Real time ML at ROKU

Krishna Chaitanya Chakka

Sr ML Engineer

Organized by HOPSWORKS

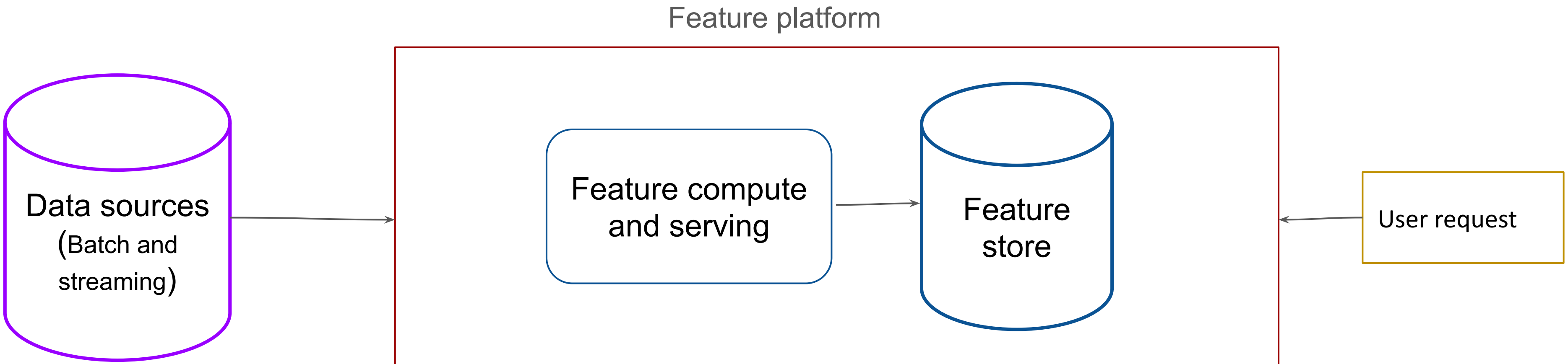
Scale at Roku

90 Million Streaming Households

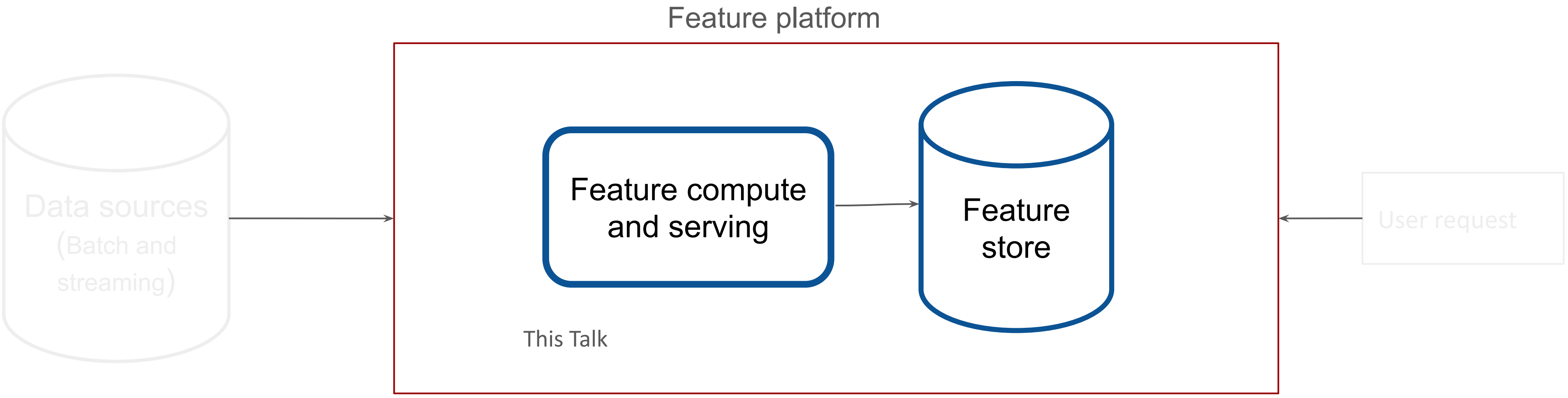
- Ads
  - Ad requests - 5B/day
- Search and Recs
  - Batch feature requests : 2M req/sec
  - Real time feature requests : 1200 req/sec (rapidly increasing this scale)

