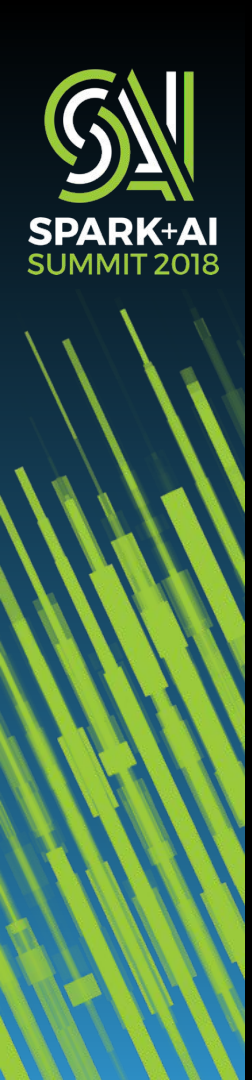


Fact Store - Netflix Recommendations

Kedar Sadekar, Netflix
Nitin Sharma, Netflix

#DevSAIS11

Agenda

- Recommendations
- Experimentation
- Fact vs Feature logging
- Fact Store Architecture
- Scaling challenges
- Future work

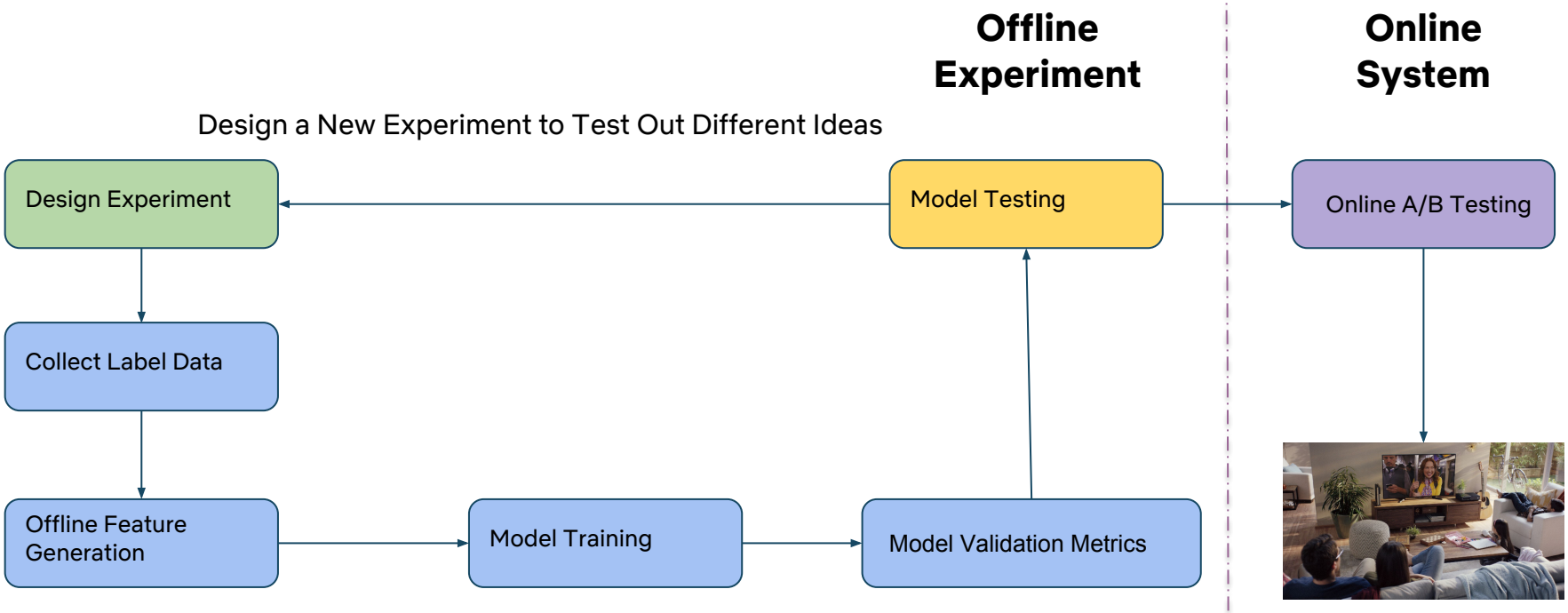
Recommendations at Netflix

- Personalized Homepage for each member
  - **Goal**: Quickly help members find content they'd like to watch
  - **Risk**: Member may lose interest and abandon the service
  - **Challenge**: Recommendations at Scale

