EuroPython 2020

Real Time Machine Learning with Python

Alejandro Saucedo | as@seldon.io

Twitter: @AxSaucedo

Hello, my name is Alejandro

@AxSaucedo

Alejandro Saucedo

Engineering Director
Seldon Technologies

Chief Scientist
The Institute for Ethical AI & ML

Head of Solutions Eng & Sci
Eigen Technologies

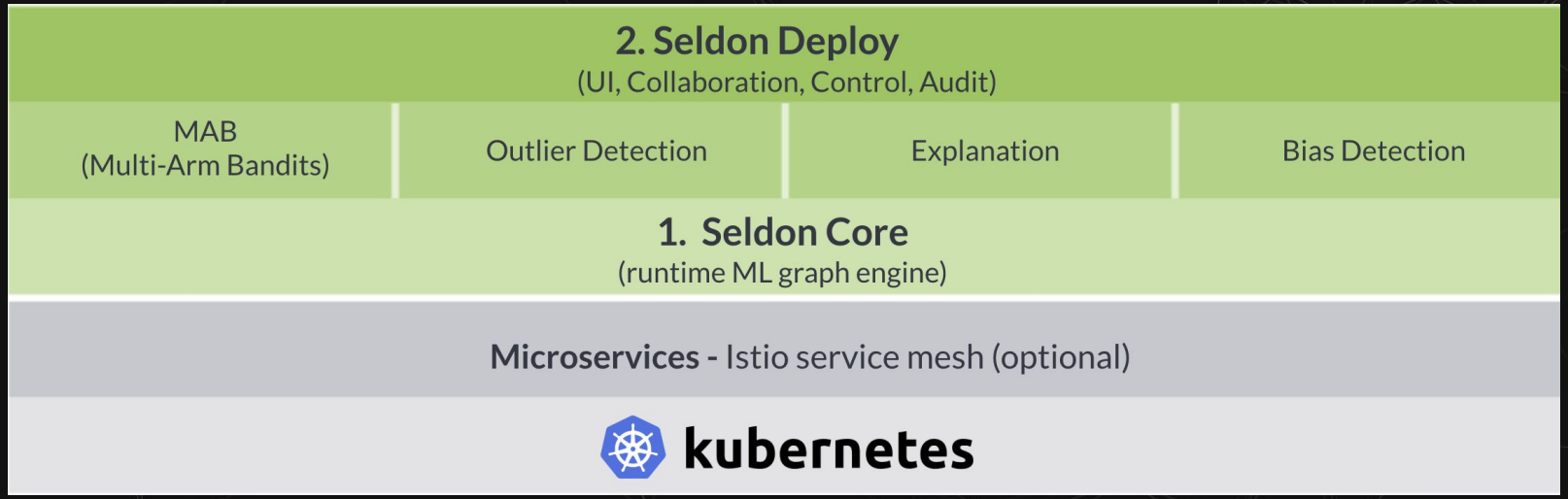
Software Engineer & DevX Lead
Bloomberg LP

Seldon: OSS Production ML Deployment

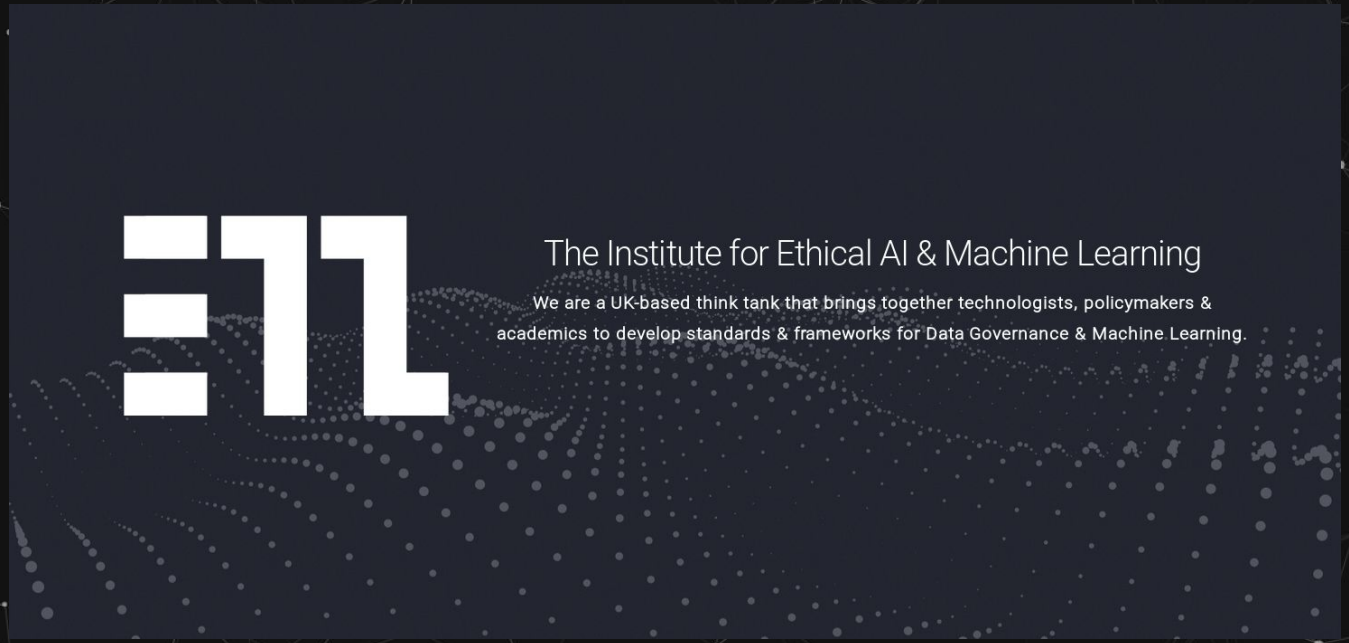
1. Package
Create REST or gRPC dockerized microservice .