Feature Platform

Feature Platform



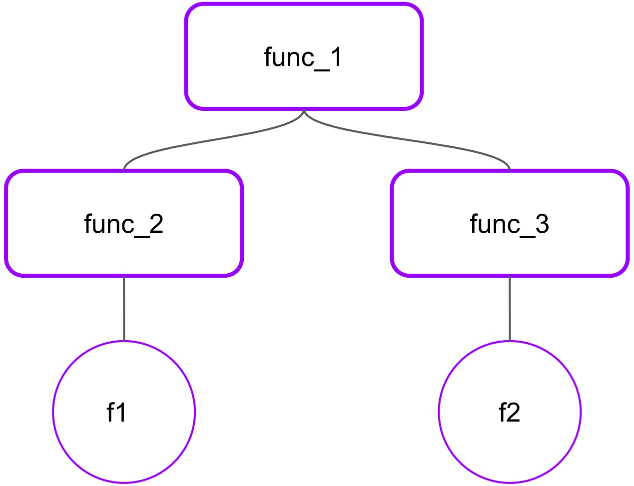
Feature Platform



Before Chronon

At Roku, custom frameworks built for different teams

Method 1: DAG based Python processors



Config

Intermediate_feature	func_1(user_activity)
final_feature_1	func_2(intermediate_feature)
final_feature_2	func_3(intermediate_feature)

Limitations

- Efficiency
  - Running python processes was slow
- No Real time feature computation
  - Have to implement java equivalent infrastructure
- No **Feature Store**
  - Compute feature transform everytime we train the model

Method 2

SQL ⇔ FLINK
(Batch) (Real time features)

Experimentation with new real time features

- Log and Wait approach
- Log new features and wait till training data is generated
- Multiple feature definitions for the same feature.

Limitations

- Longer development times
  - Wait for training data to be logged.
- Can impact production if testing larger number of features.
- Not reusable for other applications
  - Time sensitive.

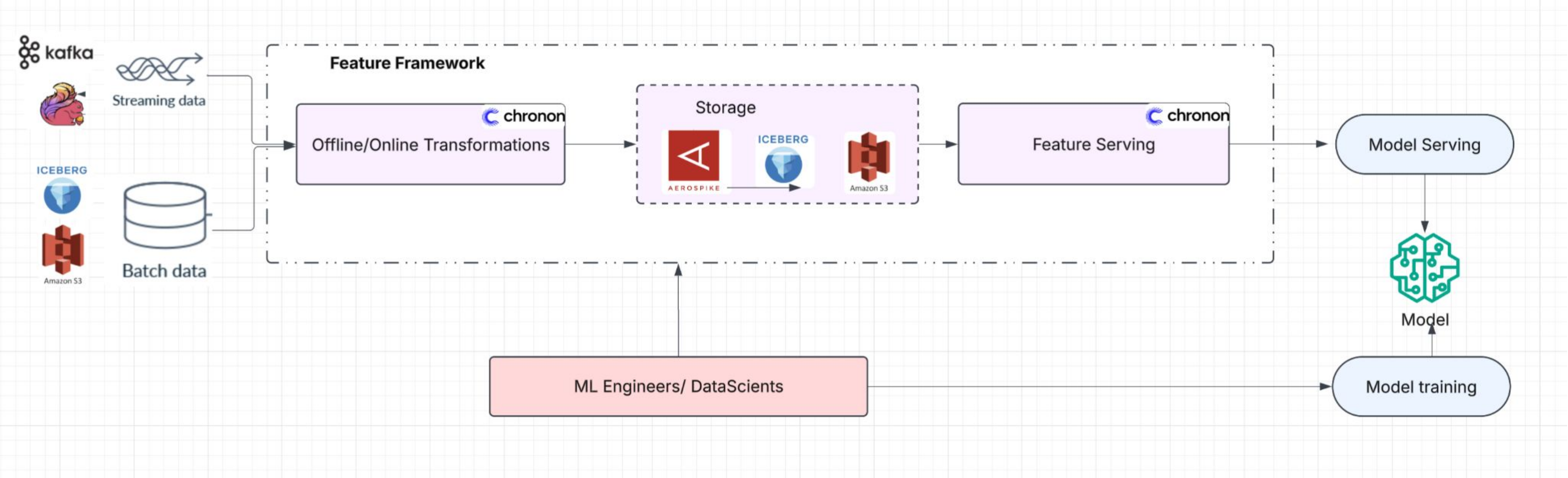
Chronon as feature platform

Chronon

- OpenSource from Airbnb
    - Active community
    - Weekly office hours quickly help in resolving issues.