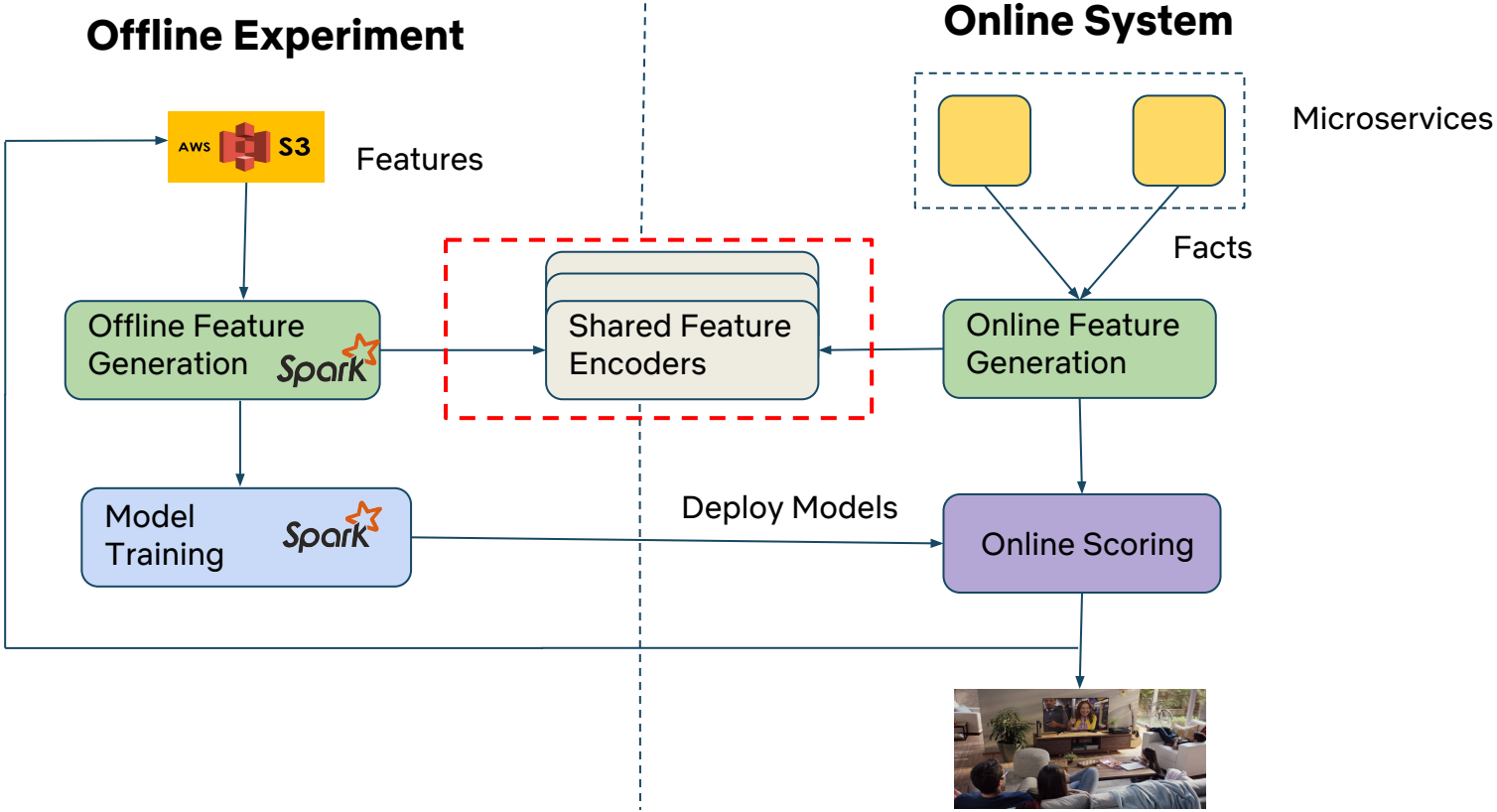
Scale @ Netflix

- 125M+ active members
- 190 countries with unique catalogs
- 450B+ unique events/day
- 700+ Kafka topics

