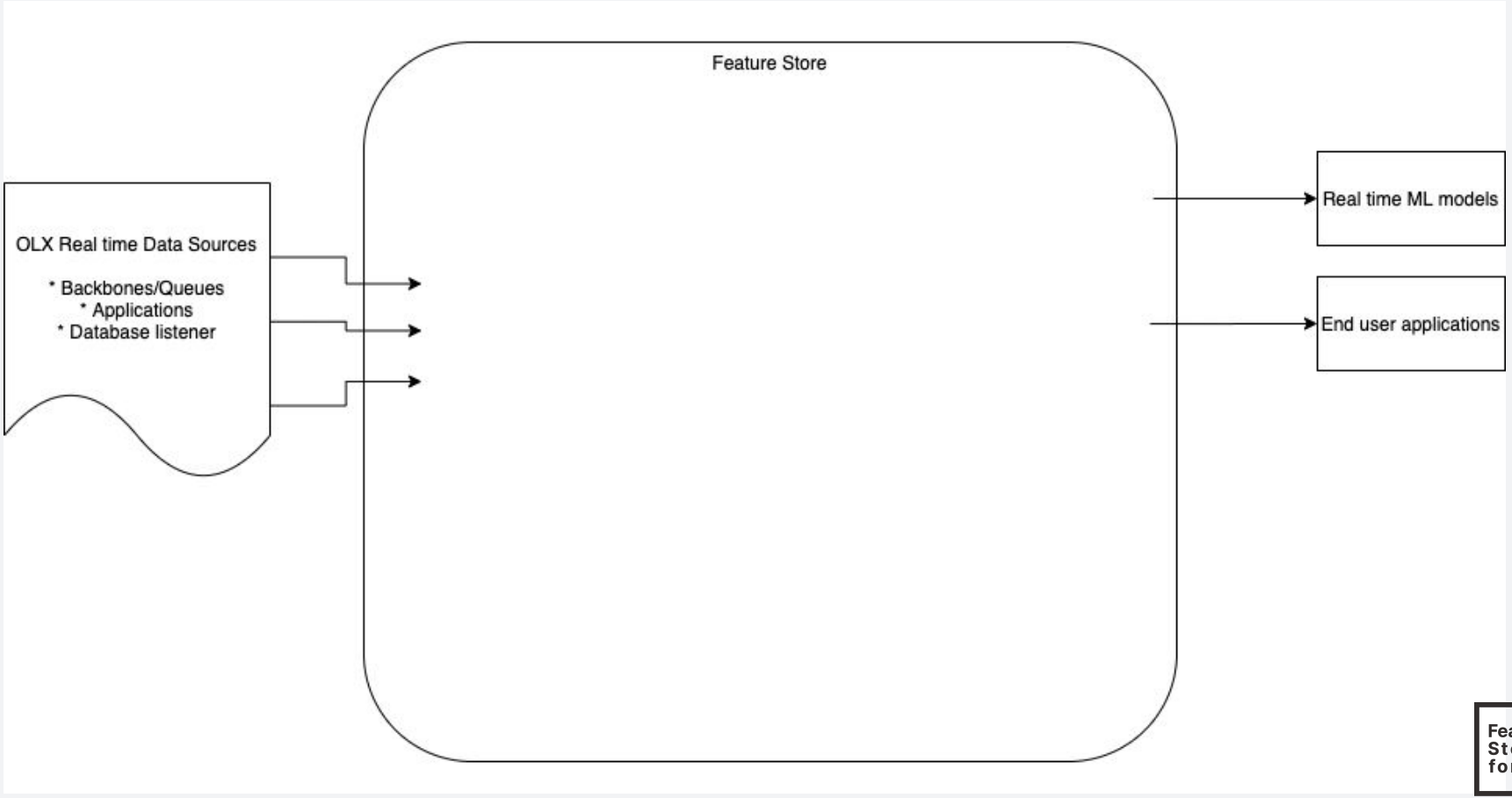
This would be the start of our Feature Store