2. Describe Deployment
Create/update Kubernetes resource manifest for deployment graph.

3. Deploy
Manage and analyze the performance of live deployments.



The Institute for Ethical AI & Machine Learning



We are part of the LFAI

Today

- Conceptual intro to stream processing
- Machine learning for real time
- Tradeoffs across tools
- Hands on use-case

Real Time Reddit Processing

@AxSaucedo

- Real time ML model for reddit comments
- 200k comments for training model
- /r/science comments removed by mods

We will be fixing the front page of the internet

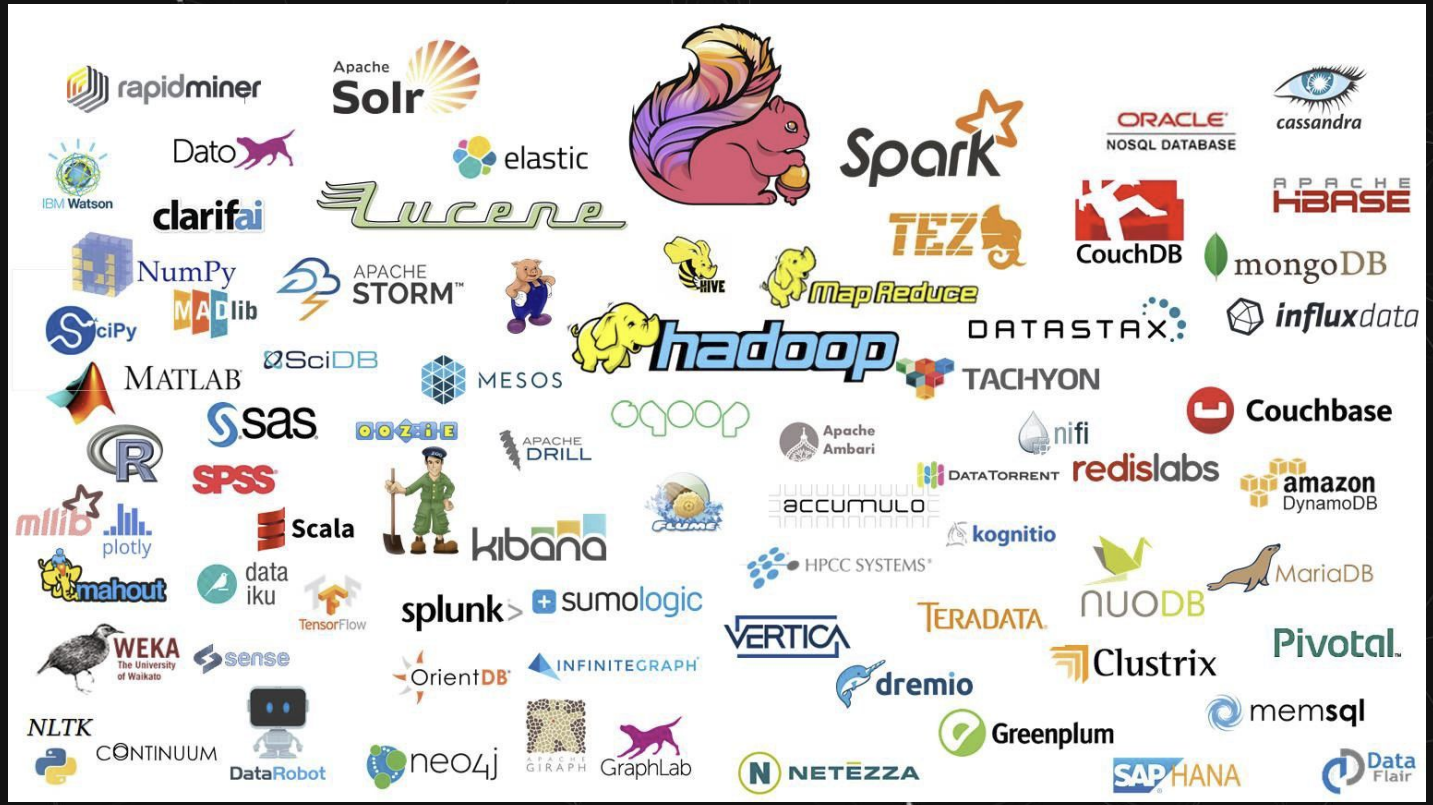
A trip to the ~~past~~ present: ETL

E - Extract
T - Transform
L - Load

Variations

- ETL - Extract Transform Load
- ELT - Extract Load Transform
- EL - Extract Load
- LT - Load Transform
- WTF - LOL

Specialised Tools



@AxSaucedo

EL

Nifi

Flume

ETL

Oozie

Airflow

...