Experimentation Cycle @ Netflix



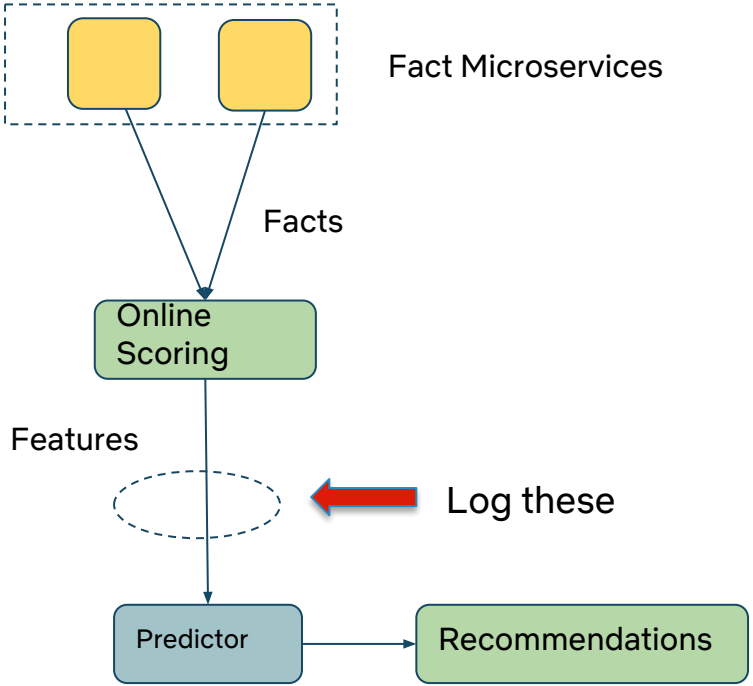
ML Feature Engineering - Architectural View



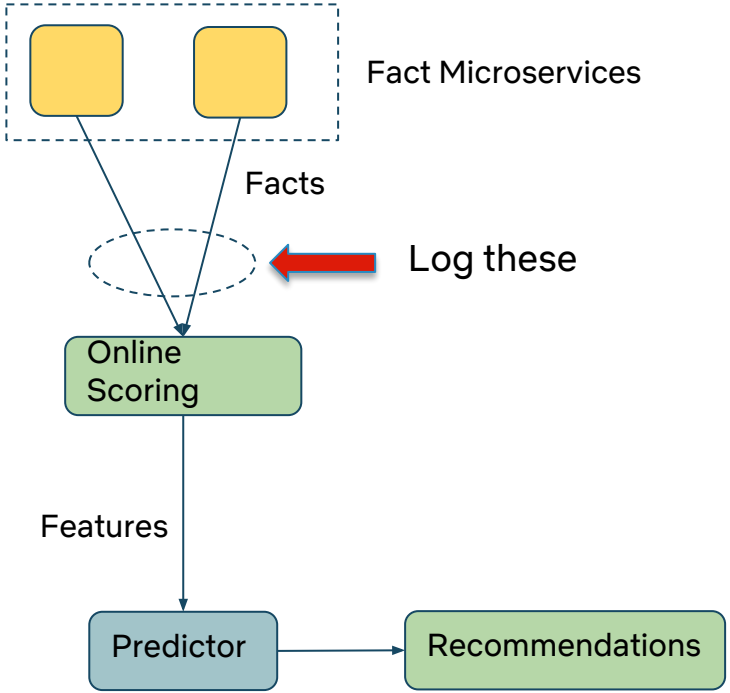
What is a Fact?