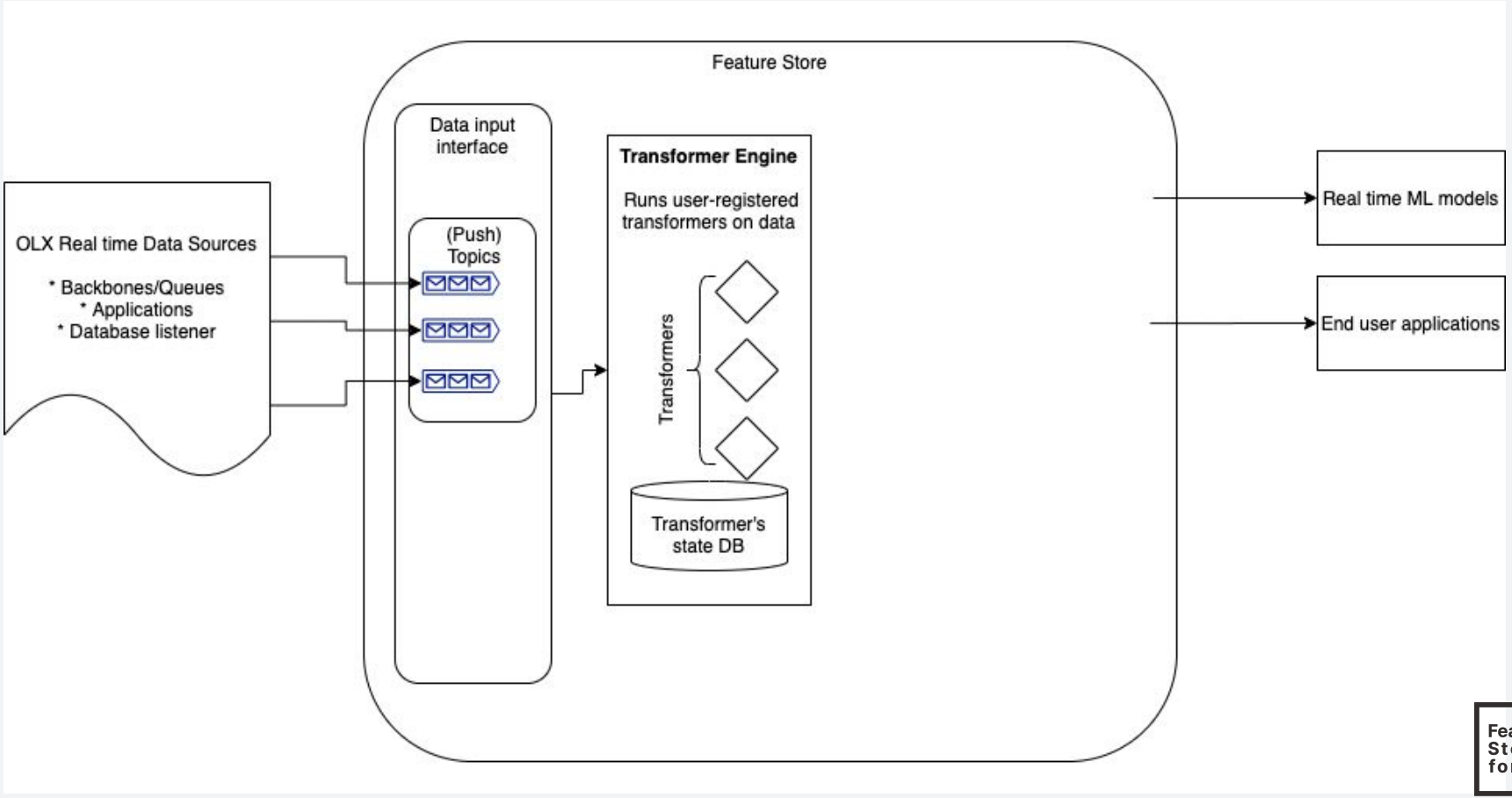


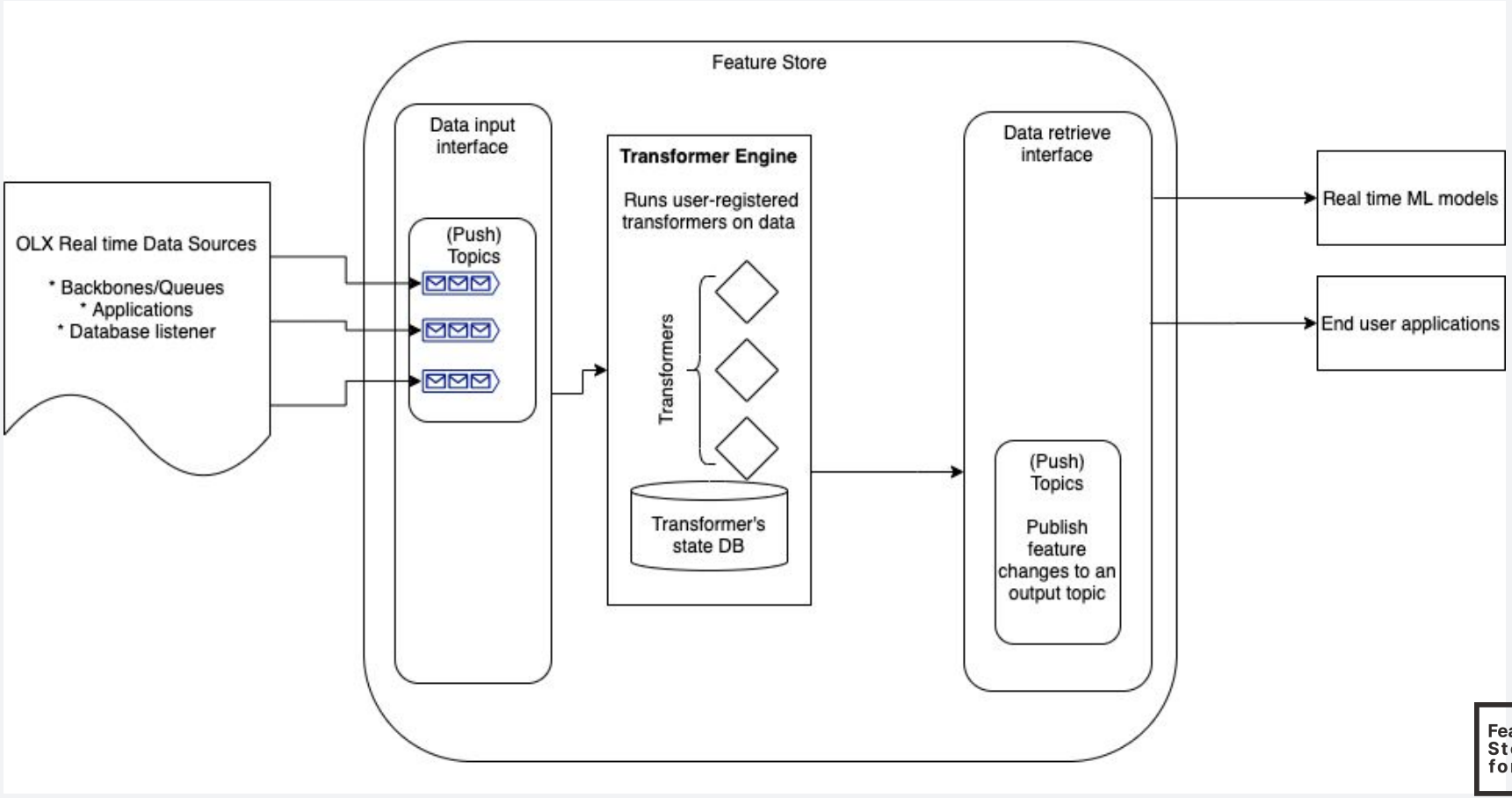
FS Conception

Feature Stores for ML

