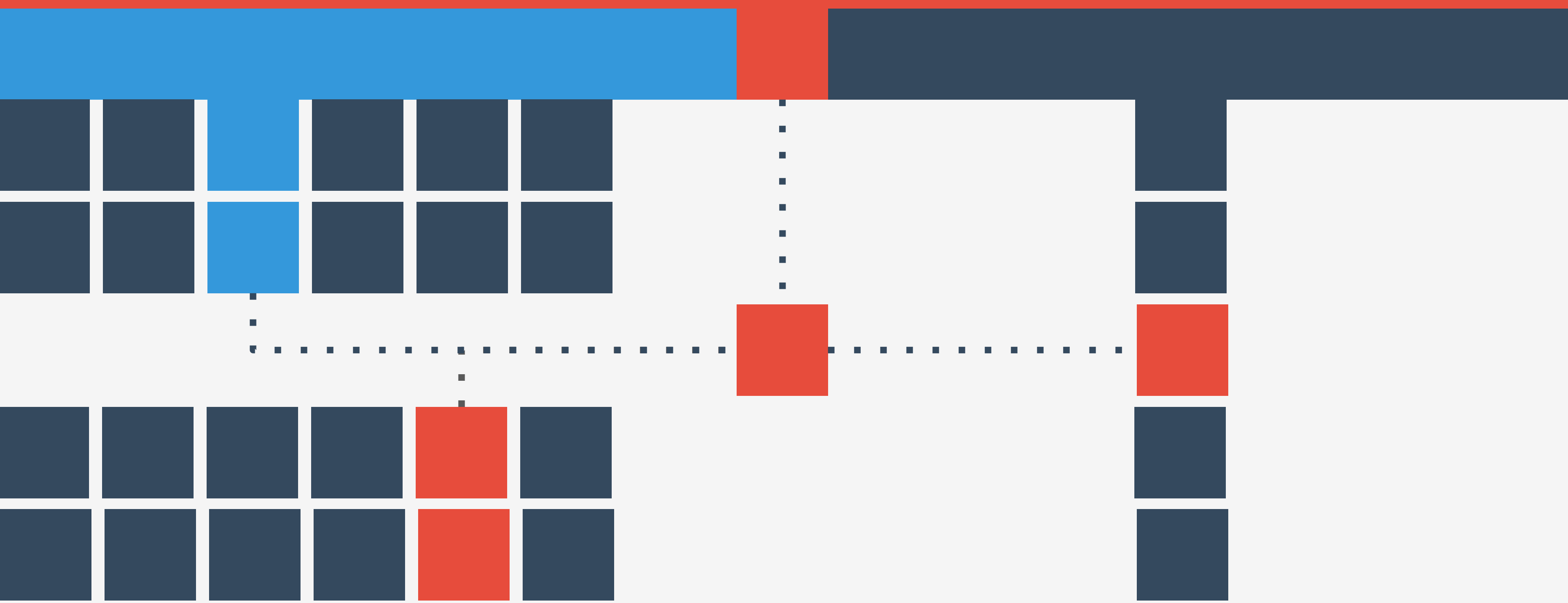
RecSys 2017 Online Ranking Tutorial

Róbert Pálovics Domokos Kelen
Dániel Berecz András A. Benczúr

Informatics Laboratory of the Hungarian Academy of Sciences



https://github.com/rpalovics/recsys-2017-online-learning-tutorial

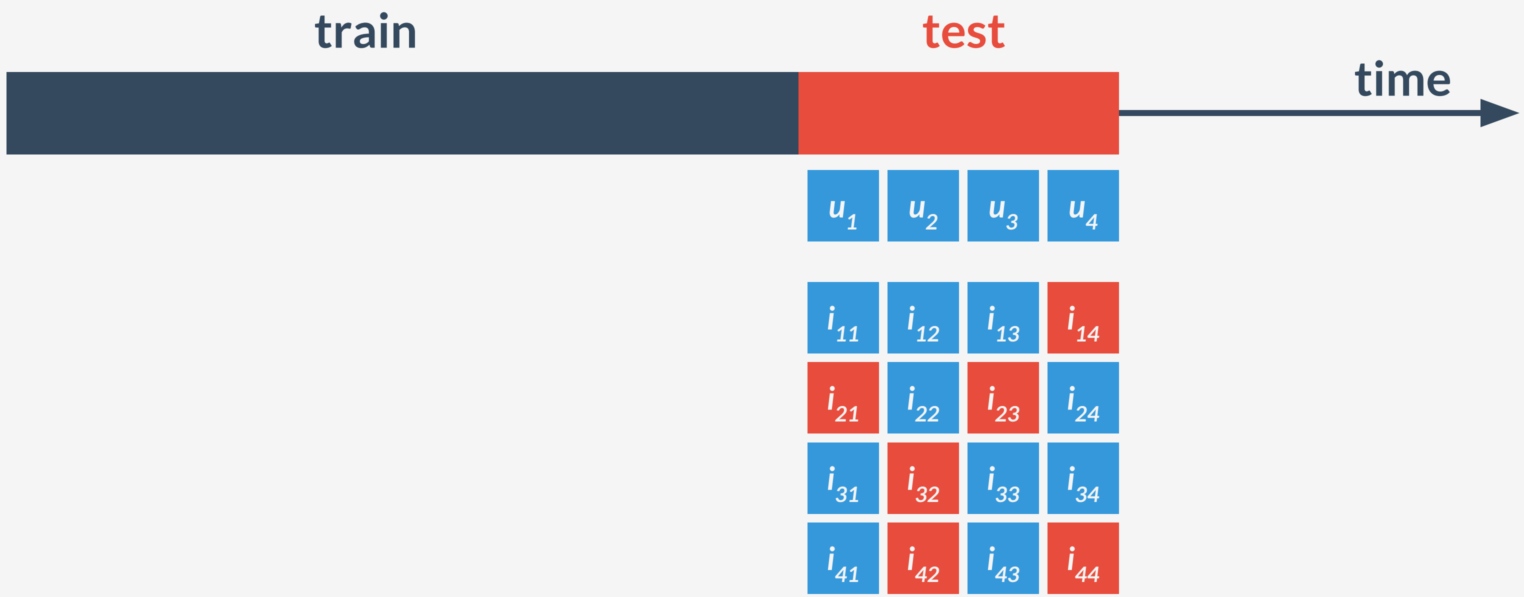
Objectives

Examples: hashtag recommendation on Twitter, news recommendation, music recommendation

