The Road to a Complete Tweet Index

Michael Busch
@michibusch
michael@twitter.com
buschmi@apache.org

Introduction

More than 2 billion search queries per day.

Introduction

500 million tweets are sent per day.

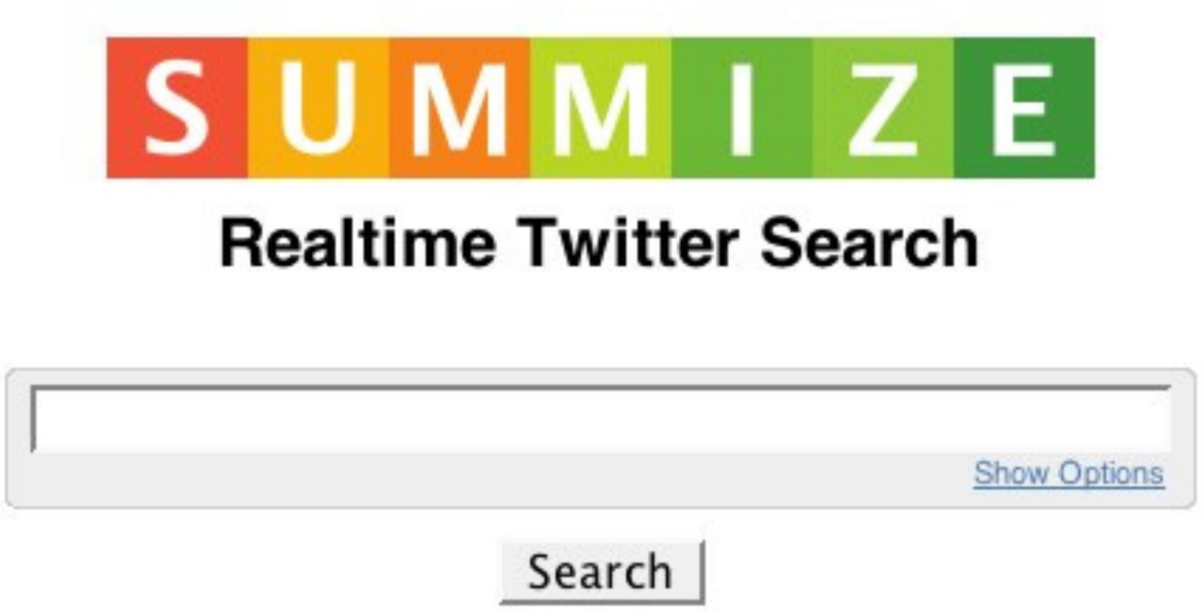
Introduction

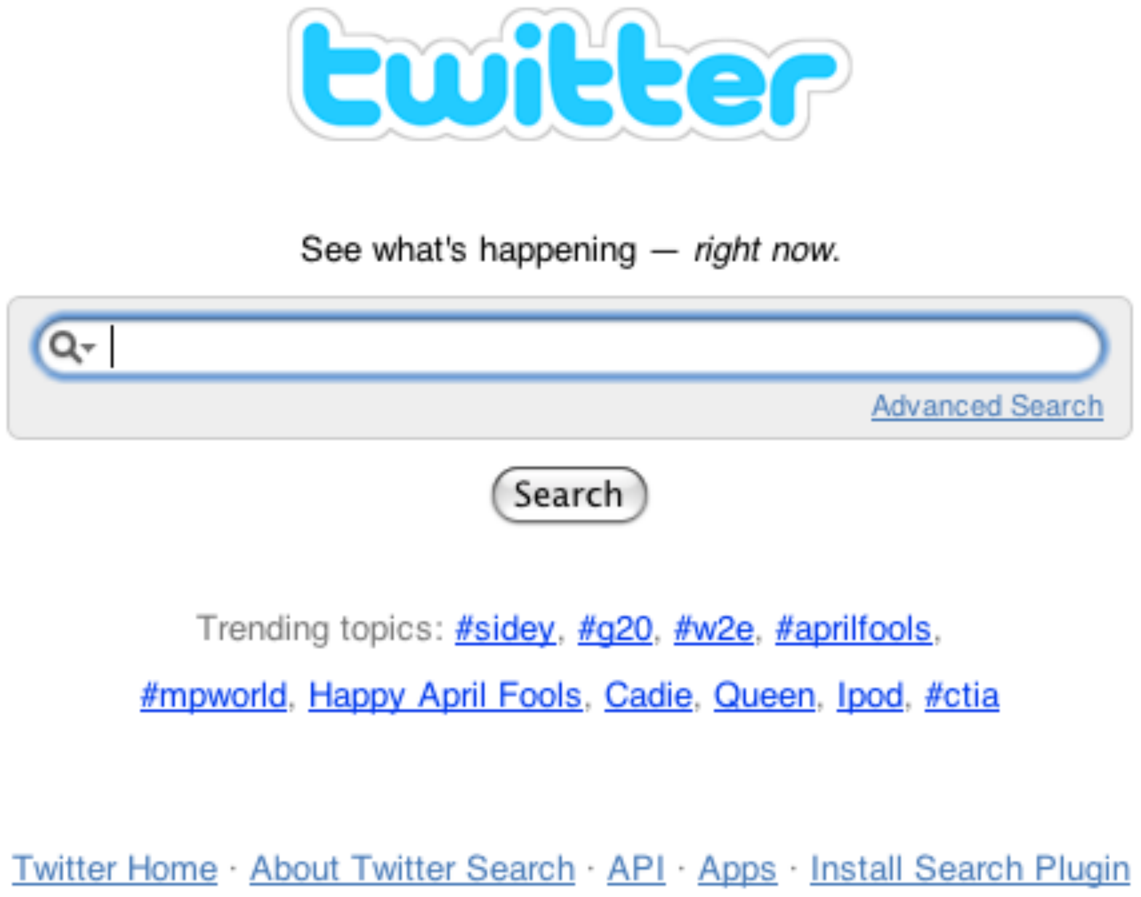
Hundreds of billions of tweets have been sent since company founding in 2006.

2010

2010

Realtime Search powered by Summize technology





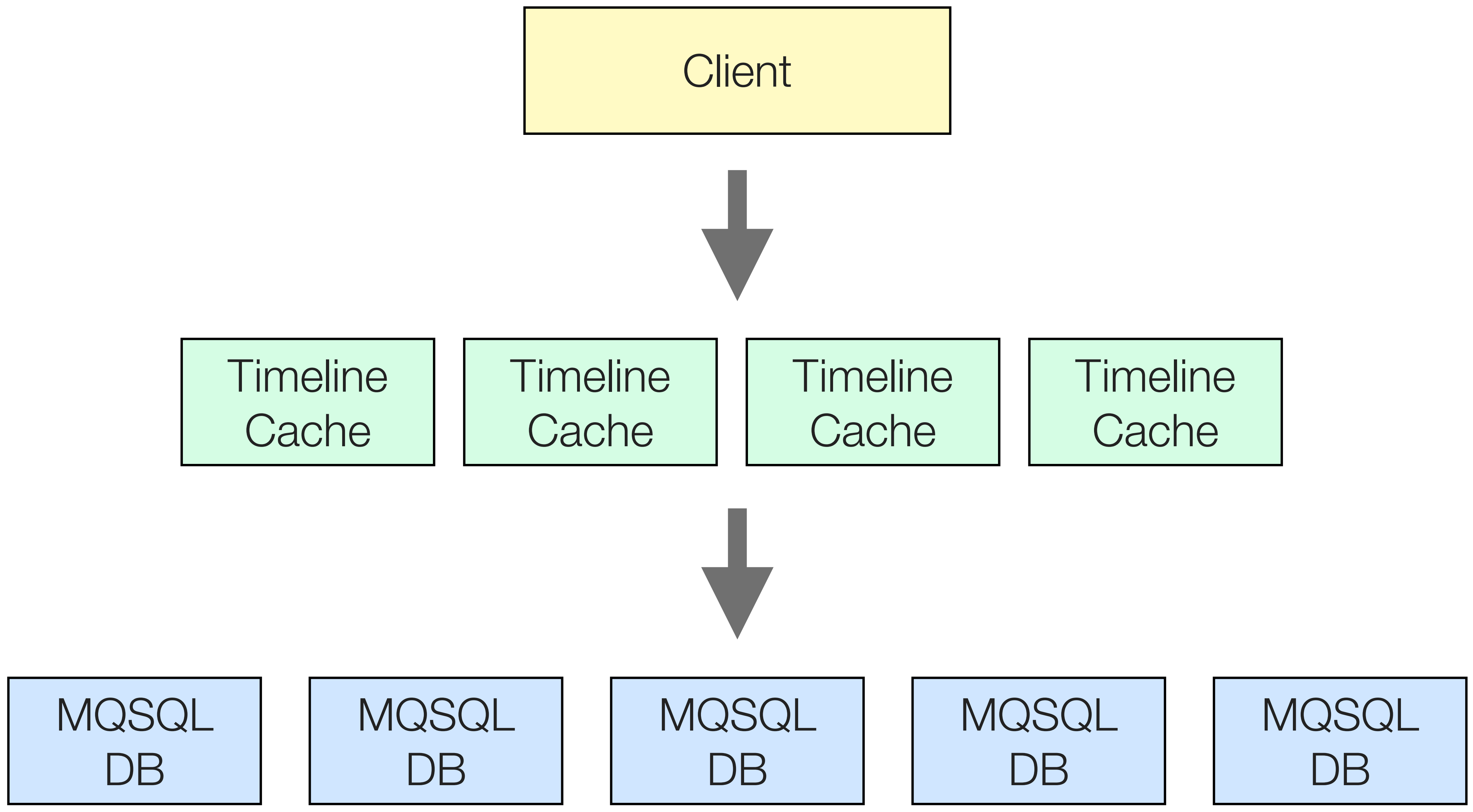
See what's happening — *right now.*

Trending topics: #sidey, #g20, #w2e, #aprilfools, #mpworld, Happy April Fools, Cadie, Queen, Ipod, #ctia

Twitter Home · About Twitter Search · API · Apps · Install Search Plugin

2010

Realtime Search powered by Summize technology



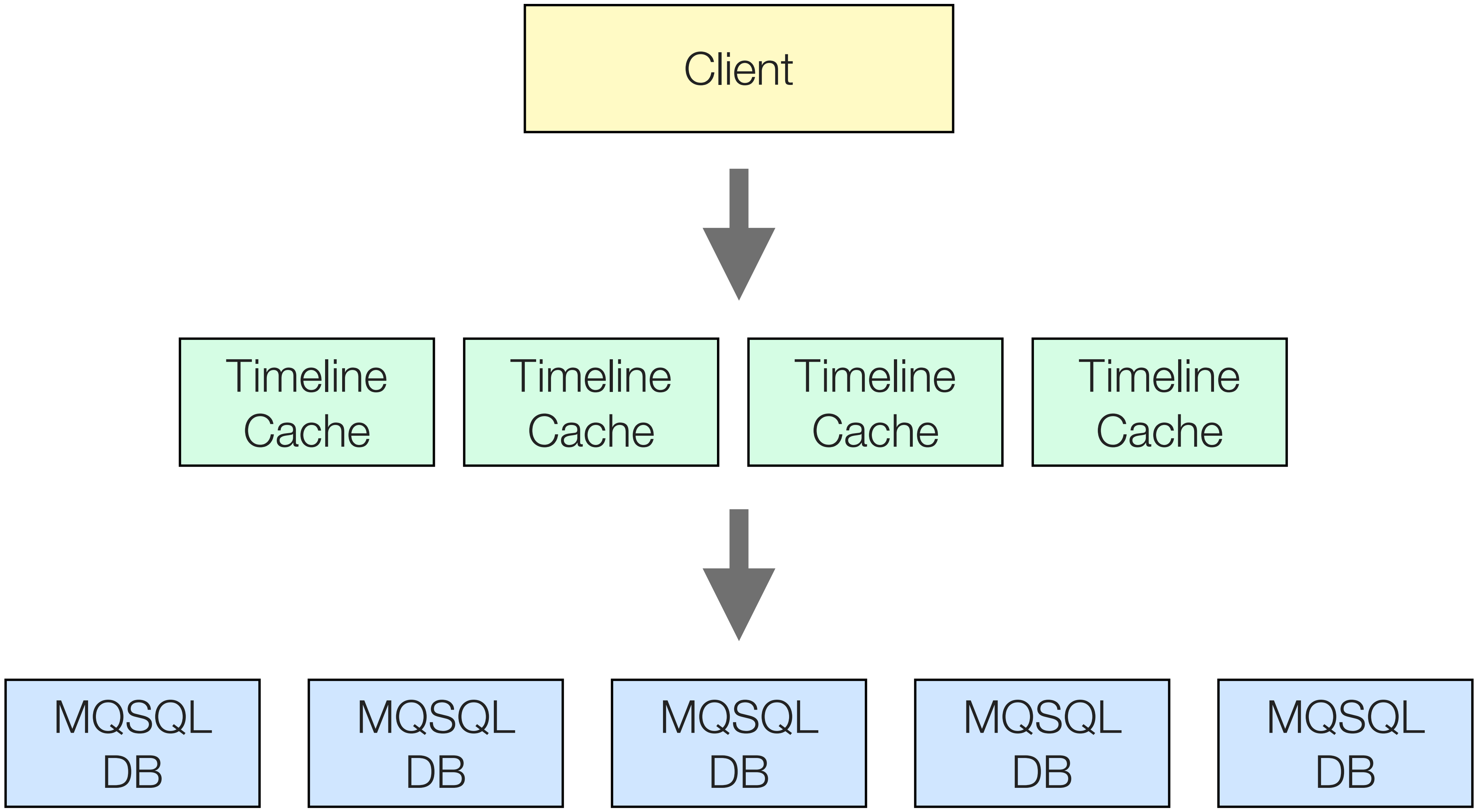
2011

2011

Twitter launches first Lucene-based search engine: **Earlybird.**

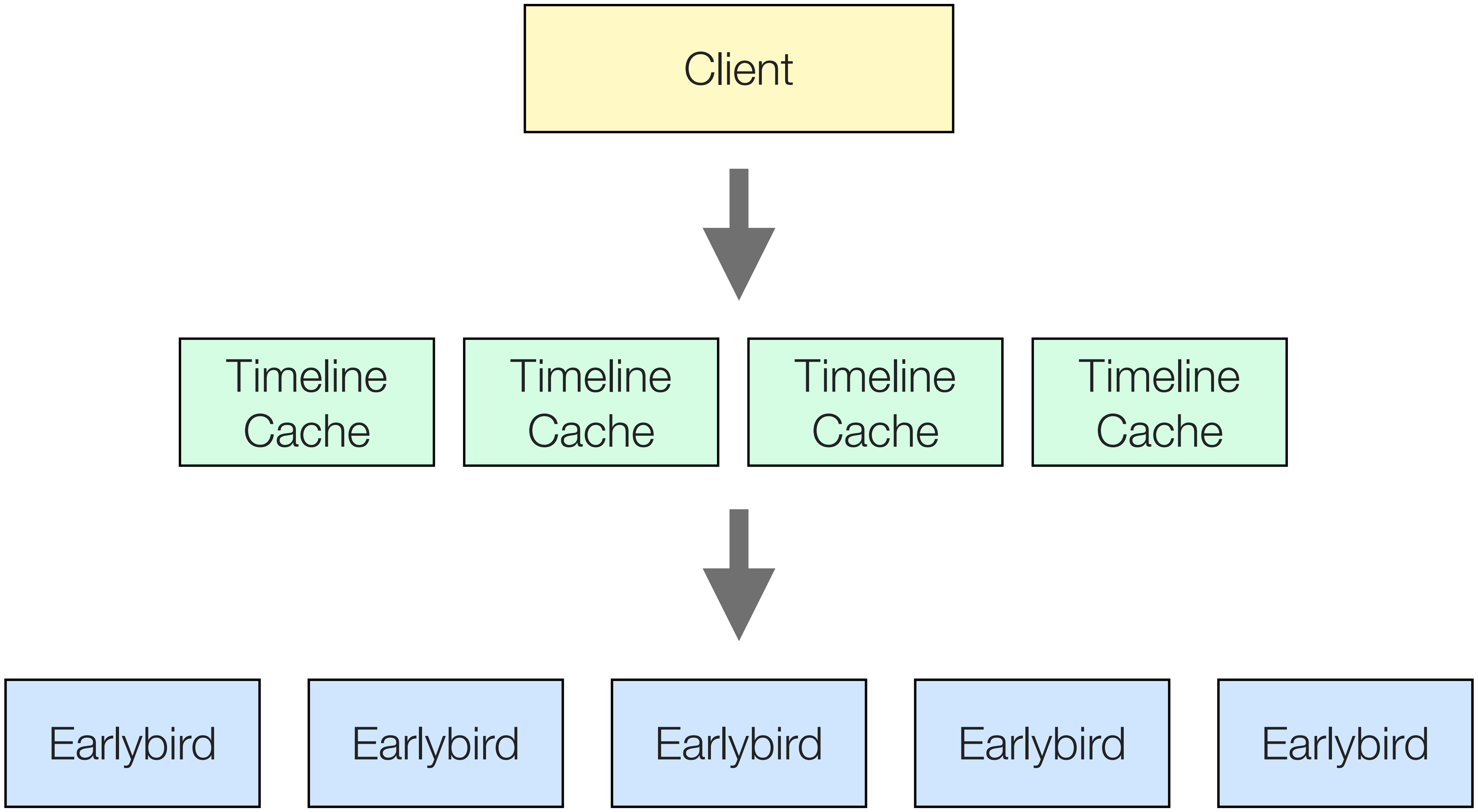
2011

Realtime Search powered by Summize technology



2011

Realtime Search powered by Earlybird

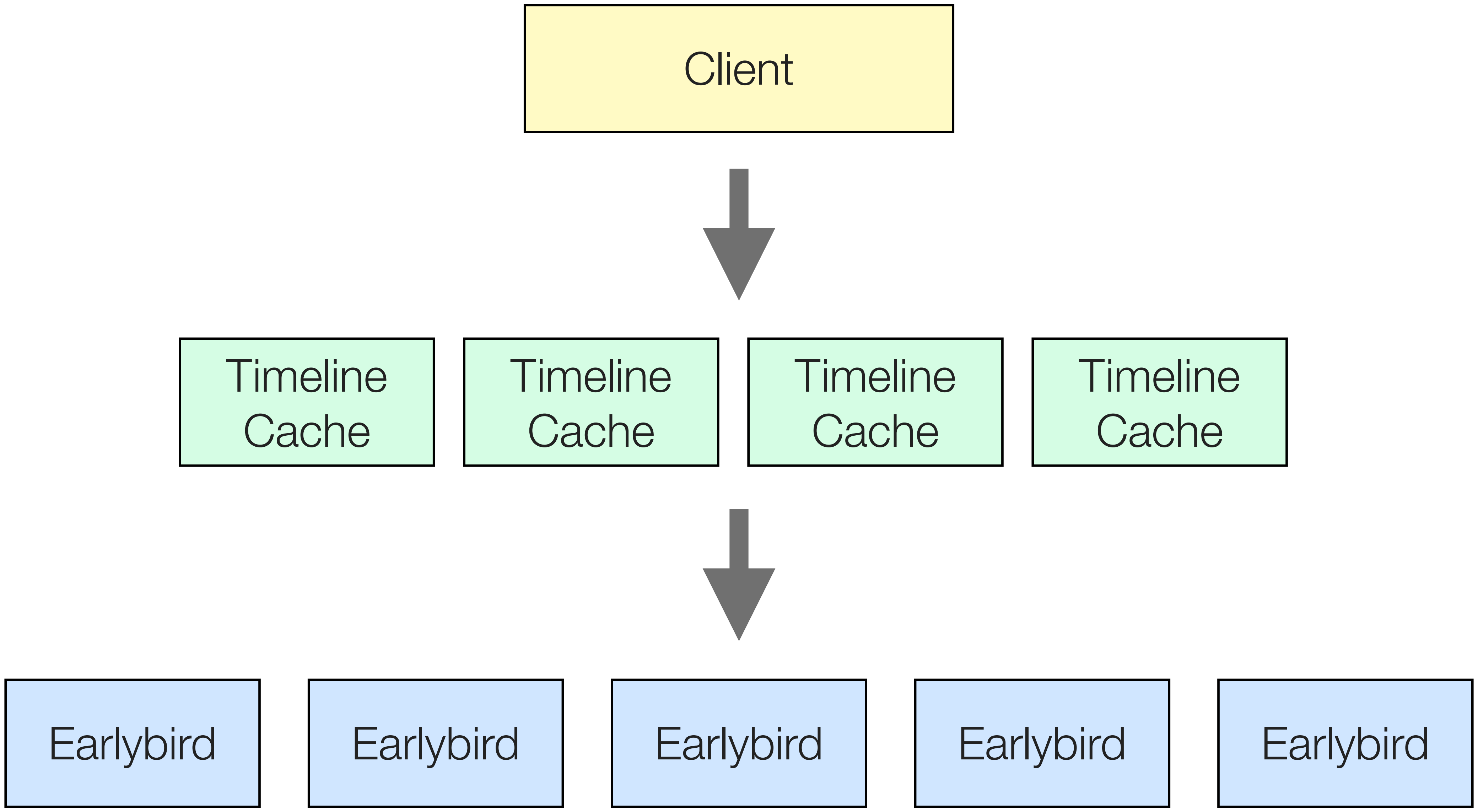


Earlybird

- In-memory index containing several days of most recent tweets
- Highly optimized for realtime search
- Limited to short documents (max. 255 tokens)
- Novel concurrency and memory models
- Concurrently writing and searching an index segment