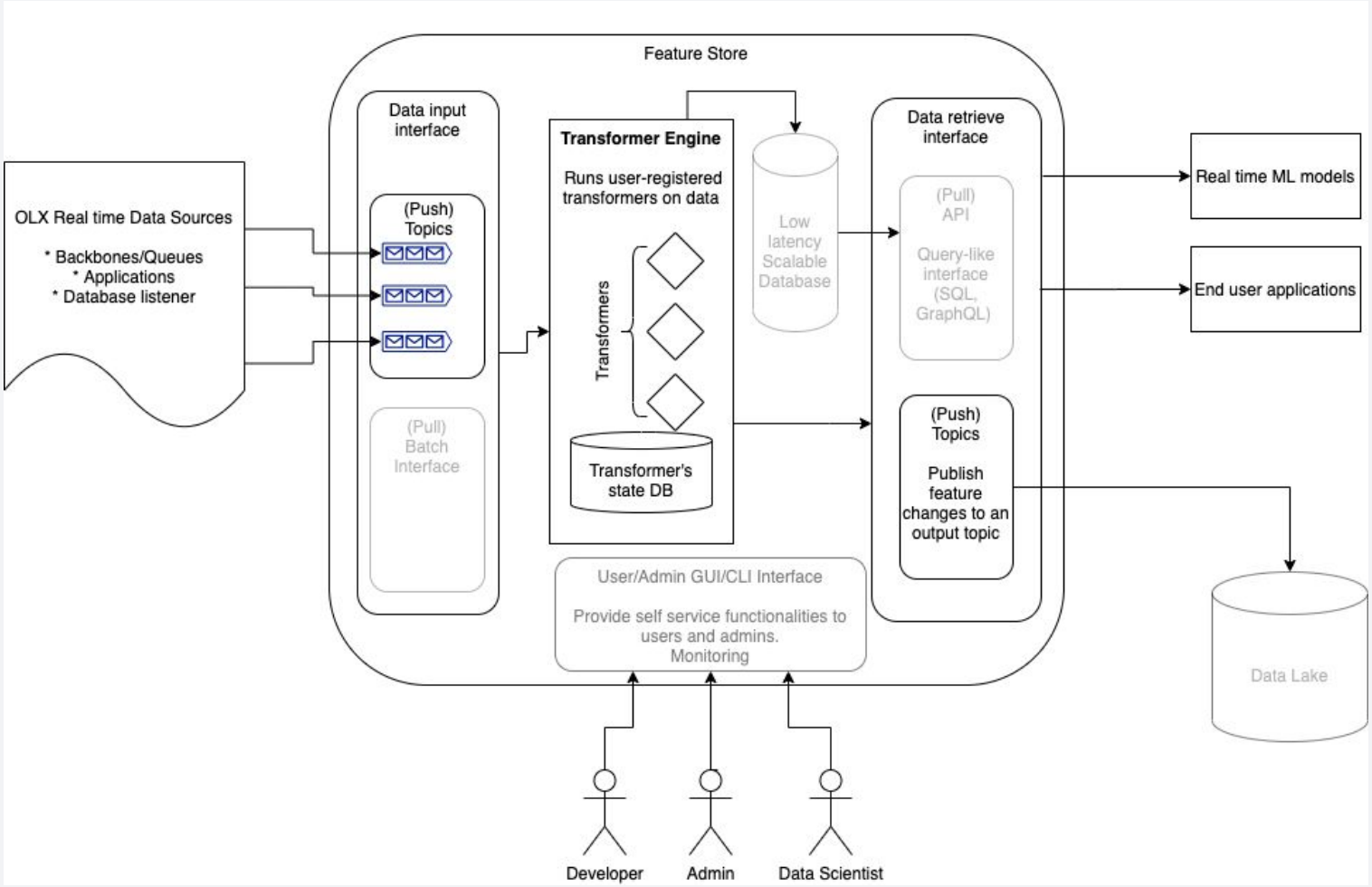
Platform that centralizes processment, offers easy to create features and promotes reusability


