Routing Trillions of Events Per Day @Twitter

#ApacheBigData 2017

Lohit VijayaRenu & Gary Steelman
@lohitvijayarenu @efsie

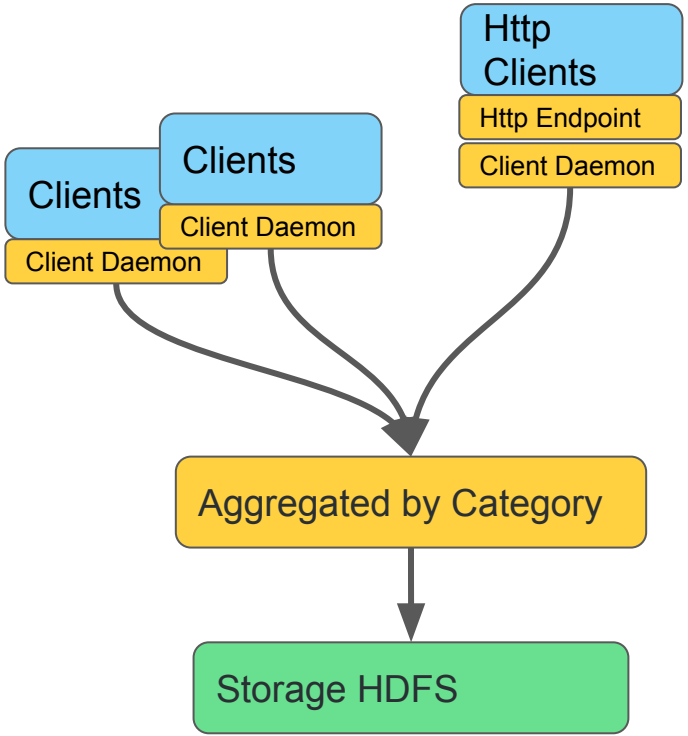
In this talk

1. Event Logs at Twitter
2. Log Collection
3. Log Processing
4. Log Replication
5. The Future
6. Questions

Overview

Life of an Event

- Clients log events specifying a Category name. Eg ads_view, login_event ...
- Events are grouped together across all clients into the Category
- Events are stored on Hadoop Distributed File System, bucketed every hour into separate directories
  - /logs/ads_view/2017/05/01/23
  - /logs/login_event/2017/05/01/23