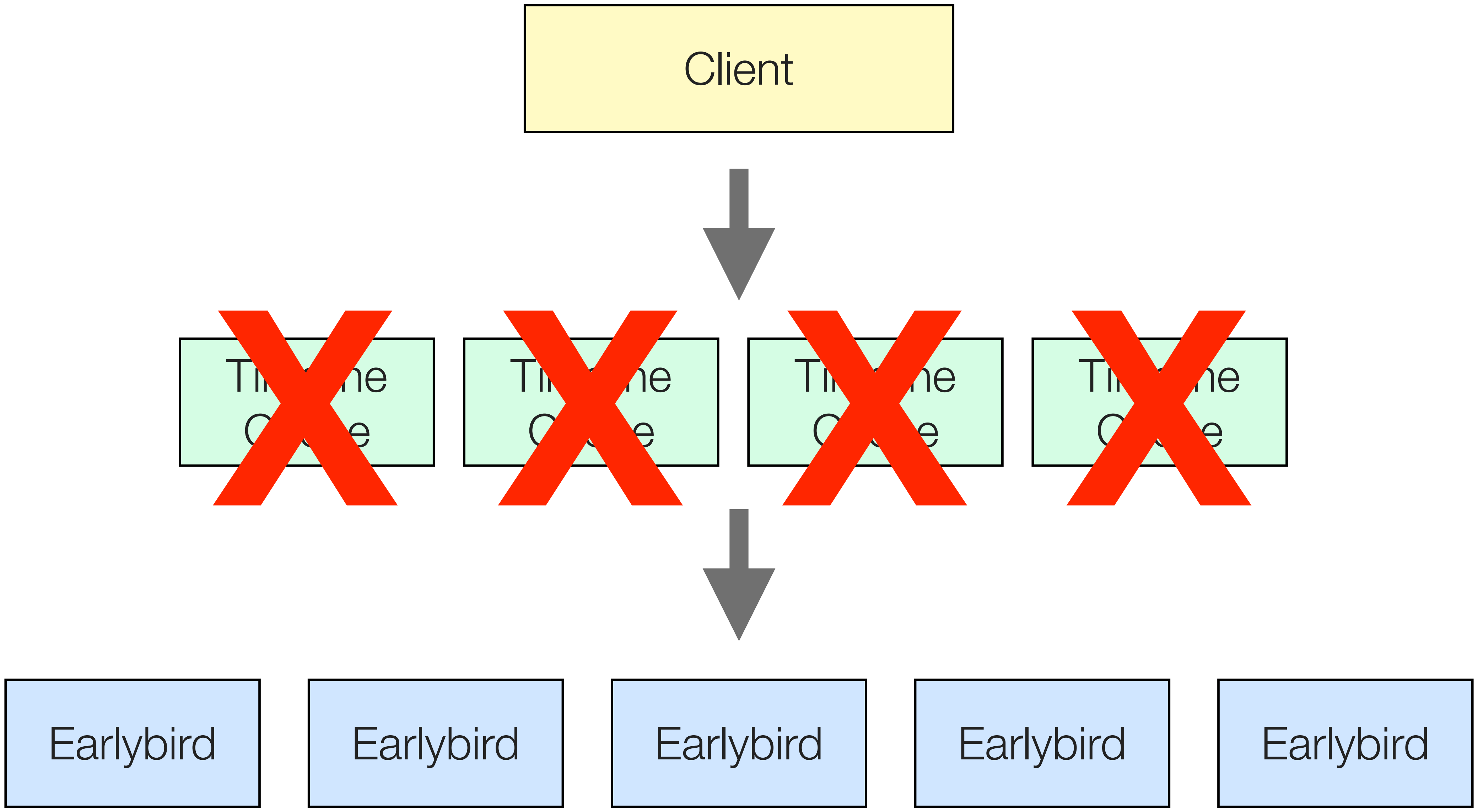
2011

Realtime Search powered by Earlybird



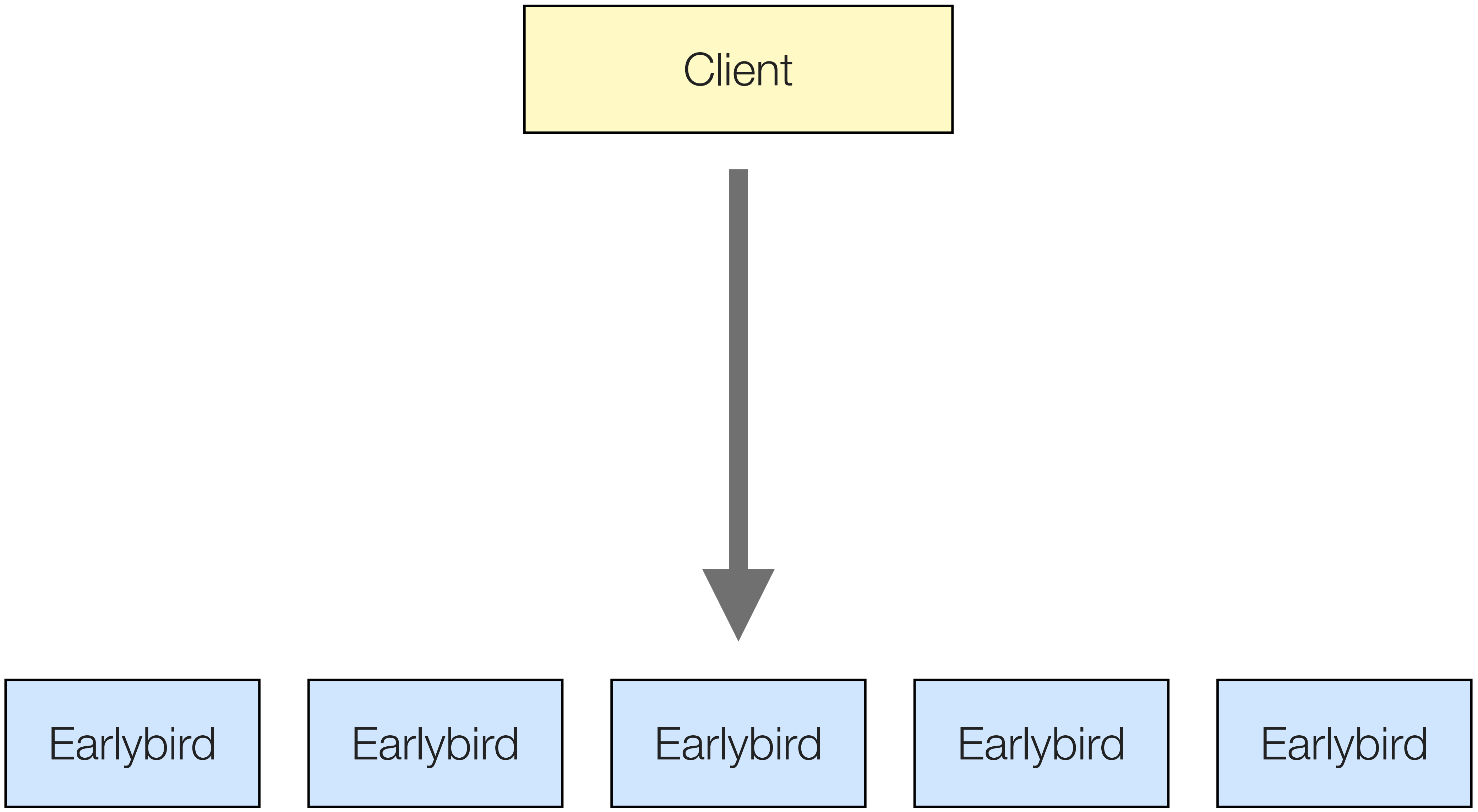
2011

Realtime Search powered by Earlybird



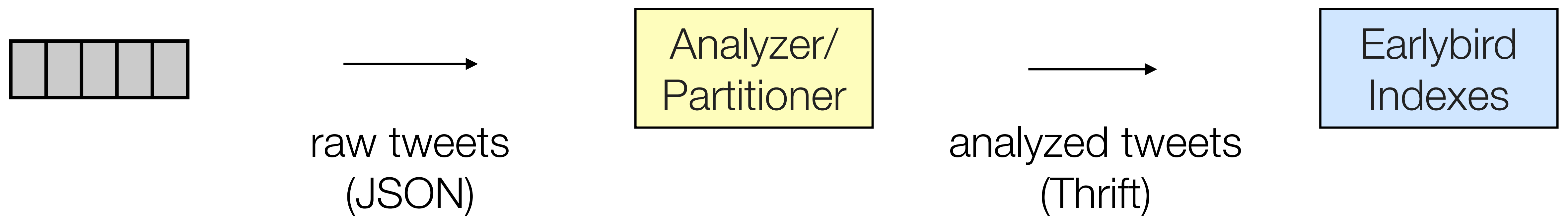
2011

Realtime Search powered by Earlybird

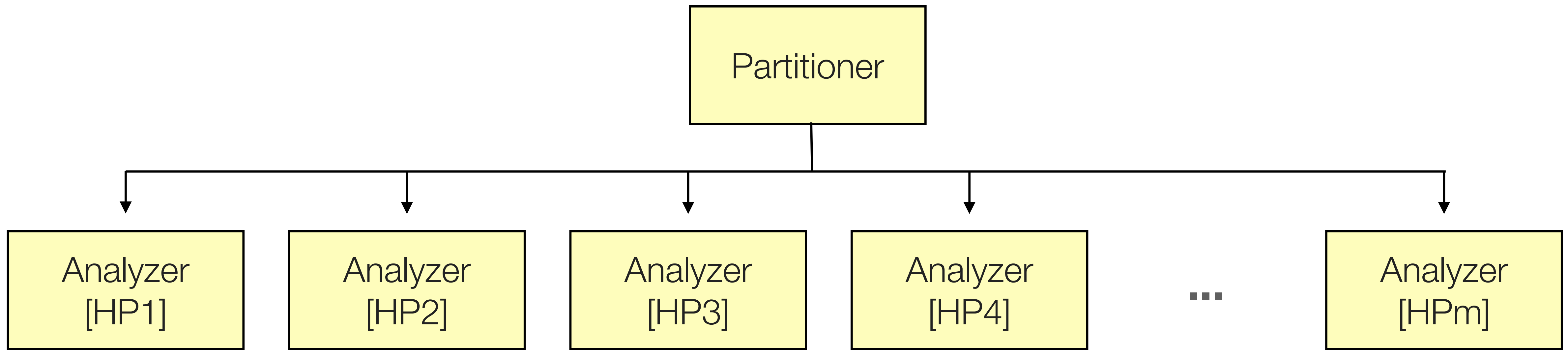


Earlybird

Realtime indexing pipeline

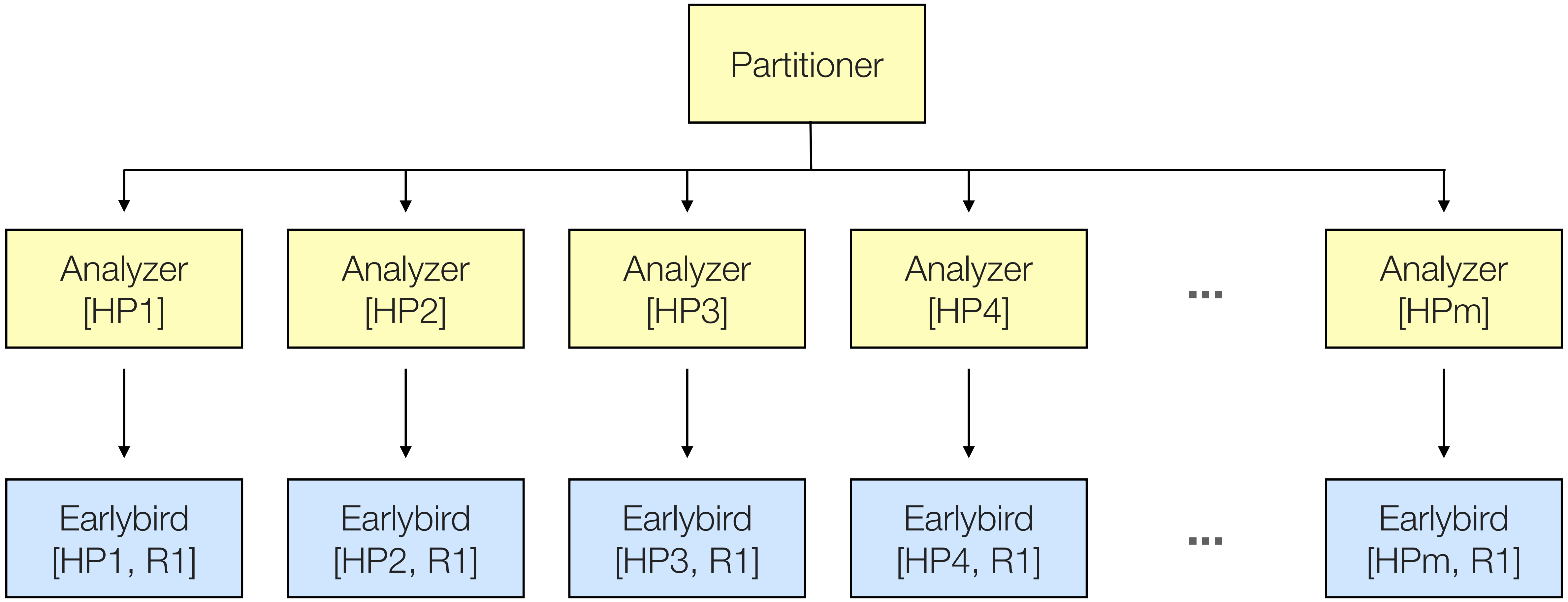


Earlybird



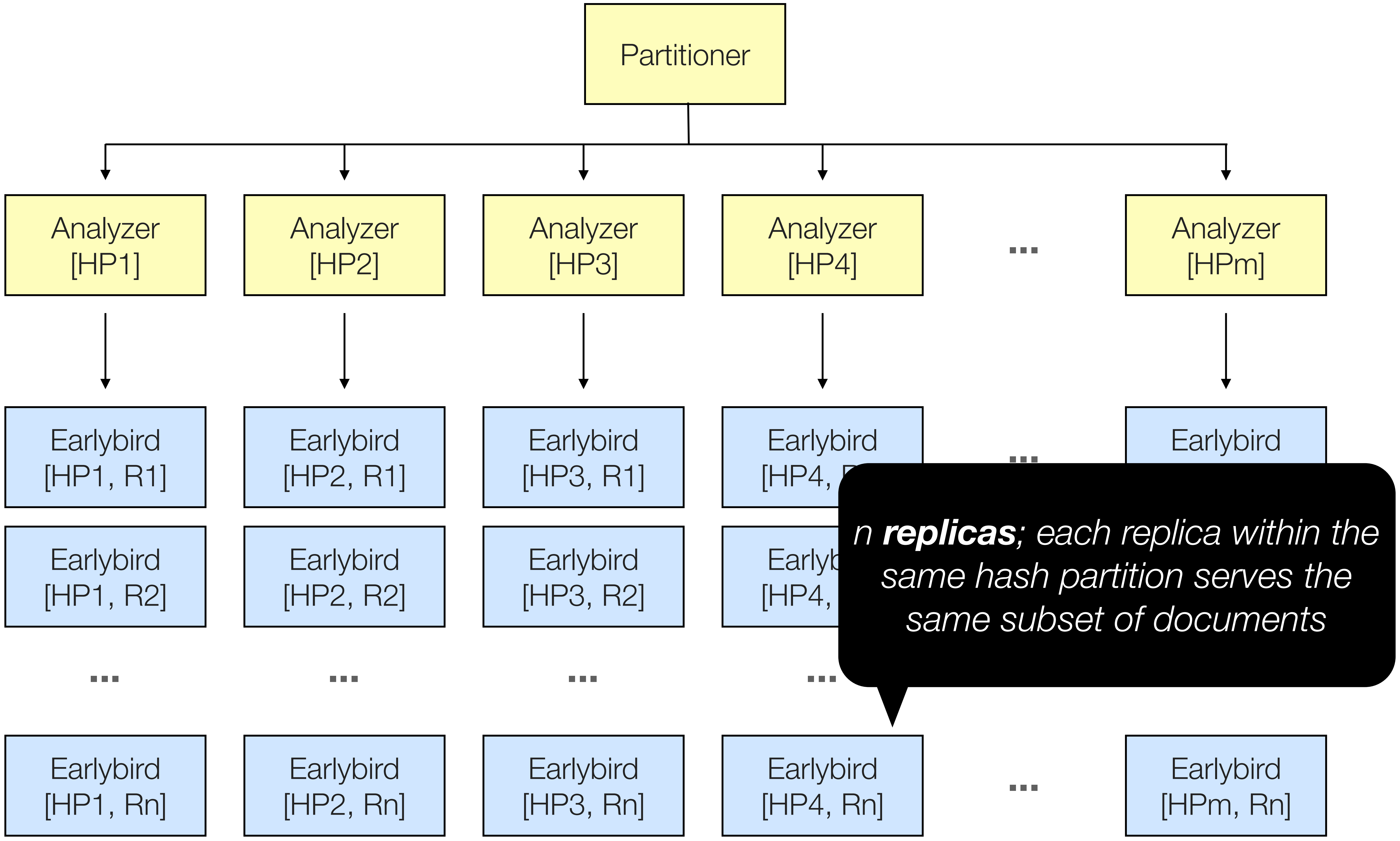
m ***hash partitions****; hash function is simply document uid % m*

Earlybird



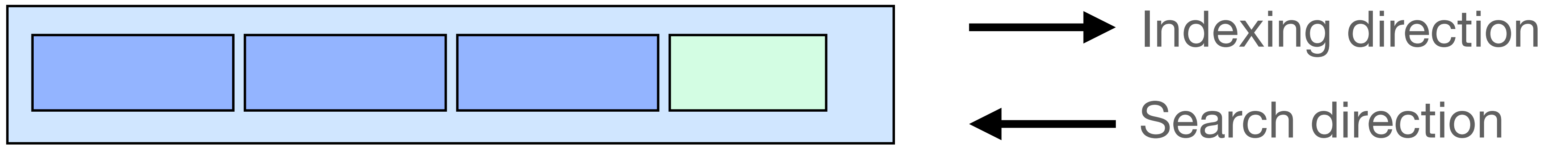
m ***hash partitions***; *hash function is simply document uid % m*

Earlybird


Partitioner
Analyzer [HP1]
Analyzer [HP2]
Analyzer [HP3]
Analyzer [HP4]
...
Analyzer [HPm]
Earlybird [HP1, R1]
Earlybird [HP2, R1]
Earlybird [HP3, R1]
Earlybird [HP4, R1]
...
Earlybird
Earlybird [HP1, R2]
Earlybird [HP2, R2]
Earlybird [HP3, R2]
Earlybird [HP4, R2]
...
Earlybird [HP1, Rn]
Earlybird [HP2, Rn]
Earlybird [HP3, Rn]
Earlybird [HP4, Rn]
Earlybird [HPm, Rn]
n ***replicas***; *each replica within the same hash partition serves the same subset of documents*

Earlybird

Fixed segment sizes

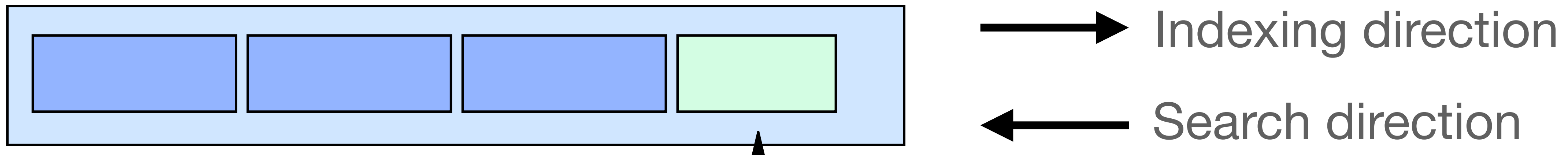


🟦 Complete segment (8M tweets)

🟩 Current incomplete segment

Earlybird

Fixed segment sizes



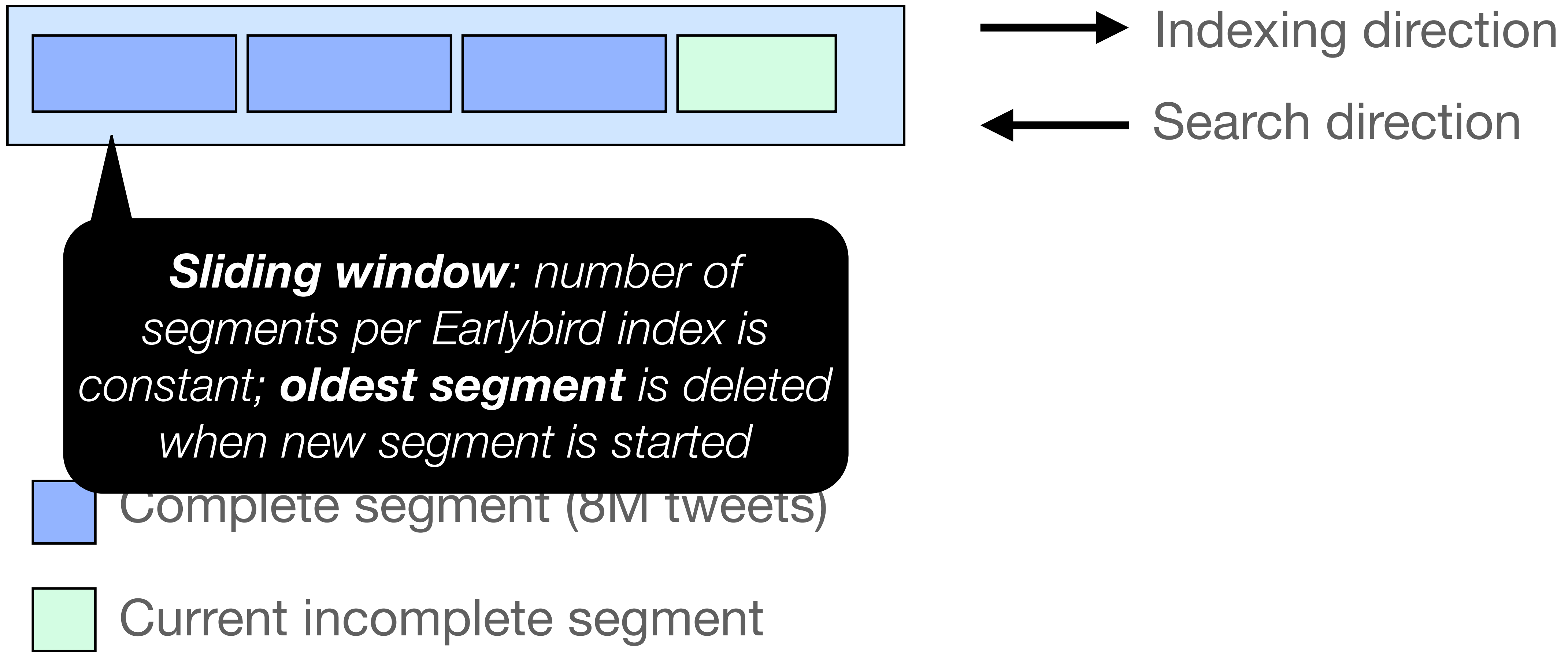
Current segment *can be* ***searched*** *while its* ***being built****; no segment flushing/merging necessary; lock-free concurrency model*

🟦 Complete segment (8M tweets)

🟩 Current incomplete segment

Earlybird

Fixed segment sizes

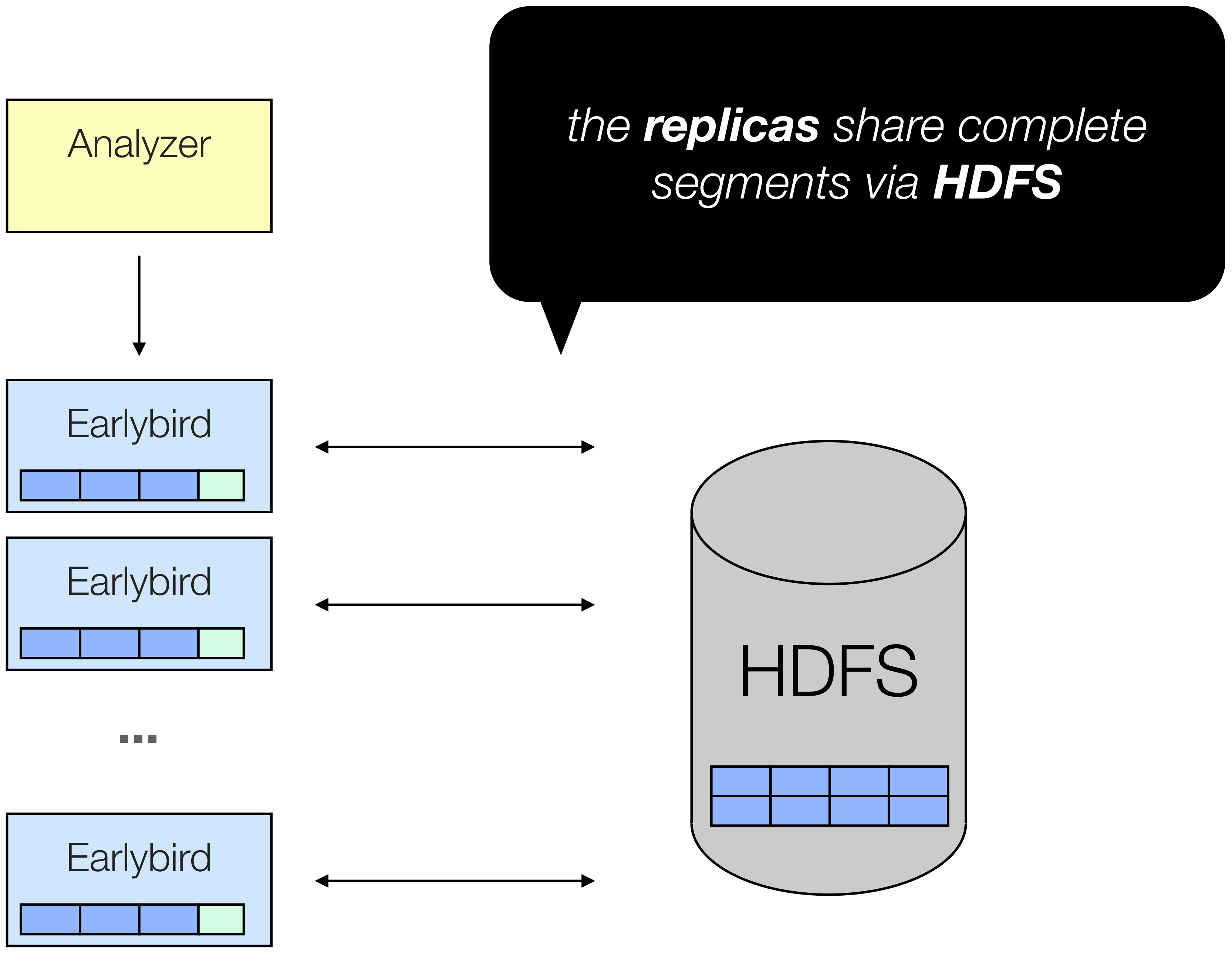

Indexing direction
Search direction
Sliding window: number of segments per Earlybird index is constant; oldest segment is deleted when new segment is started
Complete segment (8M tweets)
Current incomplete segment