- Python API with Scala Spark
    - Write chronon config once and deploy to get features
    - Increase developer velocity with model testing

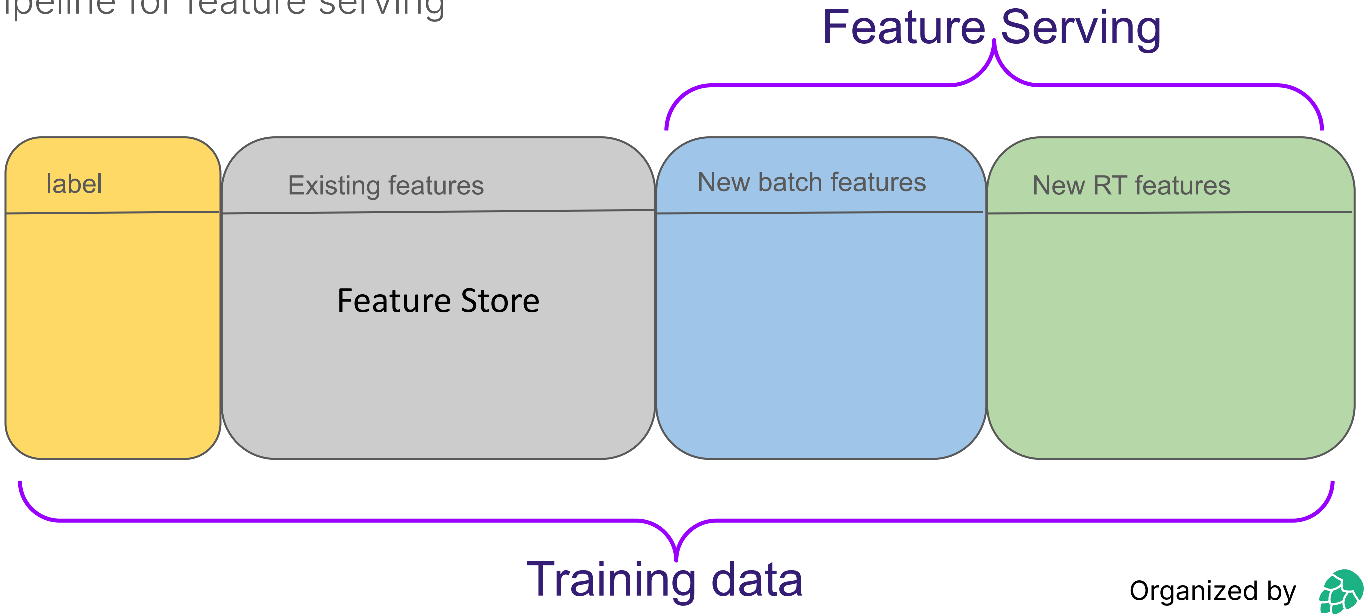
High level Architecture



Feature lifecycle

Feature Compute : Backfill

- Training data with new batch and NRT features for a window (180 days).
- Pipeline for feature serving