Feature Store
OLX Real time Data Sources
* Backbones/Queues
* Applications
* Database listener
Real time ML models
End user applications


Feature Store
Data input interface
(Push) Topics
Transformer Engine
Runs user-registered transformers on data
Transformers
Transformer's state DB
OLX Real time Data Sources
* Backbones/Queues
* Applications
* Database listener
Real time ML models
End user applications



Feature Stores for ML



Feature Stores for ML

FS Implementation

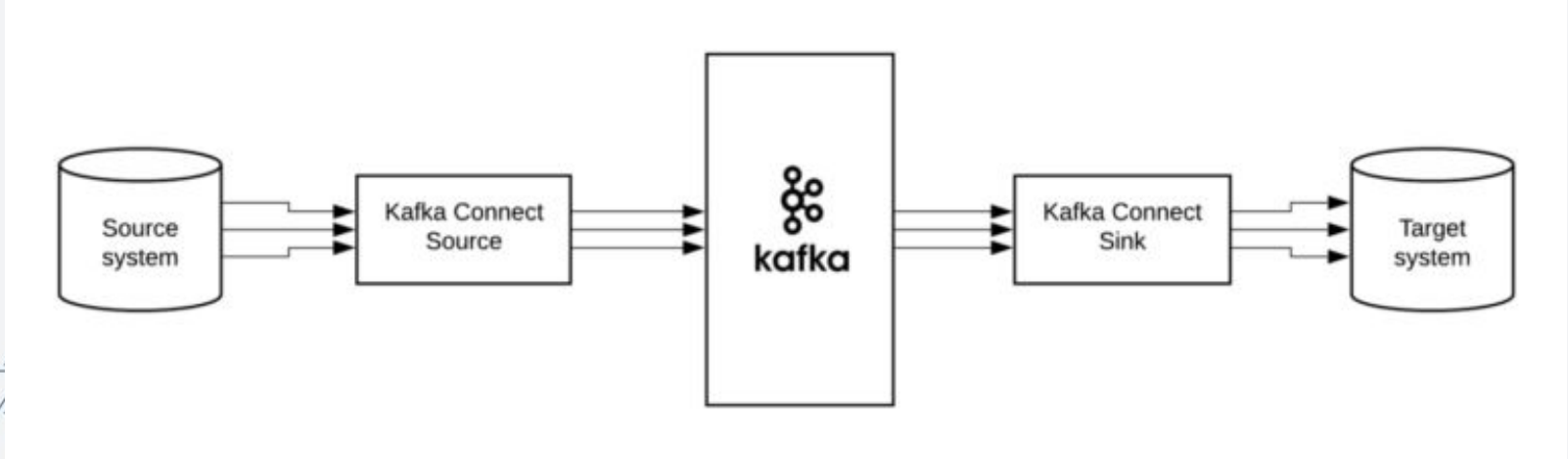
Main Concerns:

- Must be easy to create features
- Pricing
- Infrastructure Complexity

Messaging System

Kafka Managed
by AWS

Integration with external components



Kafka Connect

What about the transformations?

Raw Data

Aggregations

?