Earlybird

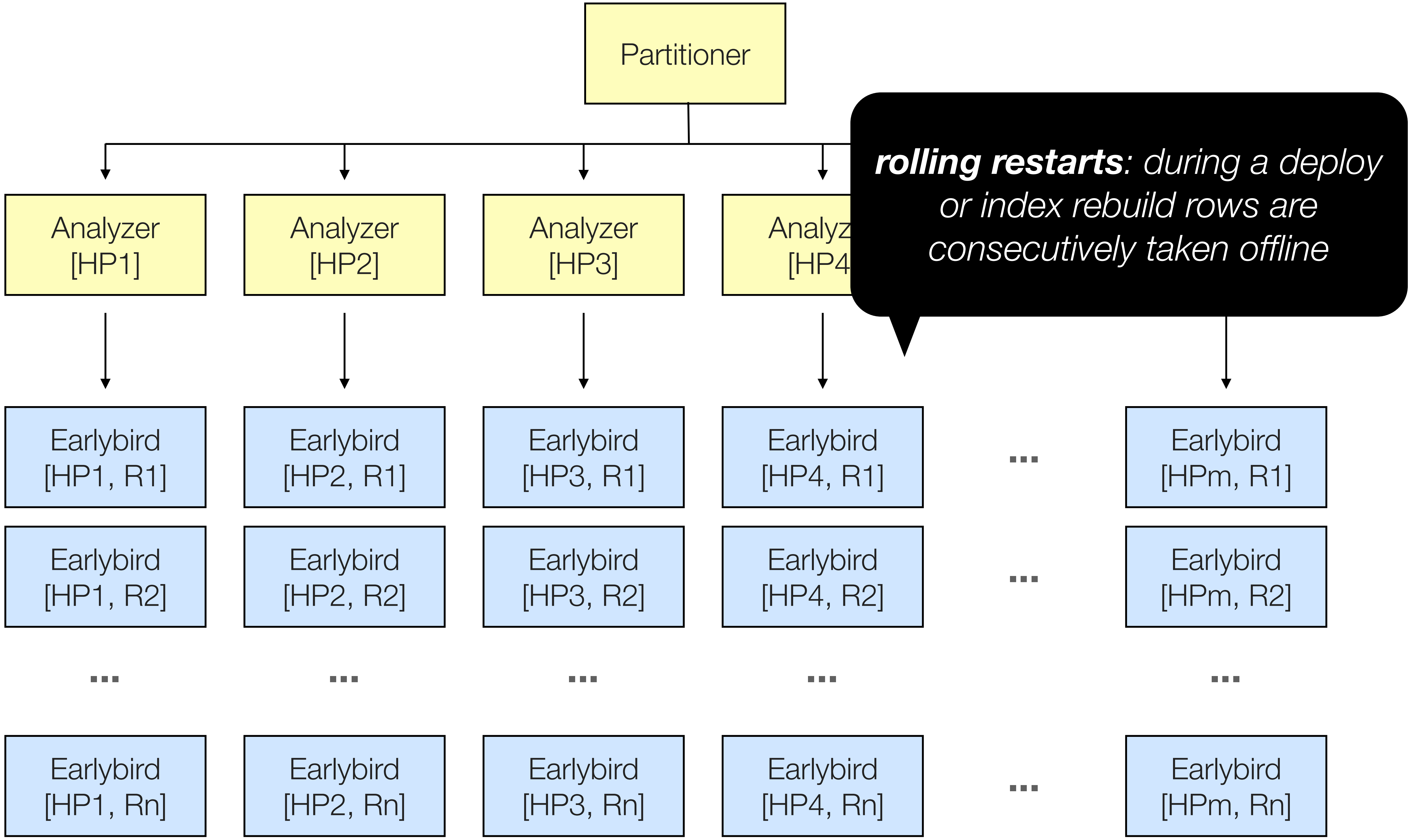
Why use a fixed segment size?

- Earlybird does not need to flush a segment to make it searchable
- No segment merges necessary - consistently high indexing throughput
- Predictable indexing and search performance
- Fixed segment sizes keep replicated Earlybird indexes in sync

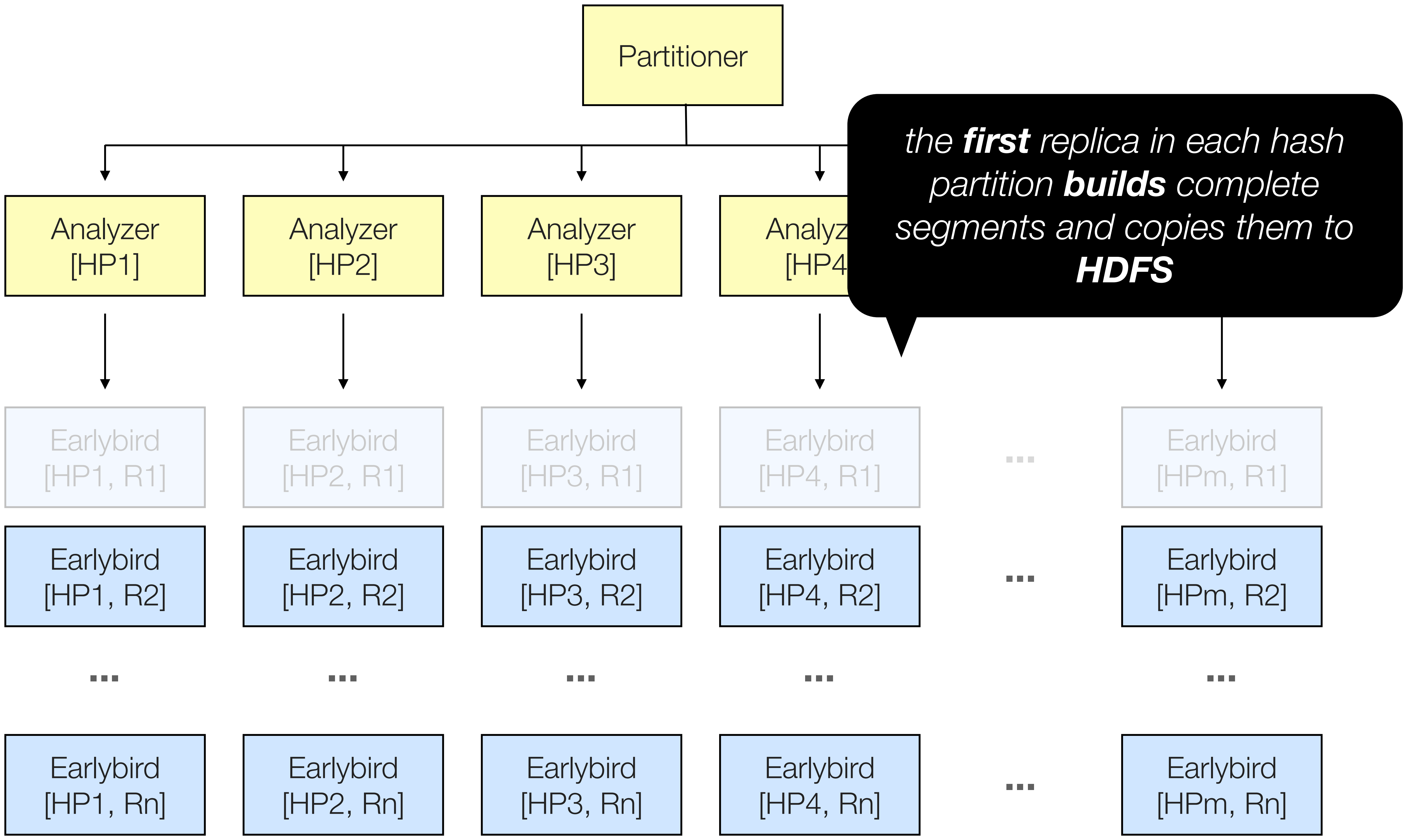
Earlybird


Analyzer
*the **replicas** share complete segments via **HDFS***
Earlybird
Earlybird
...
Earlybird
HDFS

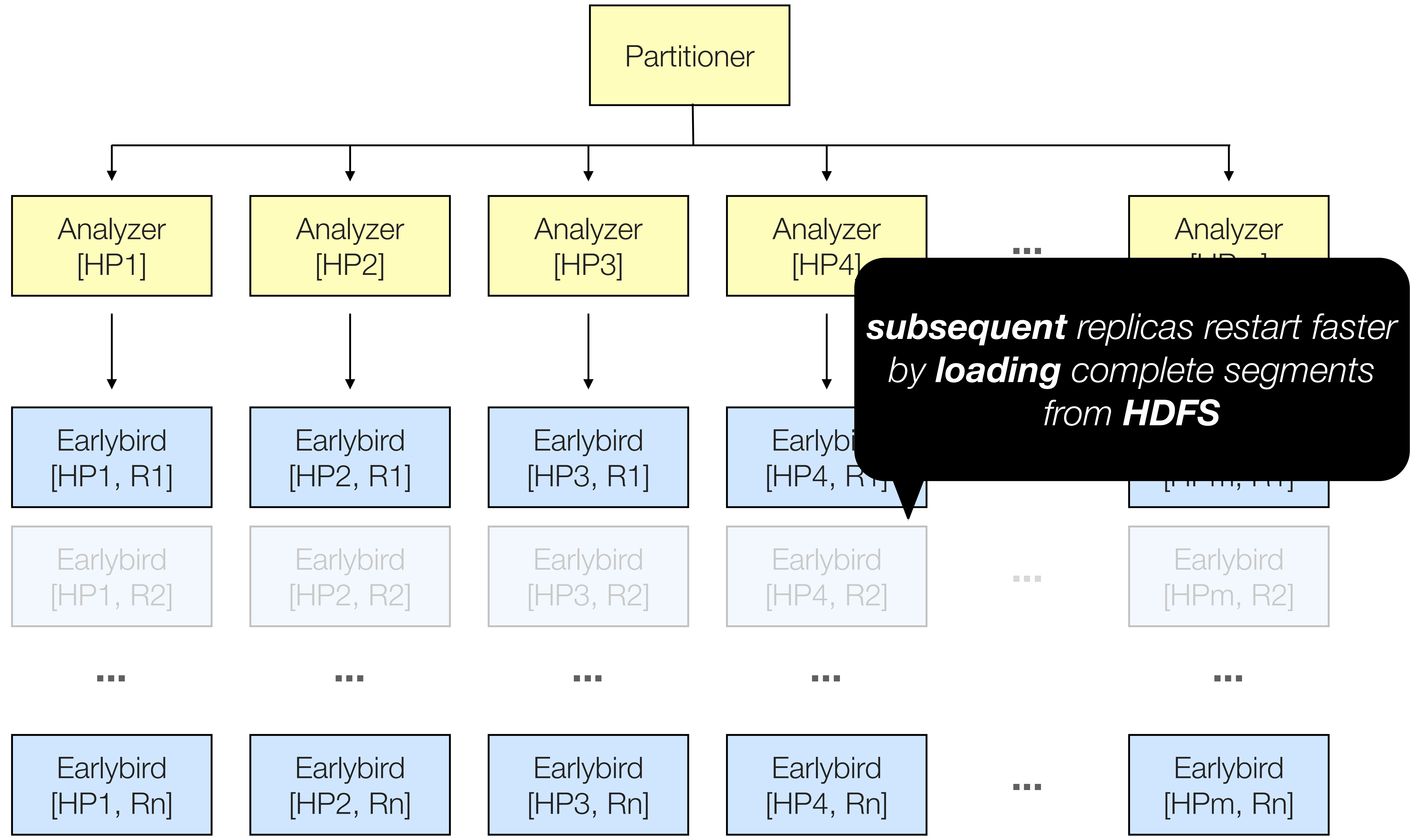
Earlybird


Partitioner
Analyzer [HP1]
Analyzer [HP2]
Analyzer [HP3]
Analyz [HP4
rolling restarts: *during a deploy or index rebuild rows are consecutively taken offline*
Earlybird [HP1, R1]
Earlybird [HP2, R1]
Earlybird [HP3, R1]
Earlybird [HP4, R1]
...
Earlybird [HPm, R1]
Earlybird [HP1, R2]
Earlybird [HP2, R2]
Earlybird [HP3, R2]
Earlybird [HP4, R2]
Earlybird [HPm, R2]
Earlybird [HP1, Rn]
Earlybird [HP2, Rn]
Earlybird [HP3, Rn]
Earlybird [HP4, Rn]
Earlybird [HPm, Rn]

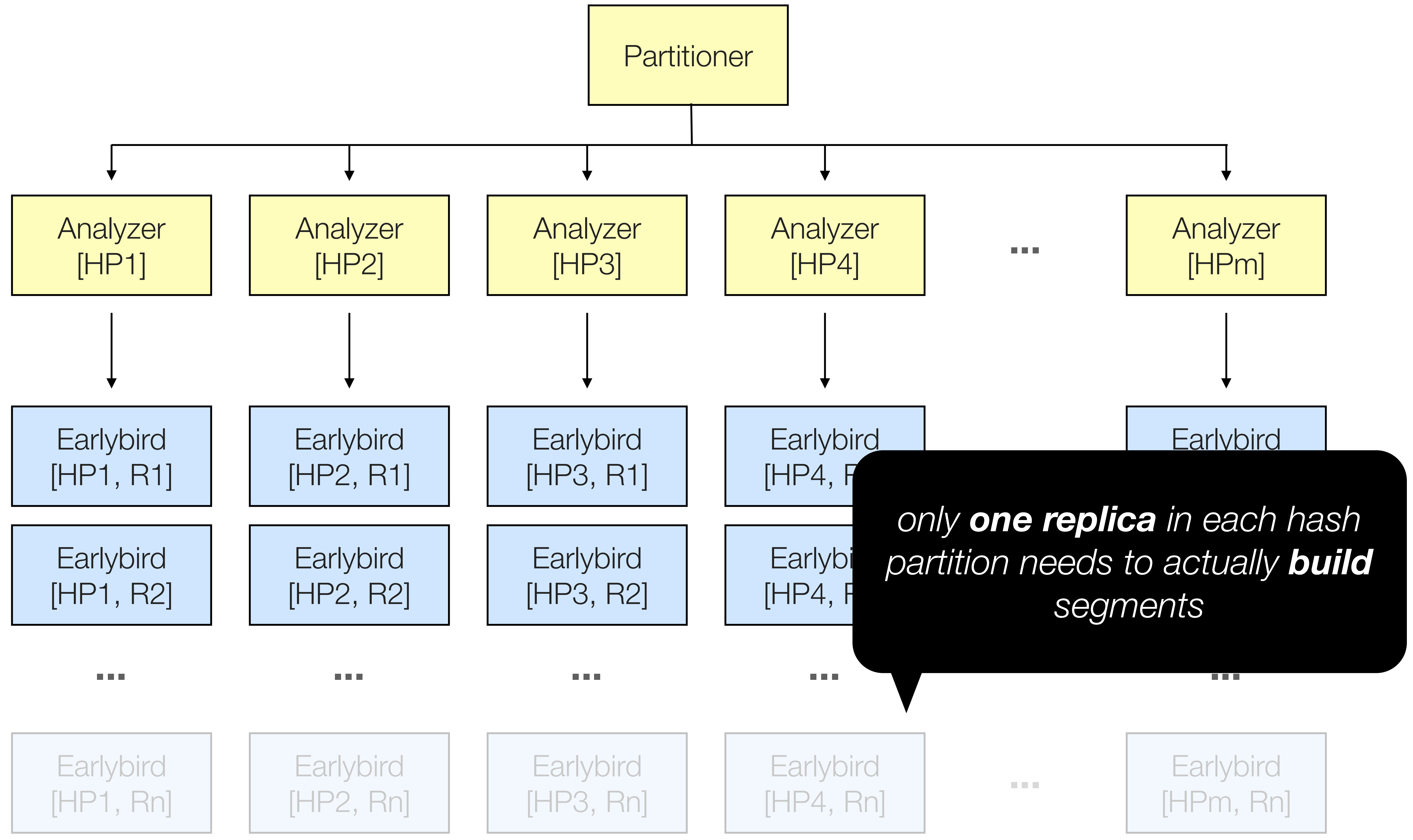
Earlybird


Partitioner
Analyzer [HP1]
Analyzer [HP2]
Analyzer [HP3]
Analyz [HP4
the ***first*** replica in each hash partition ***builds*** complete segments and copies them to ***HDFS***
Earlybird [HP1, R1]
Earlybird [HP2, R1]
Earlybird [HP3, R1]
Earlybird [HP4, R1]
Earlybird [HPm, R1]
Earlybird [HP1, R2]
Earlybird [HP2, R2]
Earlybird [HP3, R2]
Earlybird [HP4, R2]
Earlybird [HPm, R2]
Earlybird [HP1, Rn]
Earlybird [HP2, Rn]
Earlybird [HP3, Rn]
Earlybird [HP4, Rn]
Earlybird [HPm, Rn]
…

Earlybird


Partitioner
Analyzer [HP1]
Analyzer [HP2]
Analyzer [HP3]
Analyzer [HP4]
...
Analyzer
Earlybird [HP1, R1]
Earlybird [HP2, R1]
Earlybird [HP3, R1]
Earlybird [HP4, R1]
Earlybird [HP1, R2]
Earlybird [HP2, R2]
Earlybird [HP3, R2]
Earlybird [HP4, R2]
Earlybird [HPm, R2]
Earlybird [HP1, Rn]
Earlybird [HP2, Rn]
Earlybird [HP3, Rn]
Earlybird [HP4, Rn]
Earlybird [HPm, Rn]
subsequent** replicas restart faster by **loading** complete segments from **HDFS

Earlybird



only ***one replica*** *in each hash partition needs to actually* ***build*** *segments*

Earlybird

Why copy segments to HDFS?

- Rolling restarts are performed to deploy to Earlybird clusters
- A full cluster restart takes a multiple of the time it takes to restart a single hash partition
- New machines can quickly bootstrap from HDFS

2012

2012

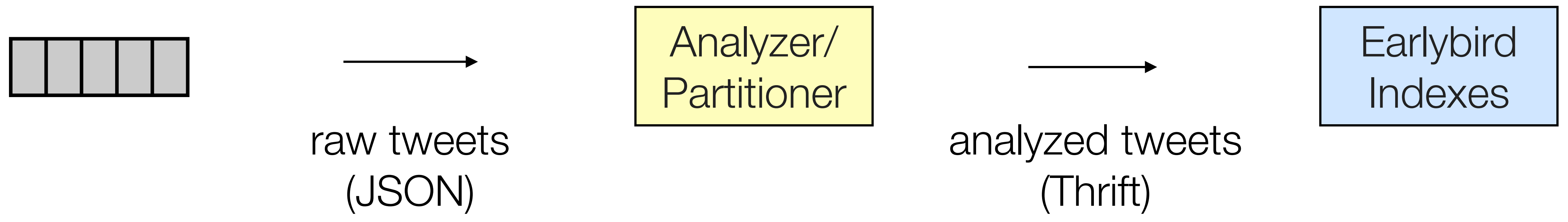
Twitter launches **in-memory historical index.**

In-Memory historical index

- In-memory index containing approx. 2 billion top Tweets of **all time**
- Best Tweets per language
- “Best” determined by relevance function
- Inverted index format identical to realtime Earlybird
- New offline ingestion pipeline