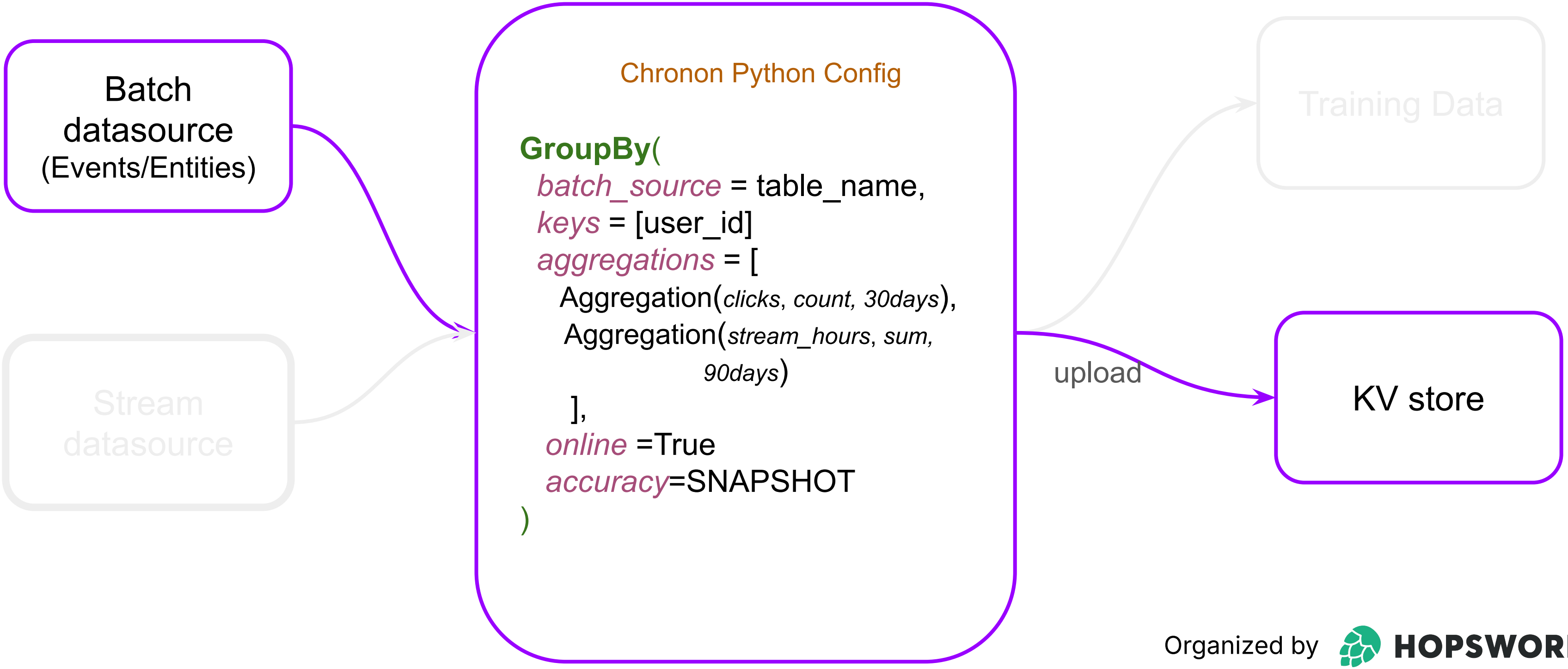


Batch features - Serving

- Example feature
  - `count_of_clicks_on_content_id_past_90days`

- Run a nightly job to compute features for all user_ids
- Upload the output to online KV store for serving