- **Implicit** recommendation
- **Top-*k*** recommendation
- **Time-aware models** and **time-aware evaluation**

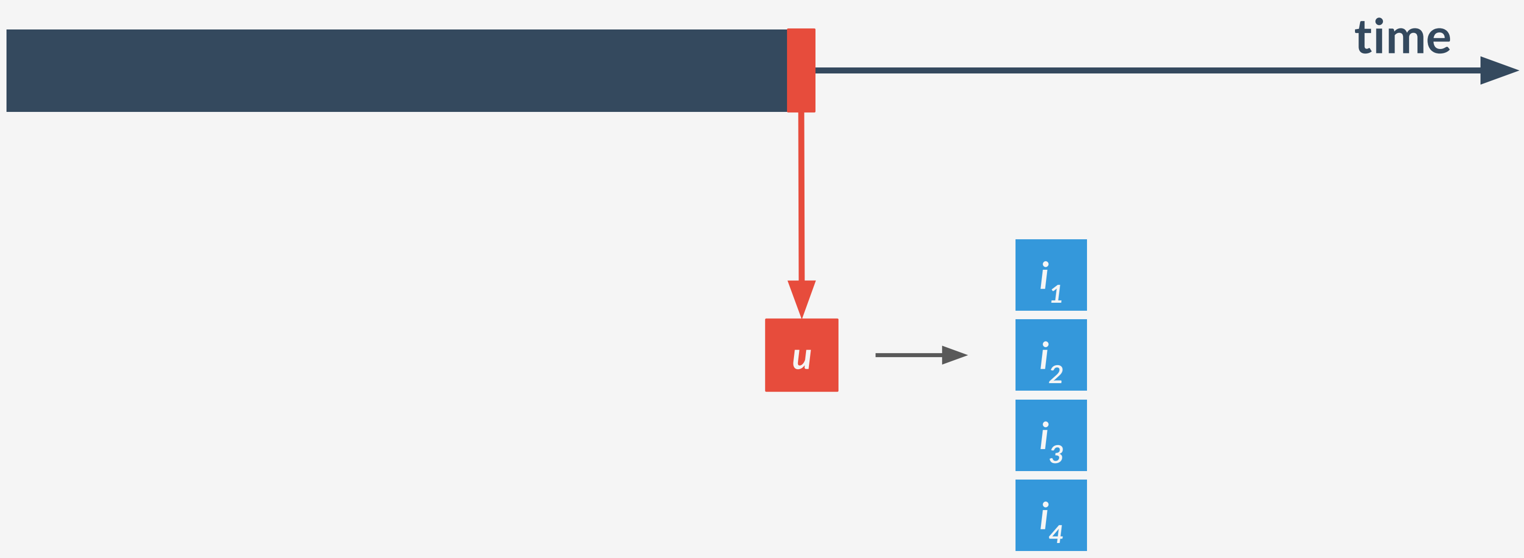
Batch [time-aware] Top-K Recommendation Task

- Learn from the past batch training set
- Recommend for each user in the test a top-*k* list
- Evaluate based on the test set



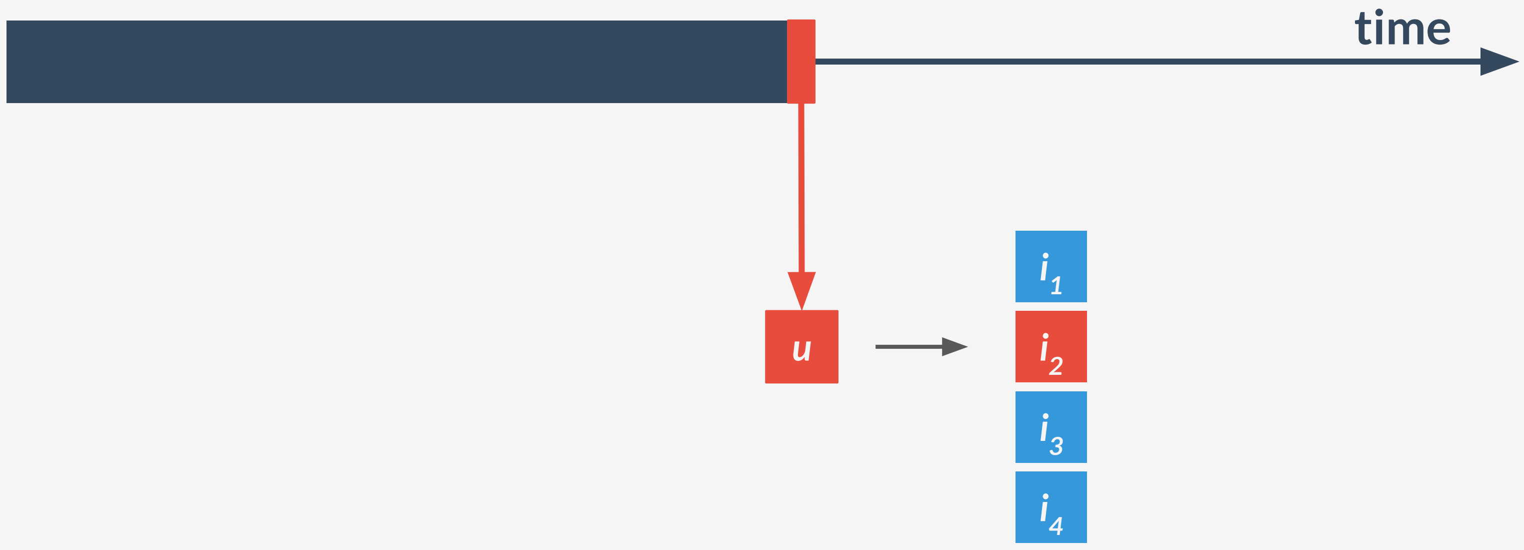
Online [time-aware] Top-K Recommendation Task