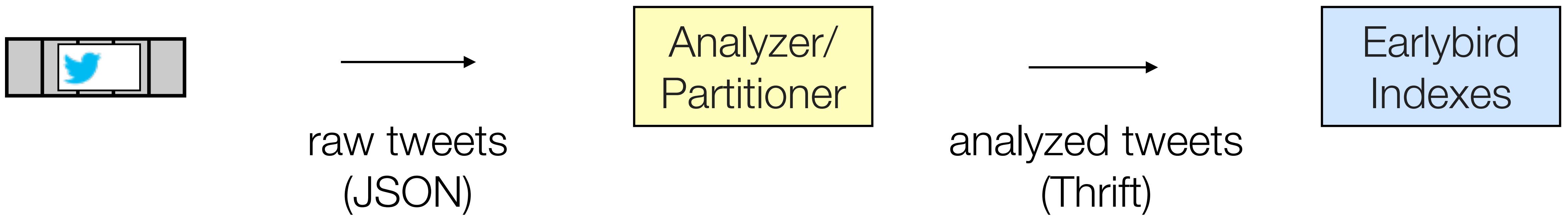
Ingestion pipeline

Online pipeline



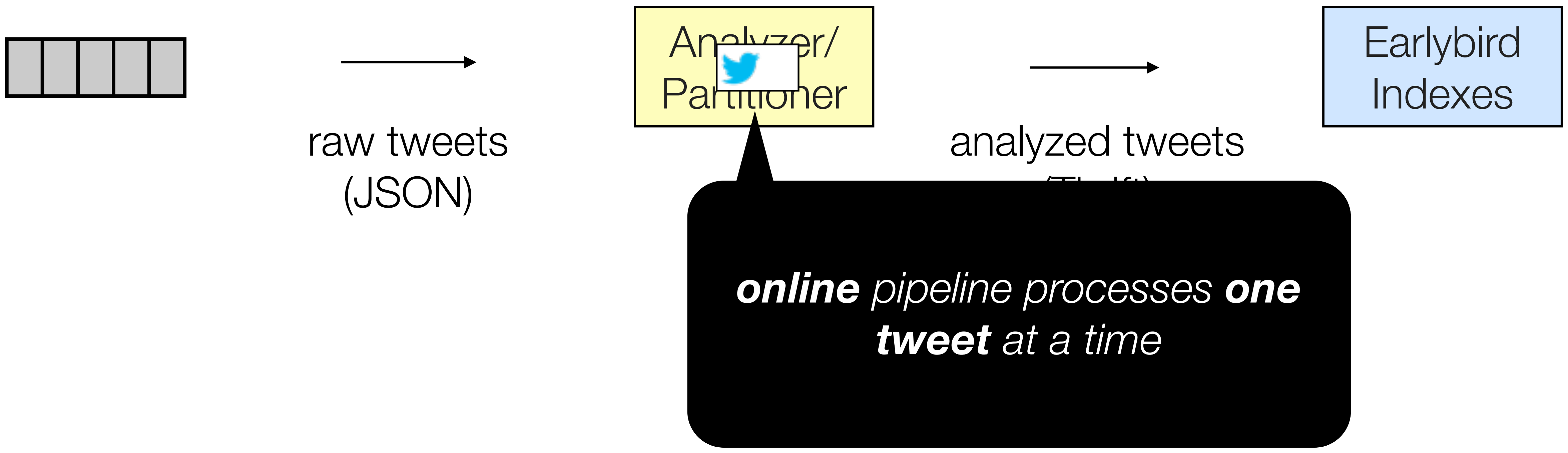
Ingestion pipeline

Online pipeline



Ingestion pipeline

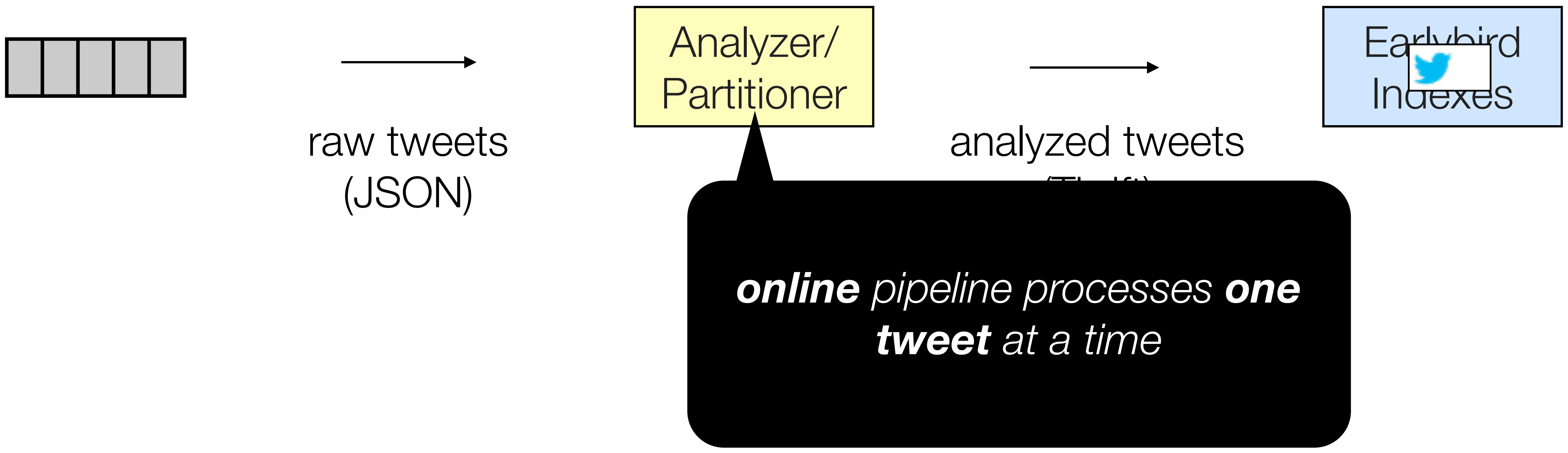
Online pipeline



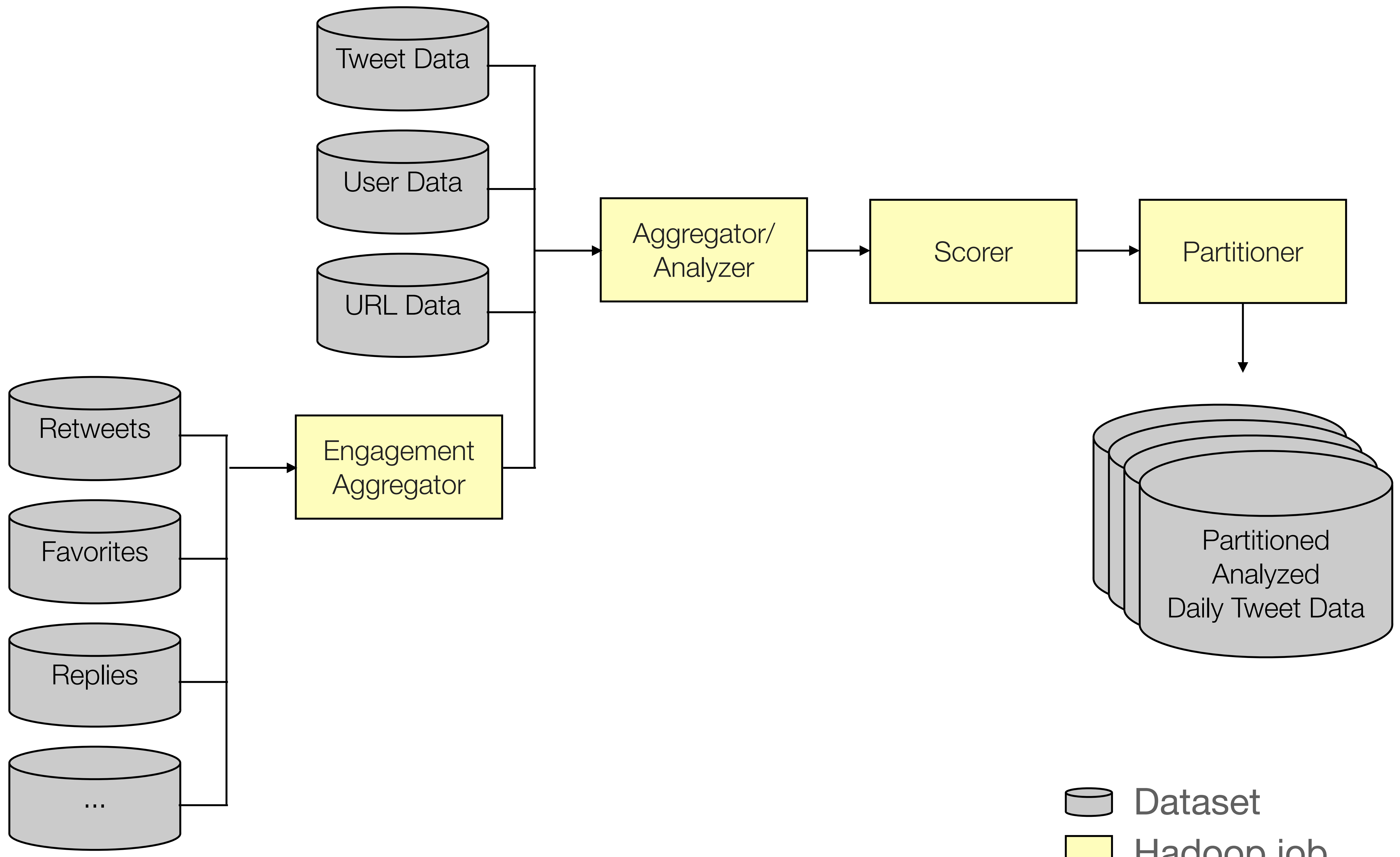
online *pipeline processes* ***one tweet*** *at a time*