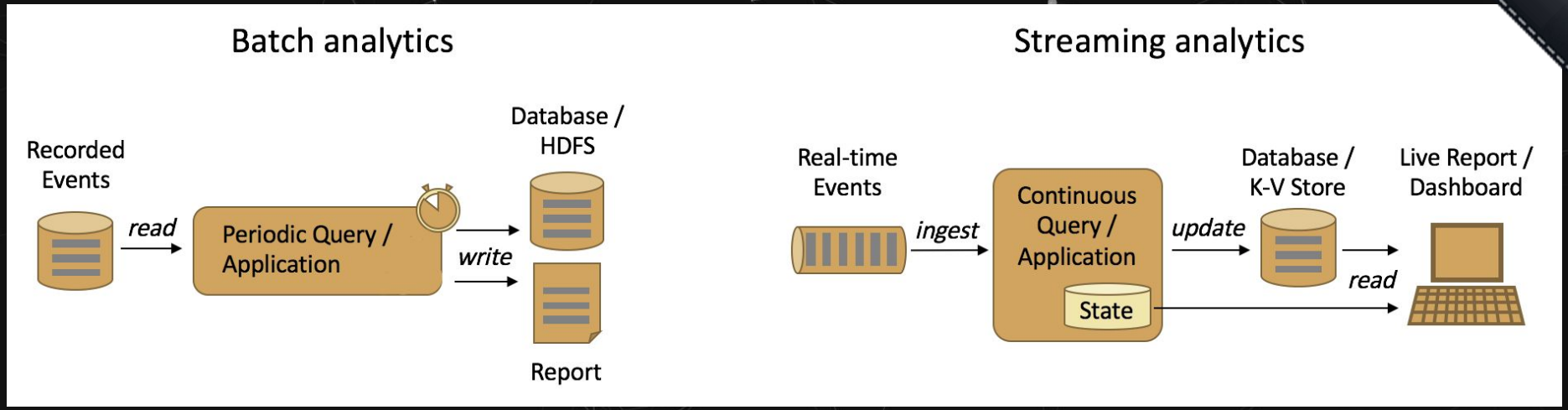
ELT

Elasticsearch

Data Warehouse

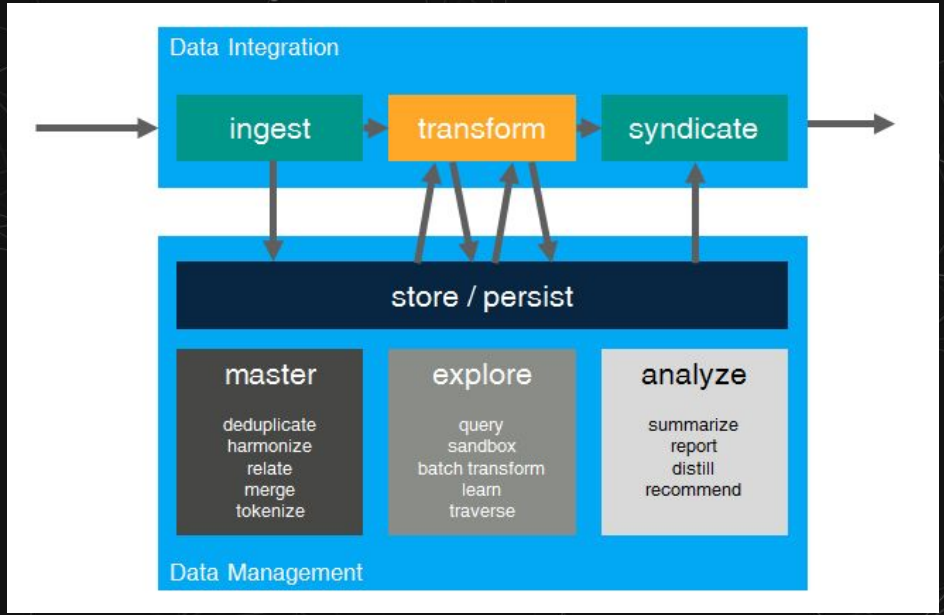
Jupyter notebook?