- We use timestamped implicit data
- Process the events in the data in temporal order
- After each event (u,i,t)
  - recommend a new top list of items
  - then update the recommender model



Online [time-aware] Evaluation

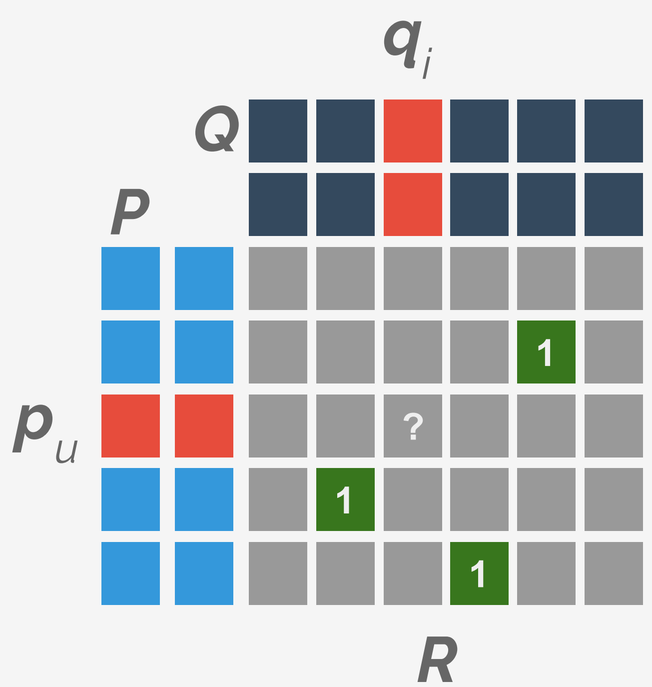
- Evaluate the given single tuple (*u*,*i*,*t*) in question against the recommended top list
- DCG = 1 / (log$_2$(rank(***i***) + 1)
- Compute timely averages, e.g. daily average DCG



Matrix Factorization

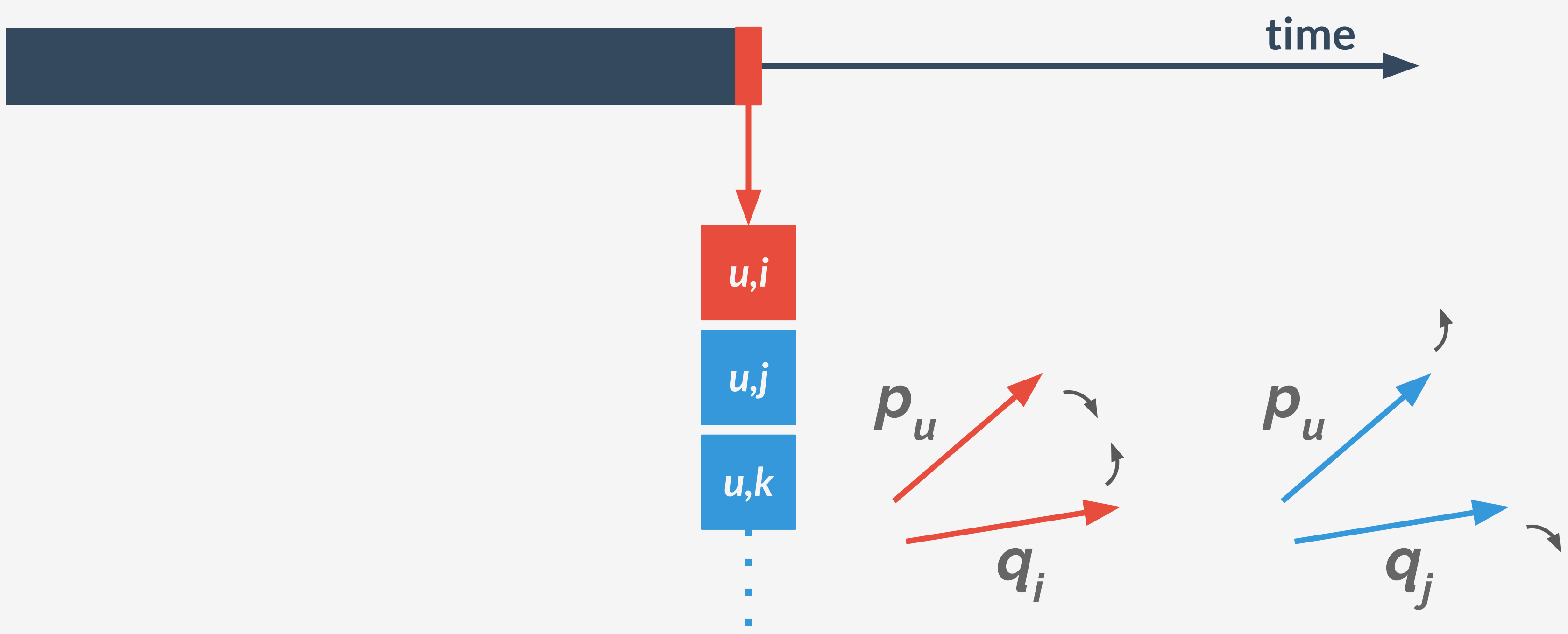
- Data
  - sparse matrix $\boldsymbol{R}$
  - $r(u, i) = 1$, if u interacted with i
- Model
  - $\boldsymbol{P}$ and $\boldsymbol{Q}$ matrices for the users and the items respectively
  - "probability" of an interaction

    $r(u, i) = \boldsymbol{p}_u \boldsymbol{q}_i$
- Learning
  - objective: MSE
  - optimization: iALS or SGD