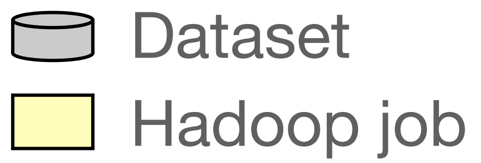
Ingestion pipeline

Online pipeline

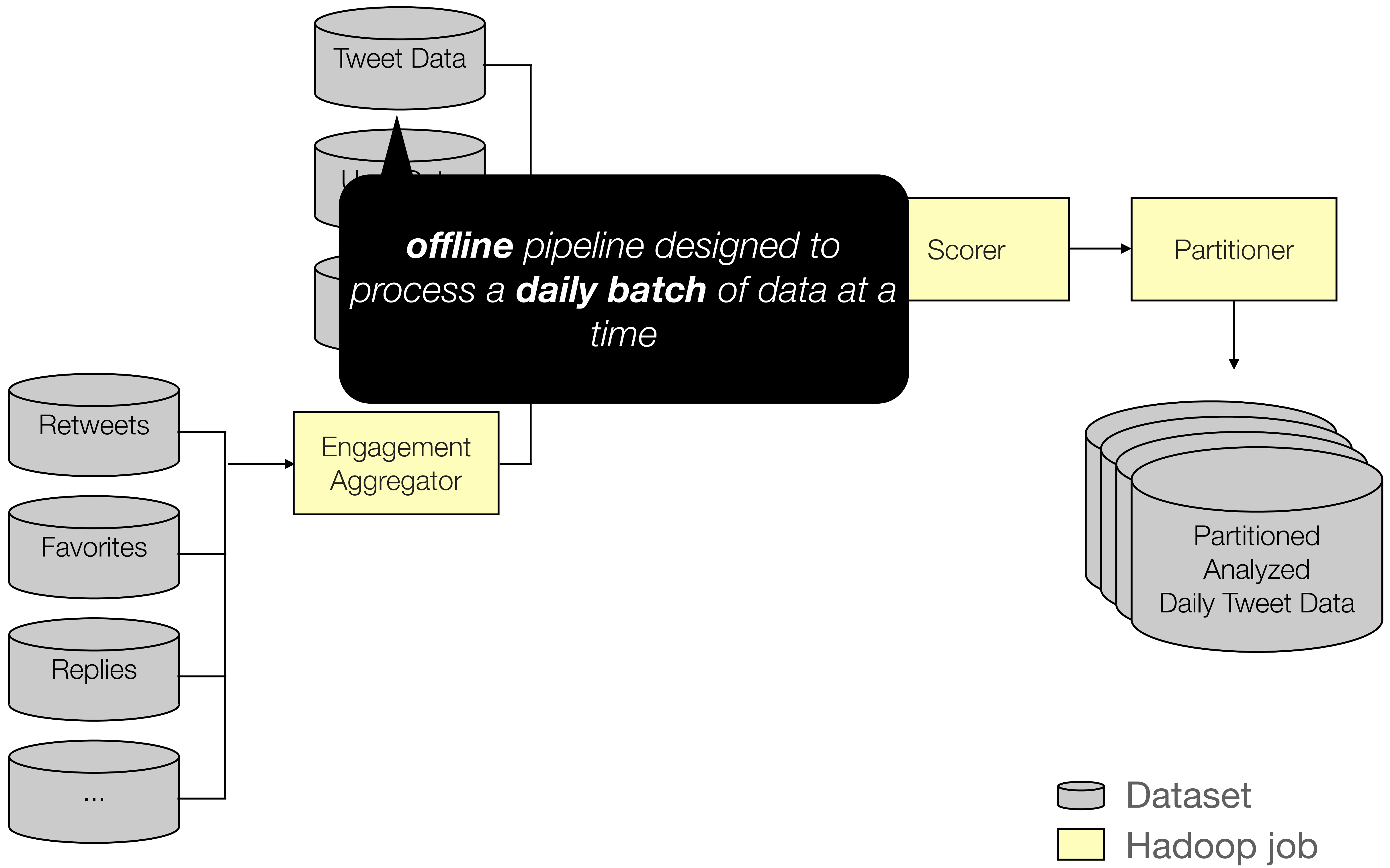


Offline Ingestion pipeline



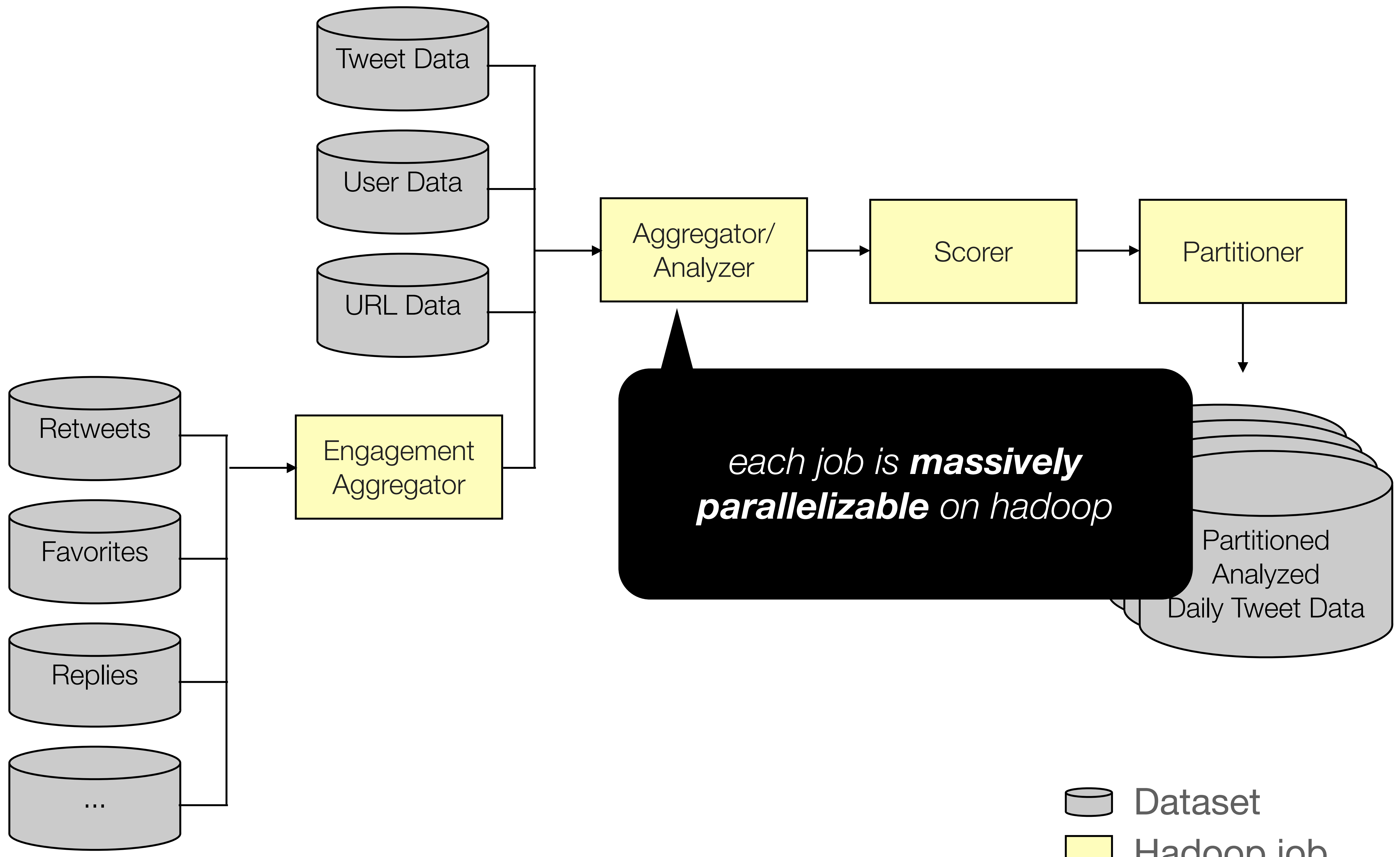


Offline Ingestion pipeline



***offline** pipeline designed to process a **daily batch** of data at a time*

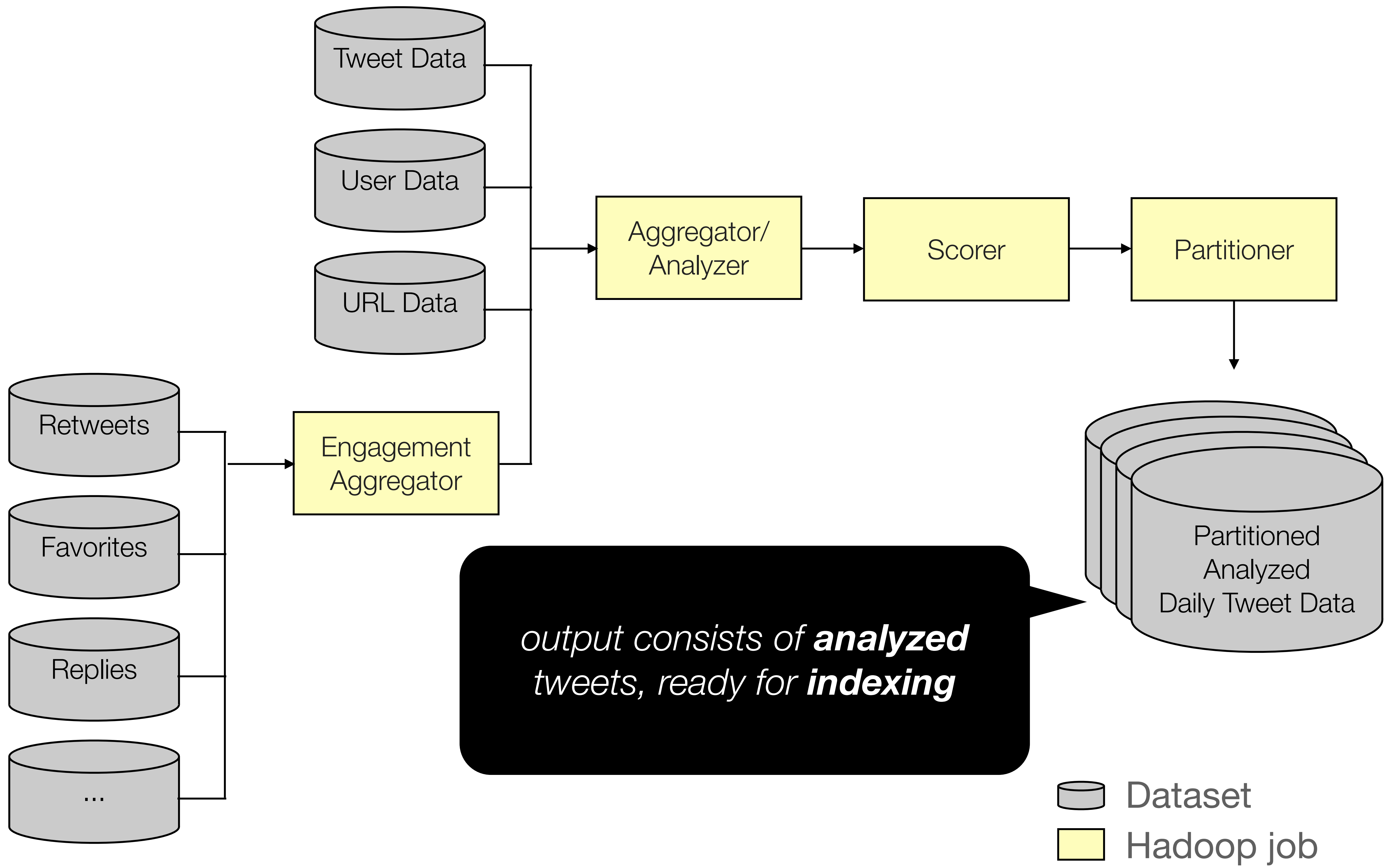
Offline Ingestion pipeline



⬭ Dataset

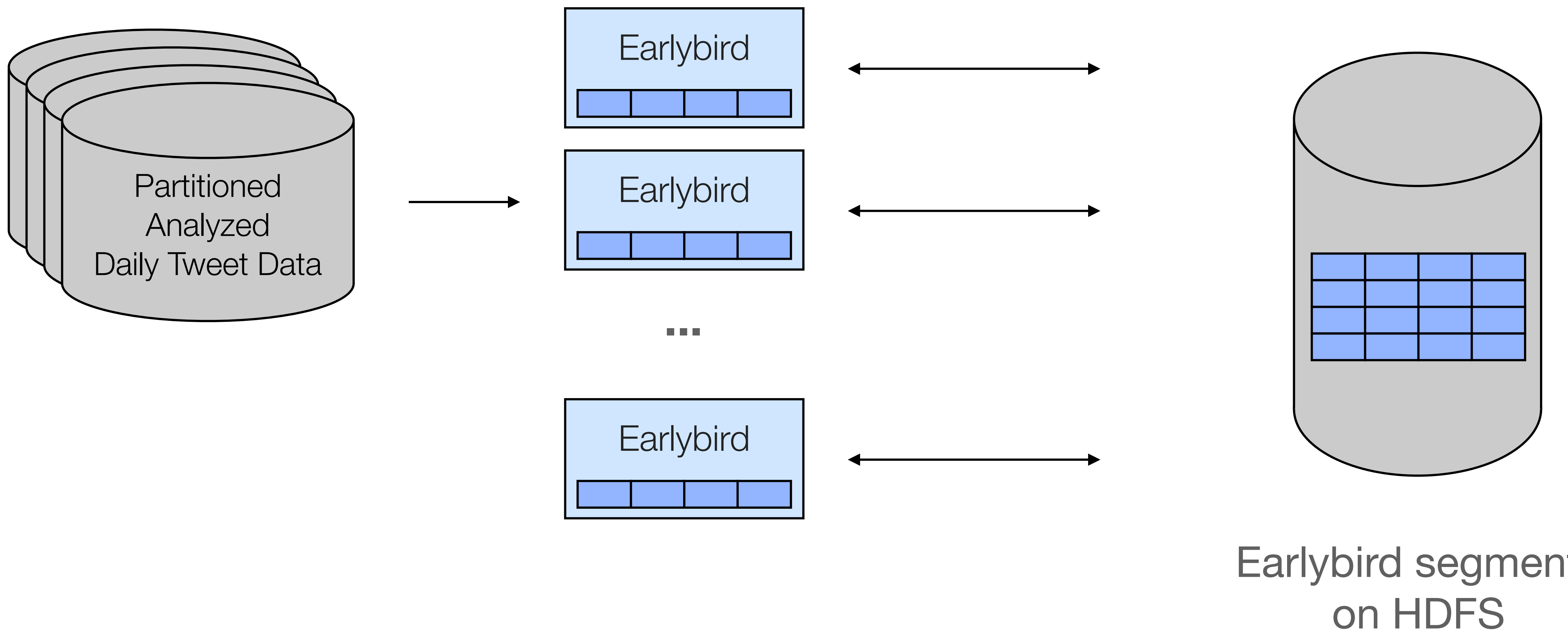
🟨 Hadoop job

Offline Ingestion pipeline

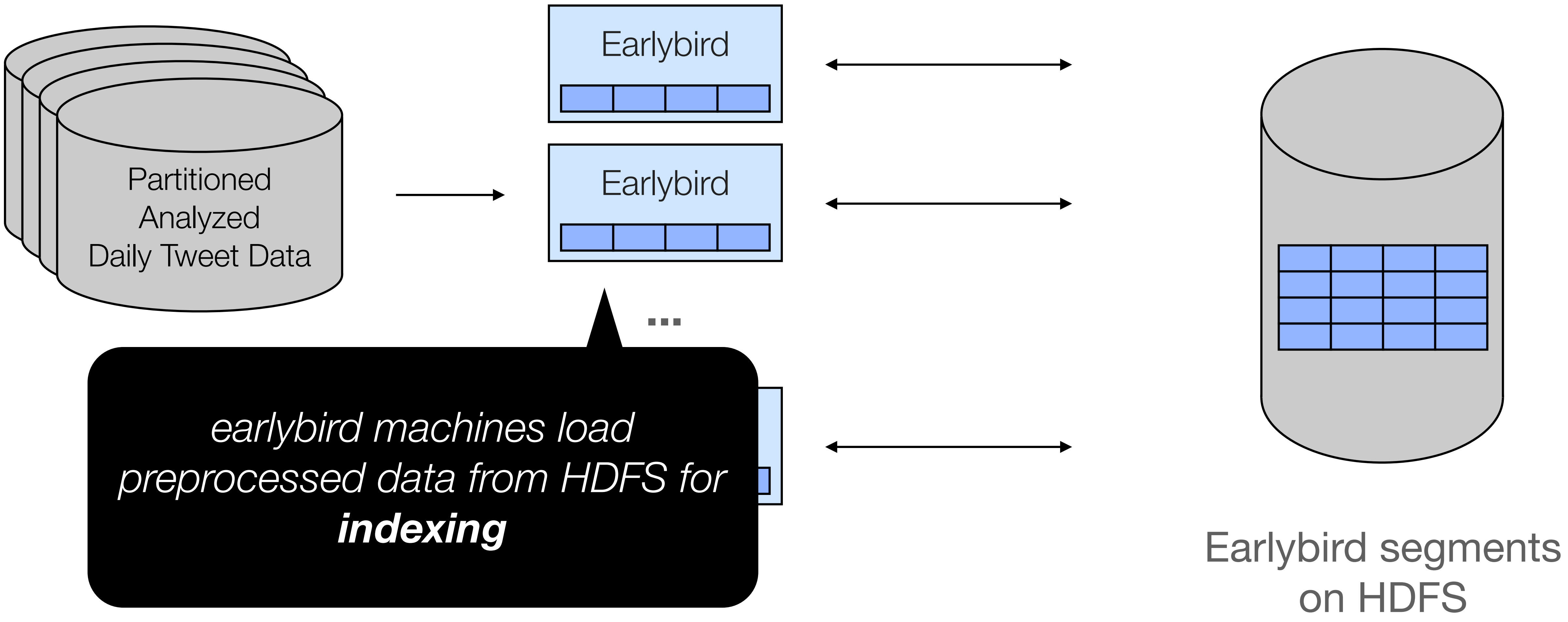


output consists of ***analyzed*** *tweets, ready for* ***indexing***

Offline Ingestion pipeline

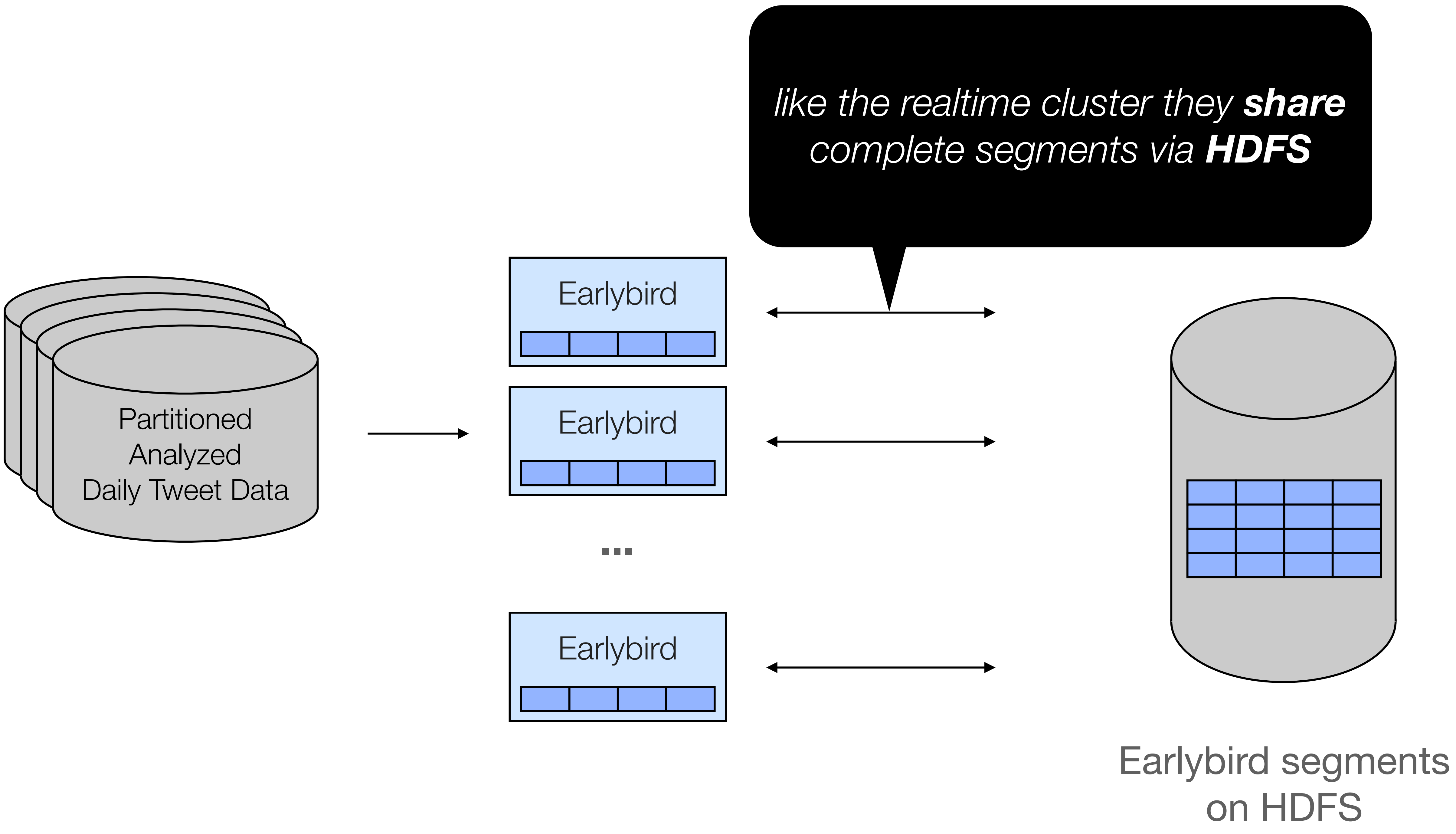


Offline Ingestion pipeline



earlybird machines load preprocessed data from HDFS for ***indexing***

Offline Ingestion pipeline


like the realtime cluster they ***share*** *complete segments via* ***HDFS***
Partitioned Analyzed Daily Tweet Data
Earlybird
Earlybird
...
Earlybird
Earlybird segments on HDFS

2013

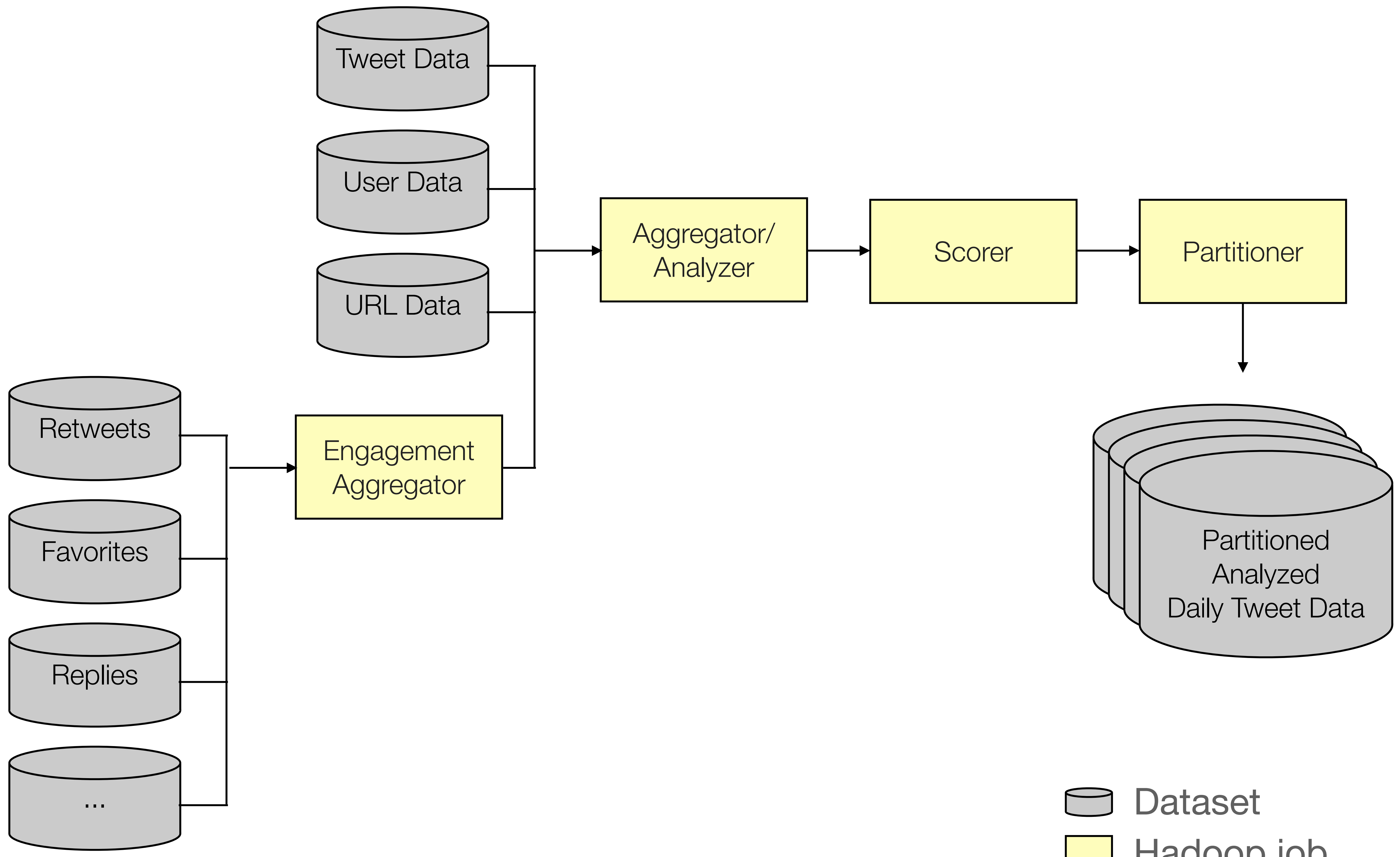
2013

Twitter launches **SSD-based historical index.**

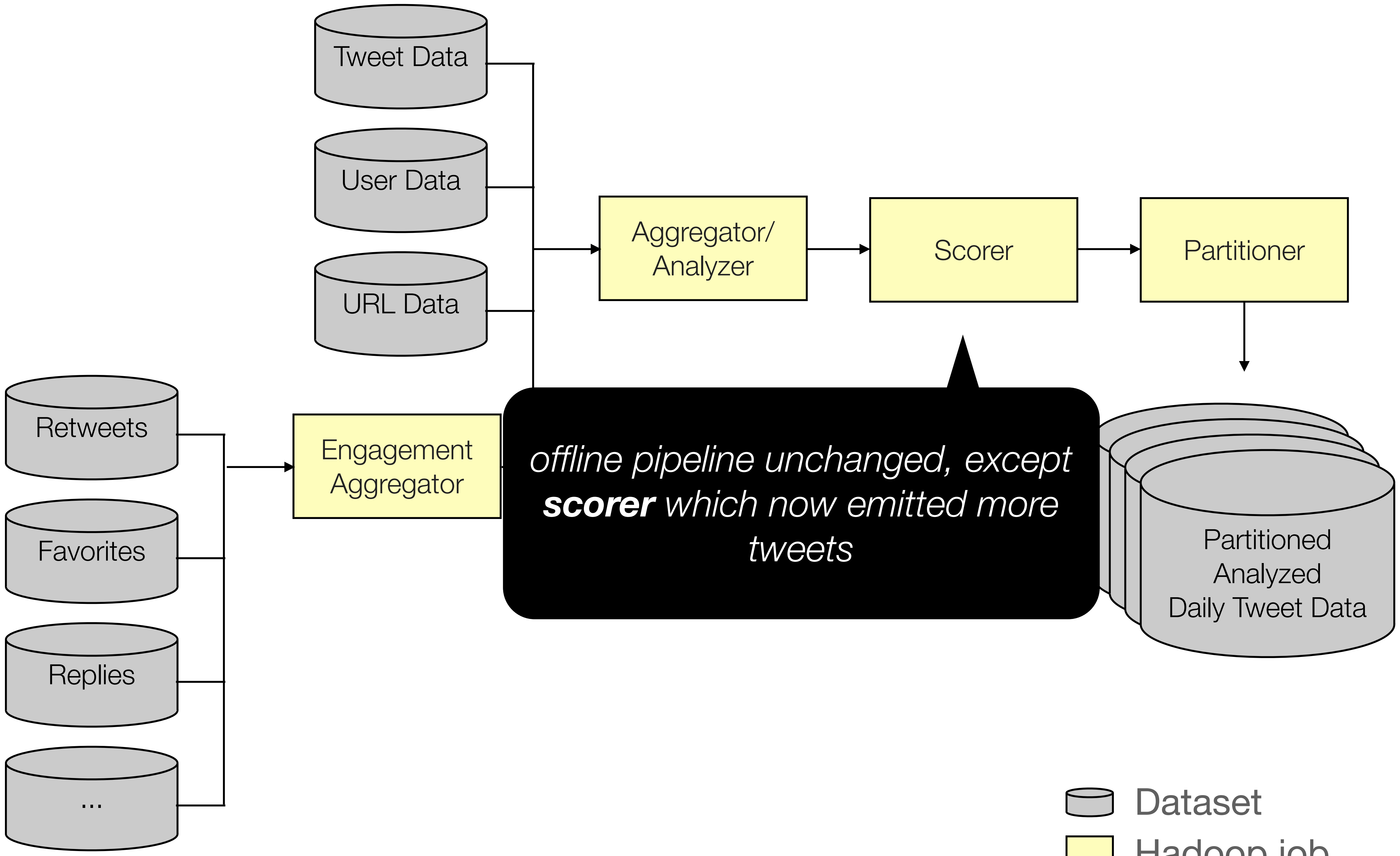
SSD-based historical index

- SSD-based index containing an order of magnitude more tweets than in-memory historical index
- Vanilla Lucene 4.x index format
- Explicit caching of forward indexes
- Rigorous hardware tuning for optimized IOPs
- “Traditional” hash-partitioned cluster layout

Offline Ingestion pipeline



Offline Ingestion pipeline

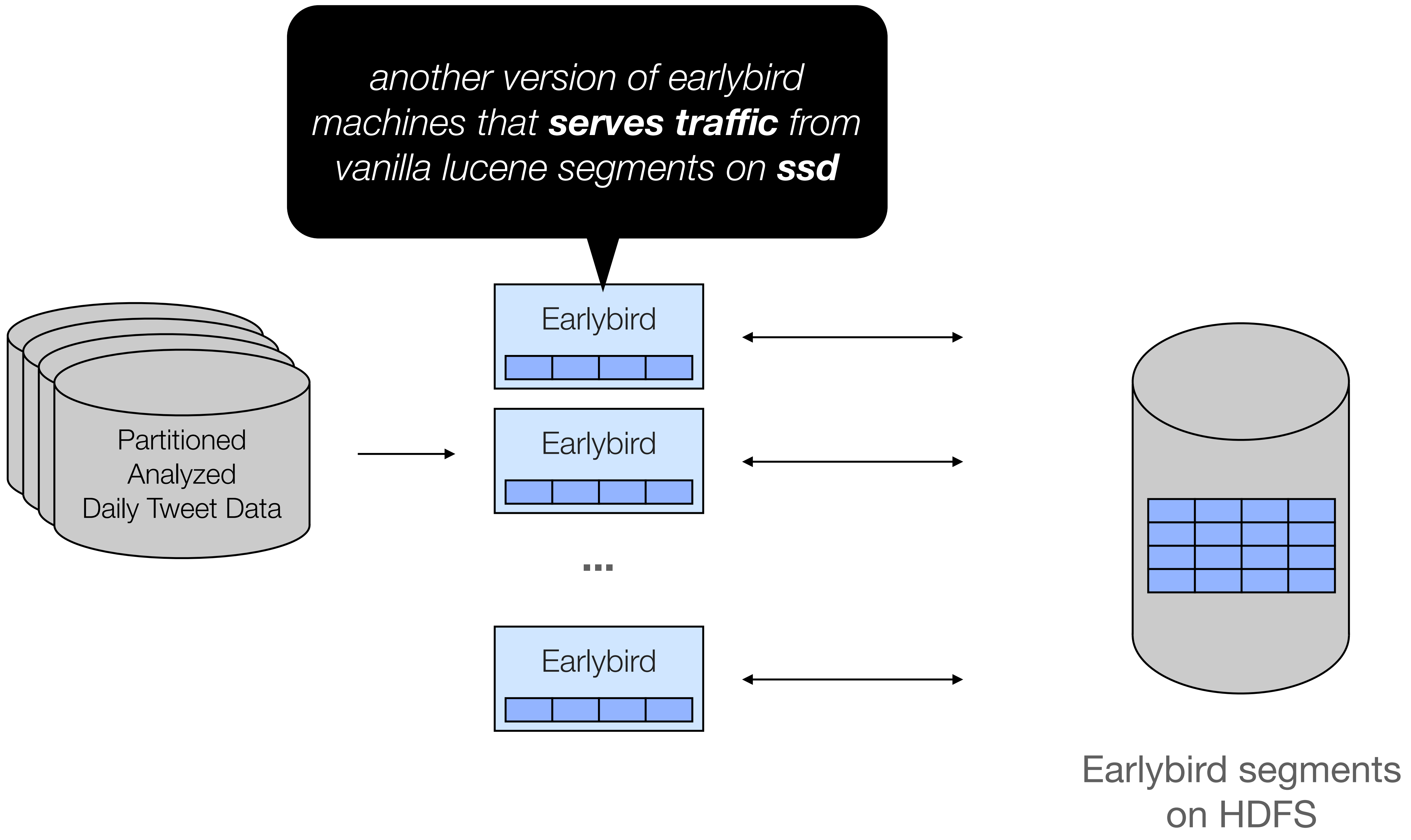


offline pipeline unchanged, except ***scorer*** *which now emitted more tweets*

▭ Dataset

🟨 Hadoop job

Offline Ingestion pipeline



*another version of earlybird machines that **serves traffic** from vanilla lucene segments on **ssd***
Partitioned Analyzed Daily Tweet Data
Earlybird
Earlybird
...
Earlybird
Earlybird segments on HDFS

Performance tuning

- Optimized hardware configuration for maximum SSD utilization
- Explicit packing and caching of Lucene DocValues for scoring
- Auxiliary skip lists using static (query independent) relevance signals
- Graceful degradation in disaster scenarios

2014

2014

Twitter launches **Full Tweet Index.**

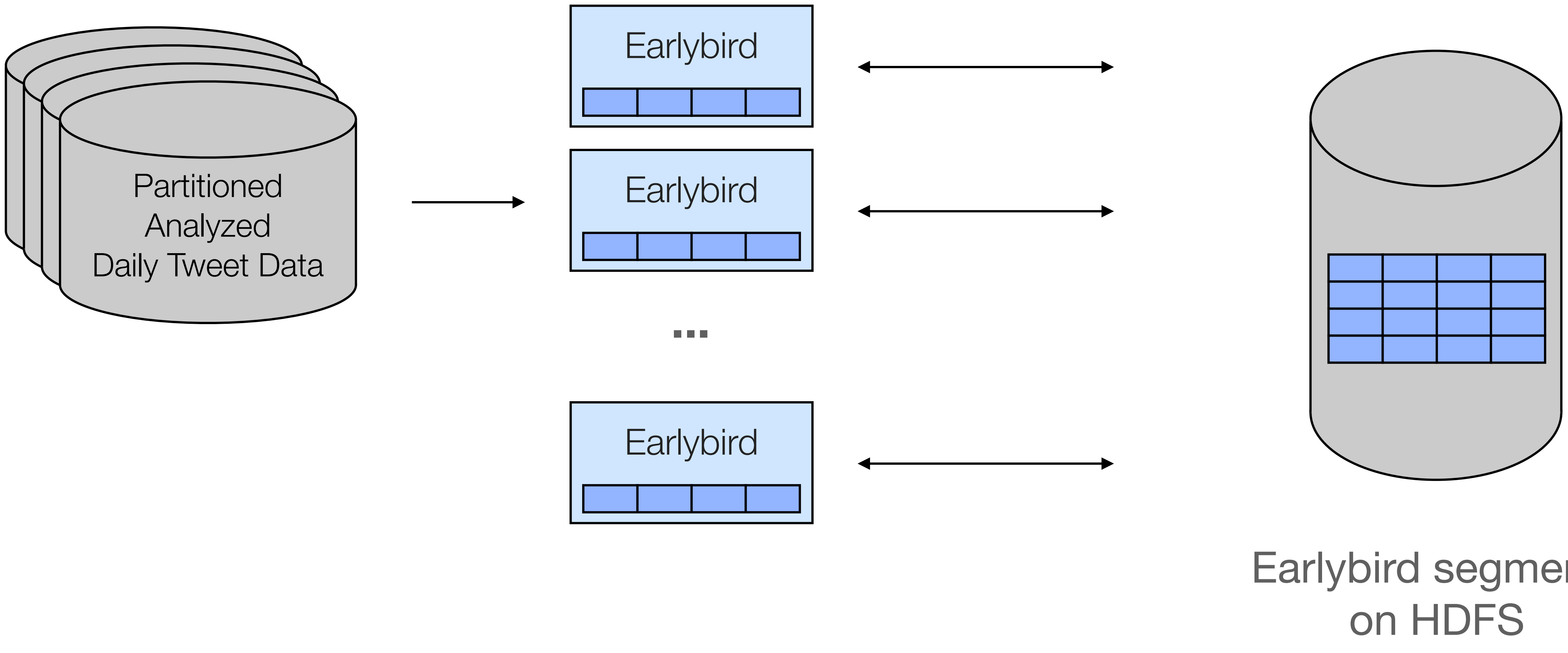
Full Tweet Index

- SSD-based index containing all Tweets ever published
- An order of magnitude bigger index - new ways of scaling necessary:
    - Faster index builds
    - Expanding the index without repartitioning

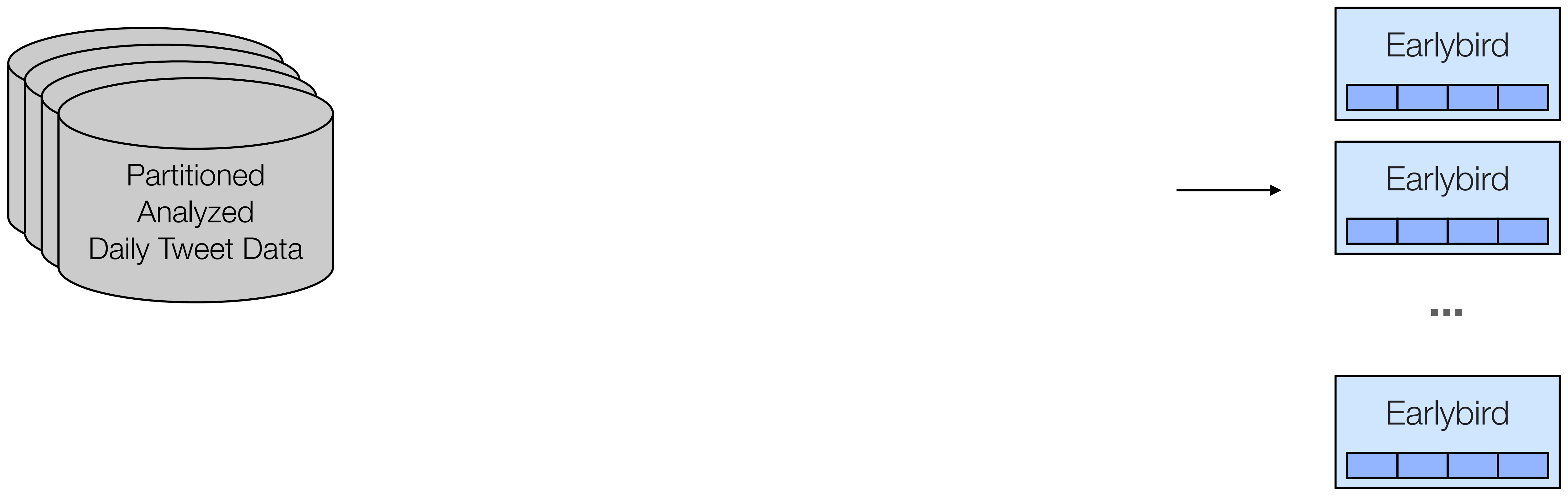
Full Tweet Index

- SSD-based index containing all Tweets ever published
- An order of magnitude bigger index - new ways of scaling necessary:
    - **Faster index builds**
    - Expanding the index without repartitioning