Discovery with some Transformation Engines

	Flink
Language	Java/SQL
Maturity	Good
Documentation	Bad
Feature Development	Regular
Beginner-friendly	Bad
Scalability	Good
Cost	Bad
Flexibility	Good

	Flink	PyFlink
Language	Java/SQL	Python
Maturity	Good	Regular
Documentation	Bad	Bad
Feature Development	Regular	Regular
Beginner-friendly	Bad	Bad
Scalability	Good	Good
Cost	Bad	Bad
Flexibility	Good	Good

	Flink	PyFlink	Kafka Streams
Language	Java/SQL	Python	Java
Maturity	Good	Regular	Good
Documentation	Bad	Bad	Good
Feature Development	Regular	Regular	Regular
Beginner-friendly	Bad	Bad	Bad
Scalability	Good	Good	Good
Cost	Bad	Bad	Regular
Flexibility	Good	Good	Good

	Flink	PyFlink	Kafka Streams	Faust
Language	Java/SQL	Python	Java	Python
Maturity	Good	Regular	Good	Bad
Documentation	Bad	Bad	Good	Bad
Feature Development	Regular	Regular	Regular	Good
Beginner-friendly	Bad	Bad	Bad	Regular
Scalability	Good	Good	Good	Bad
Cost	Bad	Bad	Regular	Regular
Flexibility	Good	Good	Good	Regular

	Flink	PyFlink	Kafka Streams	Faust	KSQLdb
Language	Java/SQL	Python	Java	Python	SQL
Maturity	Good	Regular	Good	Bad	Good
Documentation	Bad	Bad	Good	Bad	Good
Feature Development	Regular	Regular	Regular	Good	Good
Beginner-friendly	Bad	Bad	Bad	Regular	Good
Scalability	Good	Good	Good	Bad	Good
Cost	Bad	Bad	Regular	Regular	Regular
Flexibility	Good	Good	Good	Regular	Regular

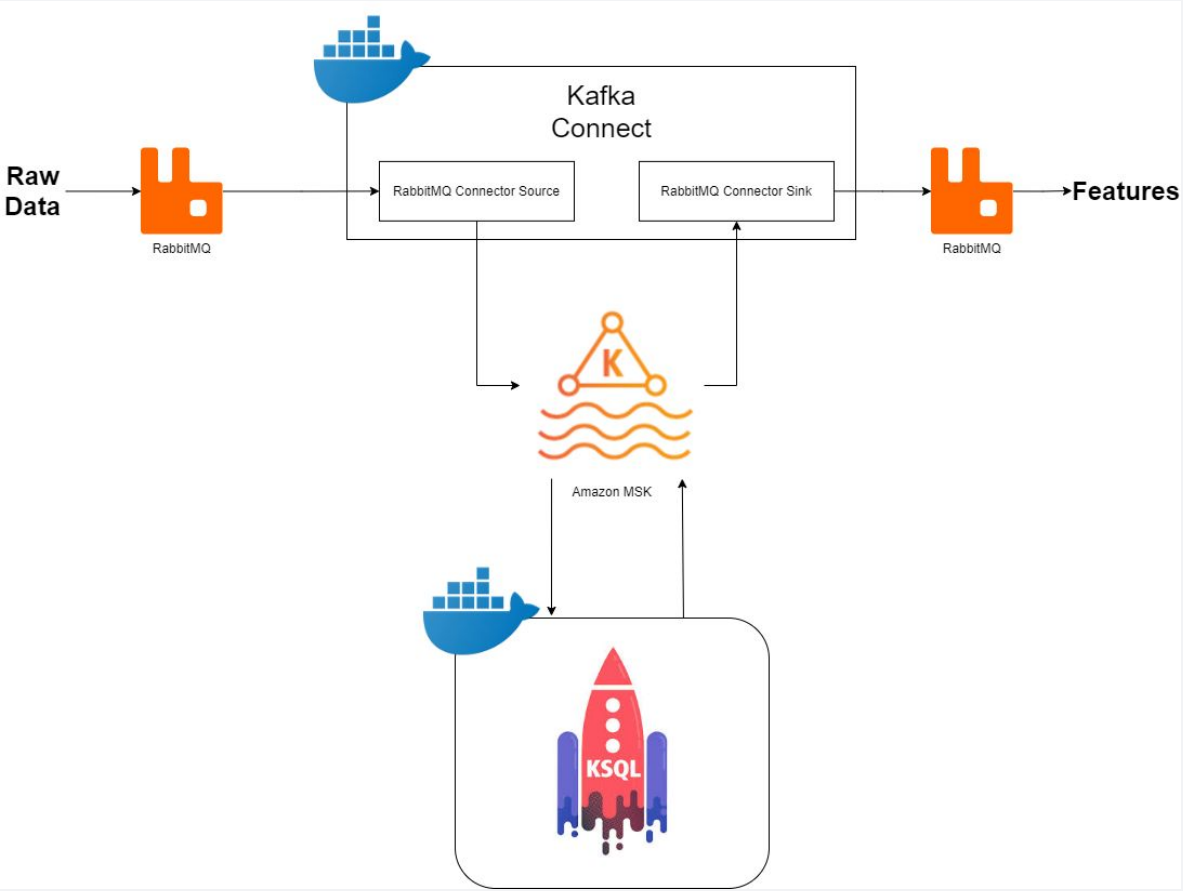
	KSQLdb
Language	SQL
Maturity	Good
Documentation	Good
Feature Development	Good
Beginner-friendly	Good
Scalability	Good
Cost	Regular
Flexibility	Regular

```
    '''
        CREATE OR REPLACE STREAM test_stateless (
            adId    VARCHAR,
            chatId  VARCHAR
        ) WITH (
            KAFKA_TOPIC='test_topic',
            VALUE_FORMAT='JSON',
            PARTITIONS=1
        )
    ''',
    '''
        CREATE OR REPLACE TABLE test_stateful as
            SELECT adId, COUNT(*)
                FROM test_stateless
                GROUP BY adId
                EMIT CHANGES
    '''
```