Event Log Stats

>1T

Trillion Events a Day

Across millions of clients

~3PB

of Data a Day

Incoming uncompressed

<1500

Nodes

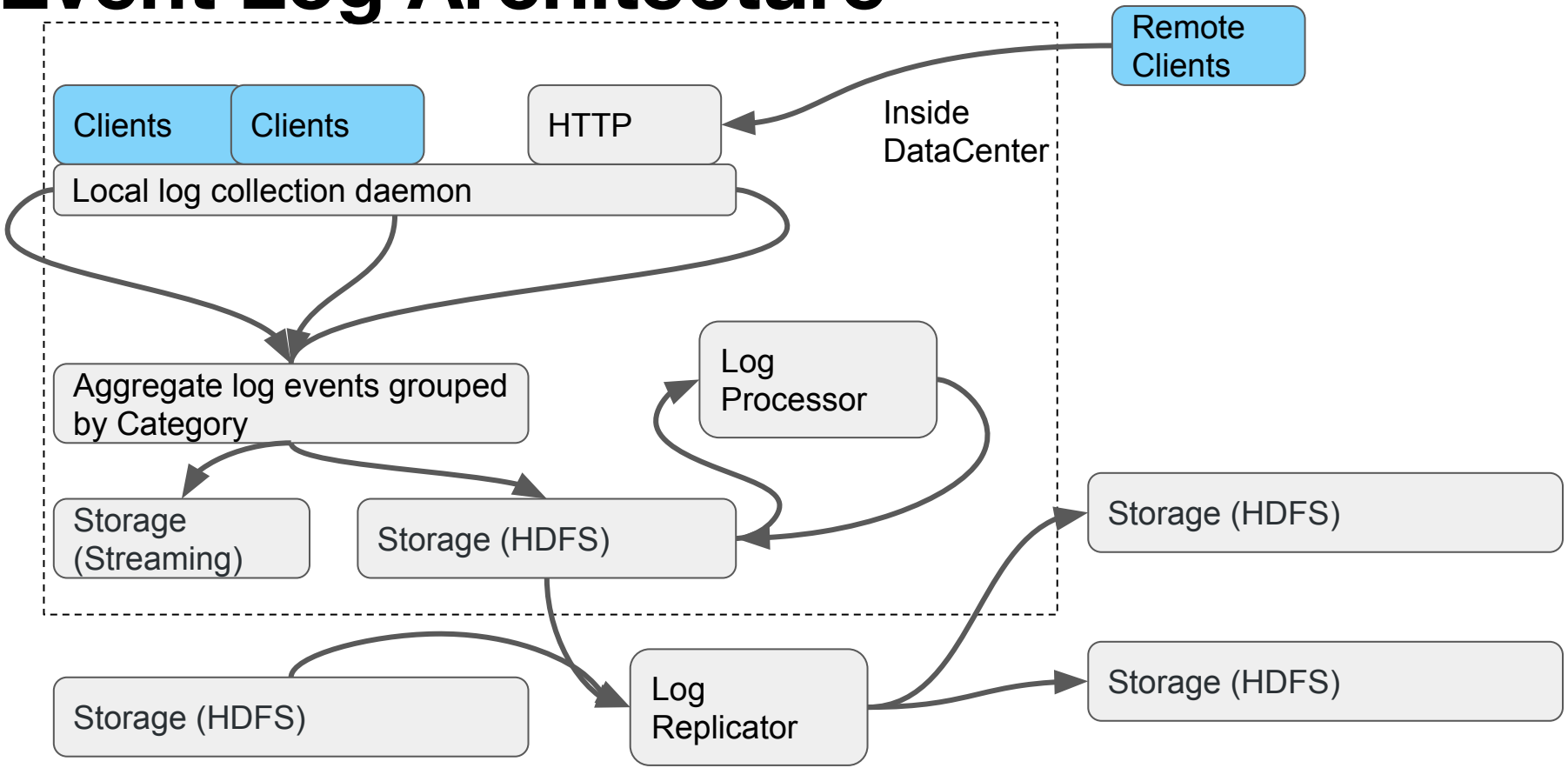
Collocated with HDFS datanodes

>600

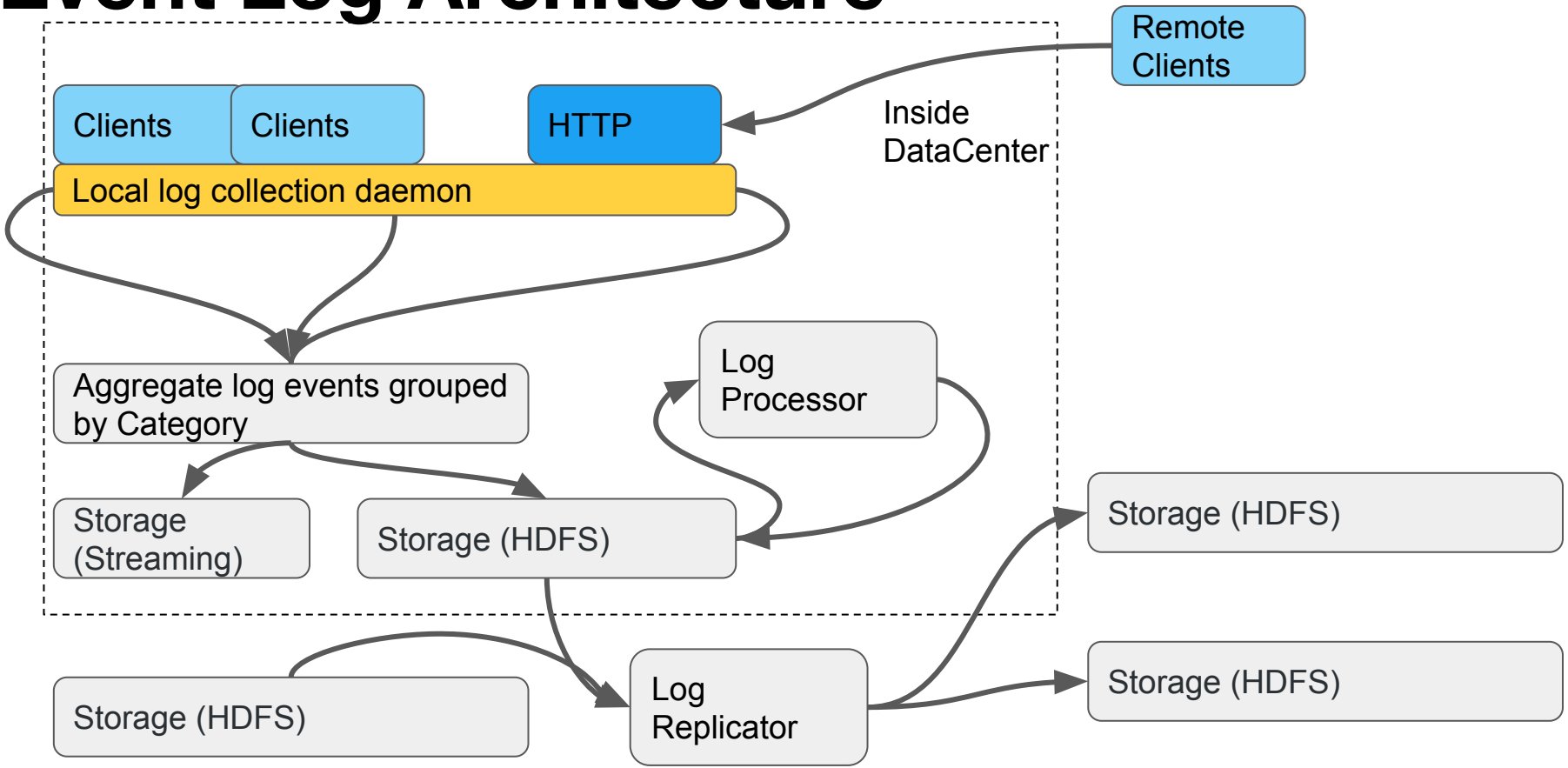
Categories

Event groups by category

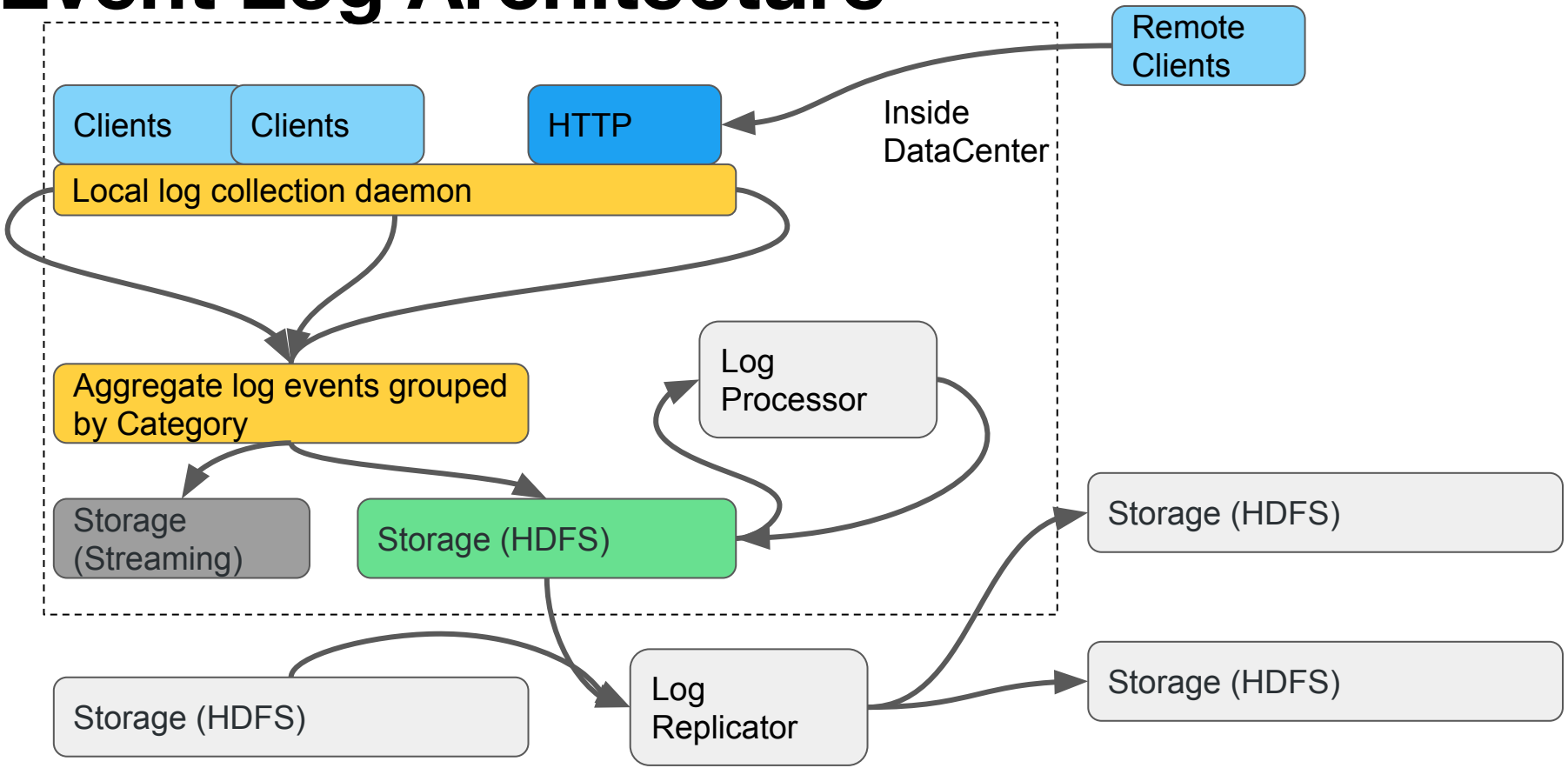
Event Log Architecture



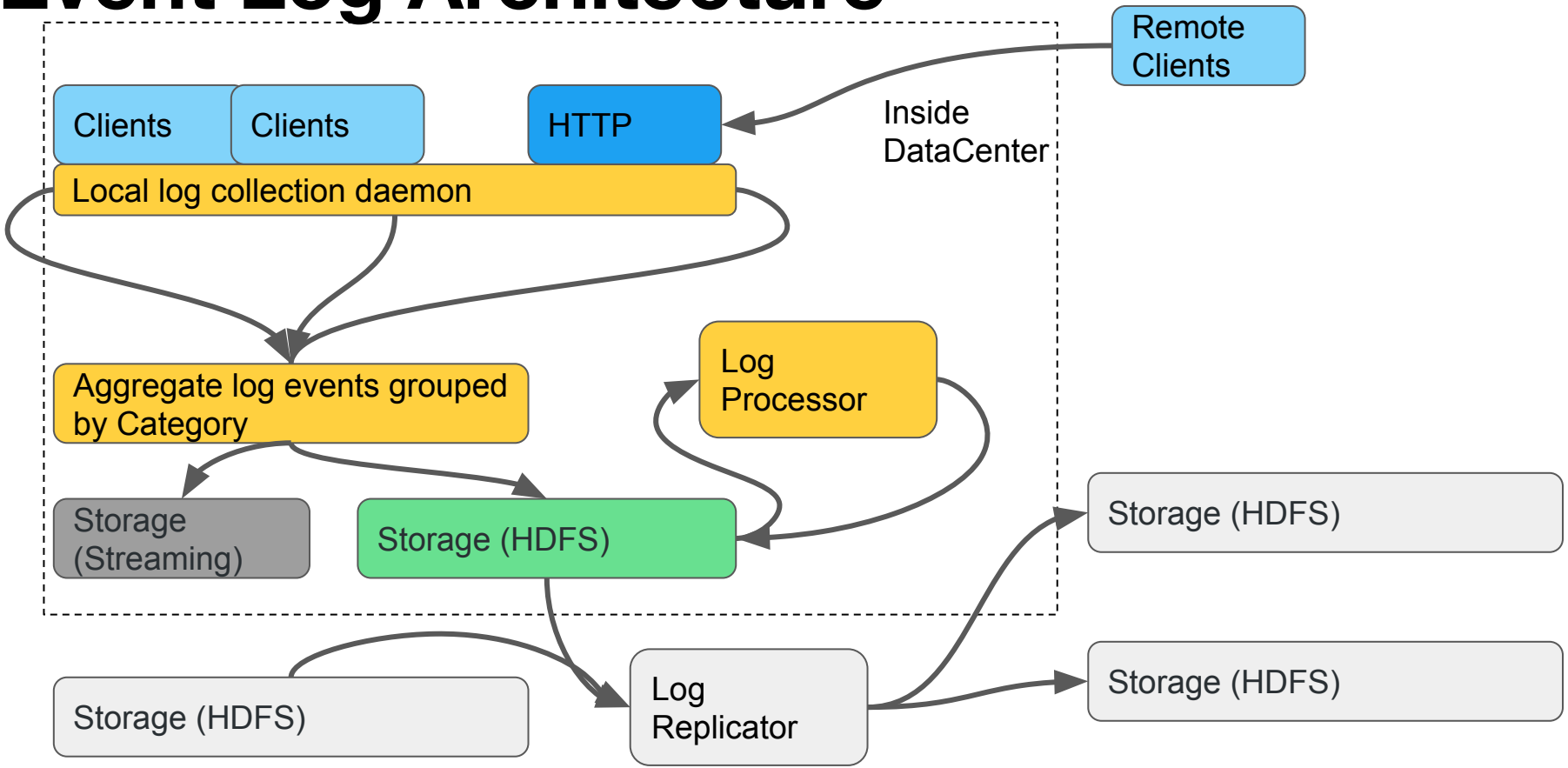
Event Log Architecture