Batch VS Streaming



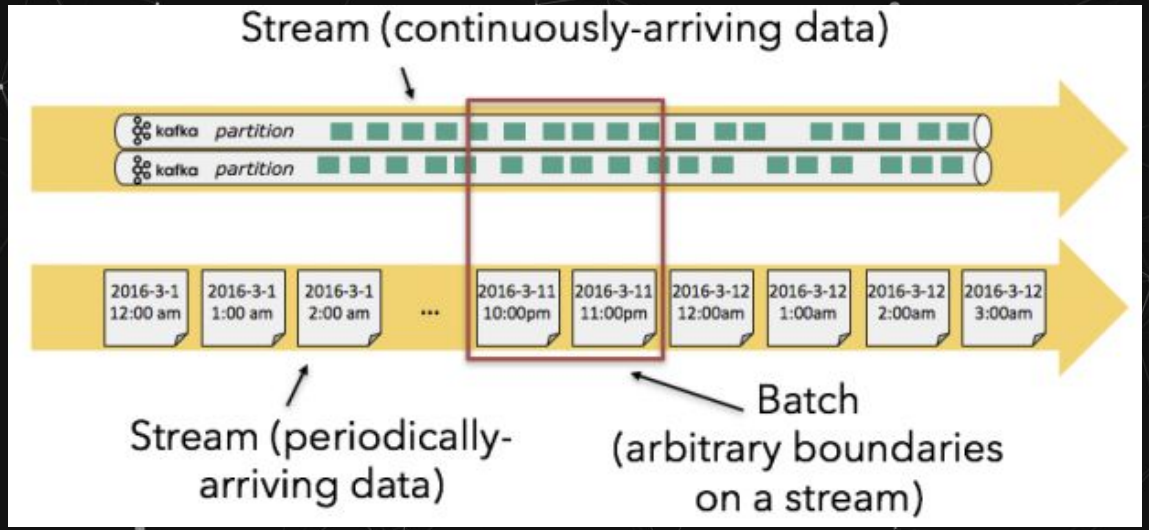
The spectrum of data processing

Batch ~~VS~~ AND Streaming



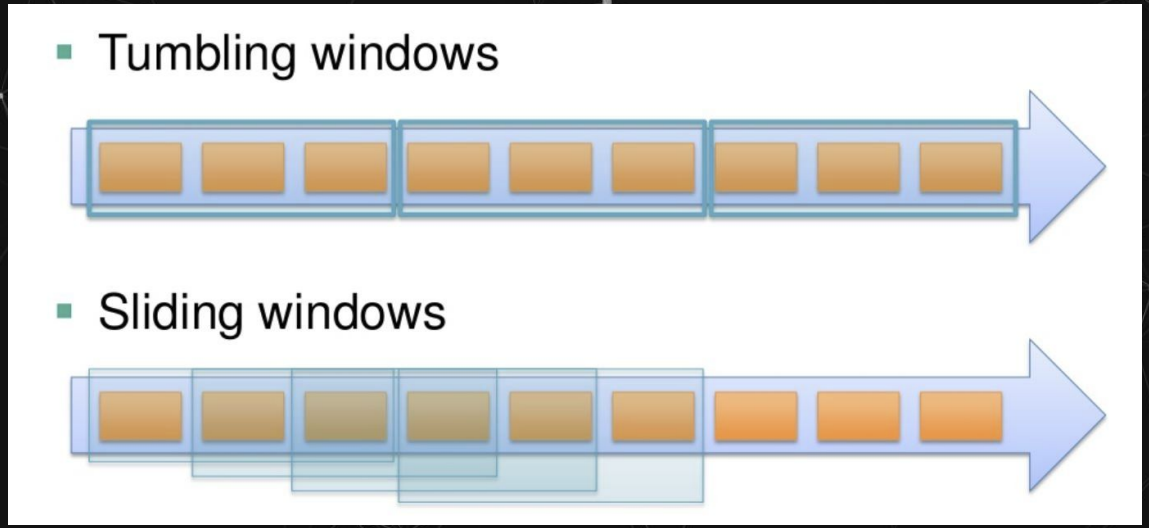
The right tool for the challenge

Unifying Worlds



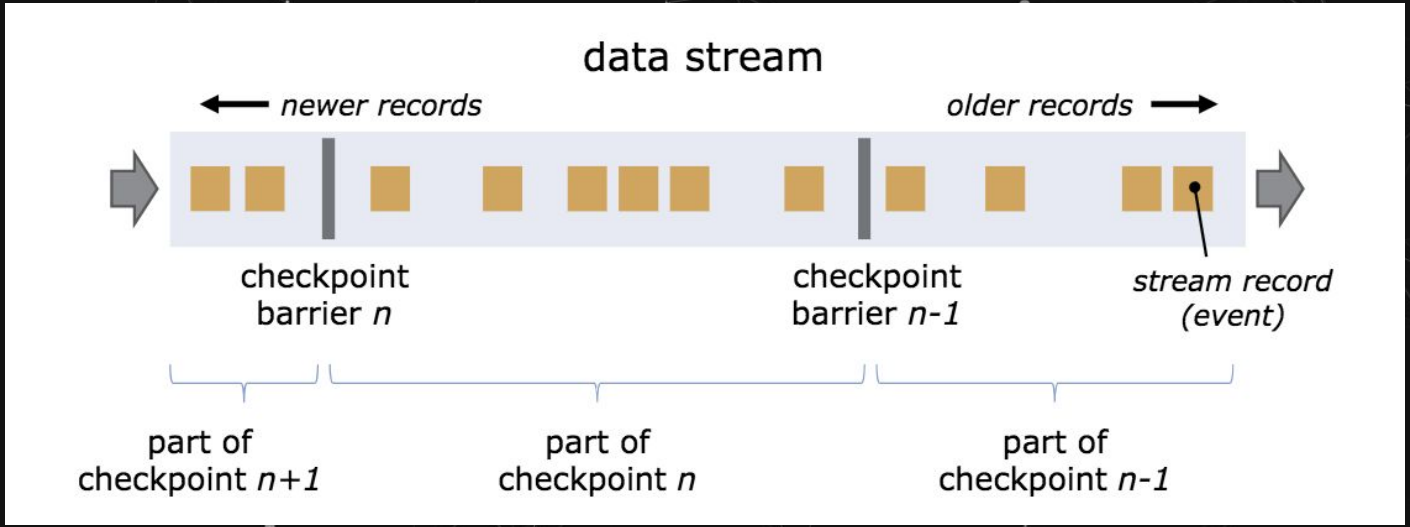
Massive drive on converging worlds

Streaming Concepts: Windows



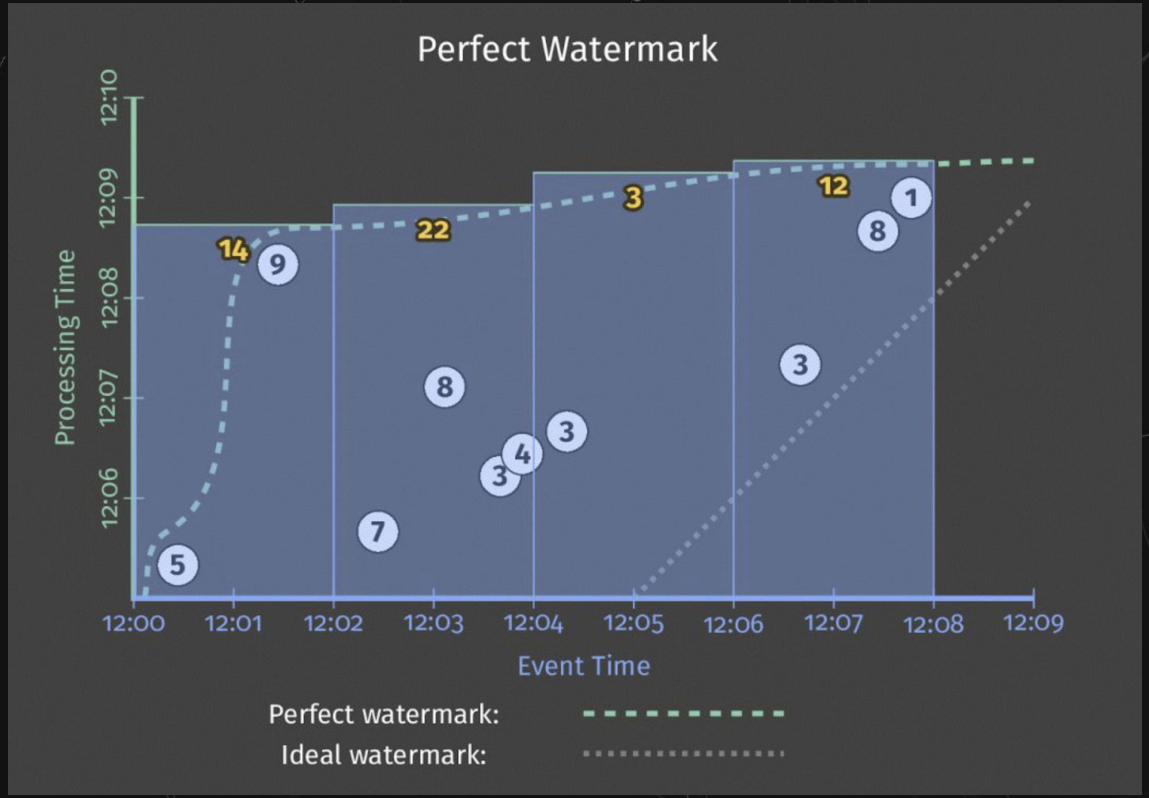
Processing of batches in real time

Streaming Concepts: Checkpoint



Keeping track of stream progress

Streaming Concepts: Watermark



Considering data that comes late in windows and stream batches

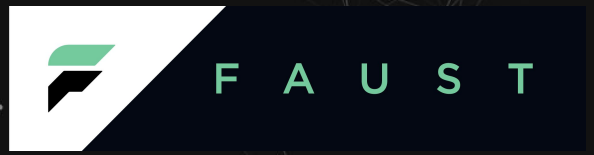
Some Stream Processing Tools

- Flink (Multiple Languages)
- Kafka Streams (Multiple Languages)
- Spark Stream (Multiple Languages)
- Faust (Python)
- Apache Beam (Python)