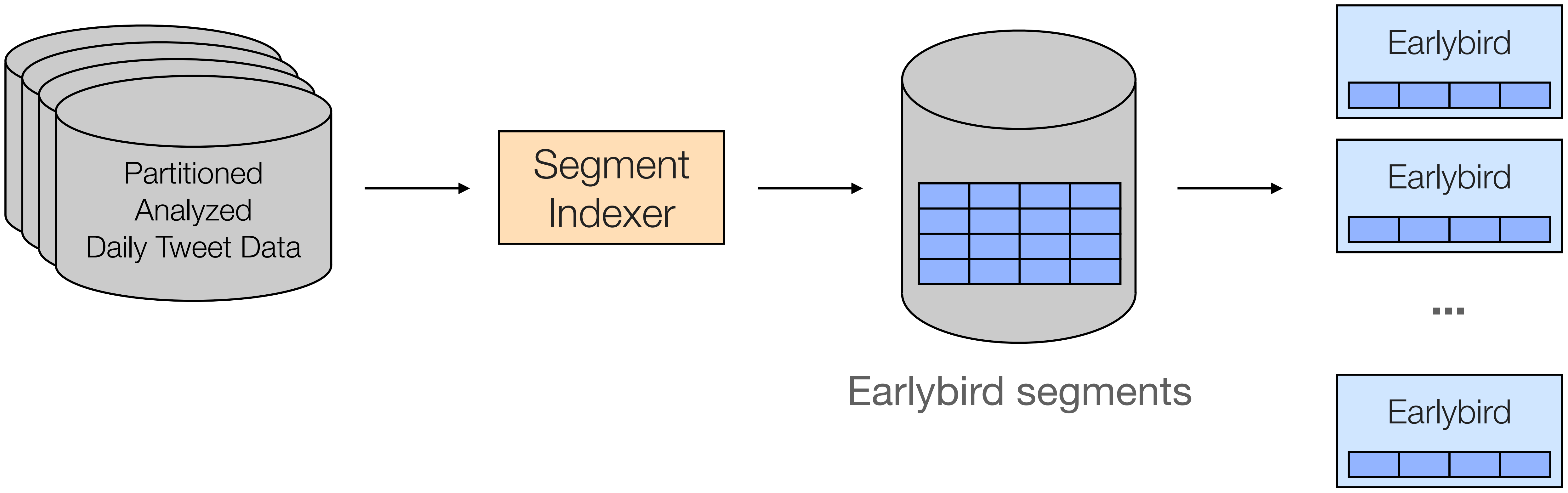
Offline Ingestion pipeline



Offline Ingestion pipeline

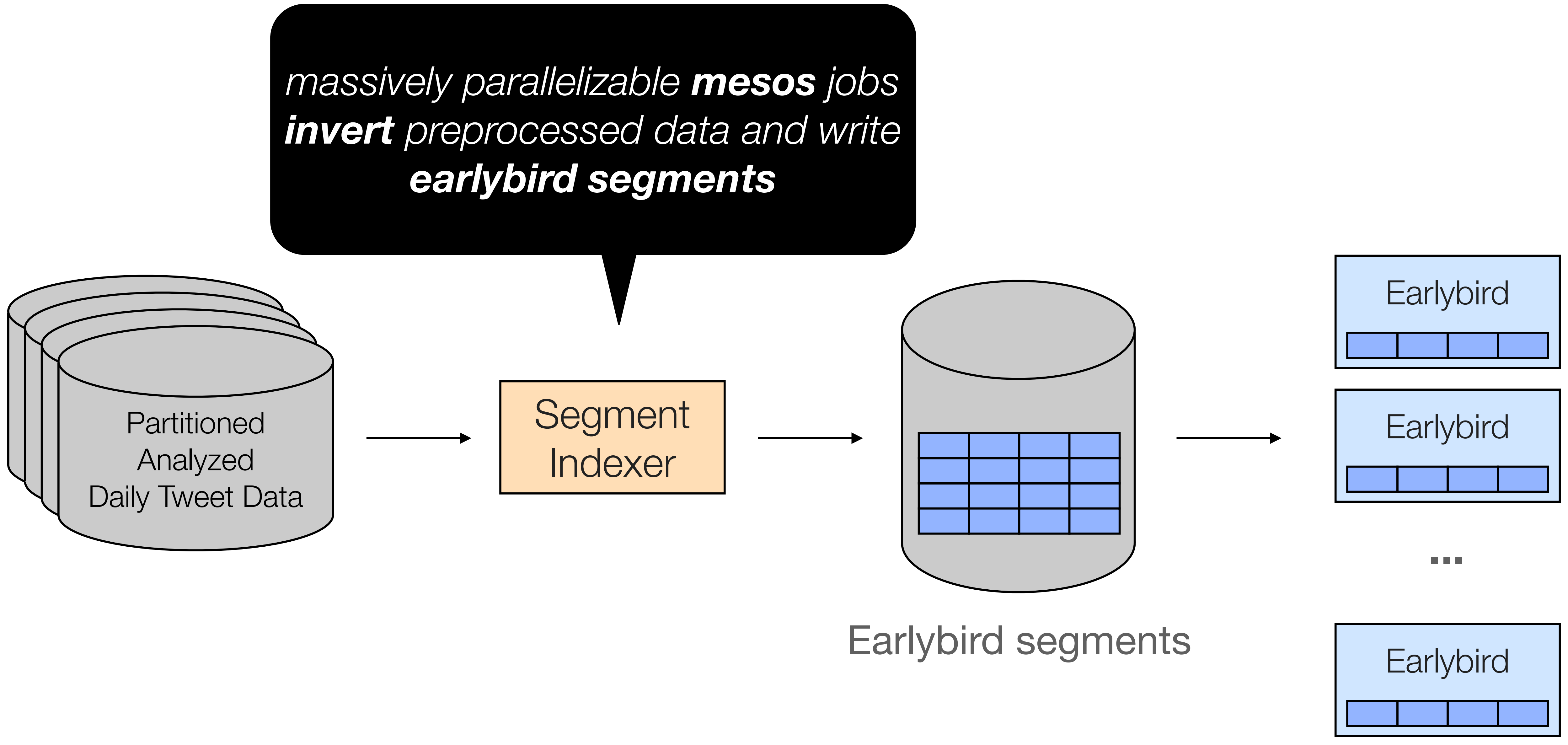


Offline Ingestion pipeline



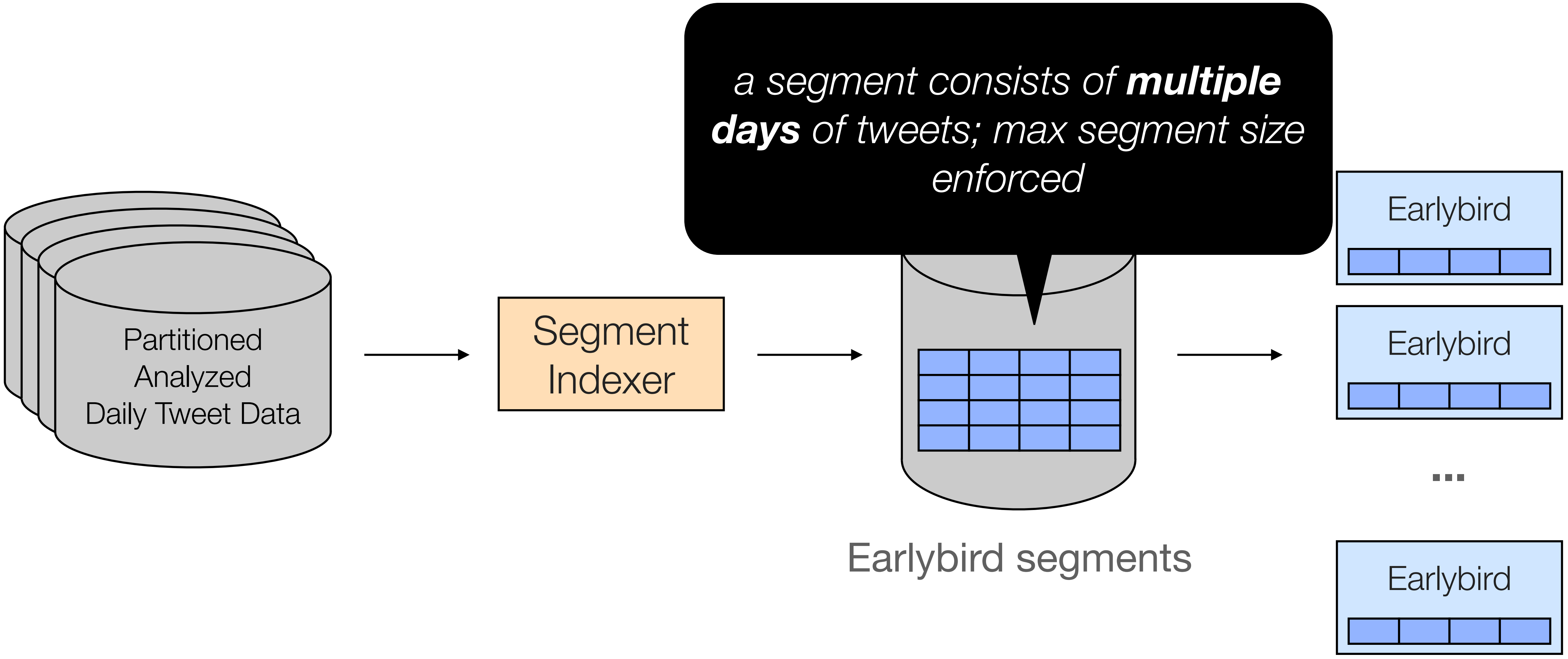
🟧 Mesos job

Offline Ingestion pipeline


*massively parallelizable **mesos** jobs **invert** preprocessed data and write **earlybird segments***
Partitioned Analyzed Daily Tweet Data
Segment Indexer
Earlybird segments
Earlybird
Earlybird
...
Earlybird

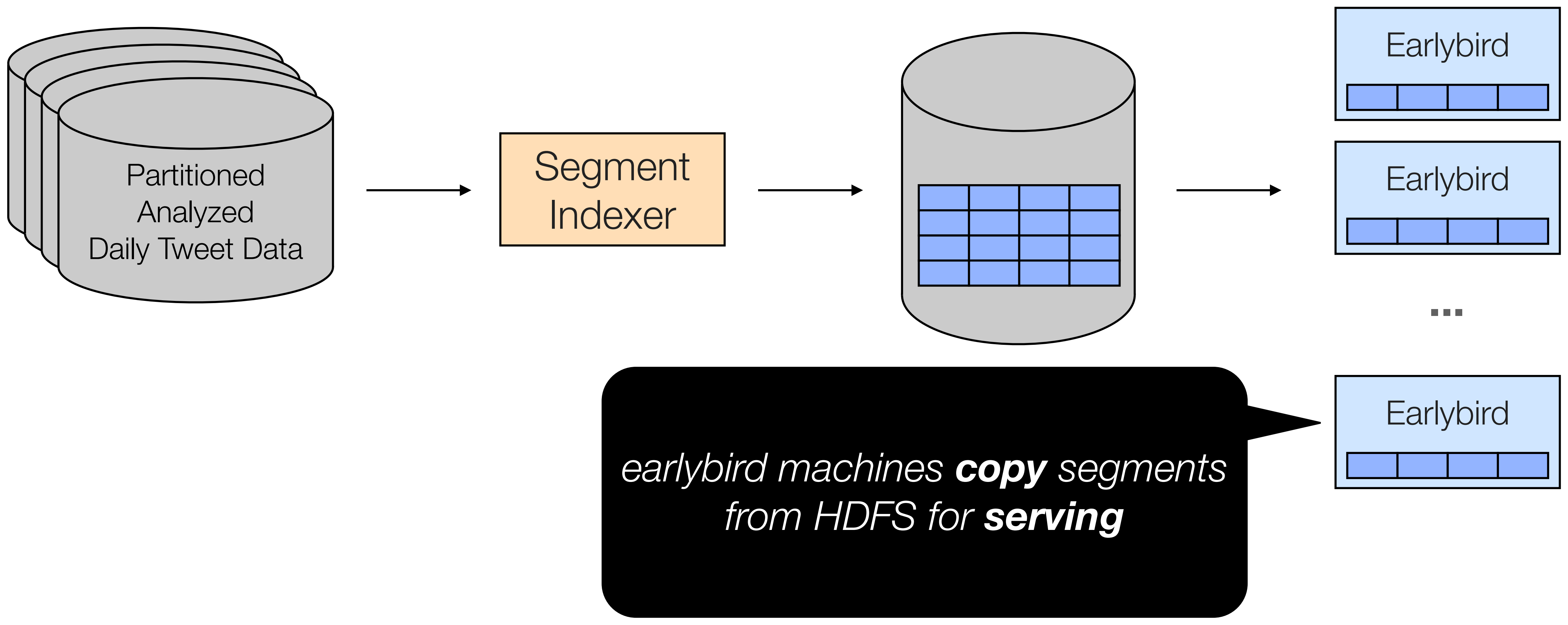
🟧 Mesos job

Offline Ingestion pipeline


Partitioned Analyzed Daily Tweet Data
Segment Indexer
a segment consists of ***multiple days*** *of tweets; max segment size enforced*
Earlybird segments
Earlybird
Earlybird
...
Earlybird

🟧 Mesos job

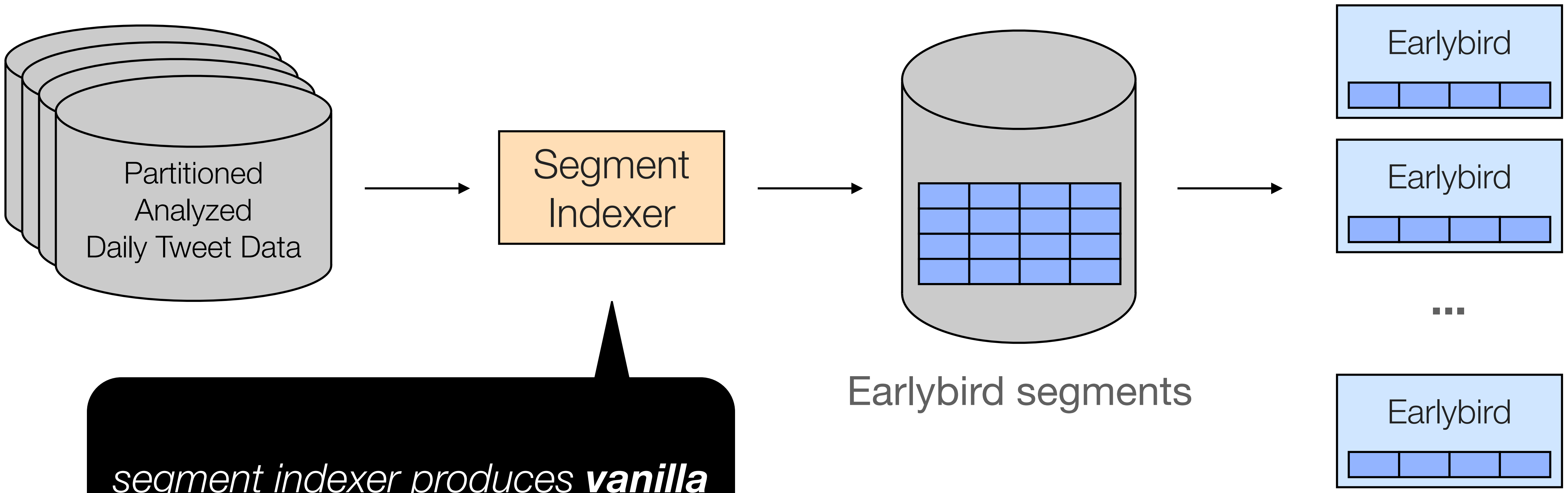
Offline Ingestion pipeline



earlybird machines ***copy*** *segments from HDFS for* ***serving***

🟧 Mesos job

Offline Ingestion pipeline



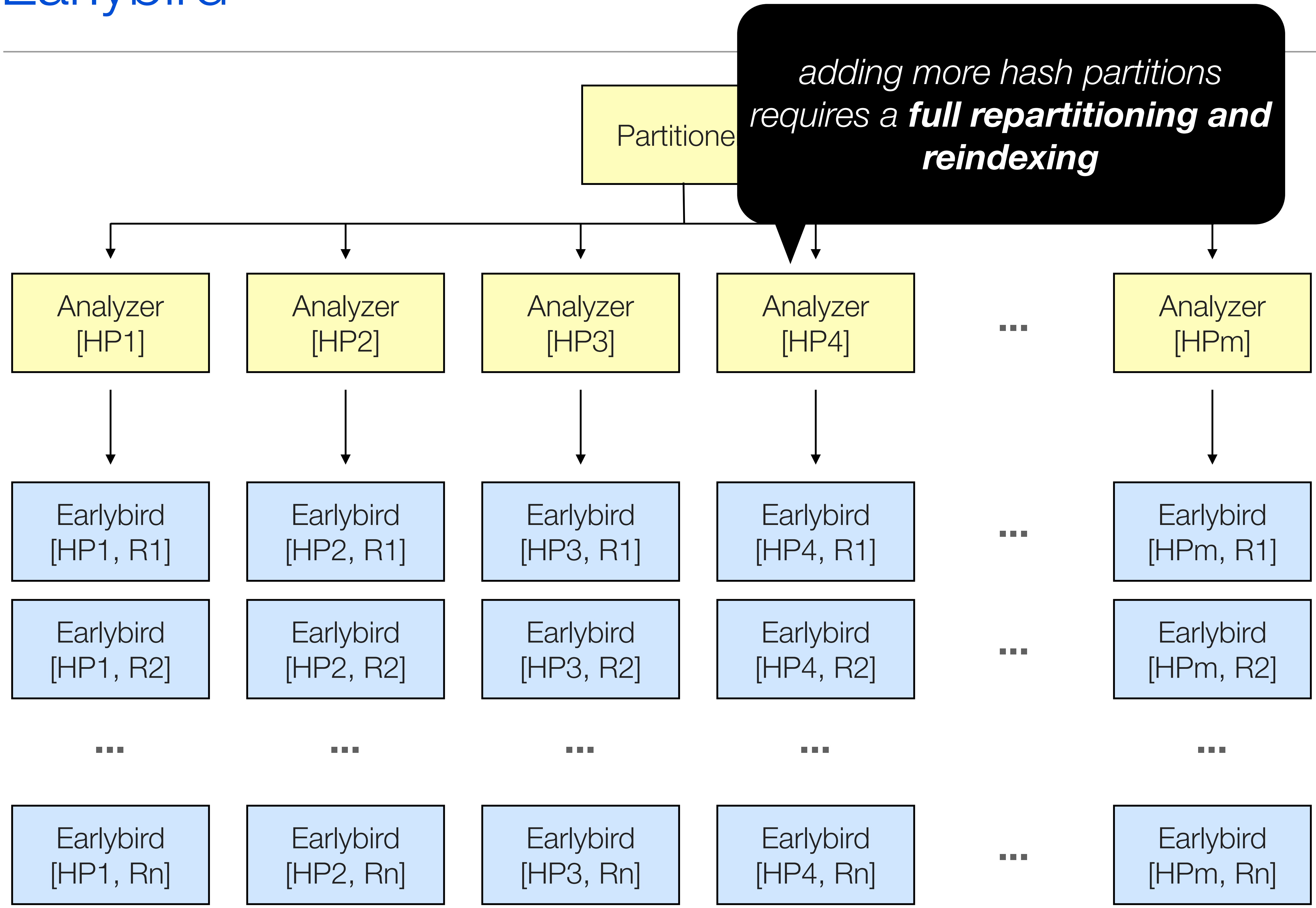
segment indexer produces ***vanilla lucene 4.x*** *segments*

🟧 Mesos job

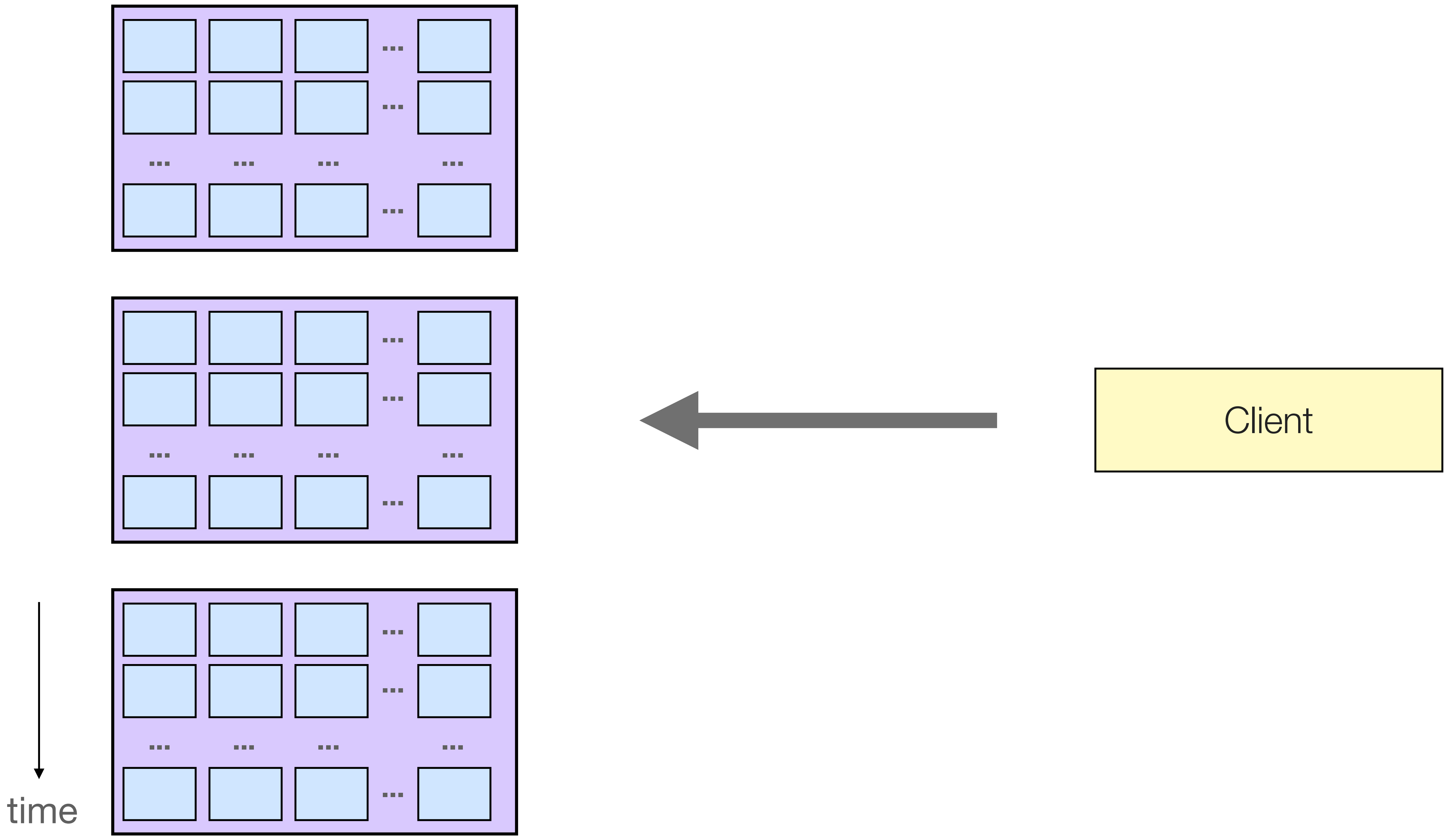
Full Tweet Index

- SSD-based index containing all Tweets ever published
- An order of magnitude bigger index - new ways of scaling necessary:
    - Faster index builds
    - **Expanding the index without repartitioning**