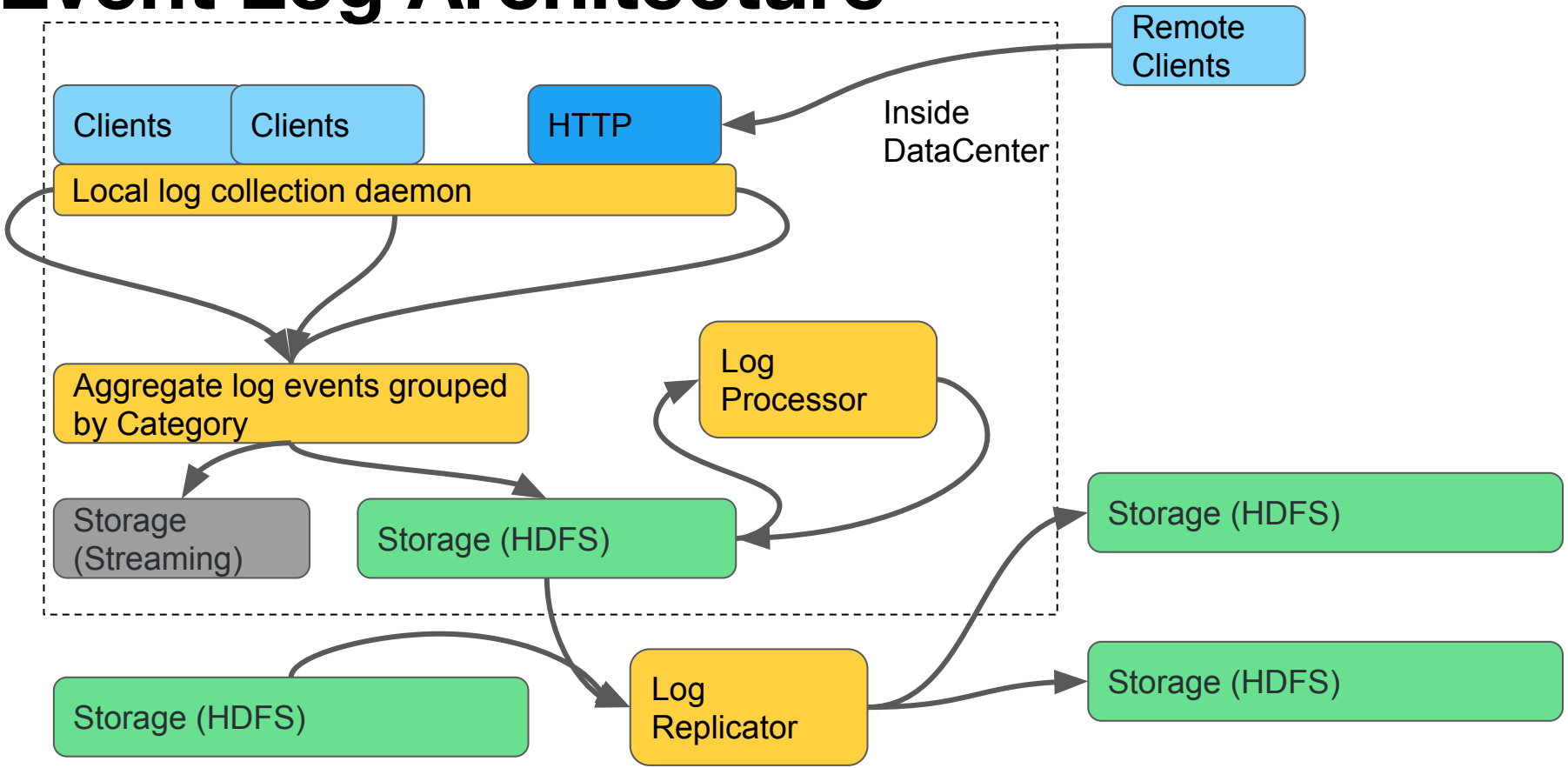
Event Log Architecture



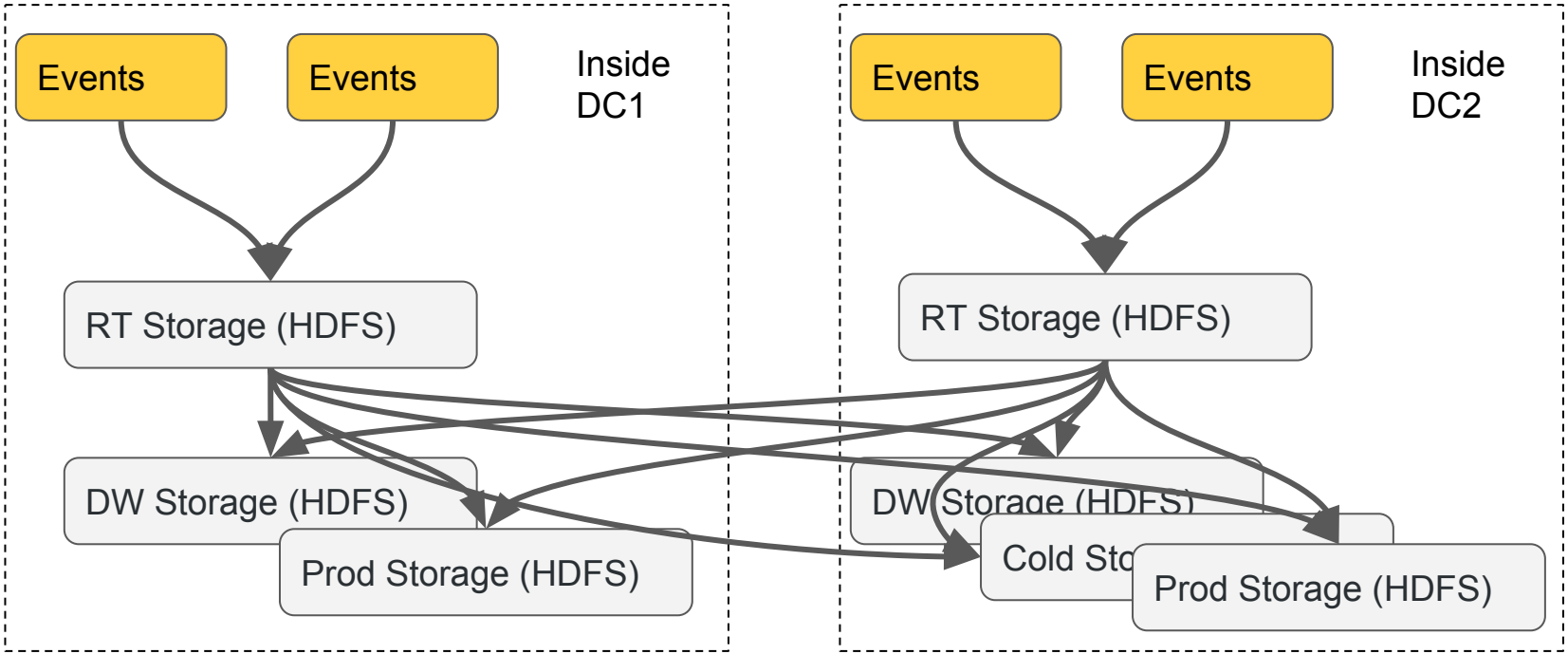
Event Log Architecture



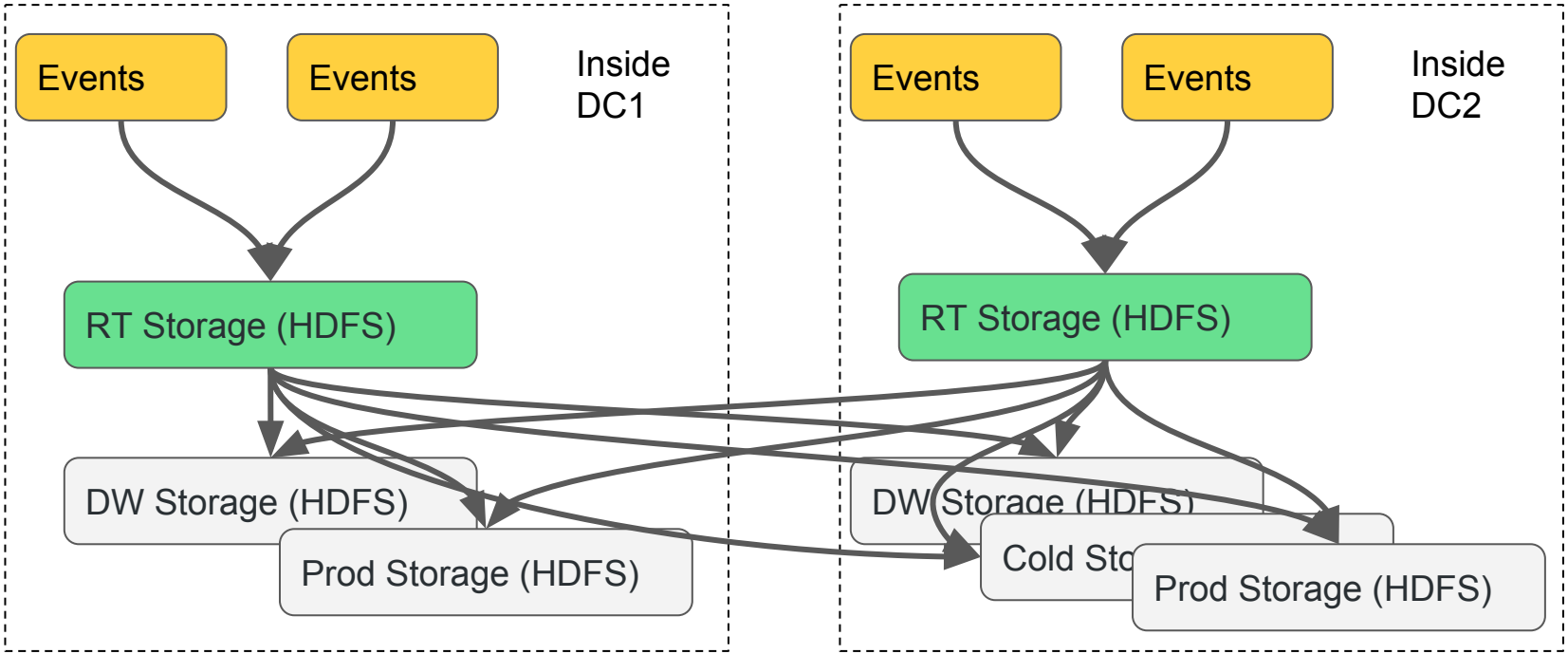
Event Log Architecture



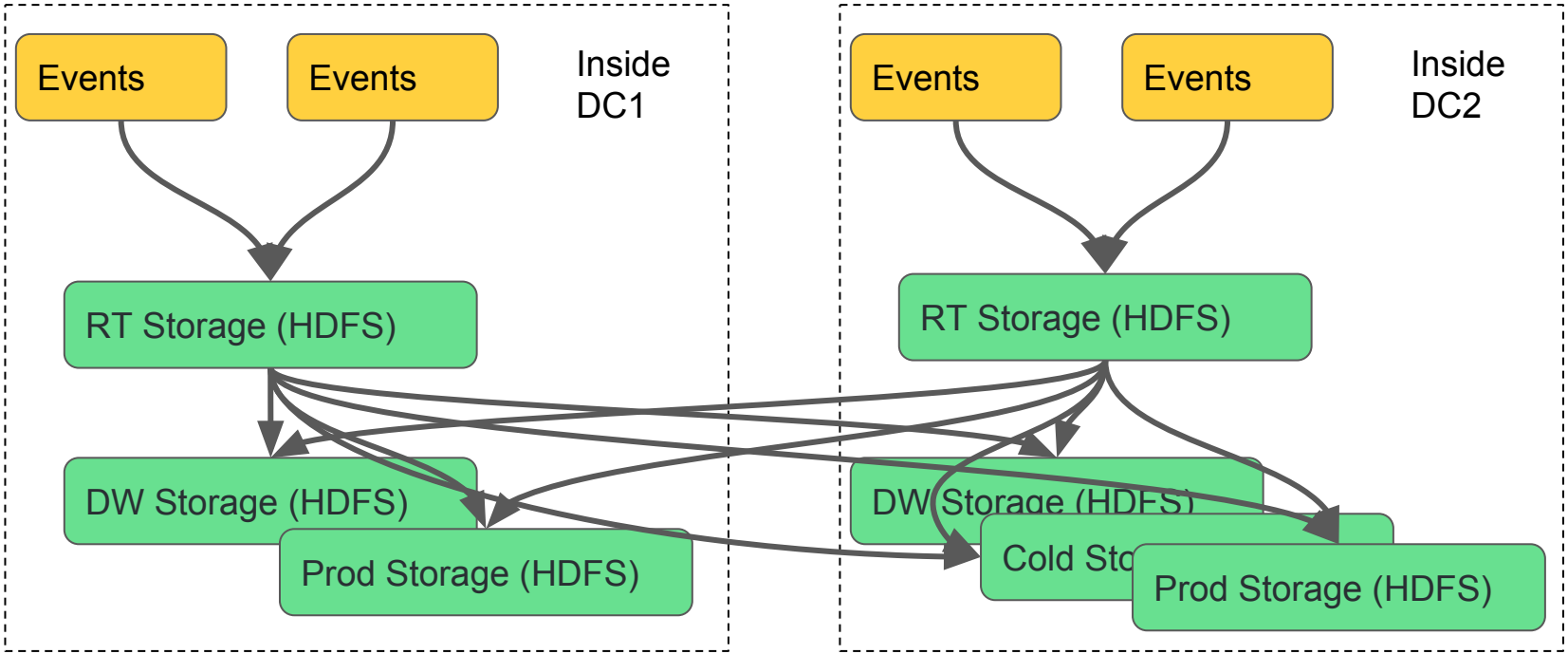
Event Log Architecture



Event Log Architecture

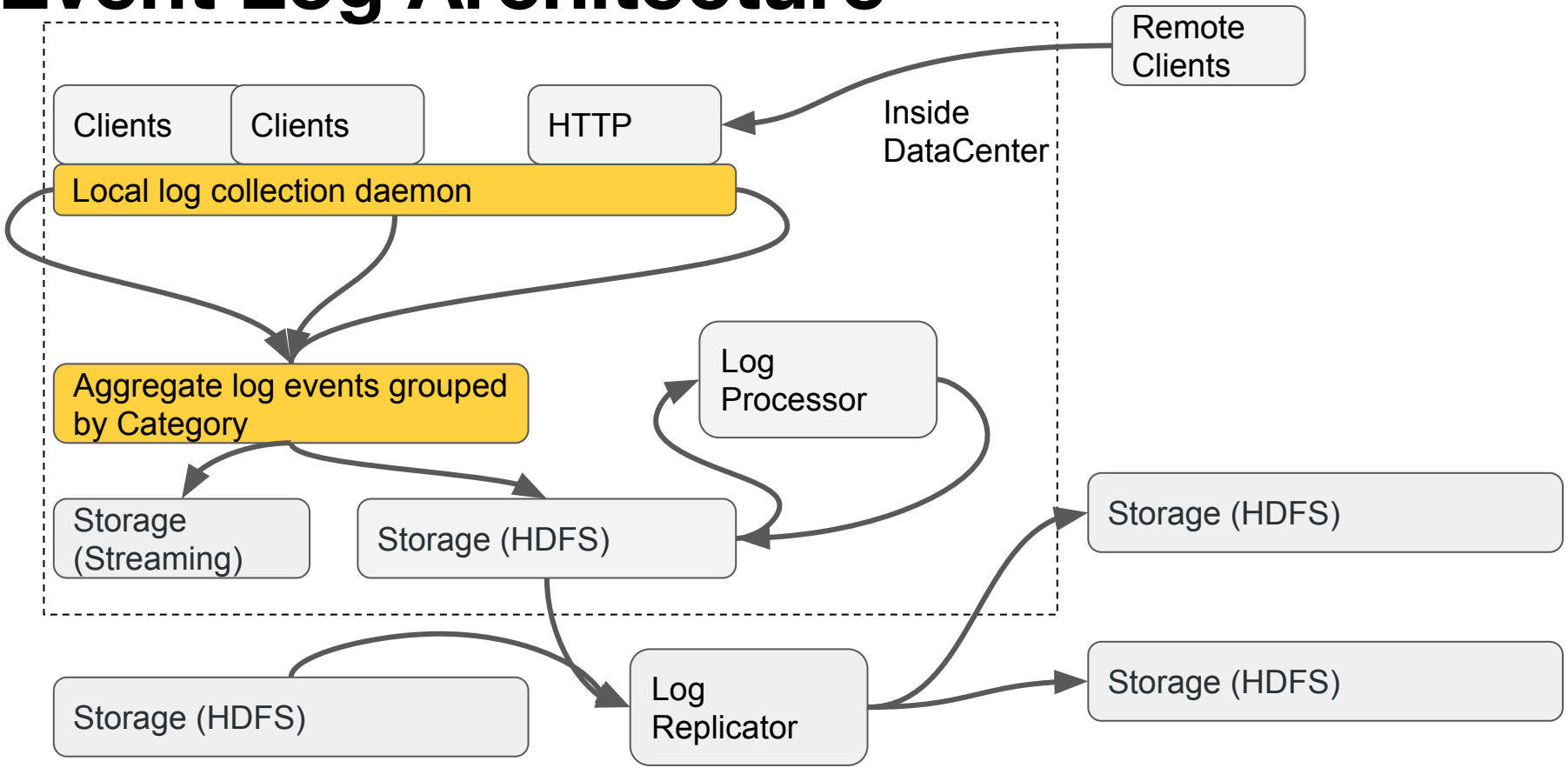


Event Log Architecture

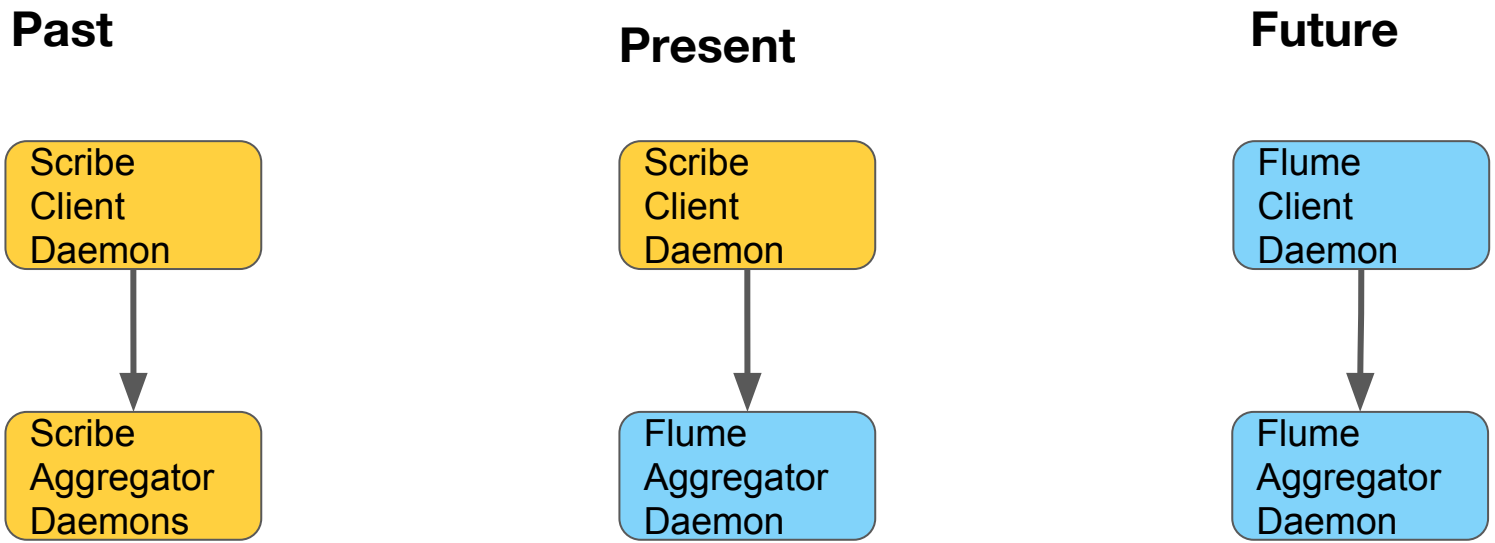