Today we're using

@AxSaucedo

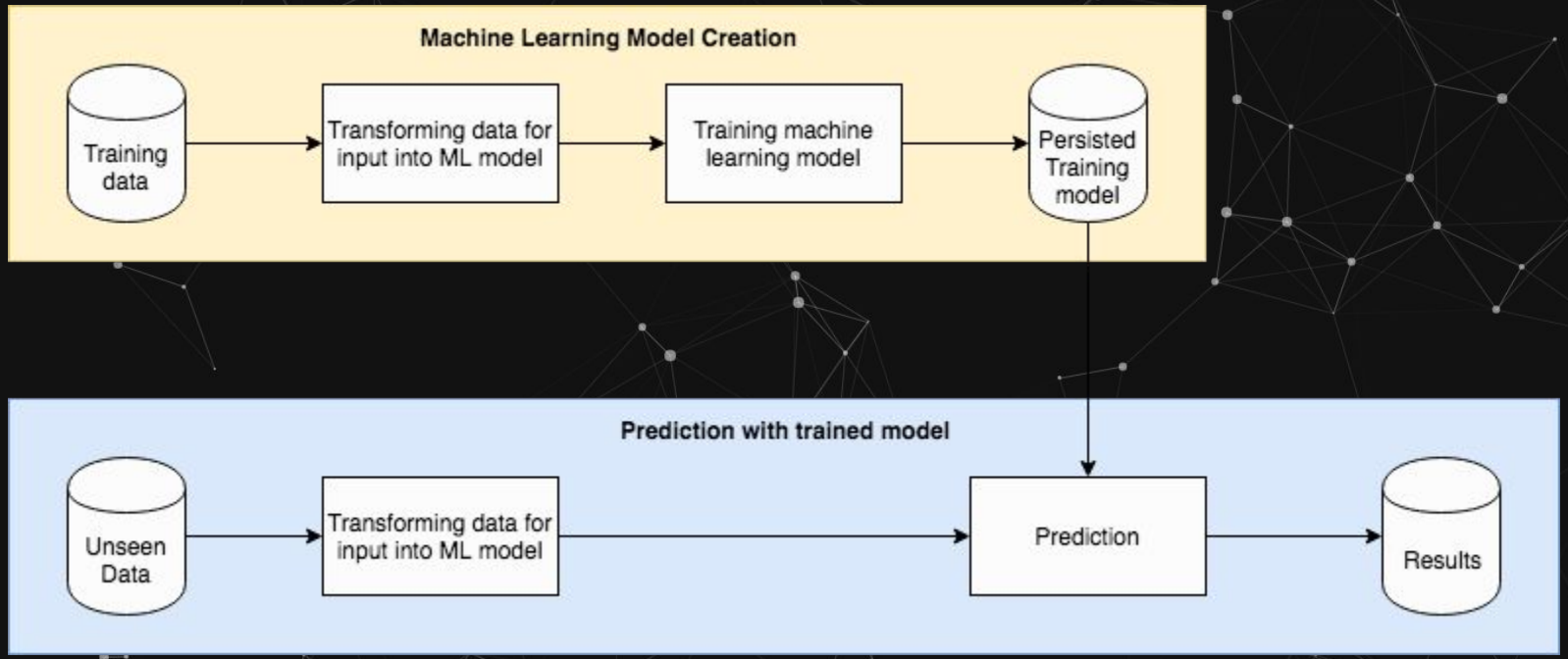
Stream Processing



ML Serving

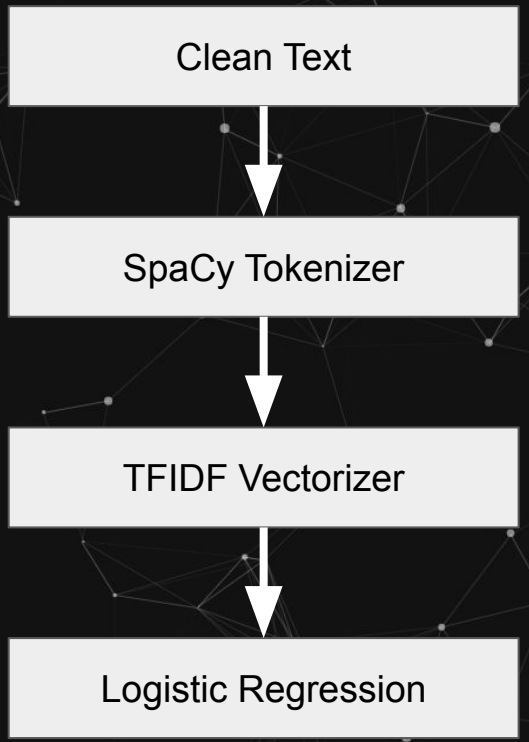
ML Training

Machine Learning Workflow

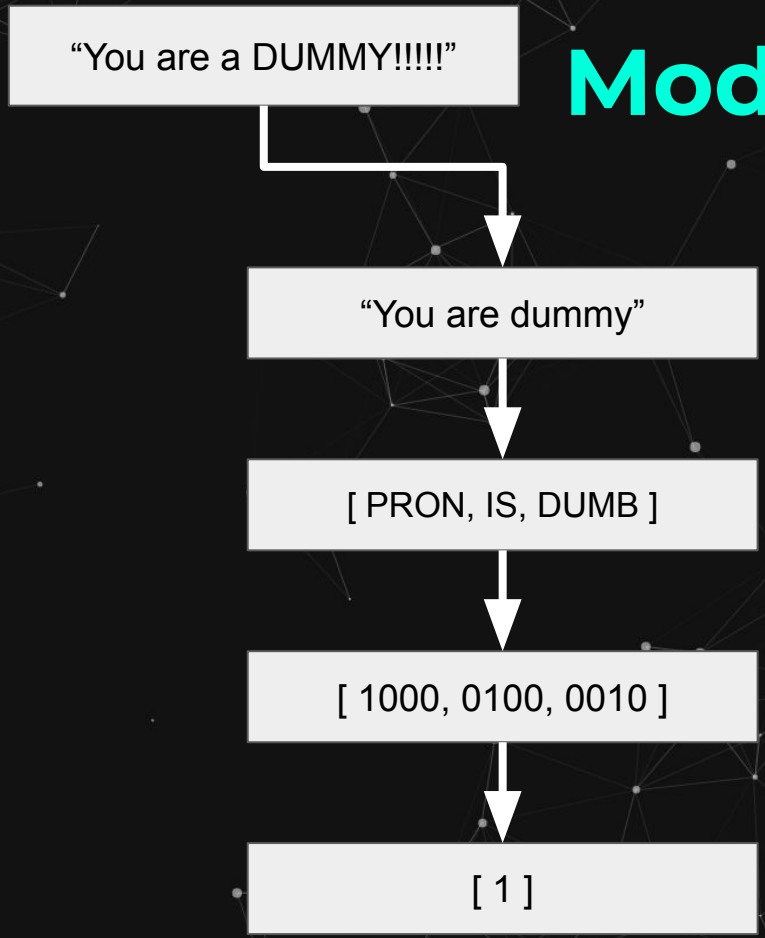


Model Training


```
clean_text_transformer = CleanTextTransformer()

spacy_tokenizer = SpacyTokenTransformer()

tfidf_vectorizer = TfidfVectorizer(
    min_df=3,
    max_features=1000,
    preprocessor=lambda x: x, tokenizer=lambda x: x,
    token_pattern=None,
    ngram_range=(1, 3), use_idf=1, smooth_idf=1,
    sublinear_tf=1)

lr_model = LogisticRegression(C=1.0, verbose=True)
```