Batch features - Serving

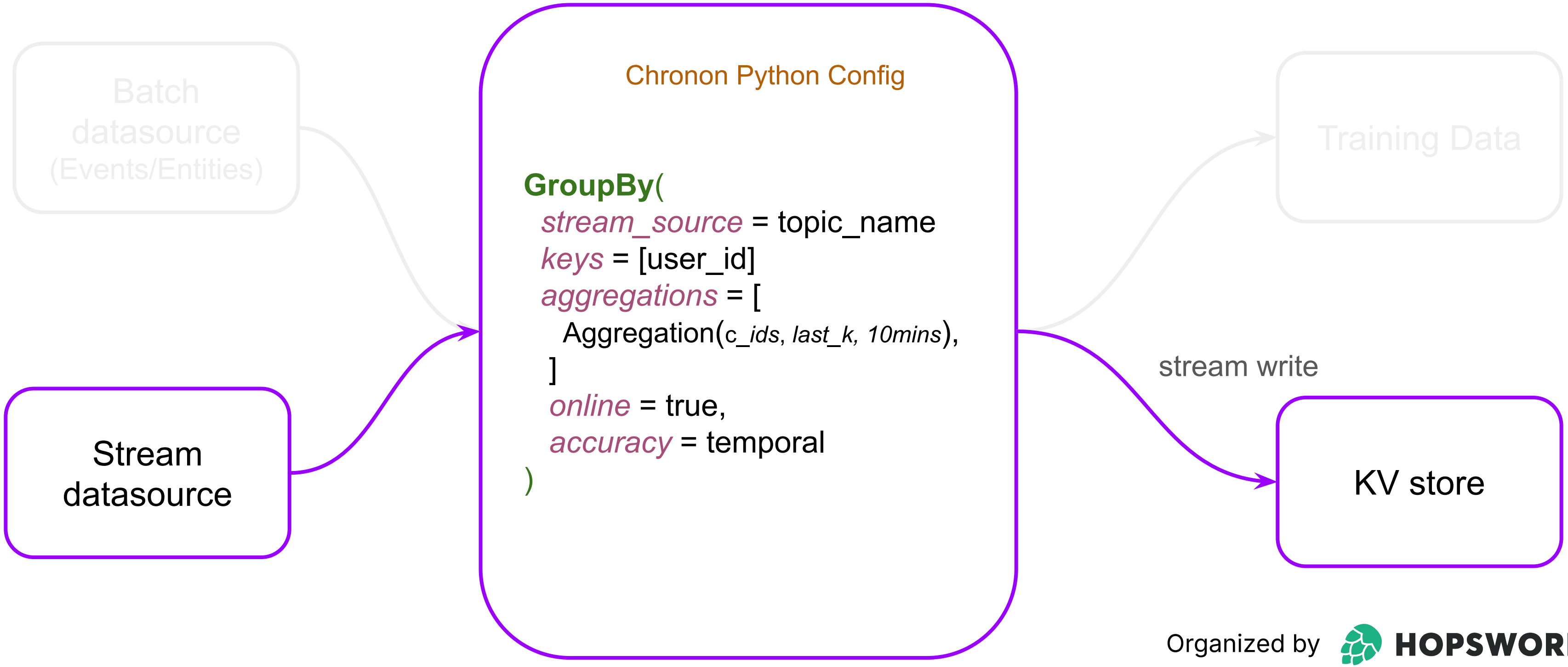


Serving - Real time features - Streaming Source

- Example feature
    - `count_of_clicks_on_content_id_past_10mins`

- Read events from flink and generate feature value for a small window (eg: 2 mins)
    - At request time, read 5 windows (2 min each) and aggregate the feature.

Real time features - Serving

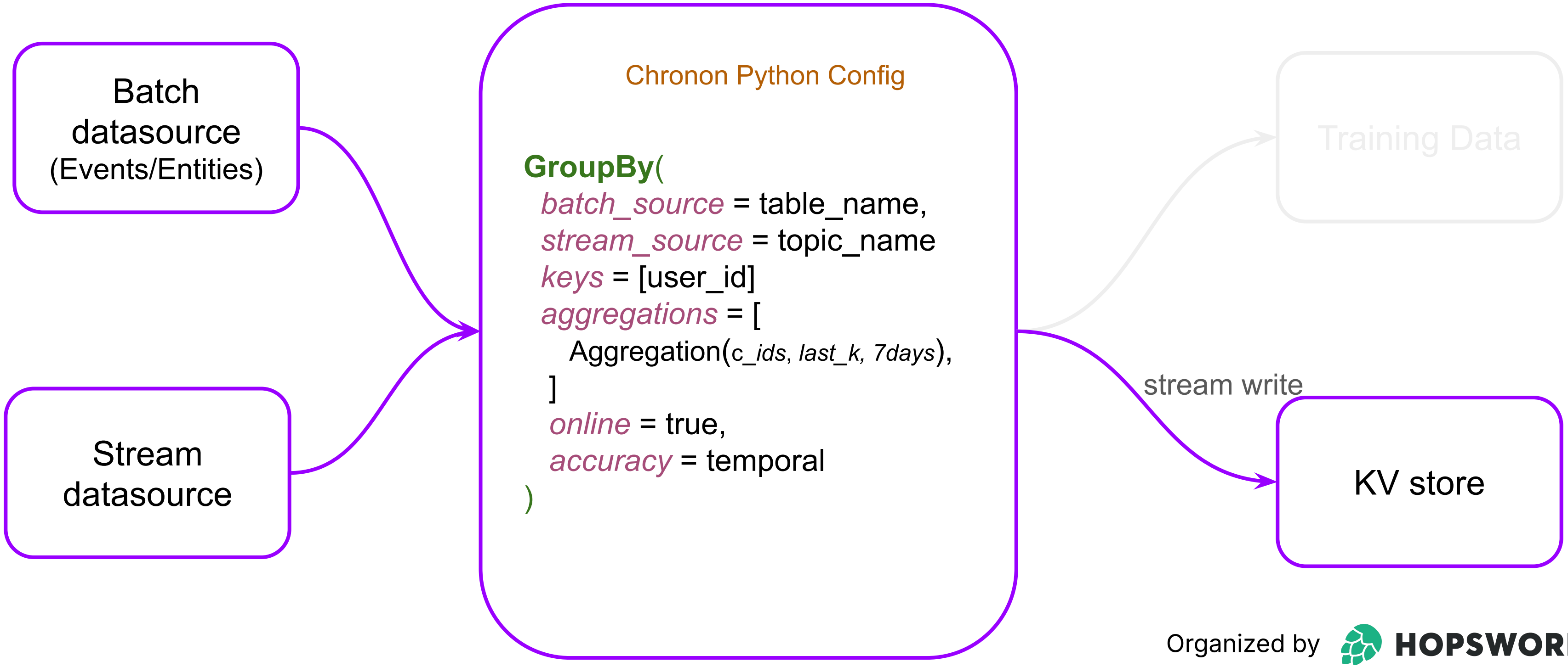


Serving - Real time features - Stream and Batch Source

- Example features
  - `count_of_clicks_on_content_id_past_7days`
  - `last_k_genres_clicked_past_2days (k = 10)`

- Read events from flink and generate feature value for a small window (eg: 2 mins)
  - At request time, read 5 windows (2 min each) from kv store
- Read daily **precomputed batch aggregates** from kv store
- Combine feature values from both sources.