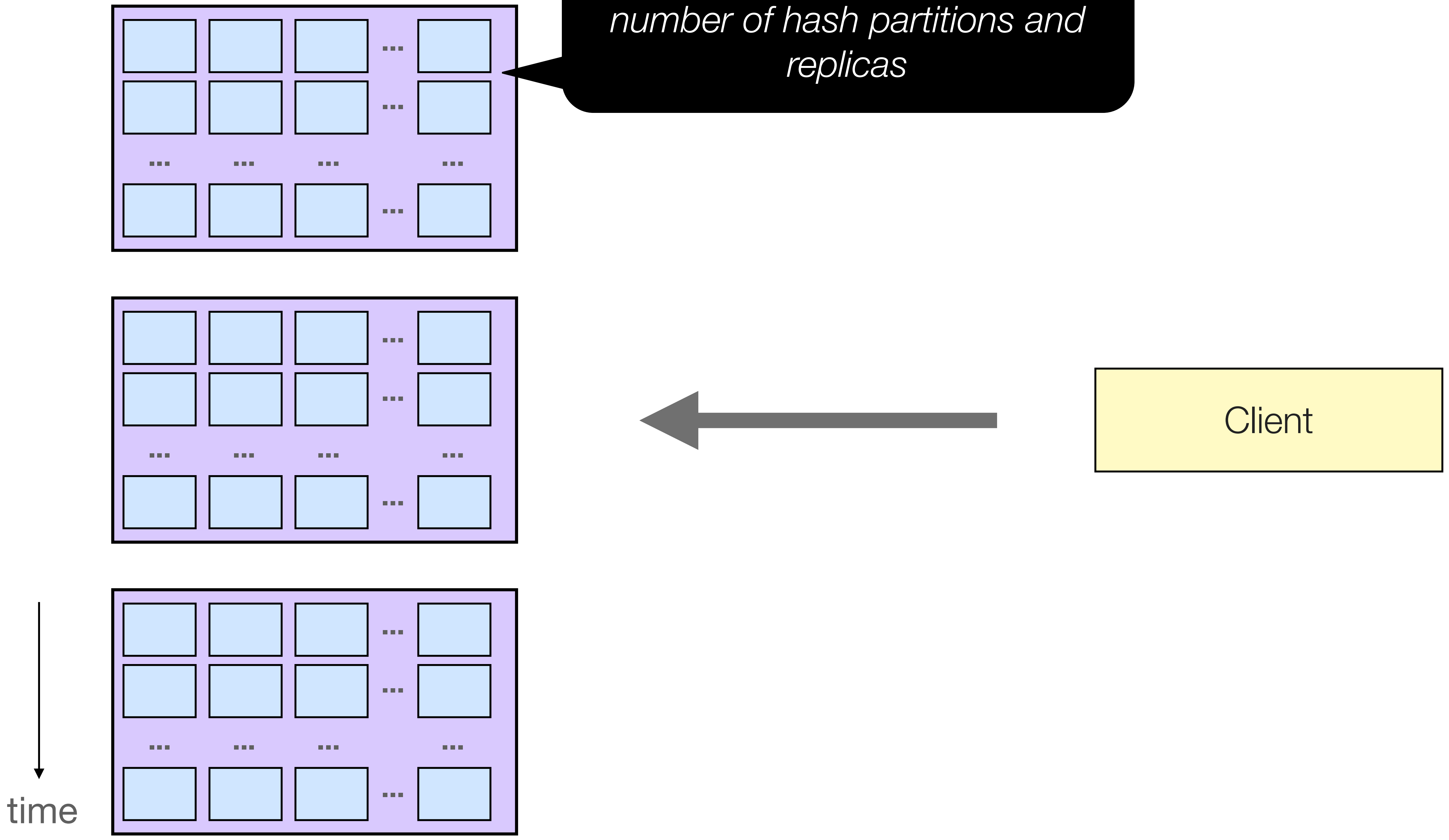
Earlybird


Partitione
*adding more hash partitions requires a **full repartitioning and reindexing***
Analyzer [HP1]
Analyzer [HP2]
Analyzer [HP3]
Analyzer [HP4]
...
Analyzer [HPm]
Earlybird [HP1, R1]
Earlybird [HP2, R1]
Earlybird [HP3, R1]
Earlybird [HP4, R1]
Earlybird [HPm, R1]
Earlybird [HP1, R2]
Earlybird [HP2, R2]
Earlybird [HP3, R2]
Earlybird [HP4, R2]
Earlybird [HPm, R2]
Earlybird [HP1, Rn]
Earlybird [HP2, Rn]
Earlybird [HP3, Rn]
Earlybird [HP4, Rn]
Earlybird [HPm, Rn]

Time Tiers

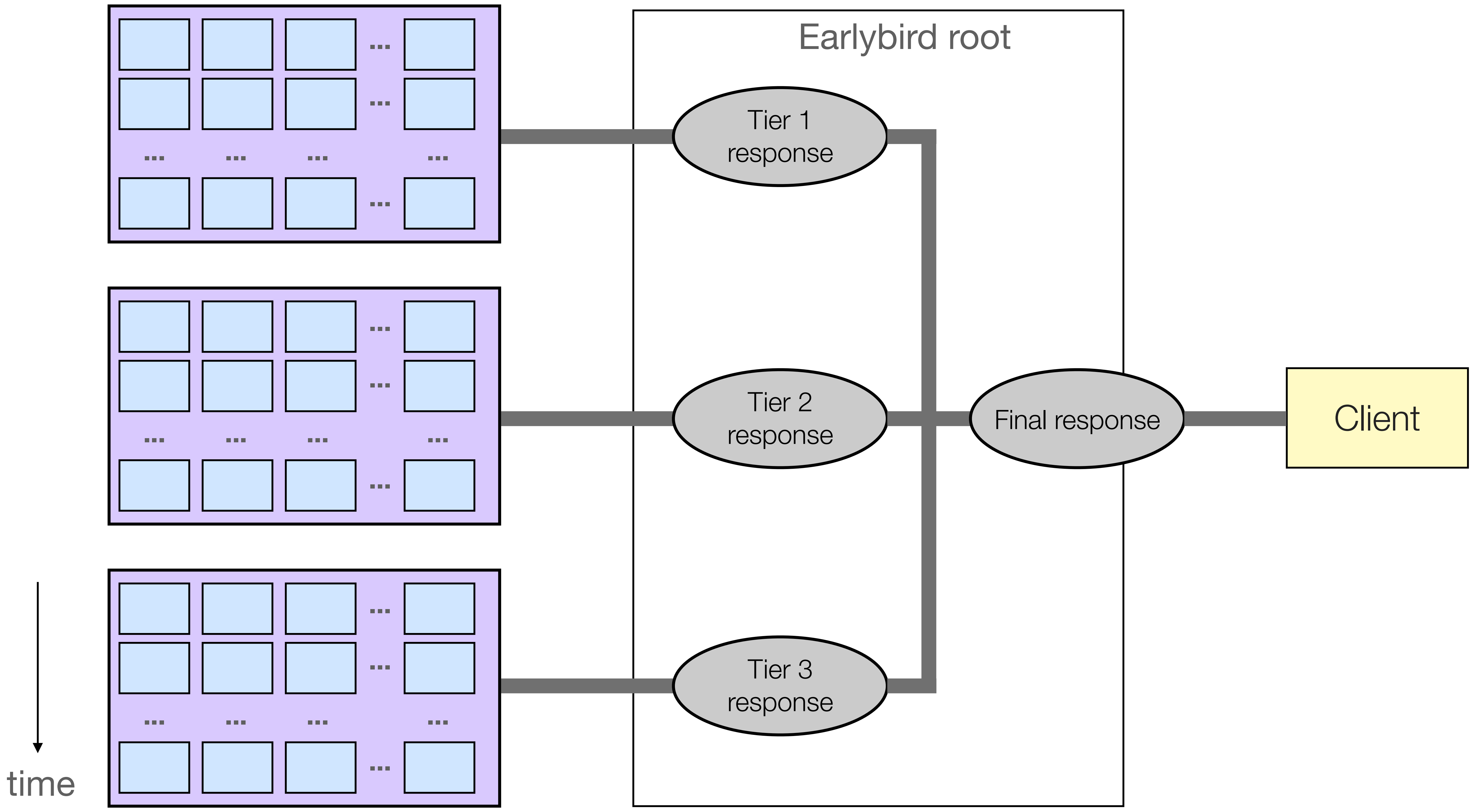


Time Tiers

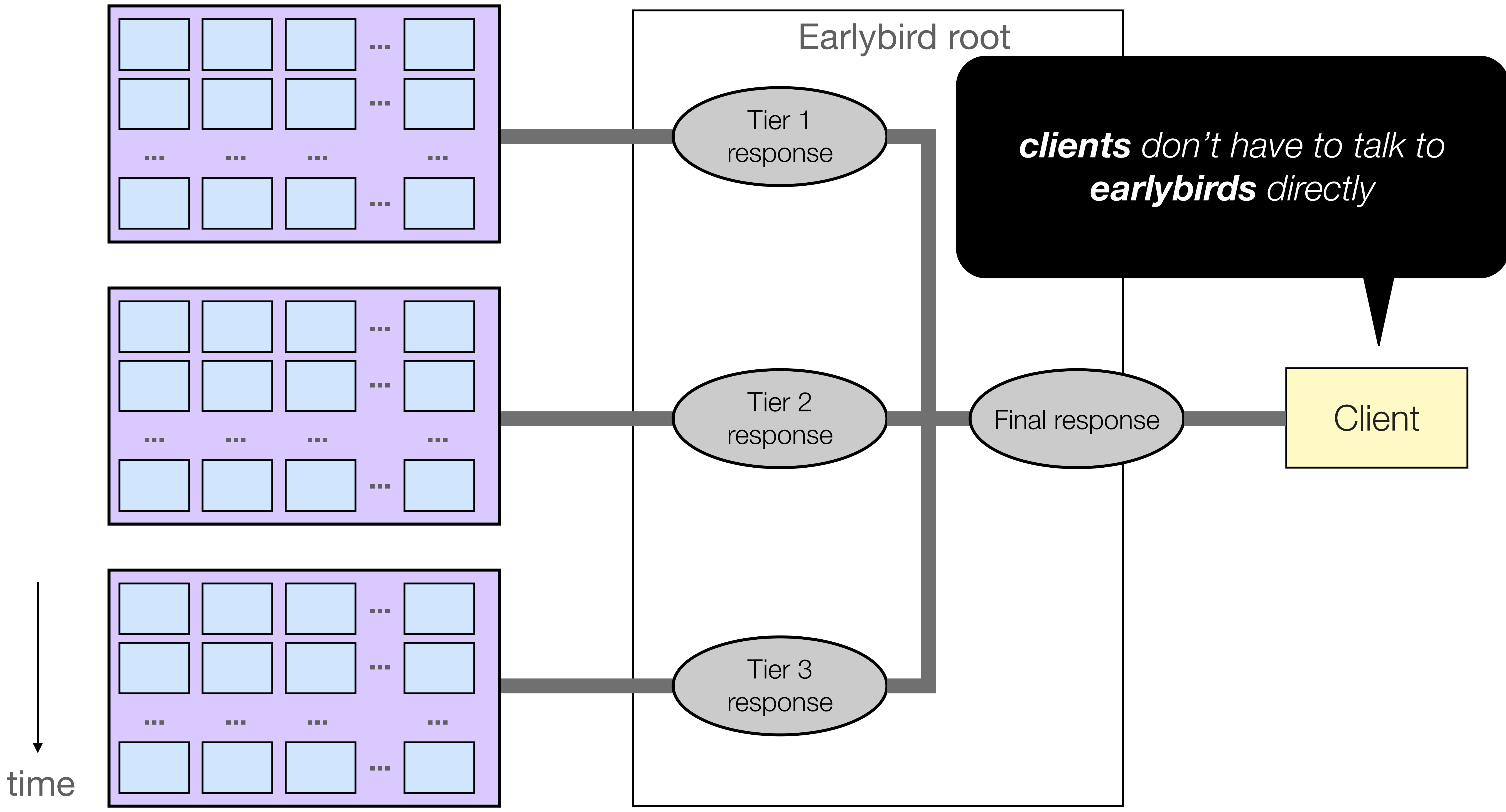


each ***tier*** *covers a* ***fixed time range****; each tier has a fixed number of hash partitions and replicas*

Search roots



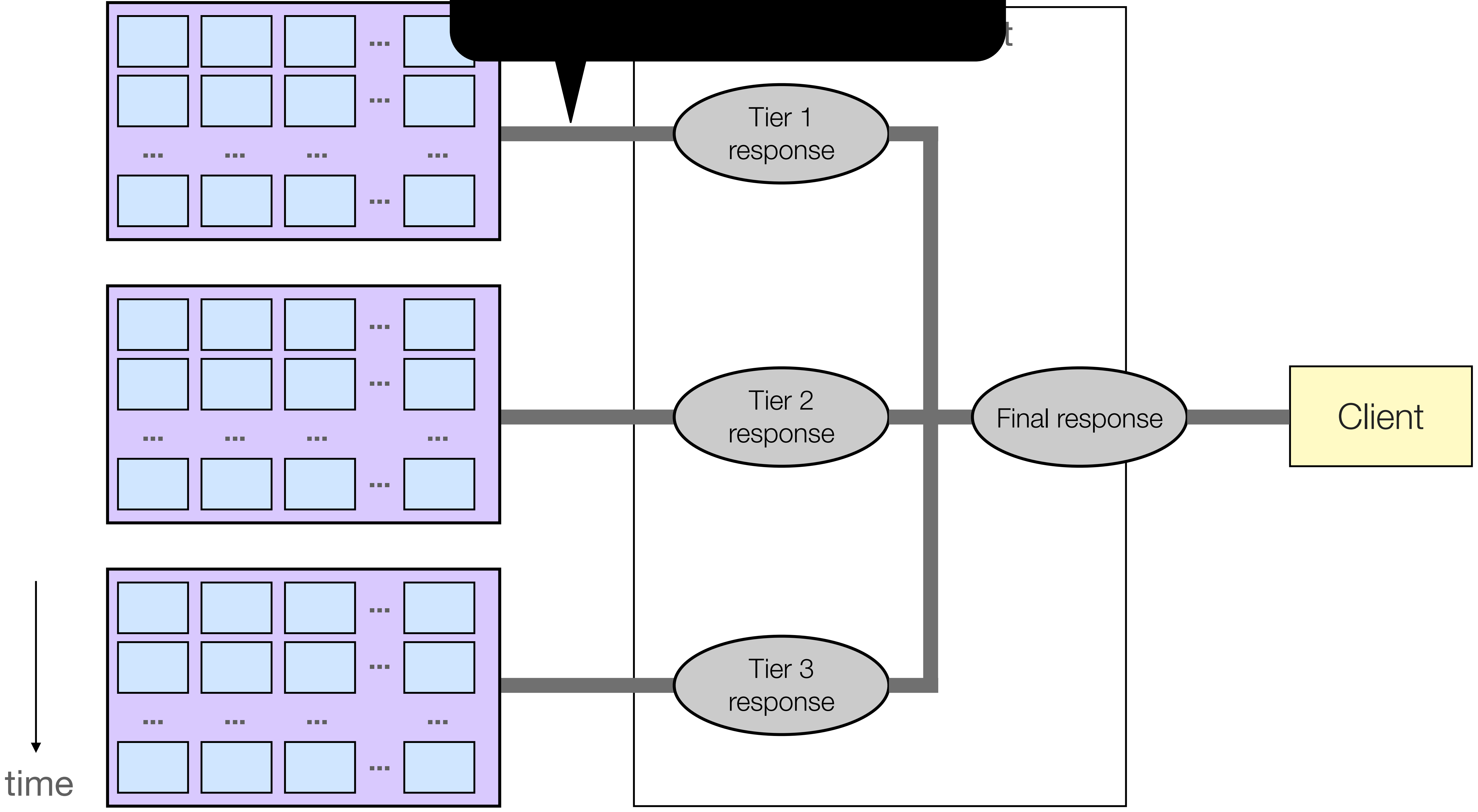
Search roots



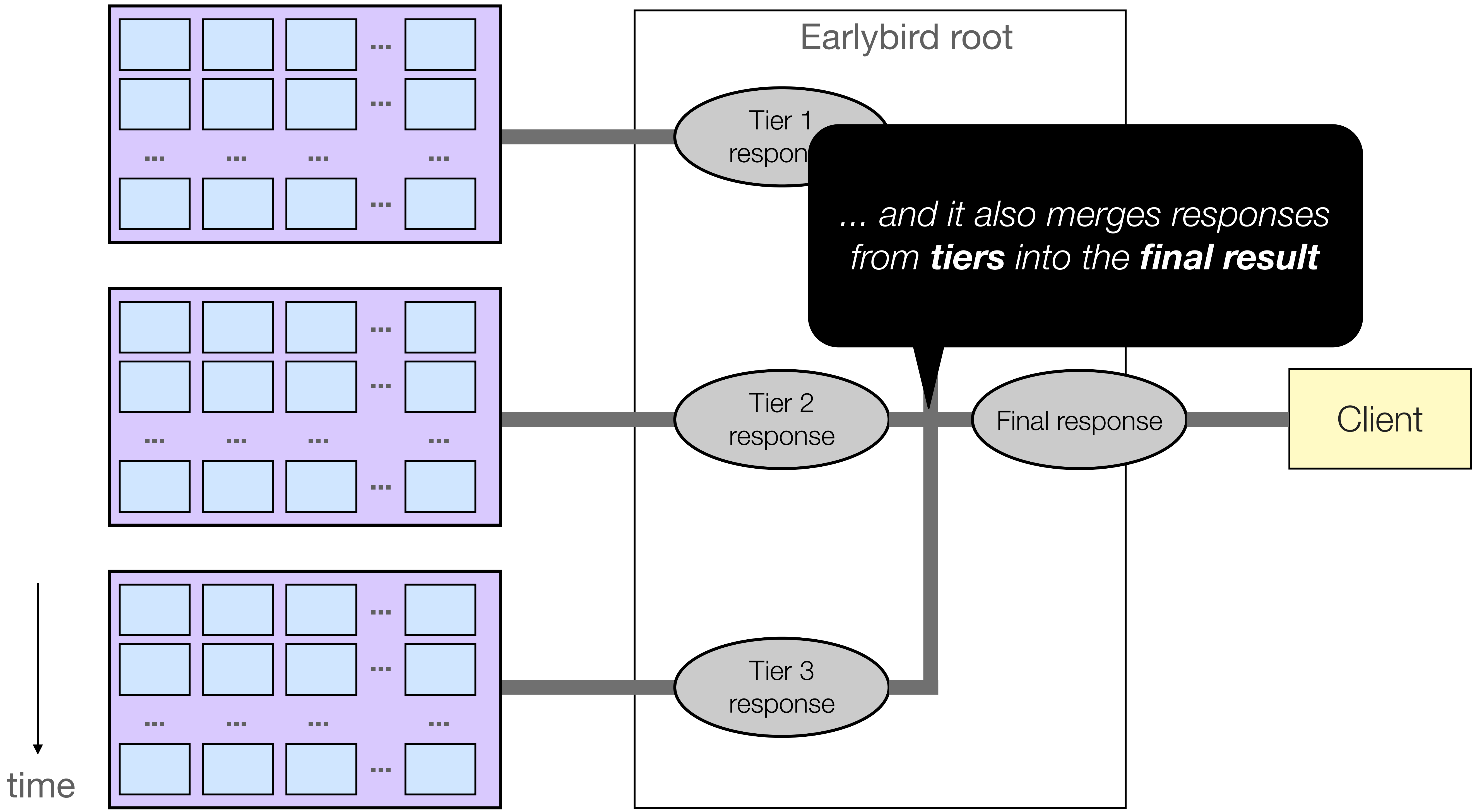
clients *don't have to talk to* ***earlybirds*** *directly*

Search roots

***earlybird root** merges results from multiple **hash partitions** ...*



Search roots

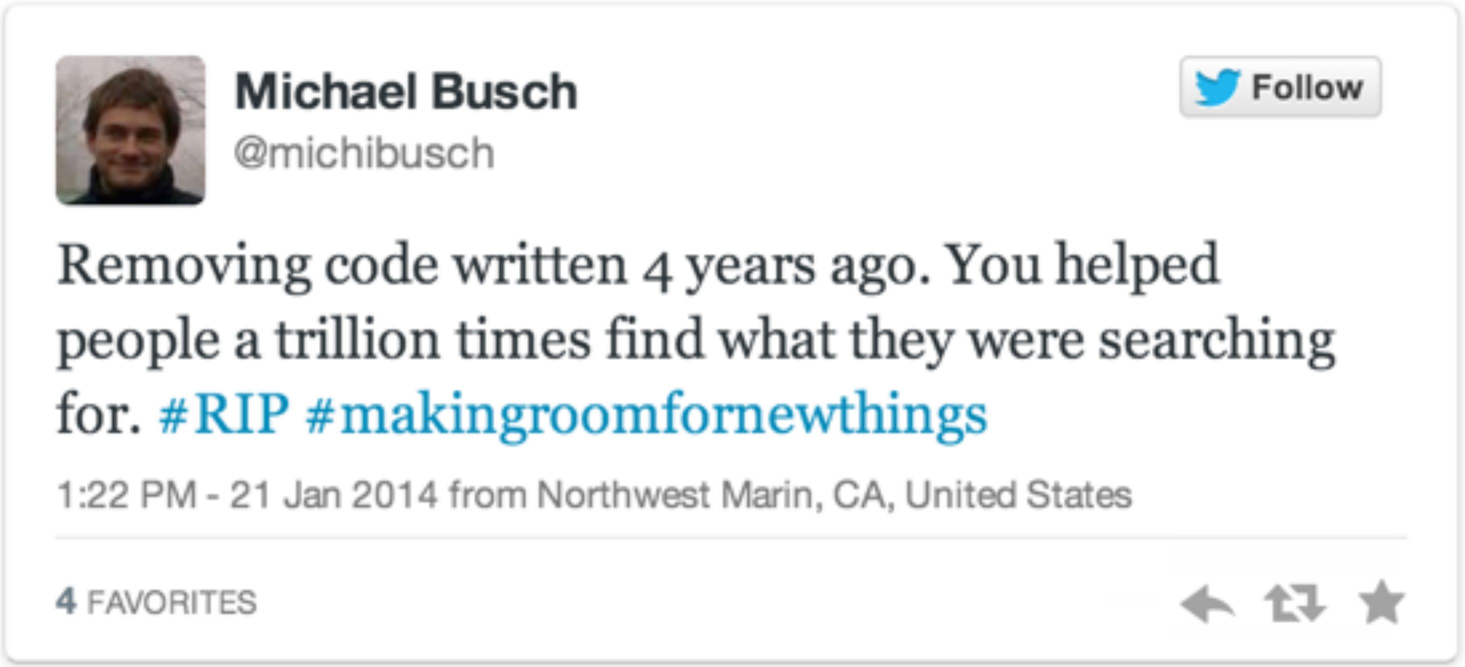


Full Tweet Index

- SSD-based index containing all Tweets ever published
- Data structures to make previous historical indexes obsolete
- New tiered cluster layout for easy index scaling
- New Mesos-based index builder system

In-memory Real-time Index

- Highly optimized for GC - all data is stored in blocked native arrays
- v1: Optimized for tweets with a term position limit of 255
- v2: Support for 32 bit positions without performance degradation
- v2: Basic support for out-of-order posting list inserts



Michael Busch
@michibusch

Removing code written 4 years ago. You helped people a trillion times find what they were searching for. #RIP #makingroomfornewthings

1:22 PM - 21 Jan 2014 from Northwest Marin, CA, United States

4 FAVORITES

2015

Outlook

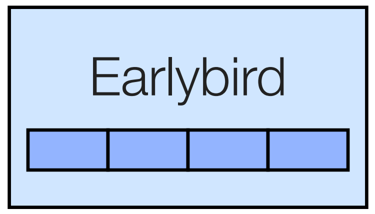
Outlook

- **Parallel indexing pipelines for faster index manipulation**
- Domain-independent core indexing library that combines real-time and offline index technology

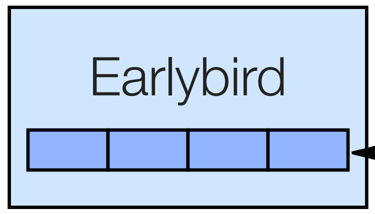
Modifying indexes

- The tiered architecture allows us to grow the index without having to reindex existing data
- But appending new fields to existing indexes or e.g. upgrading analyzers requires a full reprocessing of all data
- Idea: Introduce segment slices and parallel ingestion pipelines

Segment Slices

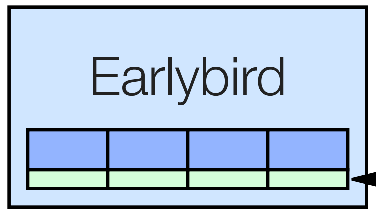


Segment Slices



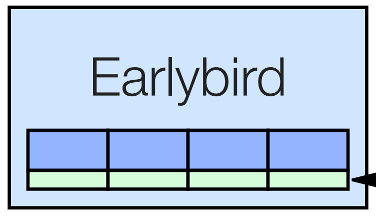
*we want to append a **new field** to an existing segment; however lucene segments are **immutable***