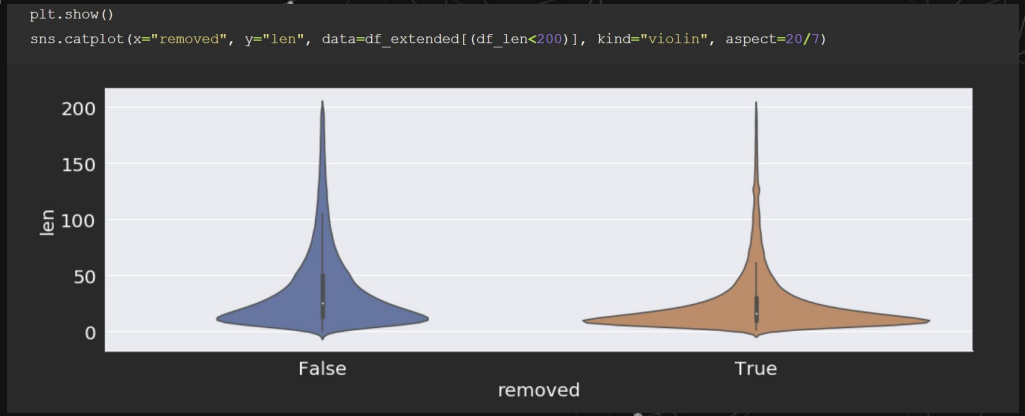
Model Training

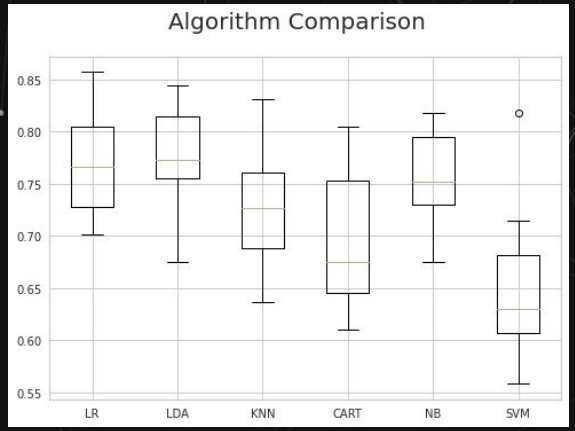



```
x_train_clean = \
    clean_text_transformer.transform(x_train)

x_train_tokenized = \
    spacy_tokenizer.transform(x_train_clean)

tfidf_vectorizer.fit(
         x_train_tokenized[TOKEN_COLUMN].values)

x_train_tfidf = \
    tfidf_vectorizer.transform(
        x_train_tokenized[TOKEN_COLUMN].values)

lr_model.fit(x_train_tfidf, y_train)

pred = lr_model.predict(x_test_tfidf)
```

More on EDA & Model Evaluatio

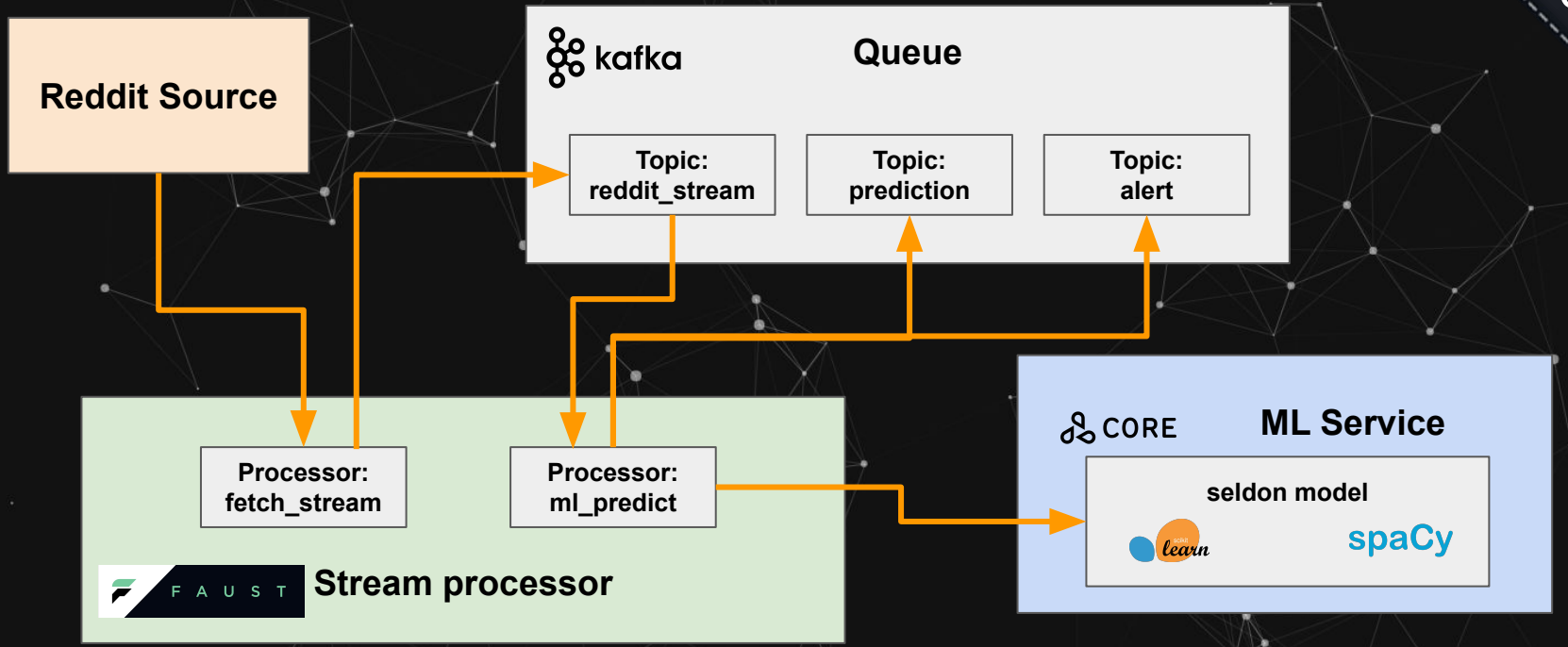

```
plt.show()
sns.catplot(x="removed", y="len", data=df_extended[(df_len<200)], kind="violin", aspect=20/7)
```