Collection

Event Log Architecture



Event Collection Overview



Event Collection

Past

Challenges with Scribe

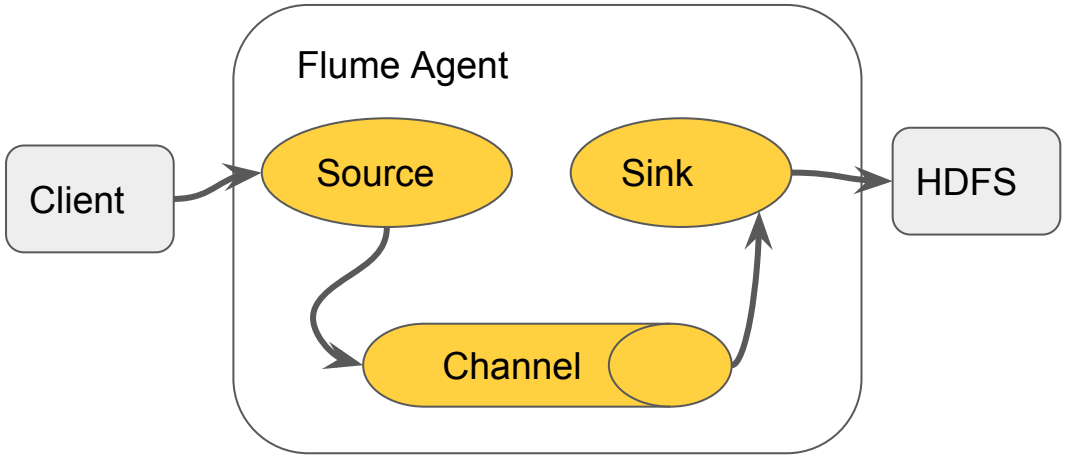
- Too many open file handles to HDFS
  - 600 categories x 1500 aggregators x 6 per hour =~ 5.4M files per hour
- High IO wait on DataNodes at scale
- Max limit on throughput per aggregator
- Difficult to track message drops
- No longer active open source development