Real time features - Serving



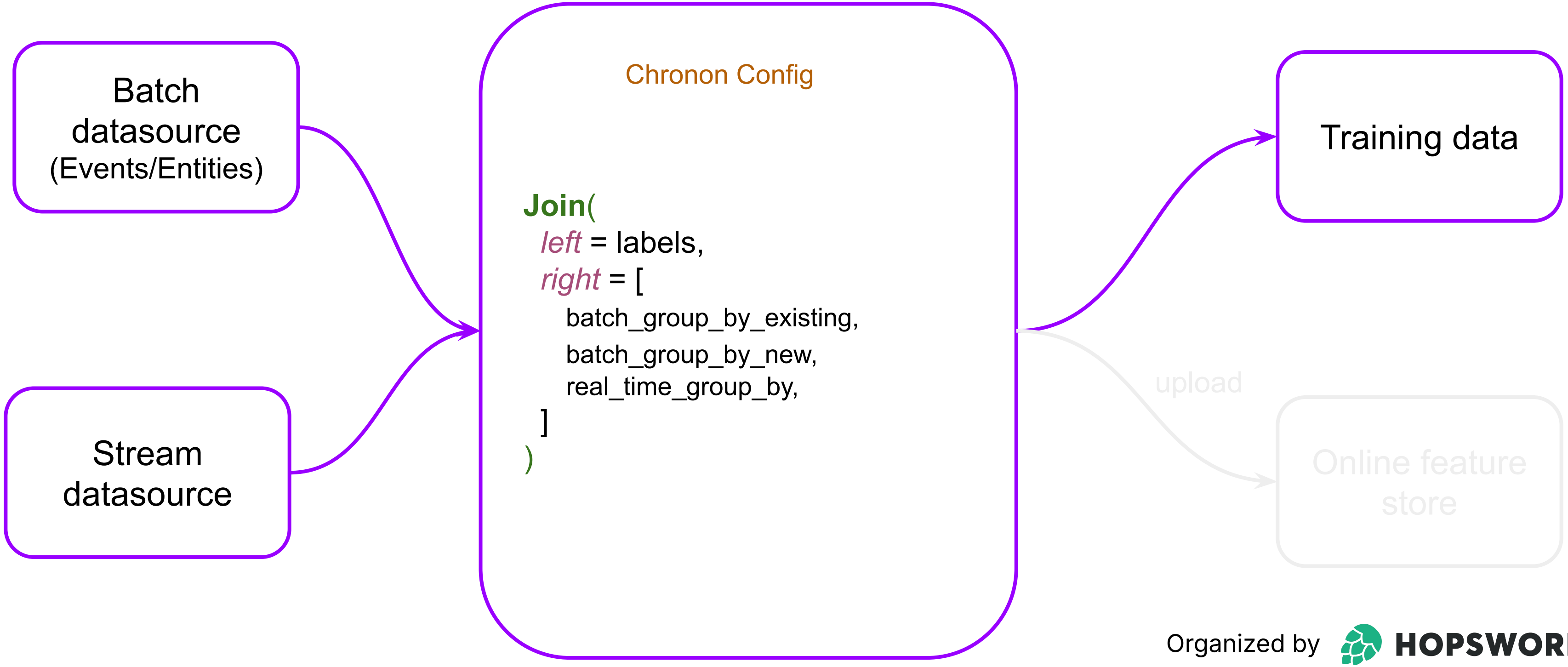
Feature Serving

- Simple API to fetch features after chronon config deployment

- **Feature Freshness** : 1-2 mins
  - Any user event is available in 1-2 mins

```
feature_fetch(
    table_name : chronon_output
    keys : user_id,
    col_names : [feat1, feat2],
)
```

