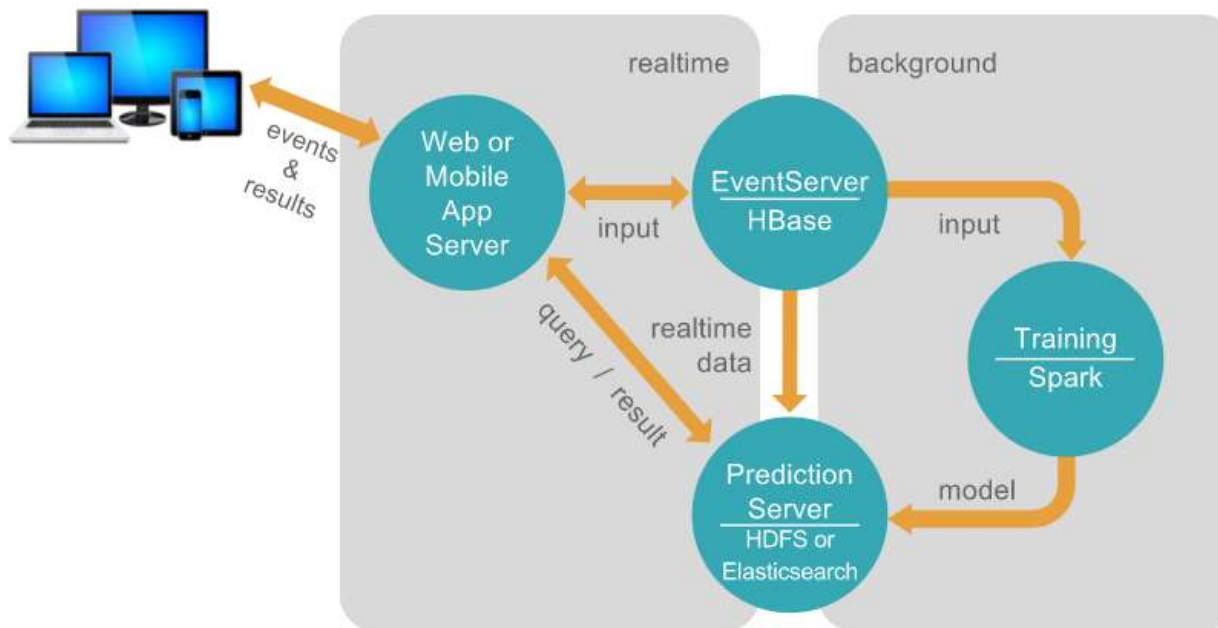
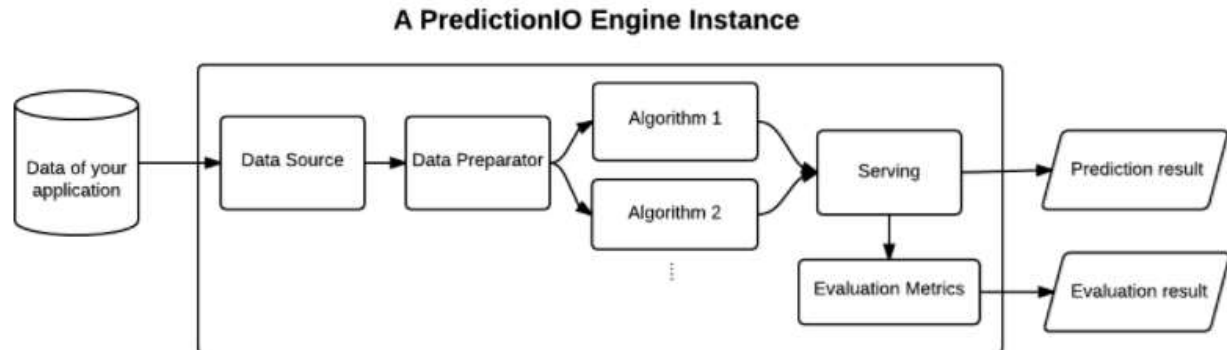


RECOMMENDATION SYSTEM USING PREDICTION IO

I/ Architecture



II/ Pipeline



- **Data Source**: loading data from event's channel & transform to desired format. Our Recommendation system includes 2 kinds of data: item, containing information of each movie/music, & event, describing interaction between 1 user & 1 item such as watching movie, listening to music, etc.
- **Data Preparator**: preprocessing data & forward it to the algorithms.
- **Algorithm**: fitting models to training data and save them to database.
- **Serving**: combining result from different & return the final recommended list to each query.

III/ Project structure

- engine.json**:
 - including configuration for the algorithm and serving.
- Datasource.scala**

- Loading item list (getItems) from channel items & event list (getEvents) from channel events.
- In getEvents, we check if each user watches less than N movies => events belonging to that user are removed.
- Currently, we don't have the evaluation part so that you can ignore the readEval method in Datasource.
- c) CFAlgorithm.scala
 - For more information, read template for E-Commerce Recommendation
 - Paper: Collaborative Filtering for Implicit Feedback Datasets.
- d) CBAAlgorithm.scala
 - For more information, read template for Content Based SVD Item Similarity Engine.
- e) TopMostAlgorithm.scala
 - Calculating total playing time for each movie based on events from events channel & ranking them based on total playing time.
- f) Serving.scala
 - Linearly combine results from CF, CB, & TopMost.
 - $FinalScore = CF \times a_1 + CB \times a_2 + TopMost \times a_3$
 - Coefficient $a_1, a_2, \& a_3$ is included in each query.