Event Collection

Present

Apache Flume

- Well defined interfaces
- Open source
- Concept of transactions
- Existing implementations of interfaces

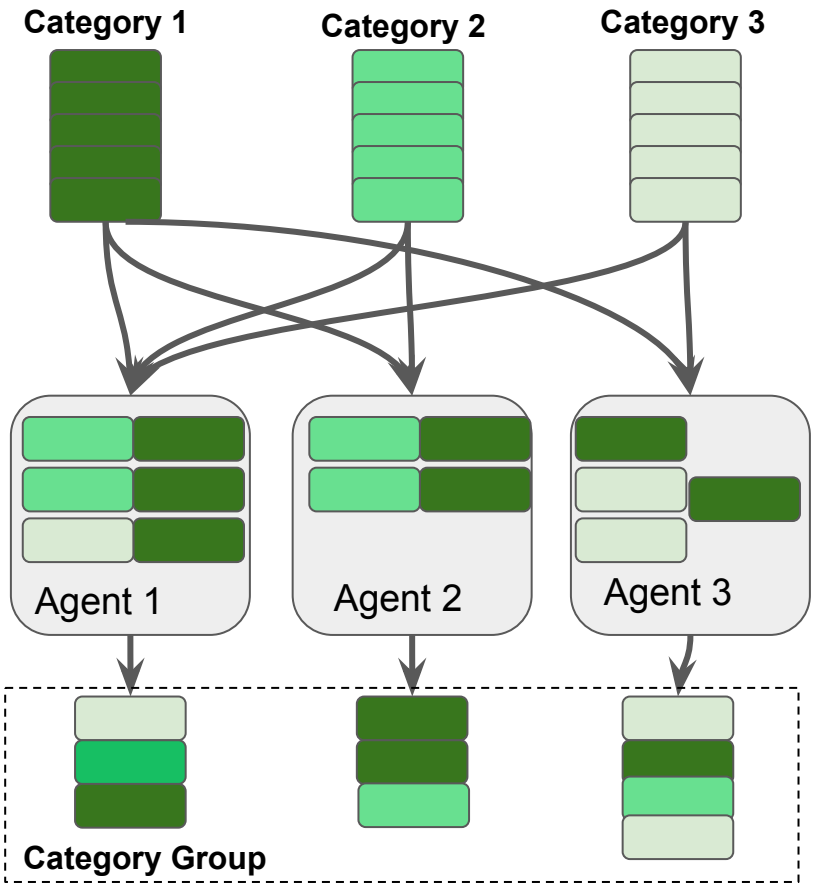


Event Collection

Present

Category Group

- Combine multiple related categories into a category group
- Provide different properties per group
- Contains multiple events to generate fewer combined sequence files