- Fact
  - Input data for feature encoders. Used to construct a feature
  - Example: Viewing history of member, my list of a member
- **Historical** Version of a fact
  - Rewindable - State of the world at that time
- **Temporal**
    - Facts are temporal i.e. they change with time
    - Each online scoring service uses the latest value of a fact

Feature Logging

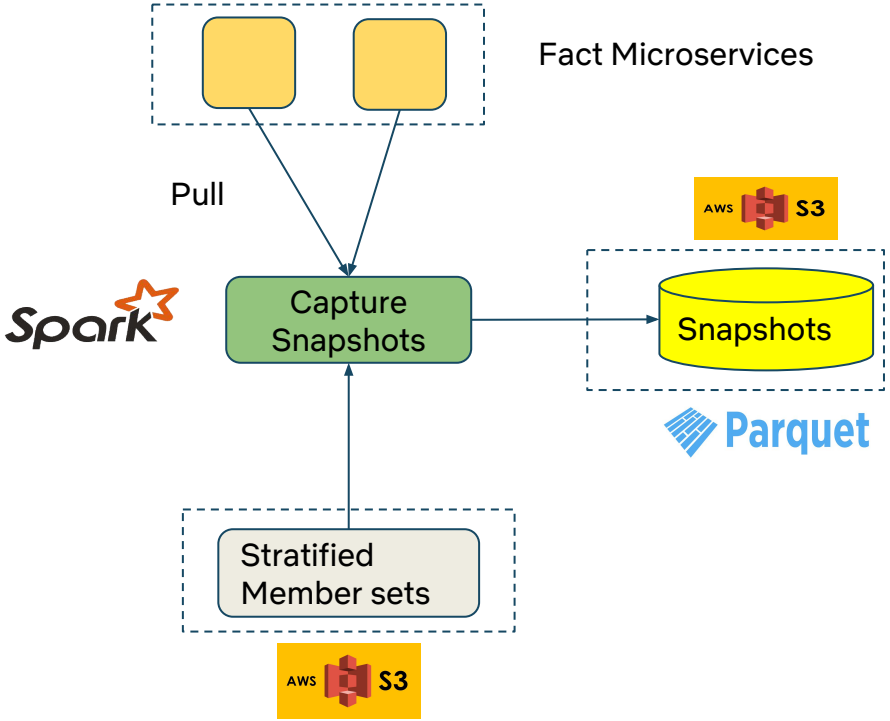


Fact Logging