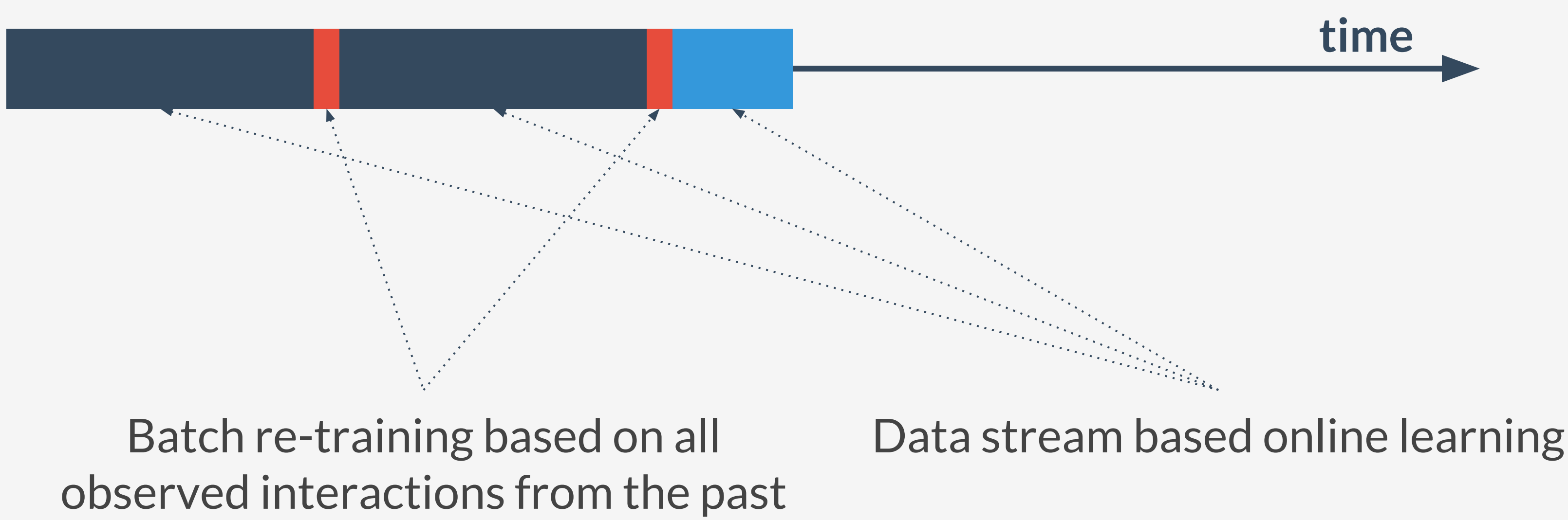
Online Matrix Factorization on Implicit Data

- Single iteration over the training data
- We process the events in temporal order
- Optimize for MSE with SGD
- Generate random negative samples for the given user



Batch then Online Matrix Factorization

- Periodically re-learn the batch model
- Between two batch model building, continue the learning of the previous batch model via online matrix factorization



Recommender frameworks

Alpenglow

free and open source C++ based framework with Python API for conjoint batch and online learning

Flink

open-source stream processing framework with batch and streaming API
MF algorithms are available only as pull requests

SparkML

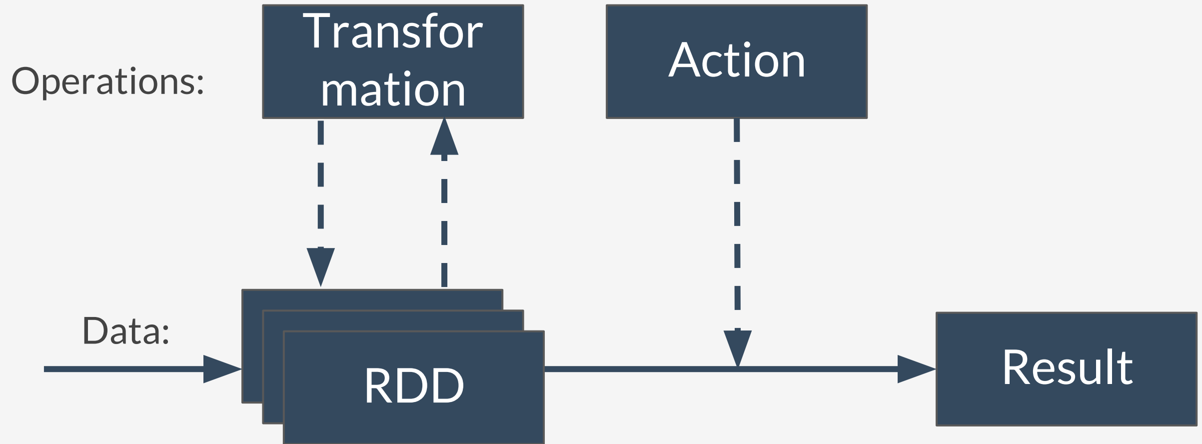
provides batch iALS baseline

Alpenglow

- Open source C++ recSys framework
- Easy-to-use Python API
- Supports conventional batch training and evaluation
- Capable of online training of recSys models
- Online trained models can adapt to concept drift
- Compatible with Jupyter/Zeppelin Notebooks and Pandas

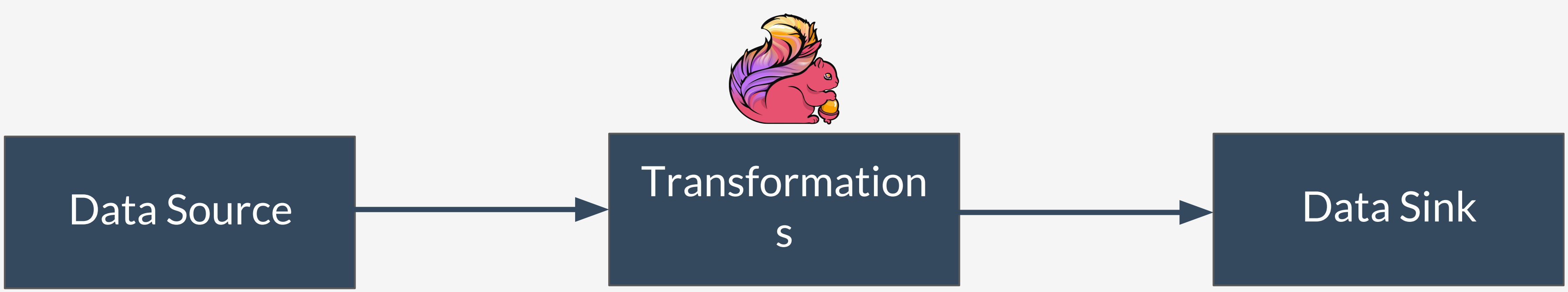
Spark

- Fast and general engine for large-scale data processing
- Basic abstraction:
  - RDD - immutable
- Basic operations:
  - Transformations
  - Actions
- Lazy evaluation
  - Optimization
  - Reduce complexity