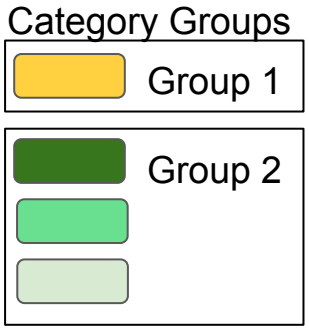


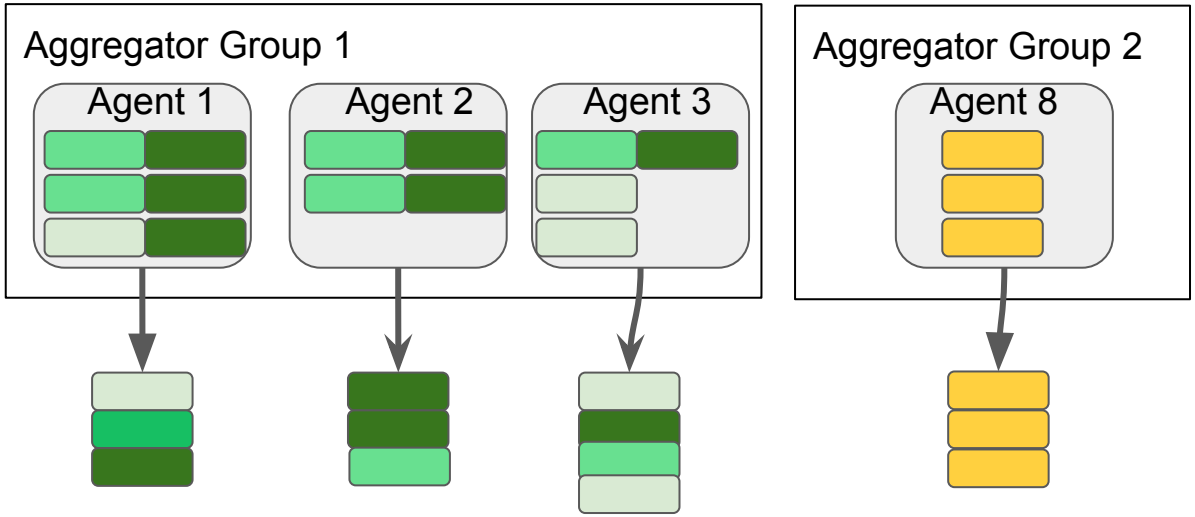
Event Collection

Present

Aggregator Group

- **A set of aggregators hosting same set of category groups**
- **Easy to manage group of aggregators hosting subset of categories**





Event Collection

Present

Flume features to support groups

- Extend Interceptor to multiplex events into groups
- Implement Memory Channel Group to have separate memory channel per category group
- ZooKeeper registration per category group for service discovery
- Metrics for category groups

Event Collection

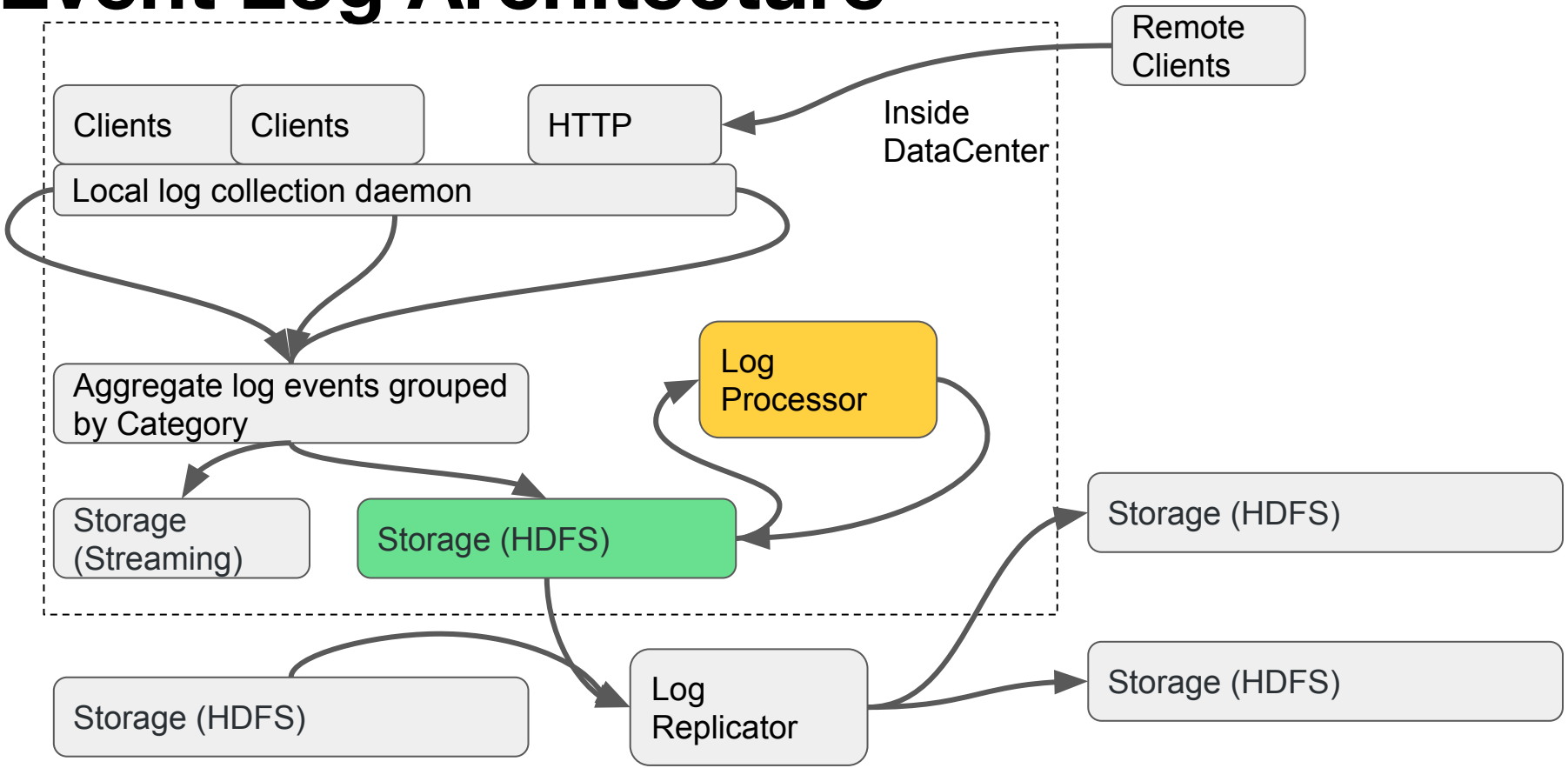
Present

Flume performance improvements

- HDFSEventSink batching increased (5x) throughput reducing spikes on memory channel
- Implement buffering in HDFSEventSink instead of using SpillableMemoryChannel
- Stream events close to network speed