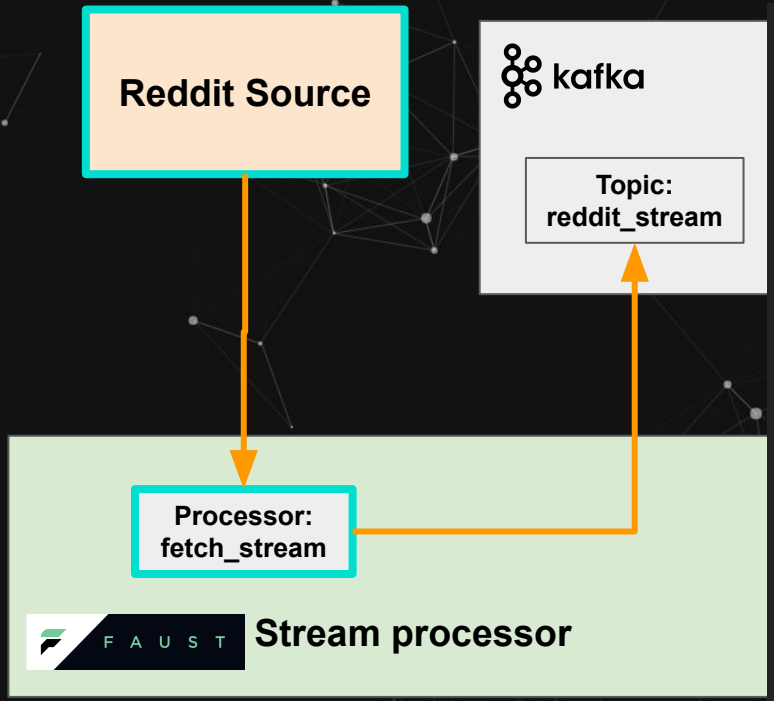


https://github.com/axsaucedo/reddit-classification-exploration/

Overview of Components

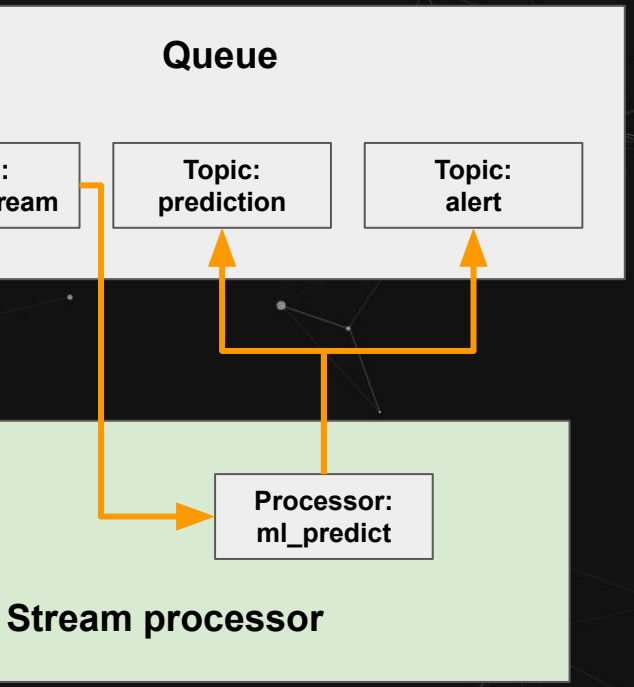


Generating comments




```python
@app.timer(0.1)
async def generate_reddit_comments():
      reddit_sample = await fetch_reddit_comment()

      reddit_data = {
          "id": reddit_sample["id"].values[0],
          "score": int(reddit_sample["score"].values[0]),
          ... # Cut down for simplicity
      }

      await app.topic("reddit_stream").send(
            key=reddit_data["id"],
            value=reddit_data)
```

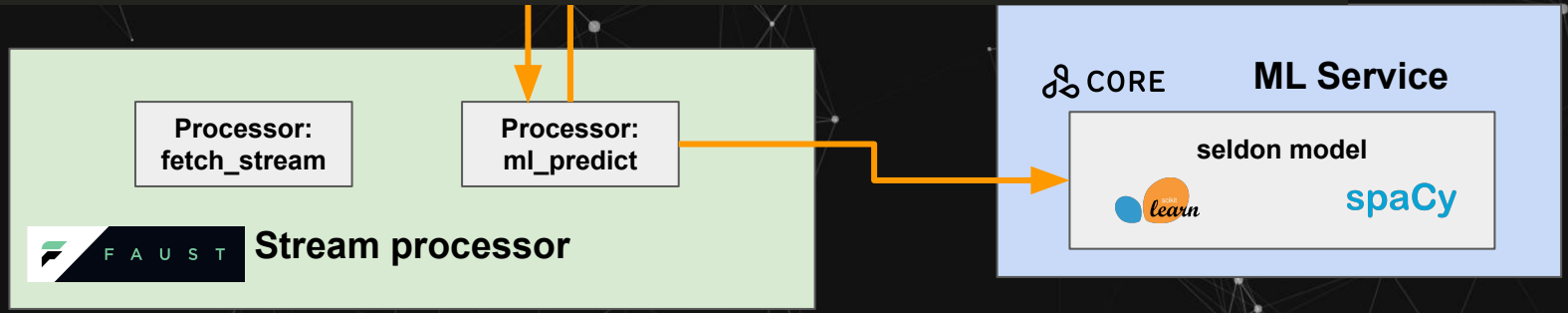
ML Stream Processing Step



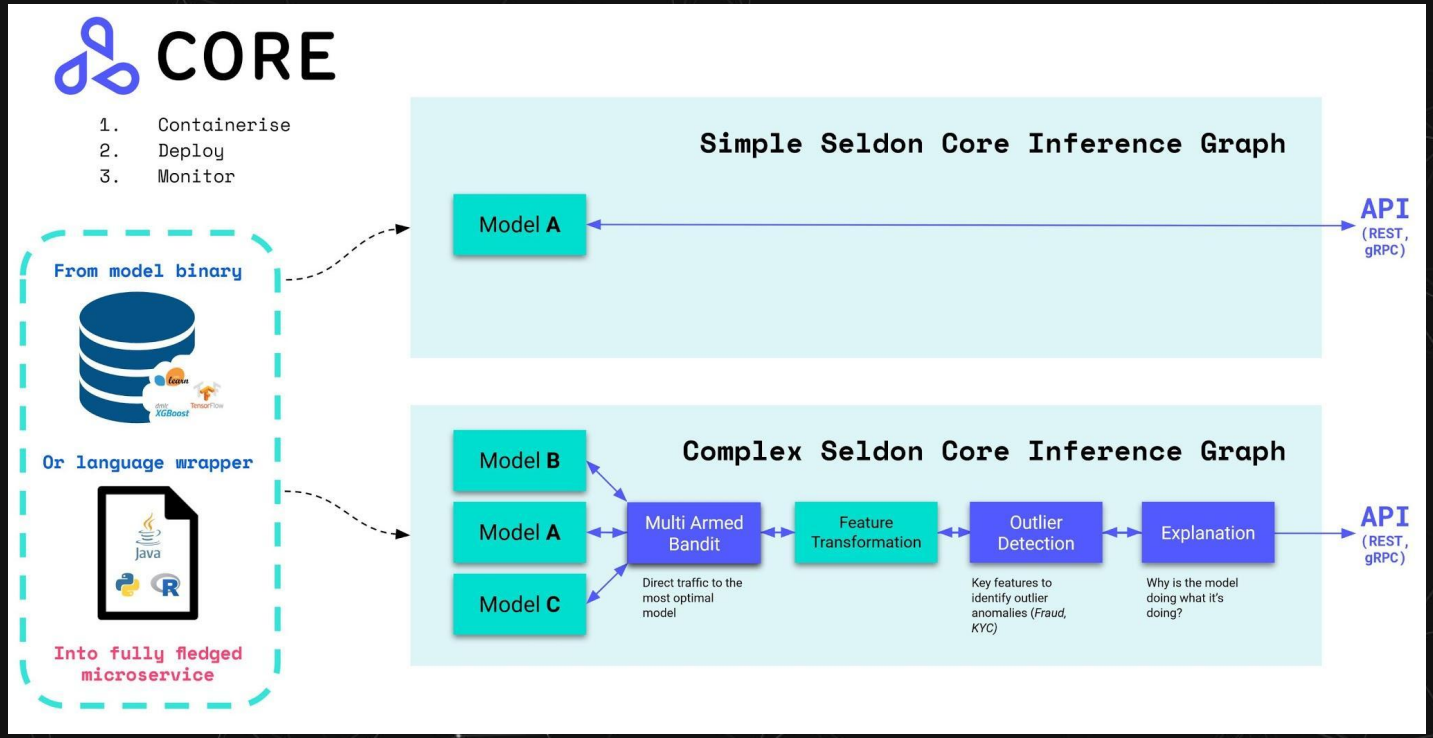

```
@app.agent(app.topic("reddit_stream"))
async def predict_reddit_content(tokenized_stream):
      async for key, comment_extended in tokenized_stream.items():
      tokens = comment_extended["body_tokens"]

      probability = seldon_prediction_req(tokens)

      data = {
                "probability": probability,
                 "original": comment_extended["body"]
      }

      await app.topic("reddit_prediction").send(
                     key=key,
                     value=data)

      if probability > MODERATION_THRESHOLD:
                await reddit_mod_alert_topic.send(
                           key=key,
                           value=data)
```