Flink

- Distributed Stream processing framework
- Basic abstraction
  - DataStream
- Building blocks
  - Data Source
  - Data Transformation
  - Data Sink



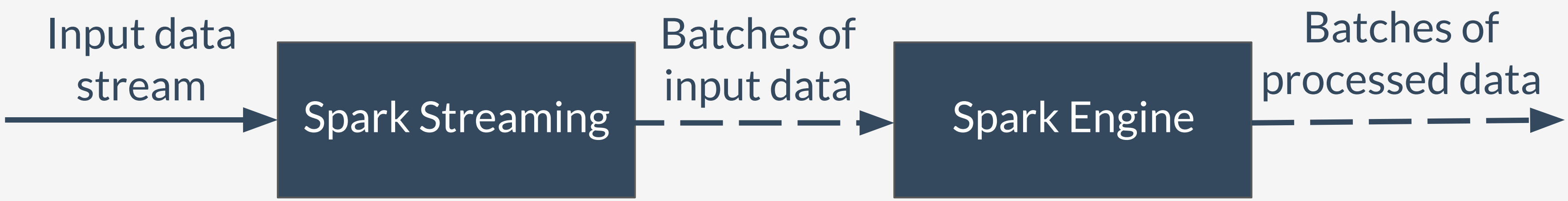
Data sets and executions models

- Two types of data:
  - Bounded
  - Unbounded

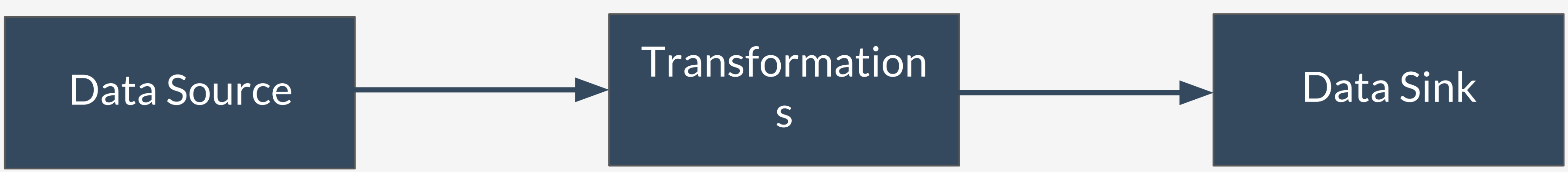
- Three types of execution model:
  - Batch - Spark
  - Micro batch - Spark Streaming
  - Streaming - Flink

Comparing Spark Streaming and Flink

- Spark Streaming



- Flink



Why distributed

Examples: too many artist in music recommendation, too many products in webshops

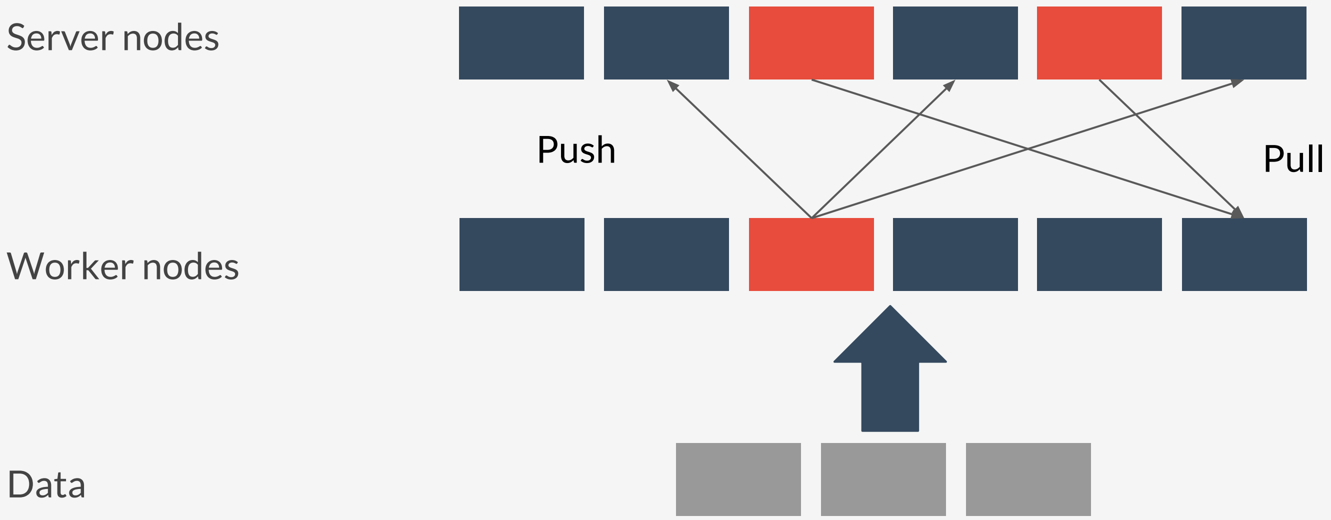
- **Factor matrices** can be too big
- **Top-*k*** recommendation can be slow