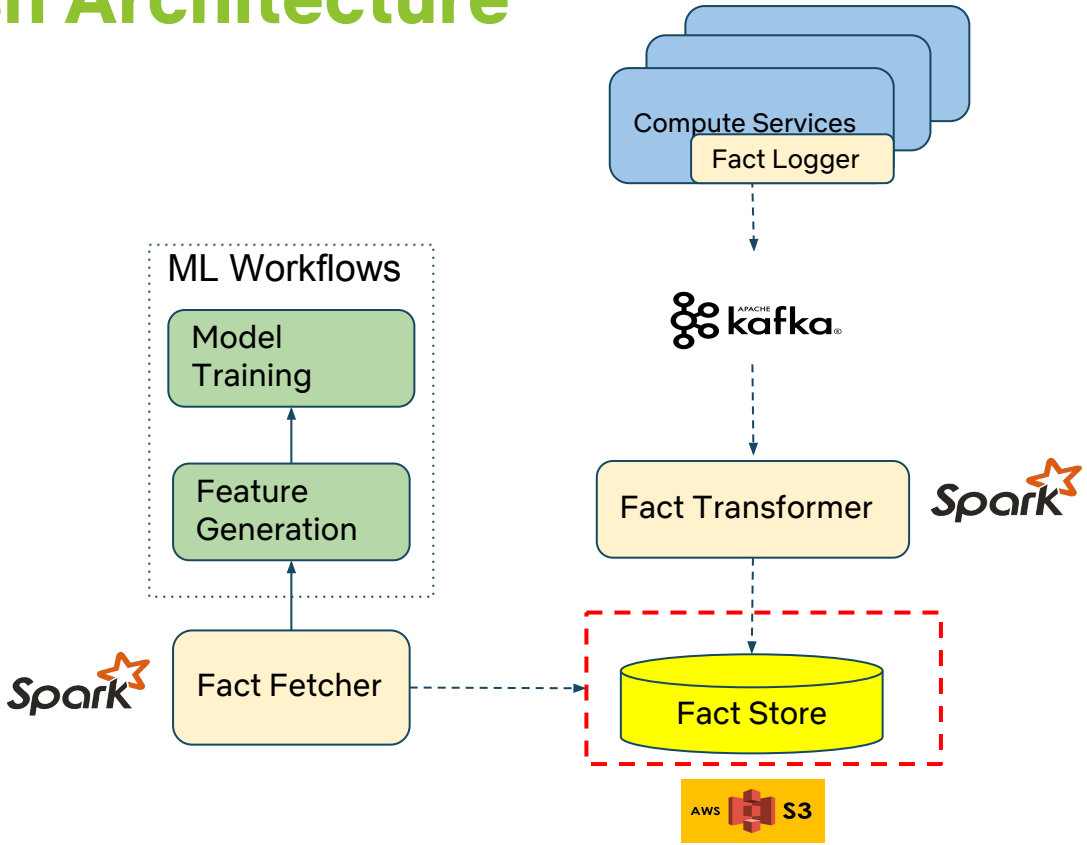
Fact Logging - Pull Architecture

- Daily snapshots of key facts
- Storage
  - S3 & Parquet
- Api to access the data
  - RDD & DataFrames
- **Cons**
  - Lacks temporal accuracy
  - Load on Microservices
  - Missing Experiment specific facts



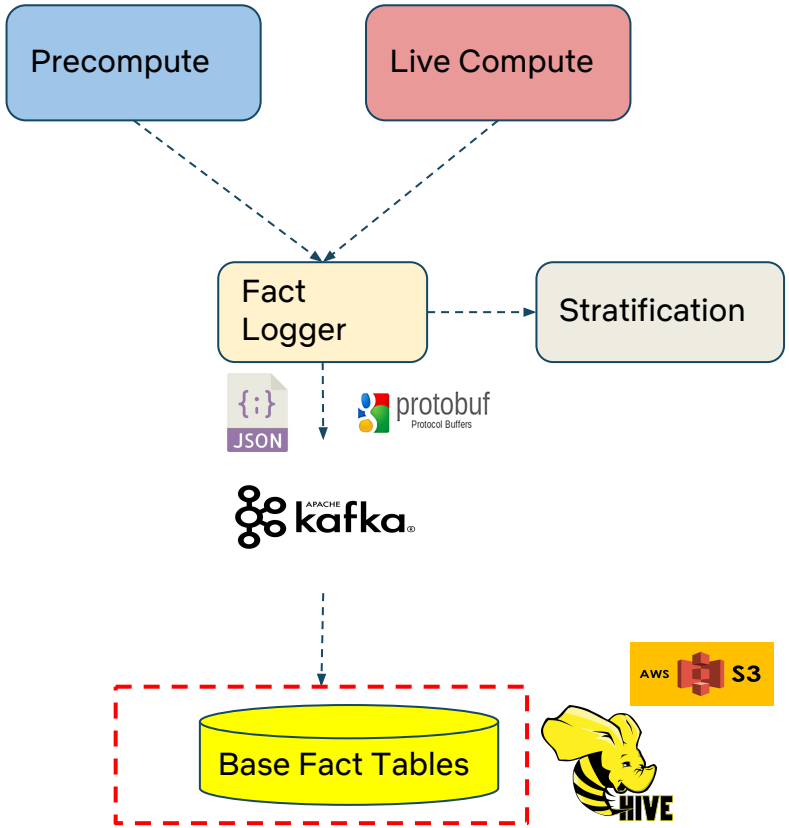
Fact Logging - Push Architecture

- Compute engines themselves control what to log
- Stratification
- Temporal accuracy