ML Model Request Step


```
sc = SeldonClient(
      gateway_endpoint="istio-ingress.istio-system.svc.cluster.local",
      deploment_name="reddit-model",
      namespace="default")

def seldon_prediction_req(tokens):
      data = np.array(tokens)
      output = sc.predict(data=data)
      return output.response["data"]["ndarray"]
```



Overview of Seldon Model Servi

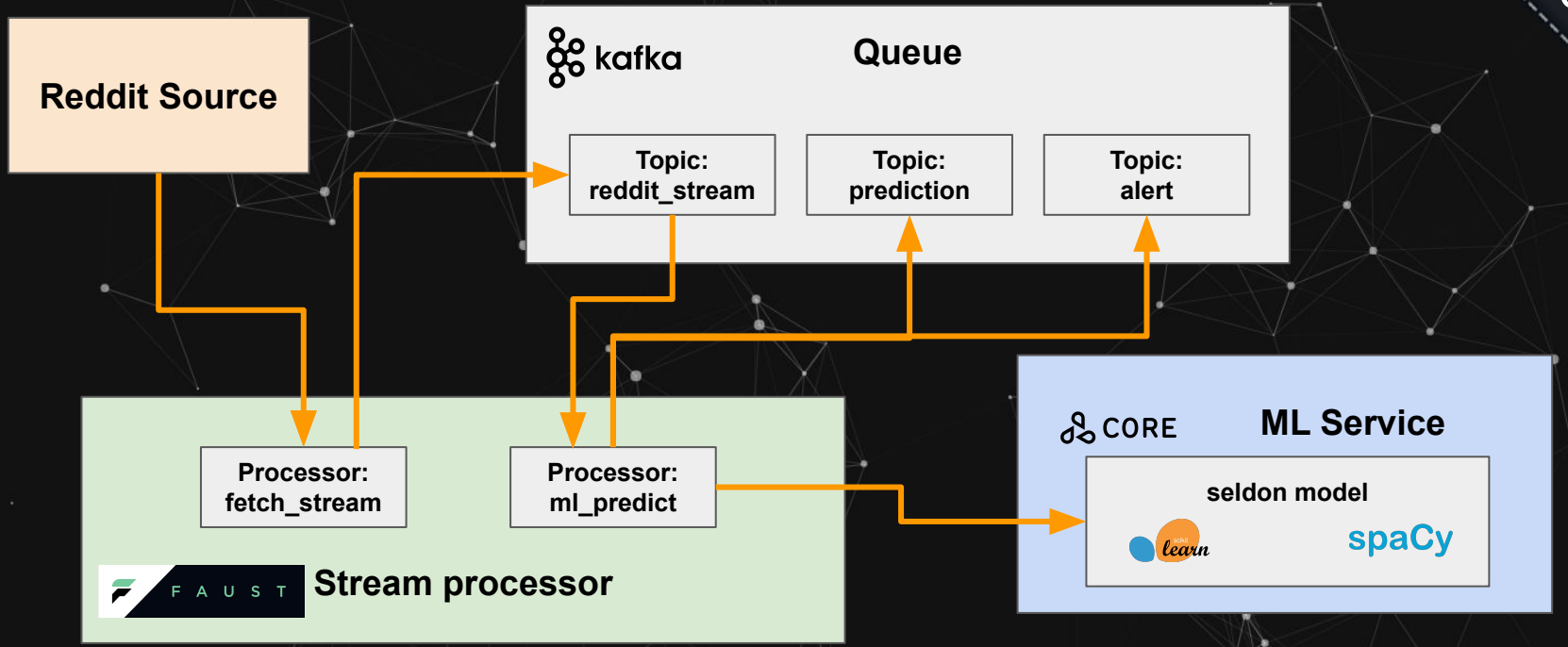


Wrapping ML models for Serving with Seldon

@AxSaucedo

```
import dill

from ml_utils import CleanTextTransformer, SpacyTokenTransformer

class RedditClassifier:

      def __init__(self):
          self._clean_text_transformer = CleanTextTransformer()
          self._spacy_tokenizer = SpacyTokenTransformer()

          with open('tfidf_vectorizer.model', 'rb') as model_file:
                self._tfidf_vectorizer = dill.load(model_file)

          with open('lr.model', 'rb') as model_file:
                self._lr_model = dill.load(model_file)

      def predict(self, X, feature_names):
          clean_text = self._clean_text_transformer.transform(X)
          spacy_tokens = self._spacy_tokenizer.transform(clean_text)
          tfidf_features = self._tfidf_vectorizer.transform(spacy_tokens)
          predictions = self._lr_model.predict_proba(tfidf_features)
          return predictions
```

Overview of Components



Recap of Today

- Conceptual intro to stream processing
- Machine learning for real time
- Tradeoffs across tools
- Hands on use-case

EuroPython 2020

Real Time Machine Learning with Python

Alejandro Saucedo | as@seldon.io

@AxSaucedo