Segment Slices



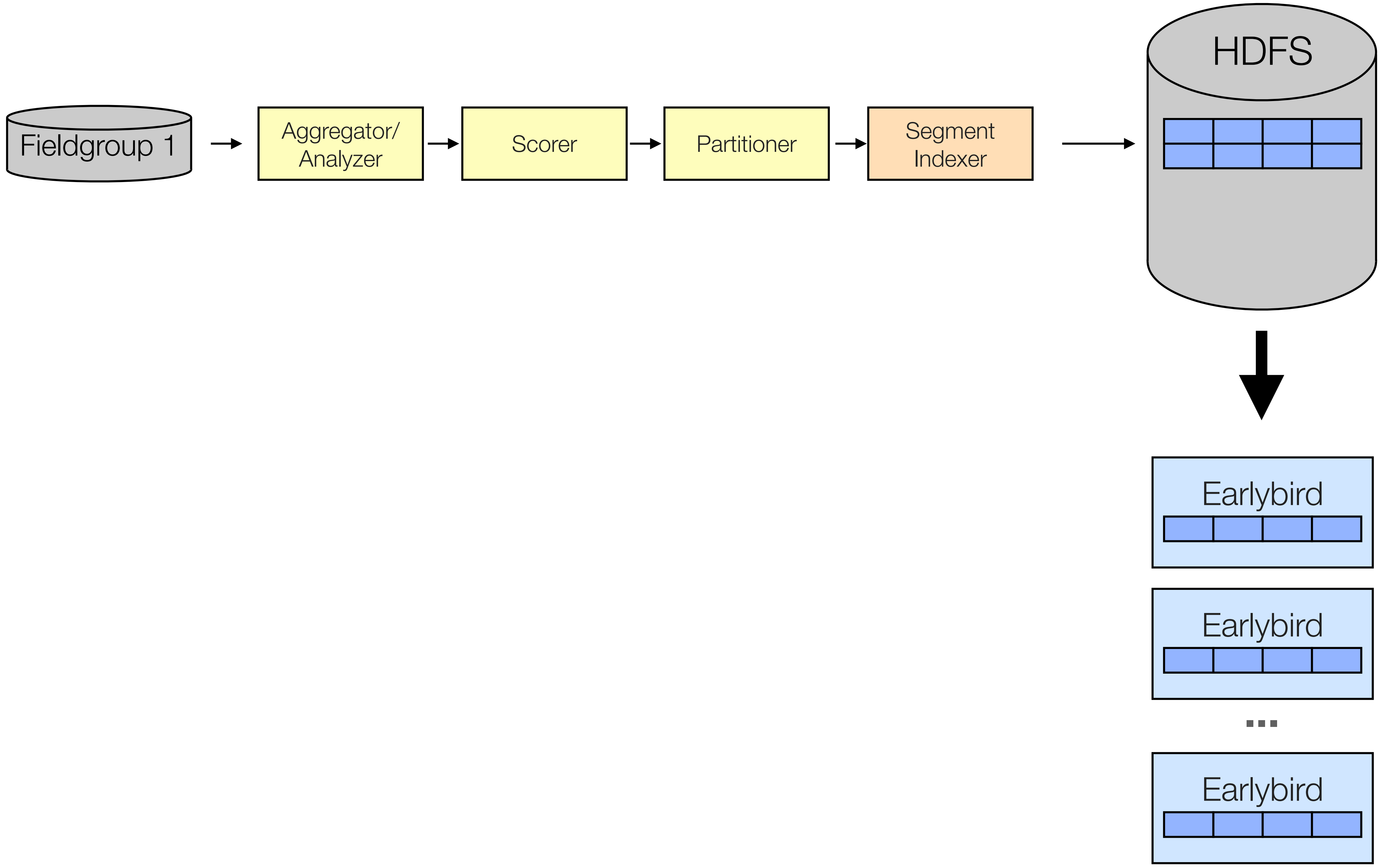
index a separate parallel segment covering the ***same doc id range***

Segment Slices

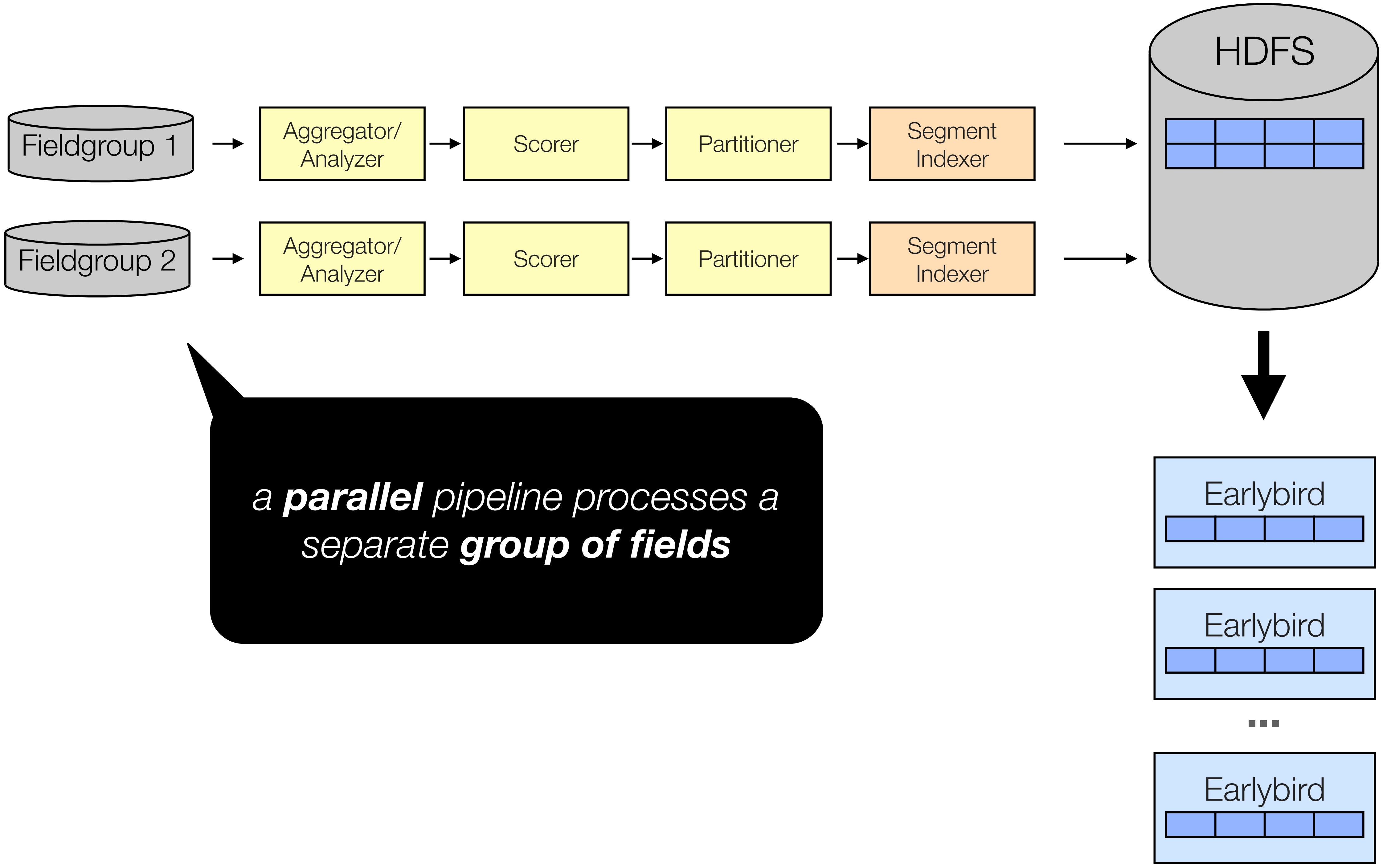


using lucene's ***ParallelIndexReader*** *we can read both segment slides as if they were a single one*

Parallel Ingestion pipeline

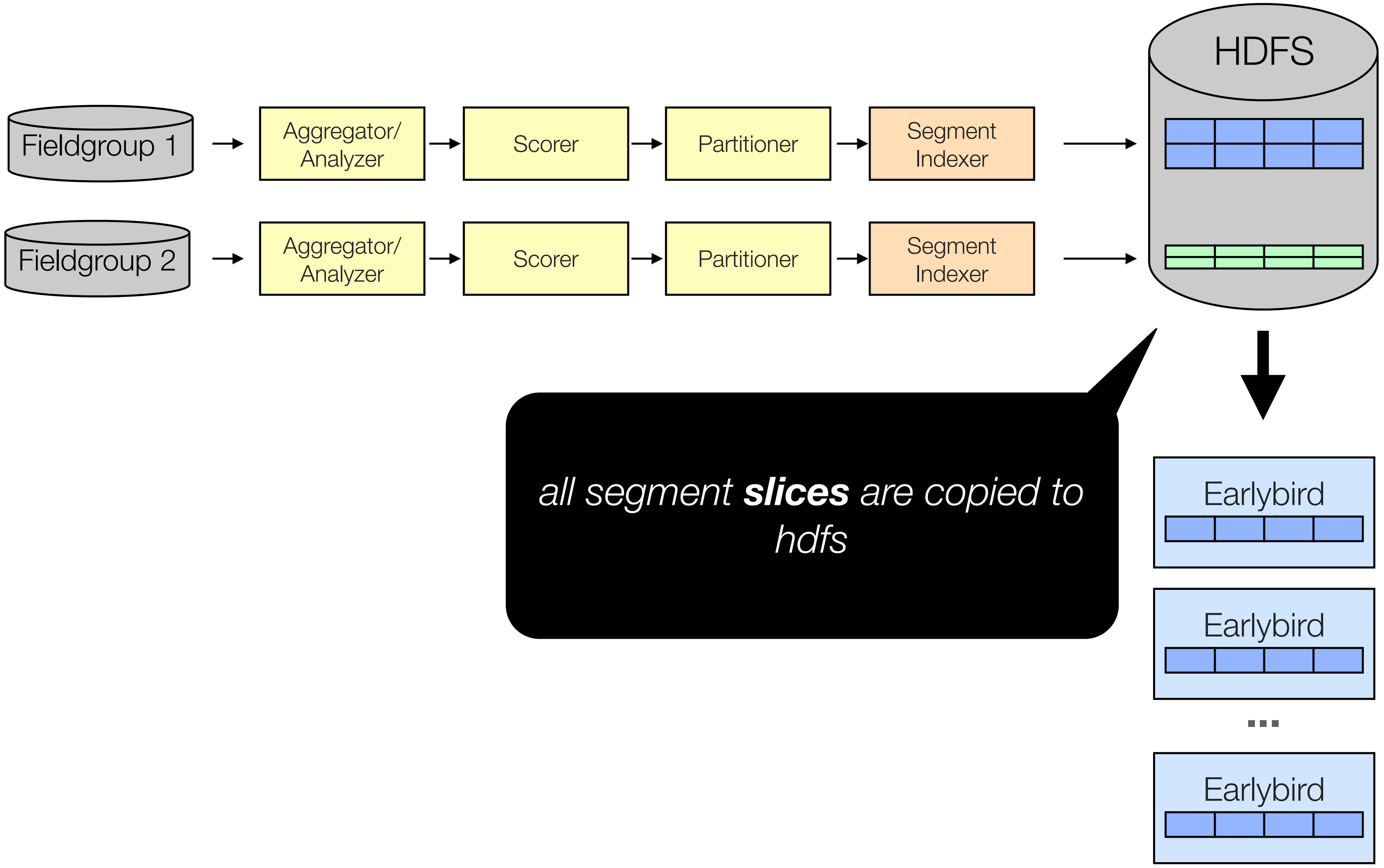


Parallel Ingestion pipeline



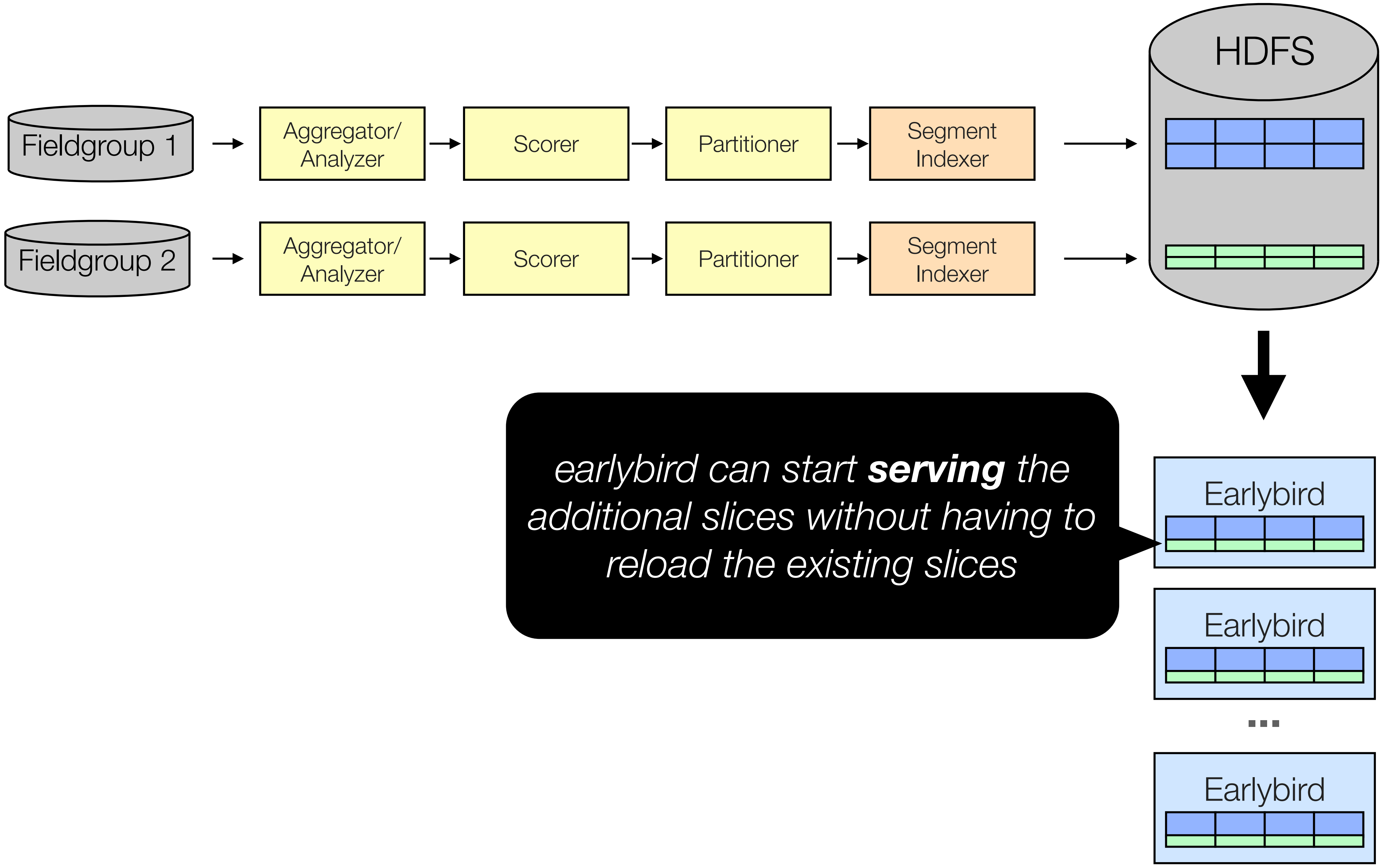
*a **parallel** pipeline processes a separate **group of fields***

Parallel Ingestion pipeline



all segment ***slices*** *are copied to hdfs*

Parallel Ingestion pipeline

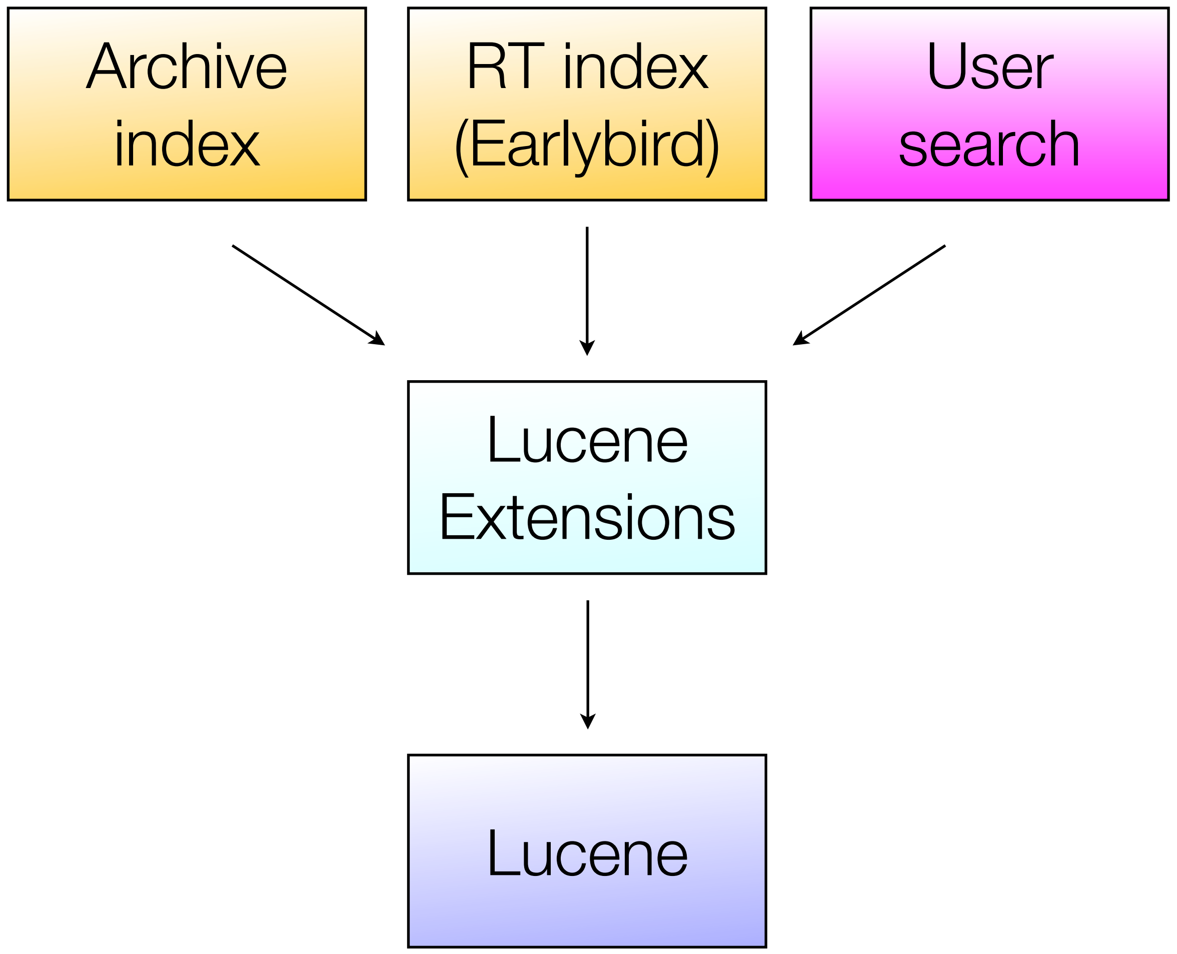


earlybird can start ***serving*** *the additional slices without having to reload the existing slices*

Outlook

- Parallel indexing pipelines for faster index manipulation
- **Domain-independent core indexing library that combines real-time and offline index technology**

Search Architecture



- New Lucene extension package
- This package is truly generic and has no dependency on an actual product/index
- It contains Twitter's extensions for real-time search, a thin segment management layer and other features

Lucene Extension Library

- Abstraction layer for Lucene index segments
- Real-time writer for in-memory index segments
- Schema-based Lucene document factory
- Real-time faceting

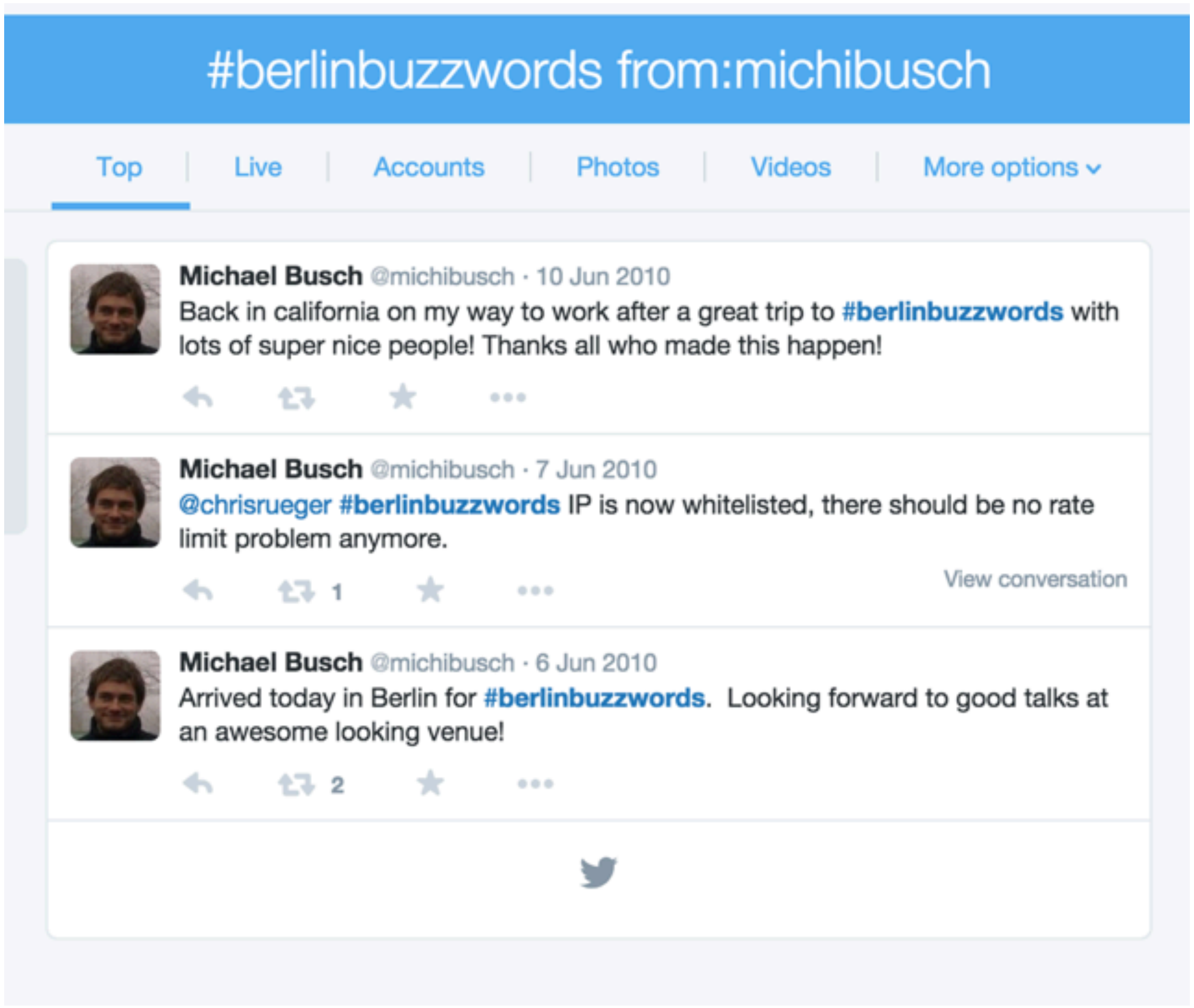
Lucene Extension Library

- API layer for Lucene segments
    - *IndexSegmentWriter
    - *IndexSegmentAtomicReader
- Two implementations
    - In-memory: RealtimeIndexSegmentWriter (and reader)
    - On-disk: LuceneIndexSegmentWriter (and reader)

Lucene Extension Library

- IndexSegments can be built ...
    - in realtime
    - on Mesos or Hadoop (Mapreduce)
    - locally on serving machines
- Cluster-management code that deals with IndexSegments
    - Share segments across serving machines using HDFS
    - Can rebuild segments (e.g. to upgrade Lucene version, change data schema, etc.)

Demo



Michael Busch @michibusch · 10 Jun 2010
Back in california on my way to work after a great trip to **#berlinbuzzwords** with lots of super nice people! Thanks all who made this happen!

Michael Busch @michibusch · 7 Jun 2010
@chrisrueger **#berlinbuzzwords** IP is now whitelisted, there should be no rate limit problem anymore.
1
View conversation

Michael Busch @michibusch · 6 Jun 2010
Arrived today in Berlin for **#berlinbuzzwords**. Looking forward to good talks at an awesome looking venue!
2