Training data



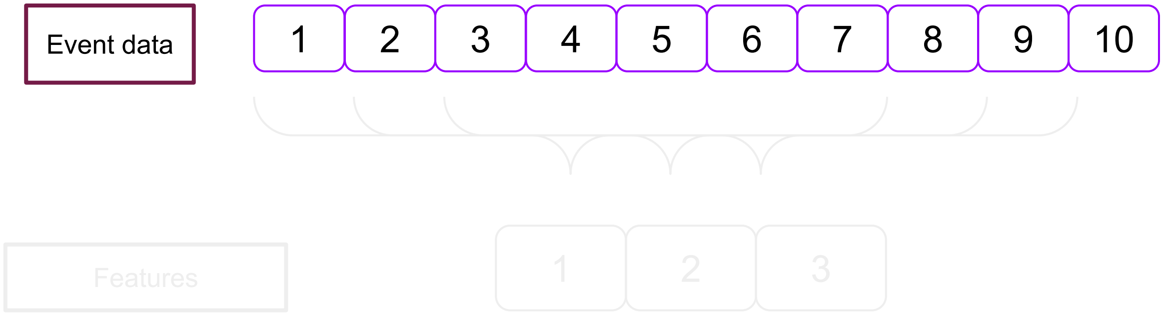
Chronon Optimizations - Bloom filter

- Left Joins
  - Bloom filter to filter out features (right parts) that is not in the label dataset
  - Huge time savings during data shuffle.

Chronon Optimizations - Windowed Backfill

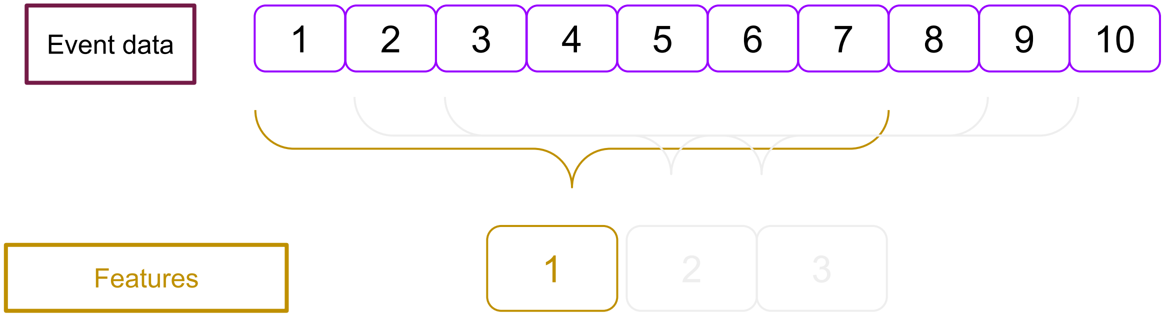
Backfill Window : 6 months, Feature window : 7 days

- Sliding window optimizations - 7day aggregation



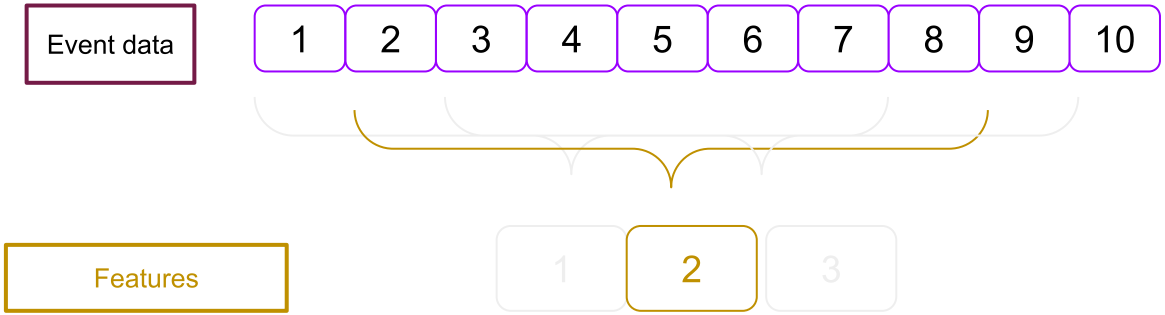
Chronon Optimizations - Windowed Backfill

- Sliding window optimizations - 7day aggregation



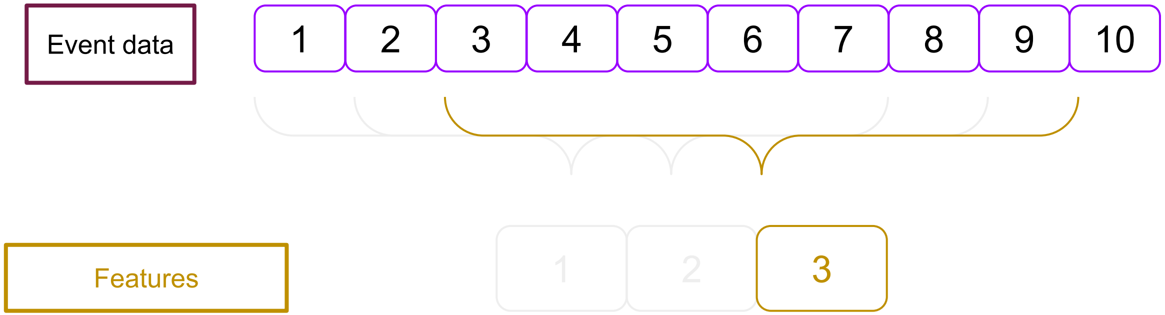
Chronon Optimizations - Windowed Backfill