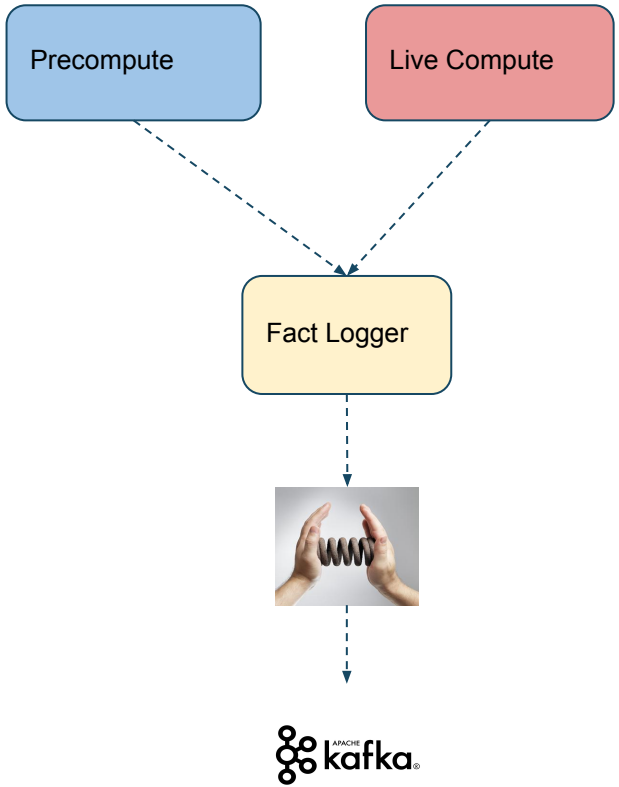
Fact Logger

- Library
- Facts
  - User Related
  - Video Related
  - Computation Specific
- Serialization
- Stratification Service
- Fact Stream
- Storage



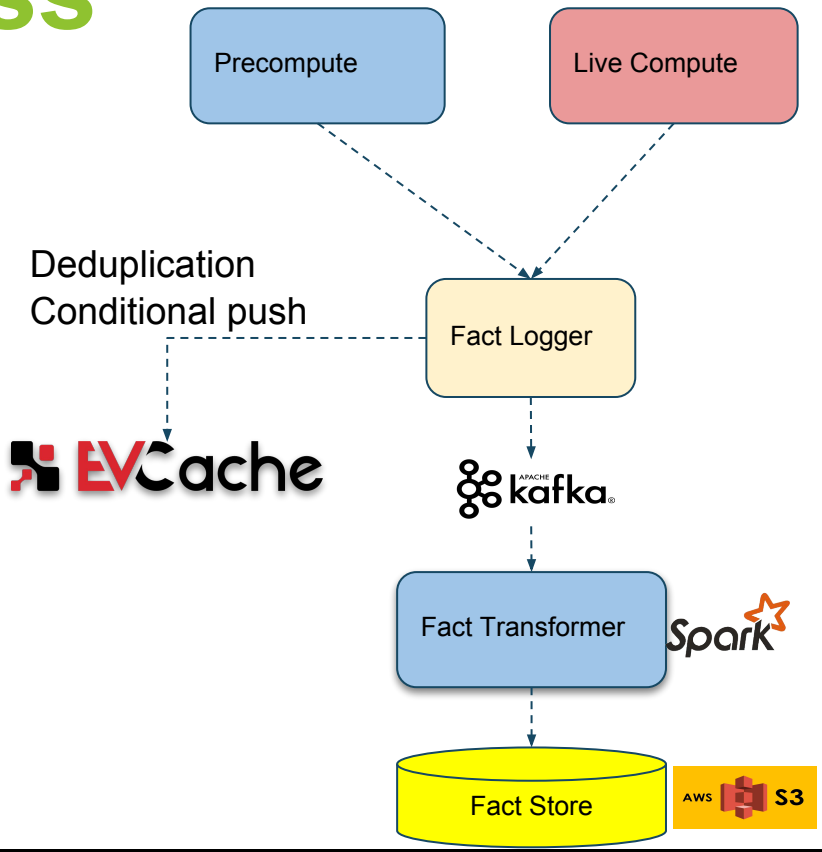
Fact Logging - Scalability

- 5-10x increase in data through Kafka
- SLA Impact; Cost Increase
- Compression - 70% decrease