Processing

Event Log Architecture



Log Processor Stats

Processing Trillion Events per Day

8

Wall Clock Hours

To process one day of data

>1PB

Data per Day

Output of cleaned, compressed, consolidated, and converted

20-50%

Disk Space

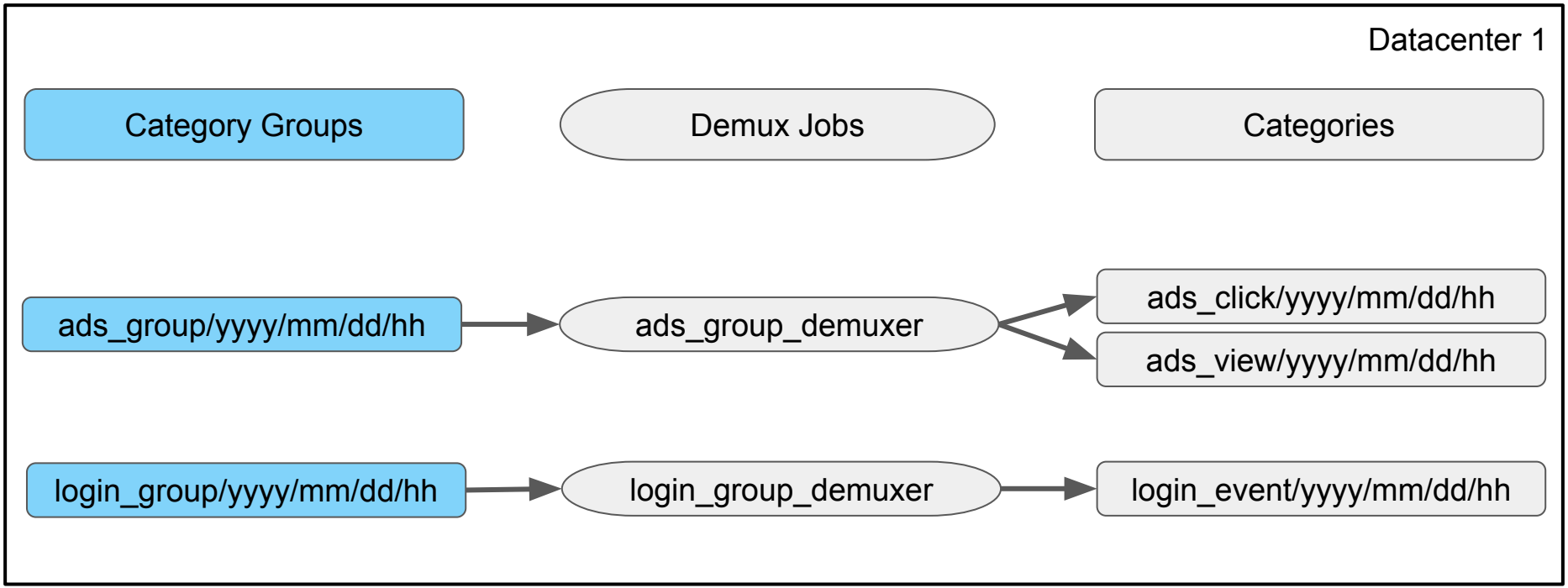
Saved by processing Flume sequence files

Log Processor Needs

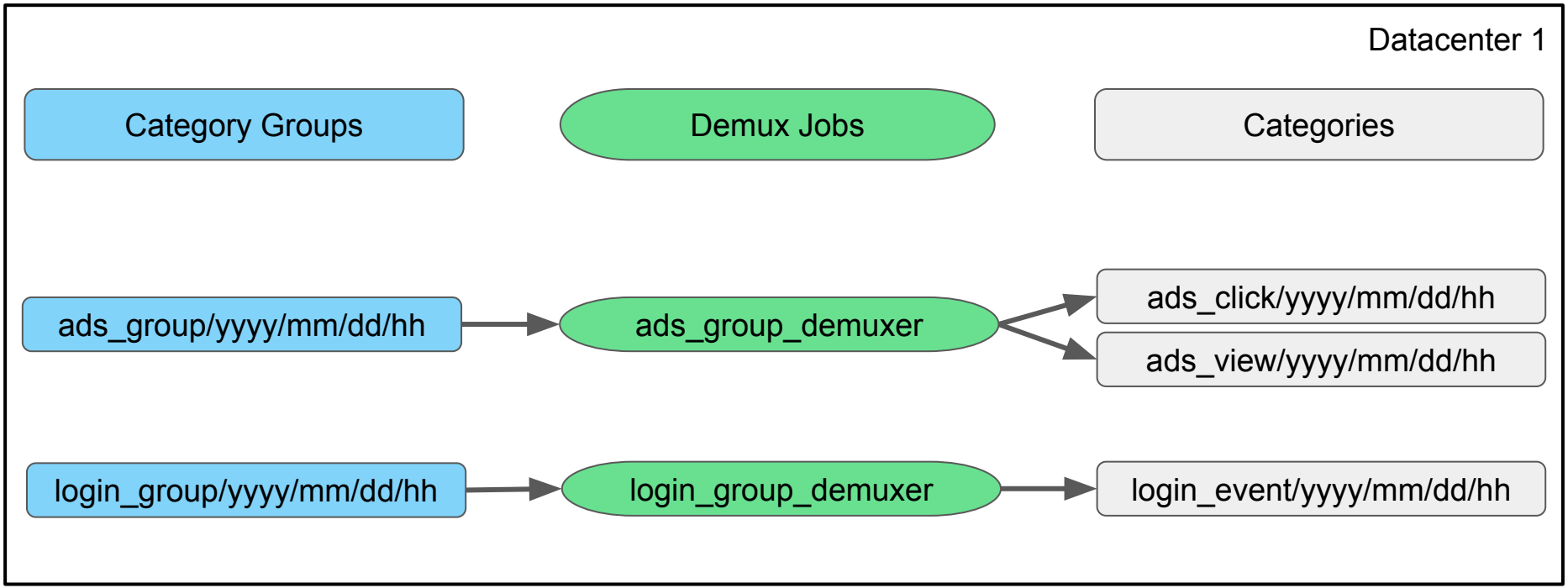
Processing Trillion Events per Day

- Make processing log data easier for analytics teams
- Disk space is at a premium on analytics clusters
- Still too many files cause increased pressure on the NameNode
- Log data is read many times and different teams all perform the same pre-processing steps on the same data sets

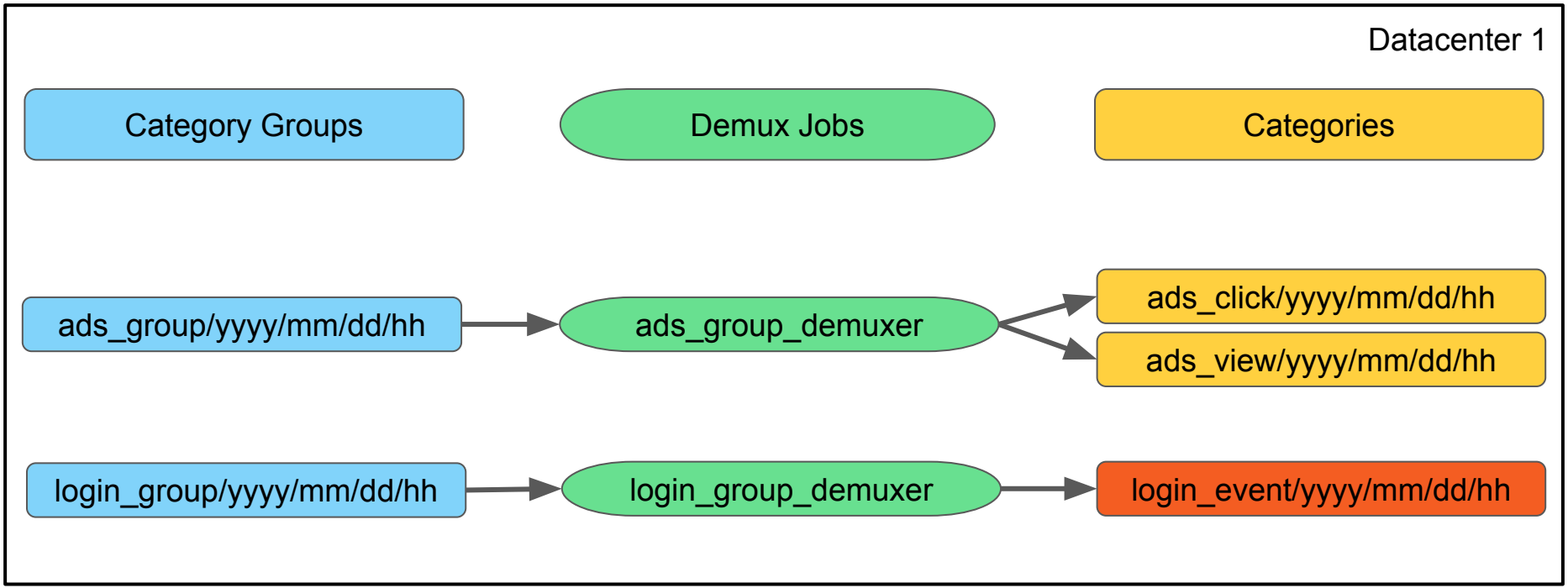
Log Processor Steps



Log Processor Steps



Log Processor Steps



Log Processor Steps

1 **Decode**
Base64 encoding from logged data

2 **Demux**
Category groups into individual categories for easier consumption by analytics teams