LEMP* for fast Top-k generation

*Teflioudi: Fast retrieval of Large Entries in a Matrix Product

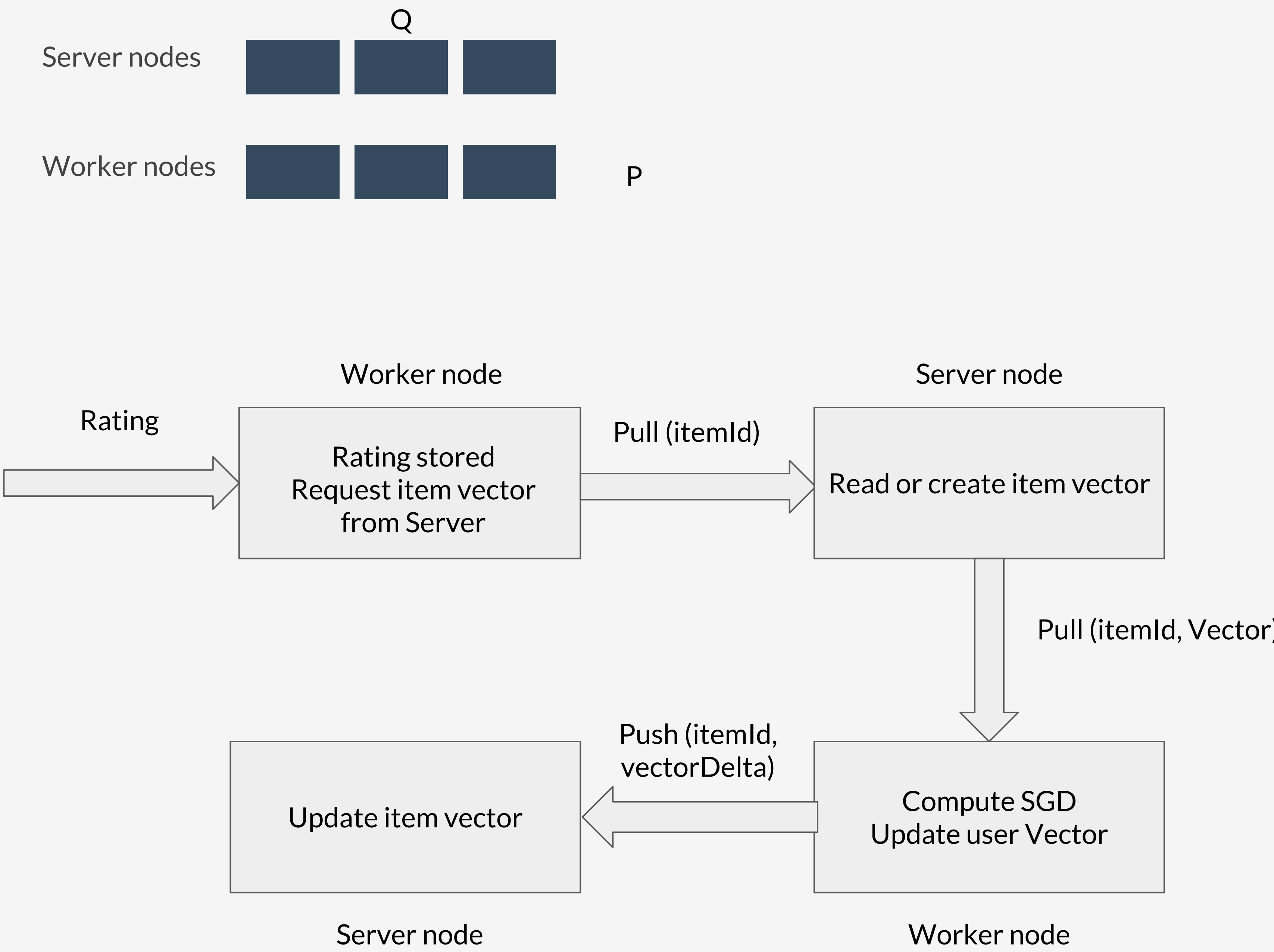
Parameter Server*

- Abstraction for model-parallel machine learning
- Architecture
  - Model on server nodes
  - Computation on worker nodes
  - Communication via push and pull messages