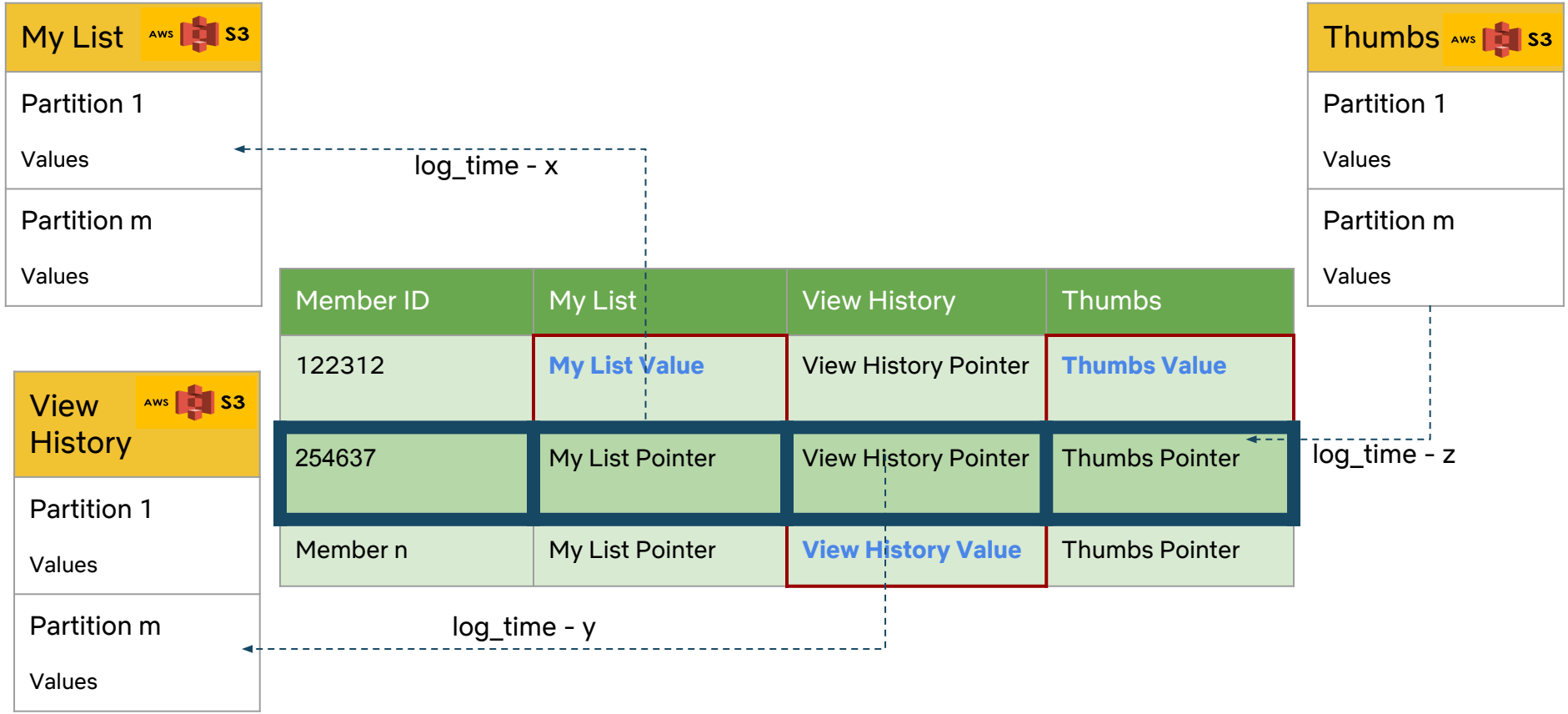


Storage & Access

- Pipeline load
  - Repeated facts
- Aggressive or not
  - Loss threshold
- Spark Job
  - Fact pointers
  - SLA