Feature Stores for ML

Raw Data

Aggregations


Kafka
Connect
Raw
Data
RabbitMQ
RabbitMQ Connector Source
RabbitMQ Connector Sink
RabbitMQ
Features
K
Amazon MSK
KSQL

Example of usage

We can create a stream to get the Raw Data

```
CREATE STREAM chatmod_message ( --noqa
    userIp VARCHAR,
    tipVisibleTo VARCHAR,
    tipValue VARCHAR,
    tipType VARCHAR,
    textMessage VARCHAR,
    senderAccountId VARCHAR,
    receiverAccountId VARCHAR,
    publicId VARCHAR,
    platform VARCHAR,
    messageType VARCHAR,
    messageTimestamp VARCHAR,
    messageId VARCHAR,
    listId VARCHAR,
    chatId VARCHAR,
    adId VARCHAR,
    senderId VARCHAR,
    senderType VARCHAR,
    isFirstMessage BOOLEAN
) WITH (
    KAFKA_TOPIC = 'chatmod_message',
    VALUE_FORMAT = 'json',
    PARTITIONS = 1,
    TIMESTAMP = 'messageTimestamp',
    TIMESTAMP_FORMAT = 'yyyy-MM-dd HH:mm:ss.SSS'
);
```

```sql
CREATE TABLE chatmod_first_message_similarity_feature WITH (KAFKA_TOPIC = 'chatmod_first_message_similarity_feature') AS --noqa
SELECT
    buyer_id AS buyer_id_key,
    AS_VALUE(buyer_id) AS buyer_id,
    MIN_LEVENSHTEIN_DISTANCE(textMessage) AS first_message_similarity
FROM chatmod_buyer_message WINDOW SESSION (2 HOURS) --noqa
WHERE isFirstMessage = true
GROUP BY buyer_id EMIT CHANGES; --noqa
```

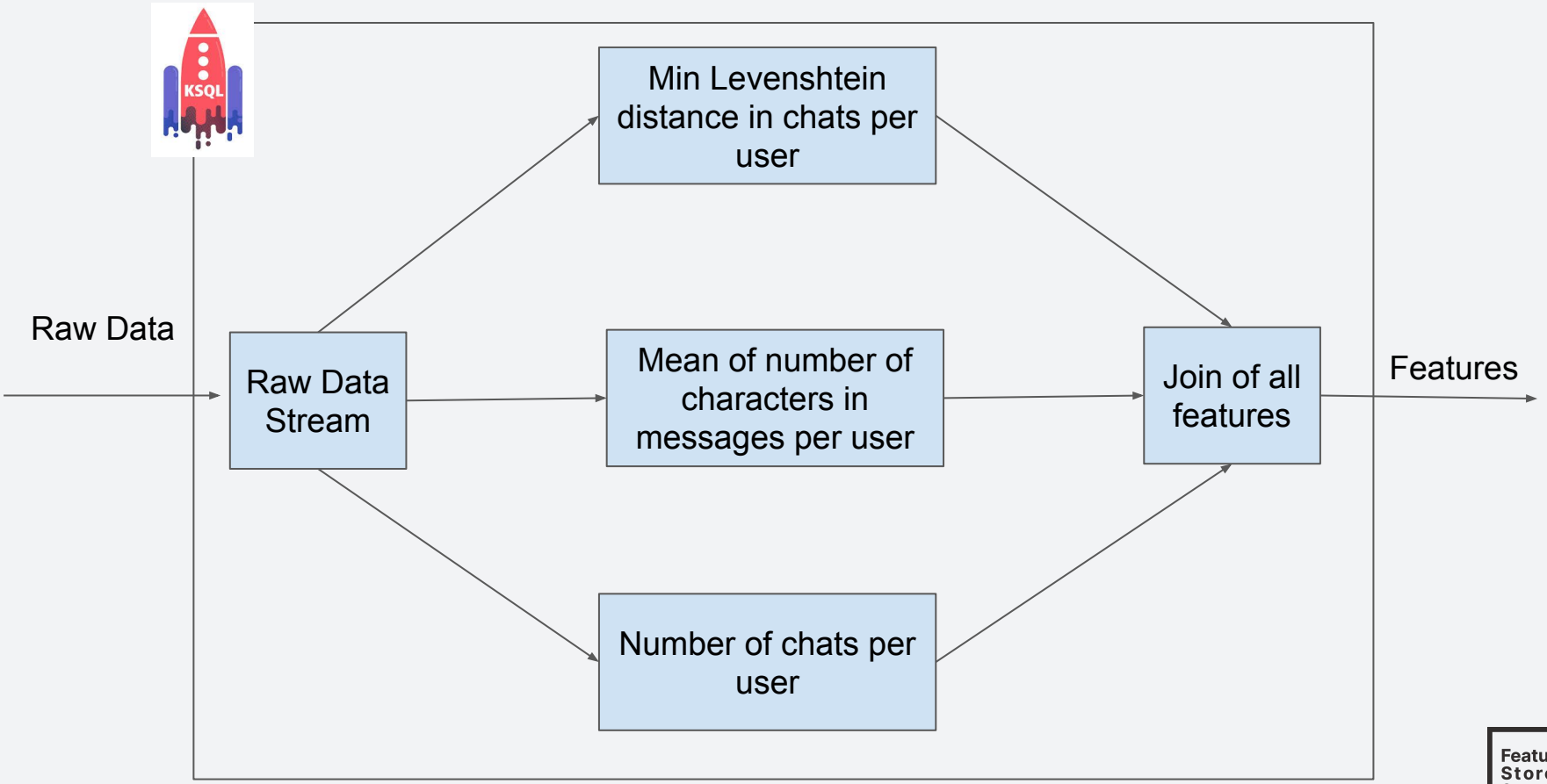
We can create a TABLE that aggregates data