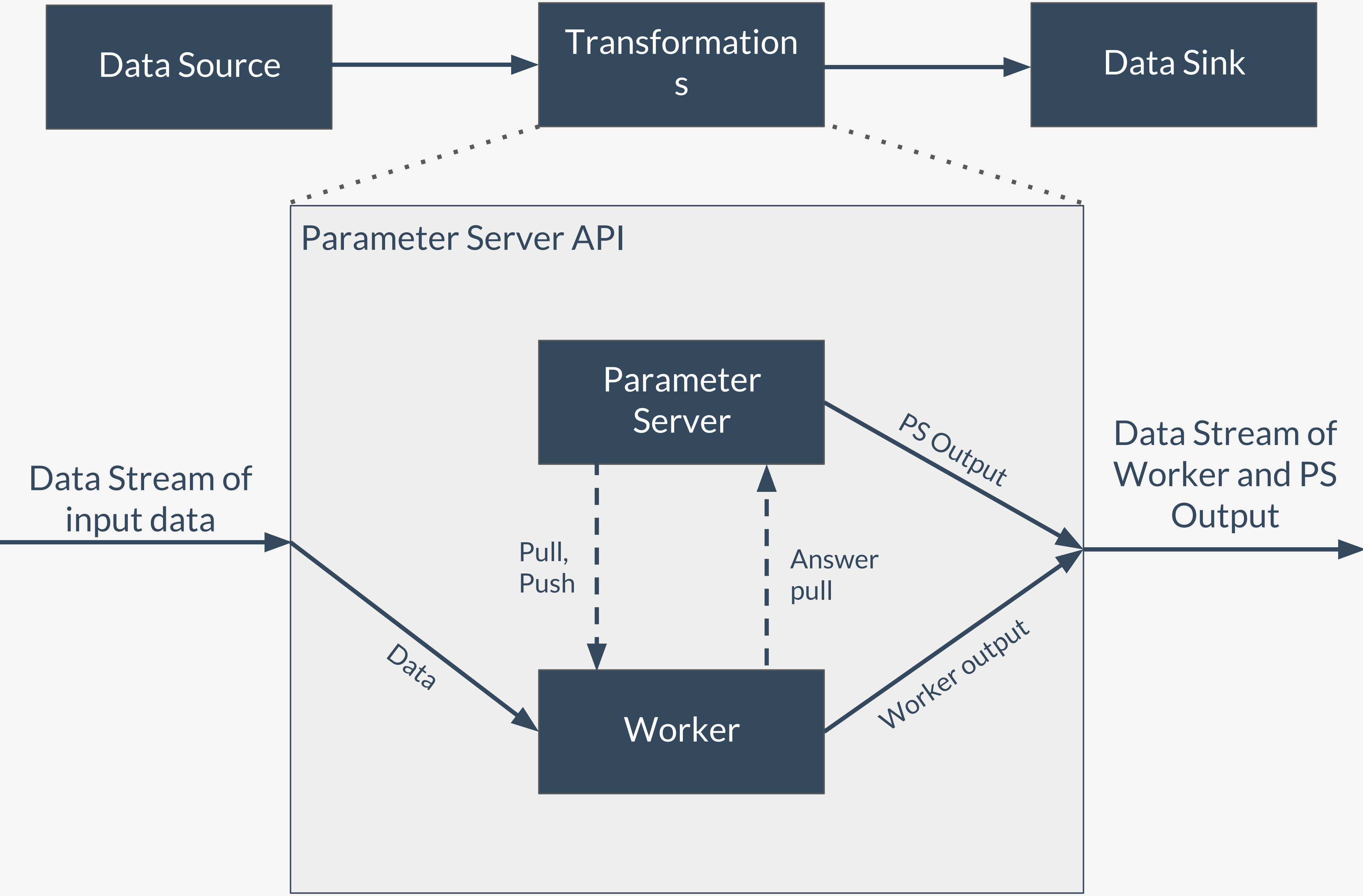


Mu Li: Scaling Distributed ML with Parameter Server

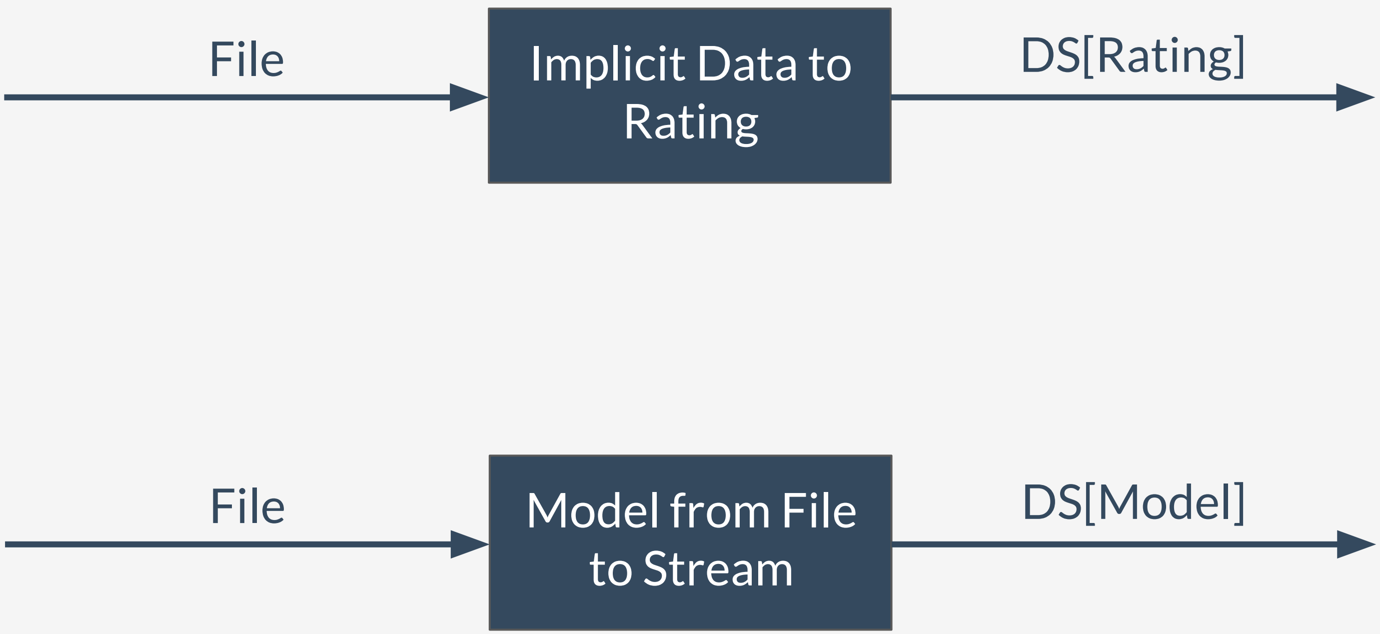
PS Online Matrix Factorization WorkFlow