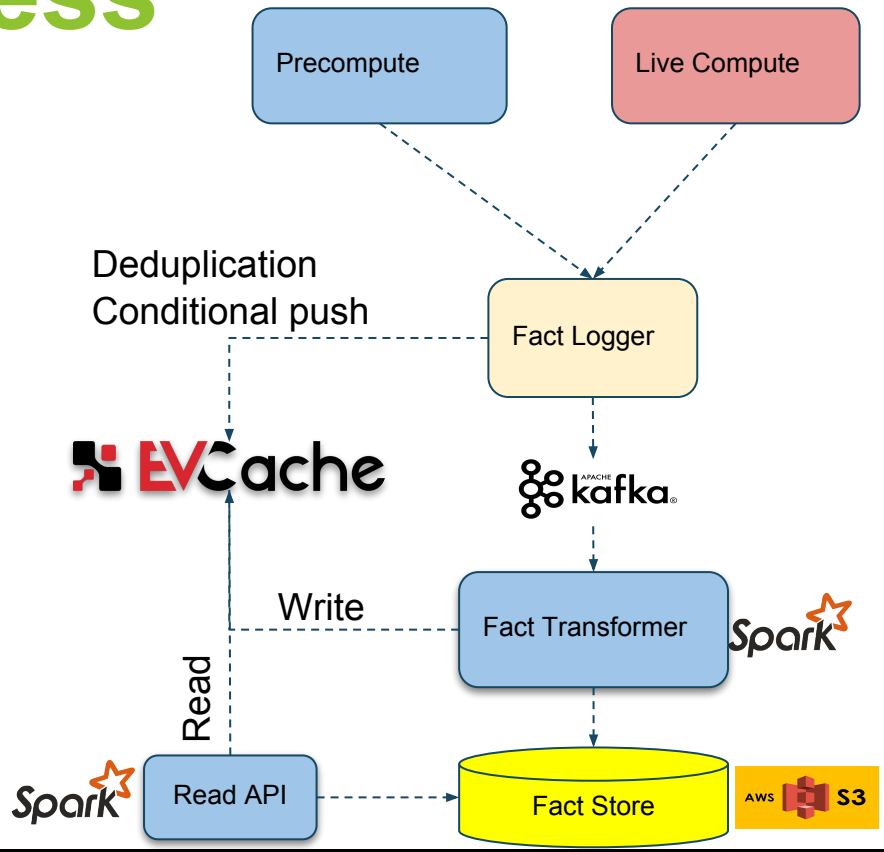


API Lookback



Storage & Access

- Query performance
  - Slow moving facts
- Point query
  - Connector
- Query time reduction
  - Hours to minutes



Performance: Storage

- Partitioning scheme
  - Noisy neighbor

- Storage format
  - Exploratory vs production

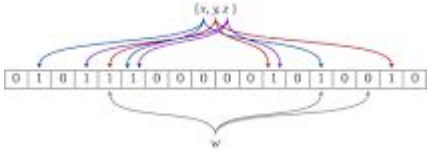
Apache Spark SQL Protocol Buffers

- Fast & Slow lane
  - Lookback limit

EVCache AWS S3

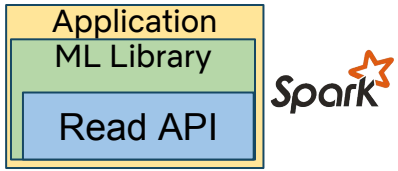
Performance: Spark reads

- Bloom Filters
  - Reduce scan



- Cache Access
  - EVCache, Spectator

- MapPartitions vs UDF
  - Eager vs Lazy
  - SPARK-11438, SPARK-11469, SPARK-20586



Future Work

- Structured with schema evolution
  - Best of both (POJO & Spark SQL), Iceberg
- Streaming vs Batch
  - Multiple lanes, accountability, independent scale
- Duplication
  - Storage vs Runtime cost

Questions?