- Sliding window optimizations - 7day aggregation



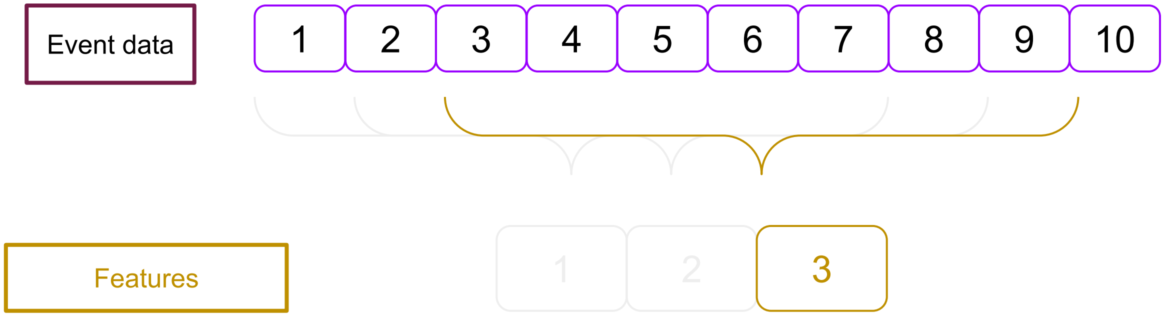
Chronon Optimizations - Windowed Backfill

- Sliding window optimizations - 7day aggregation



Chronon Optimizations - Windowed Backfill

- Sliding window optimizations - 7day aggregation



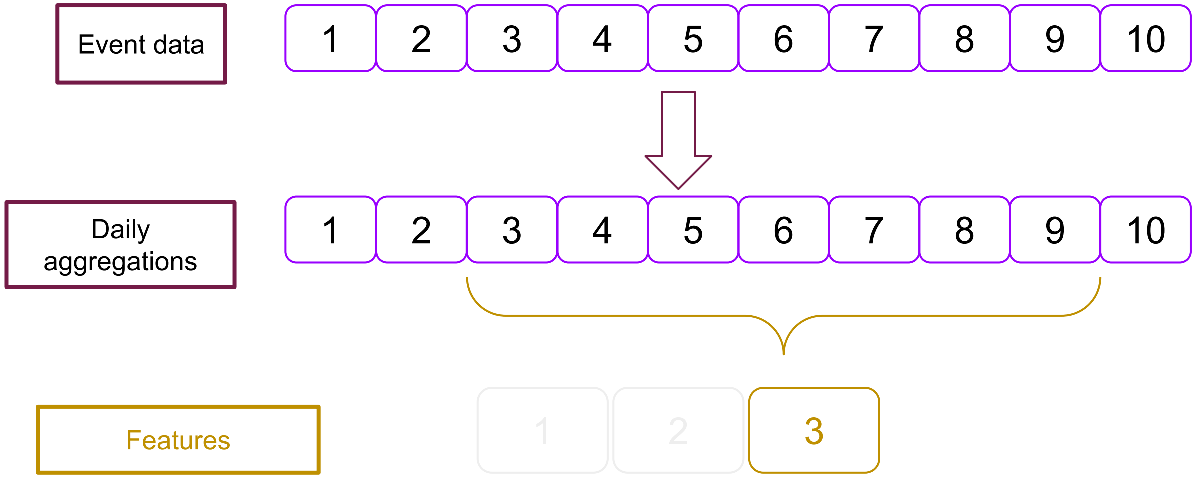
Event level daily aggregation is computed only once per backfill.

Features = Aggregation over daily partial aggregates.

Chronon : Roku contributions

WIP : Incremental Aggregations - Batch

- Sliding window optimizations - 7day aggregation



Instead of events, compute features from intermediate daily aggregations

Chronon : Roku contributions

- Bazel
  - Spark 3.5

- Bazel publish artifacts

Thank You