User Defined Function

```
CREATE TABLE chatmod_first_message_similarity_feature WITH (KAFKA_TOPIC = 'chatmod_first_message_similarity_feature') AS --noqa
SELECT
    buyer_id AS buyer_id_key,
    AS_VALUE(buyer_id) AS buyer_id,
    MIN_LEVENSHTEIN_DISTANCE(textMessage) AS first_message_similarity
FROM chatmod_buyer_message WINDOW SESSION (2 HOURS) --noqa
WHERE isFirstMessage = true
GROUP BY buyer_id EMIT CHANGES; --noqa
```

We can generate a stream that reads from this Table and saves into a topic

```
9
10 CREATE OR REPLACE STREAM chatmod_first_message_similarity_feature_windowed ( --noqa
11     buyer_id VARCHAR,
12     buyer_id_key VARCHAR KEY,
13     first_message_similarity INTEGER
14 ) WITH (
15     KAFKA_TOPIC='chatmod_first_message_similarity_feature',
16     VALUE_FORMAT='JSON',
17     WINDOW_TYPE='SESSION'
18 );
```


KSQL
Raw Data
Raw Data Stream
Min Levenshtein distance in chats per user
Mean of number of characters in messages per user
Number of chats per user
Join of all features
Features

Declaring Features

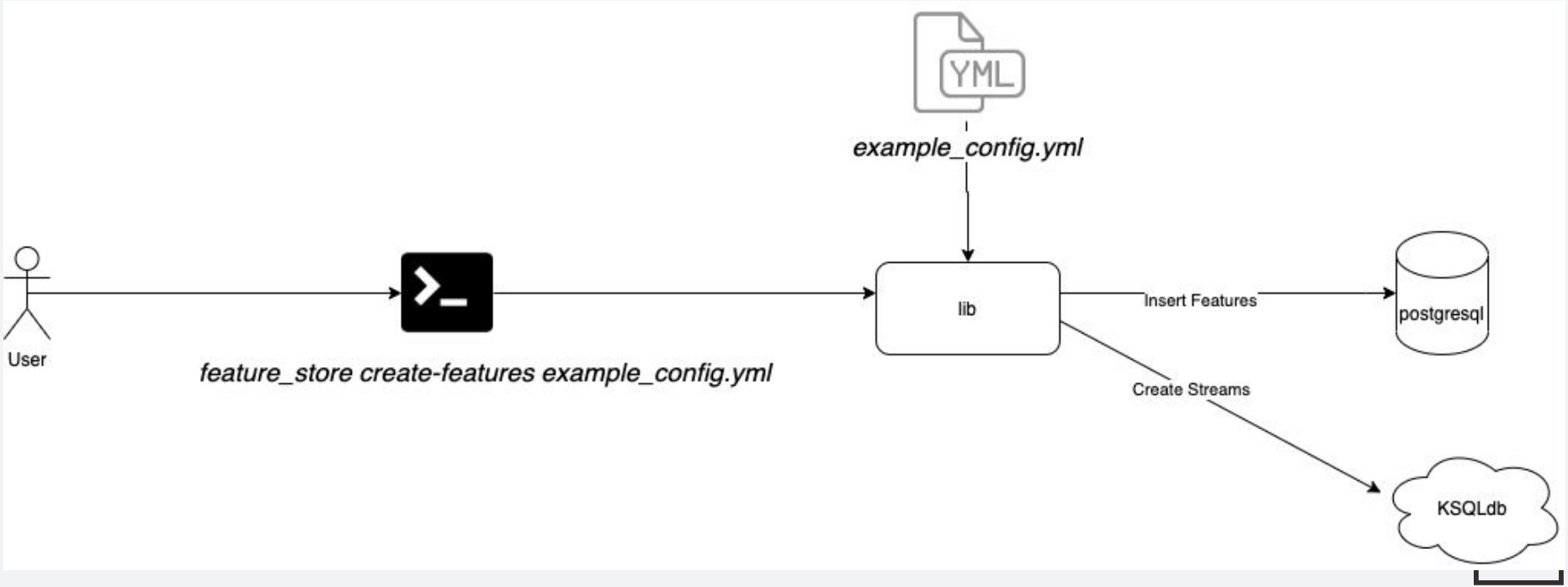
```yaml
product:
  name: test_name_6
  team: test_team
  description: This is the product description

connectors:
  connector_1:
    name: source_connector
    connector_type: source
    topic_name: topic_source_connector
  connector_2:
    name: sink_connector
    connector_type: sink
    topic_name: topic_sink_connector

features:
  feature_1:
    name: first_feature_6
    description: This is my first feature
    feature_streams:
    - stream_1
    - stream_2
    sql_file: example_sql_scripts/file_1.sql
    input_topic: topic_1
    output_topic: topic_2
    is_external: True
  feature_2:
    name: second_feature_6
    description: This is my second feature
    feature_streams:
    - stream_3
    - stream_4
    sql_file: example_sql_scripts/file_2.sql
    input_topic: topic_2
    output_topic: topic_3
    is_external: True
    dependencies:
    - feature_1