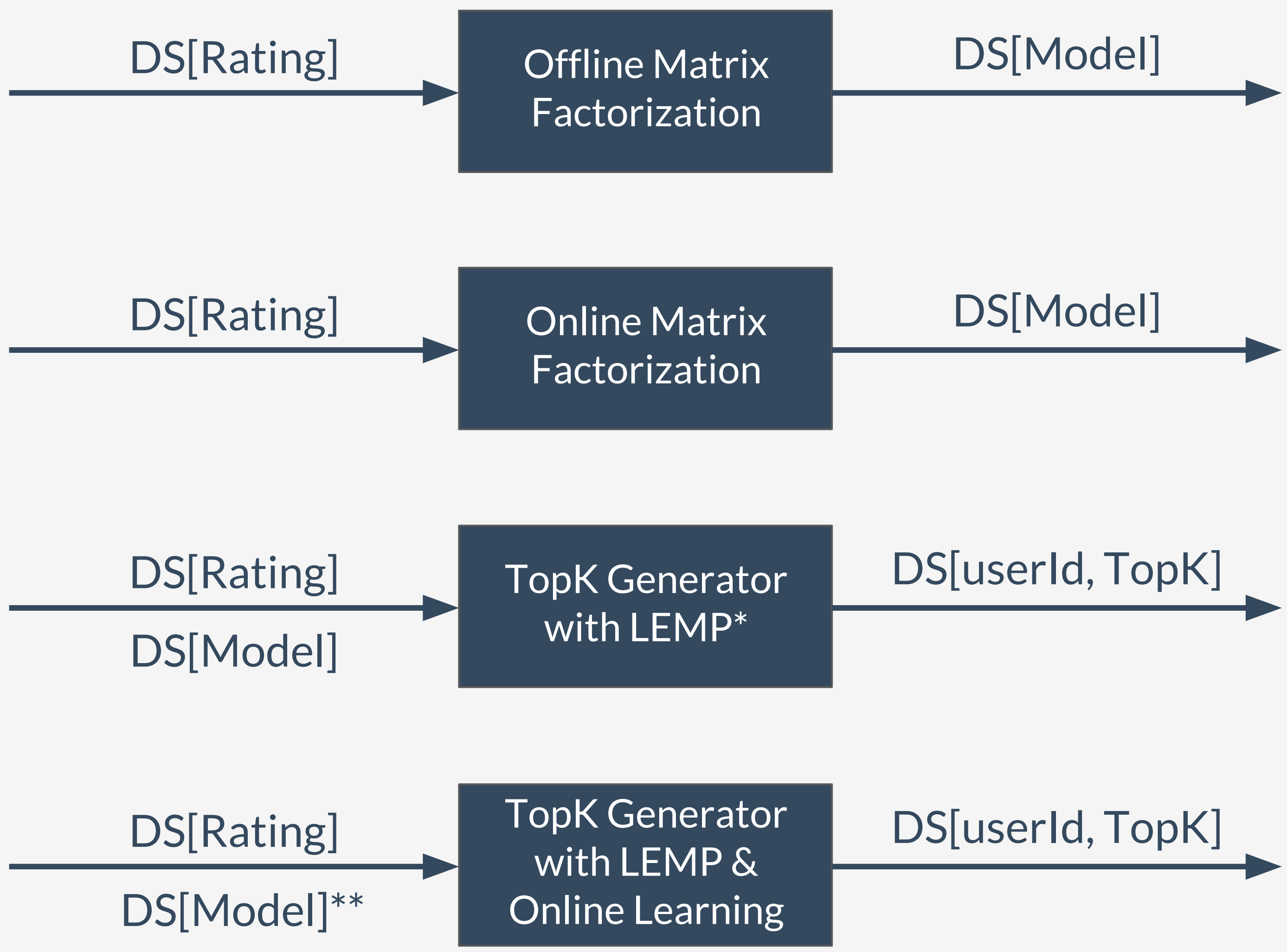
Our solution



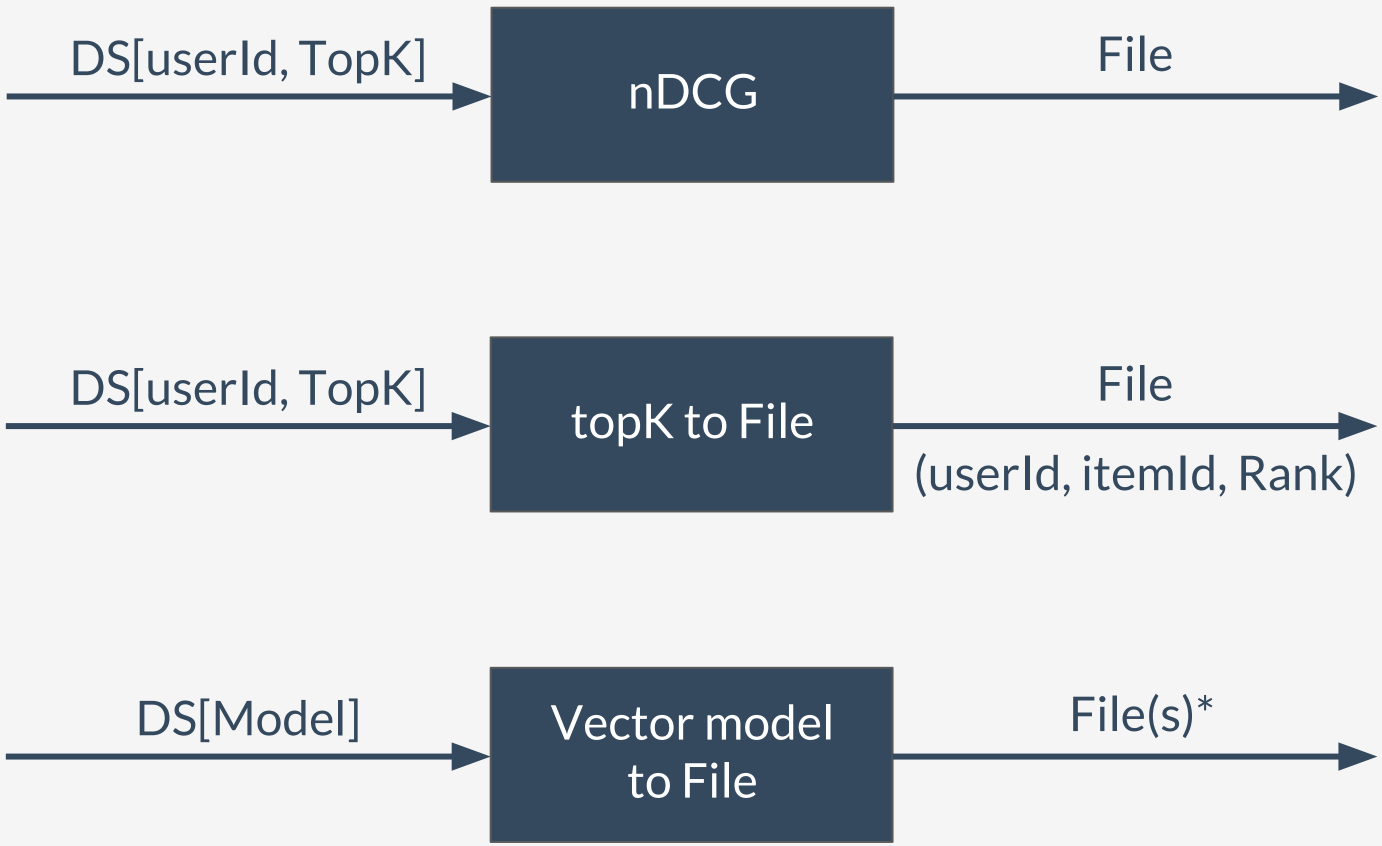
Data Sources



Transformations



Data Sinks



*Options: item and user vectors in separate files or not