```

Feature Declaration Interface



Results

Messages Input per day

5 M

Features

8 created

Feature Output per day

1.5 M

Models using the Feature Store

2 models

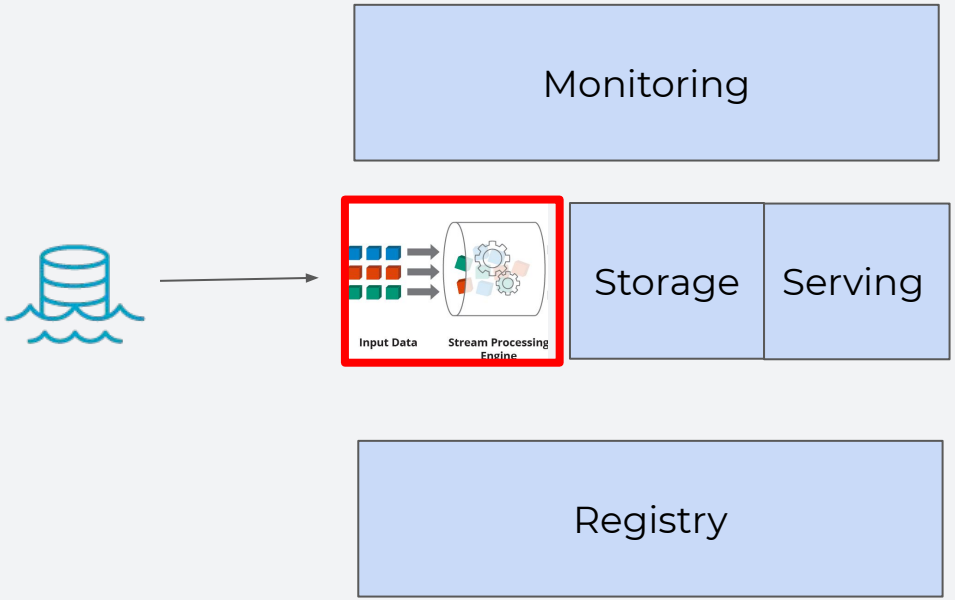
Fraudulent messages detected per day

300-400

Complaint reduced by

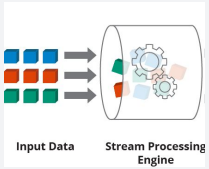
22-26%

Roadmap



Creation of features in batch, from datalake

Monitoring



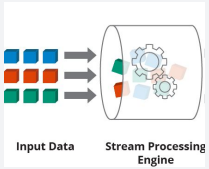
Storage | Serving

Registry

Feature Versioning

Feature Governance

Monitoring



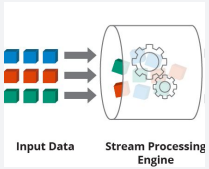
Storage

Serving

Registry

Export features into Datalake, for training

Monitoring



Storage | Serving

Registry

Feature serving with database and API

Thank you!

Do you have any questions?

Augusto Acioli Pinho Vanderley - OLX Brasil

https://www.linkedin.com/in/augusto-vanderley/?locale=en_US