3 **Clean**
Corrupt, empty, or invalid records so data sets are more reliable

4 **Compress**
Logged data to the highest level to save disk space. From LZO level 3 to LZO level 7

5 **Consolidate**
Small files to reduce pressure on the NameNode

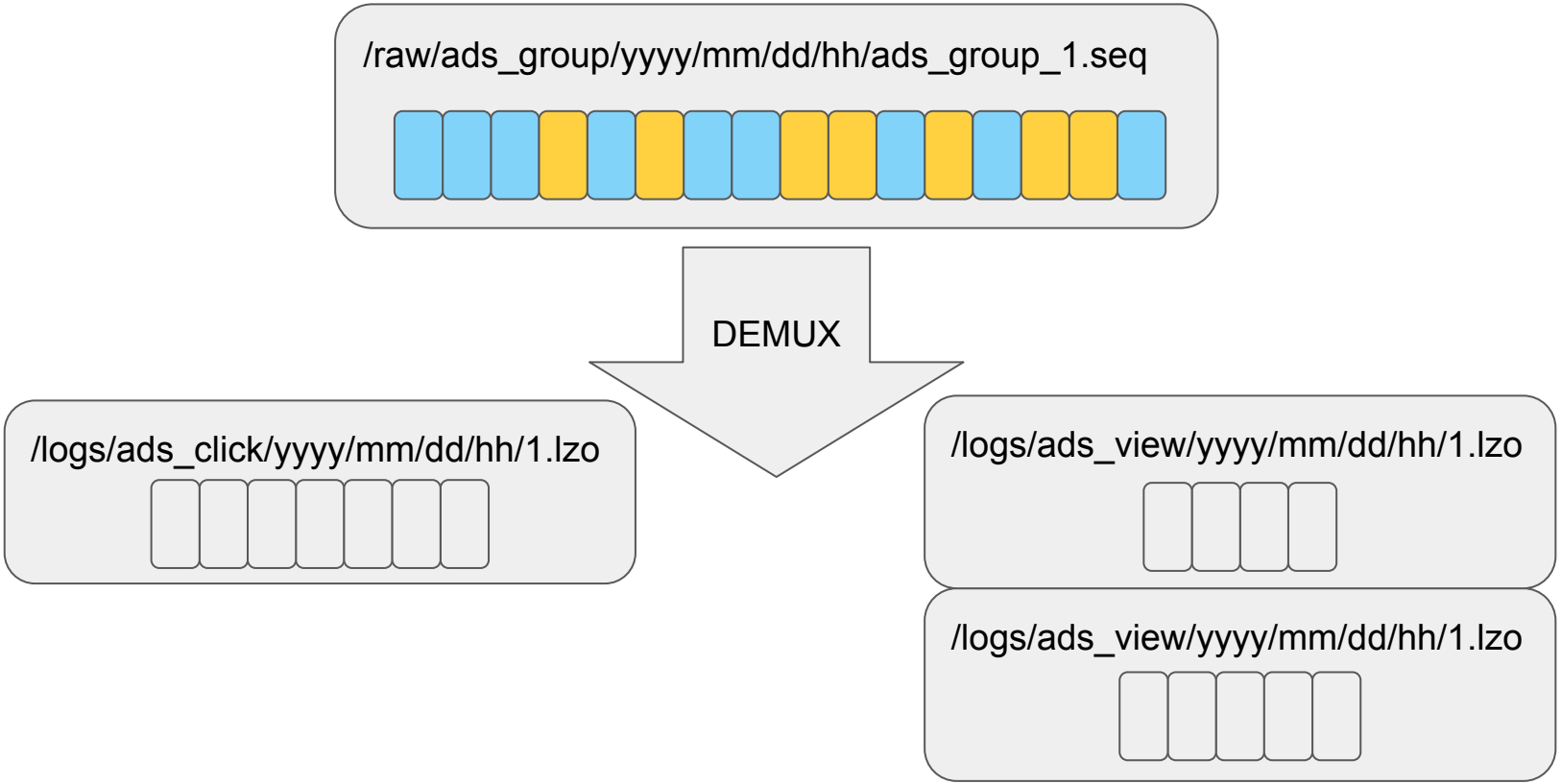
6 **Convert**
Some categories into Parquet for fastest use in ad-hoc exploratory tools

Why Base64 Decoding?

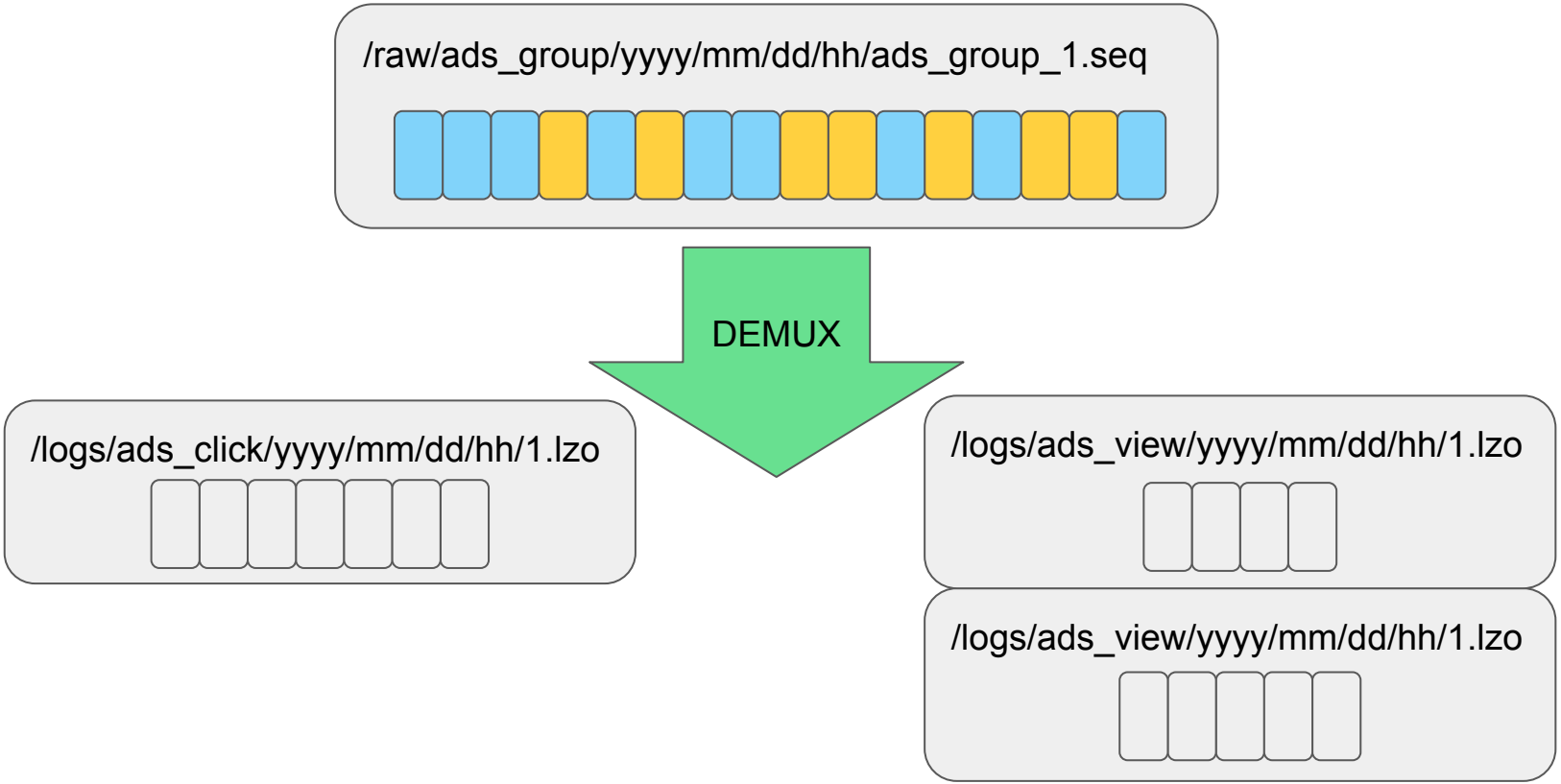
Legacy Choices

- Scribe's contract amounts to sending a binary blob to a port
- Scribe used new line characters to delimit records in a binary blob batch of records
- Valid records may include new line characters
- Scribe base64 encoded received binary blobs to avoid confusion with record delimiter
- Base 64 encoding is no longer necessary because we have moved to one serialized Thrift object per binary blob

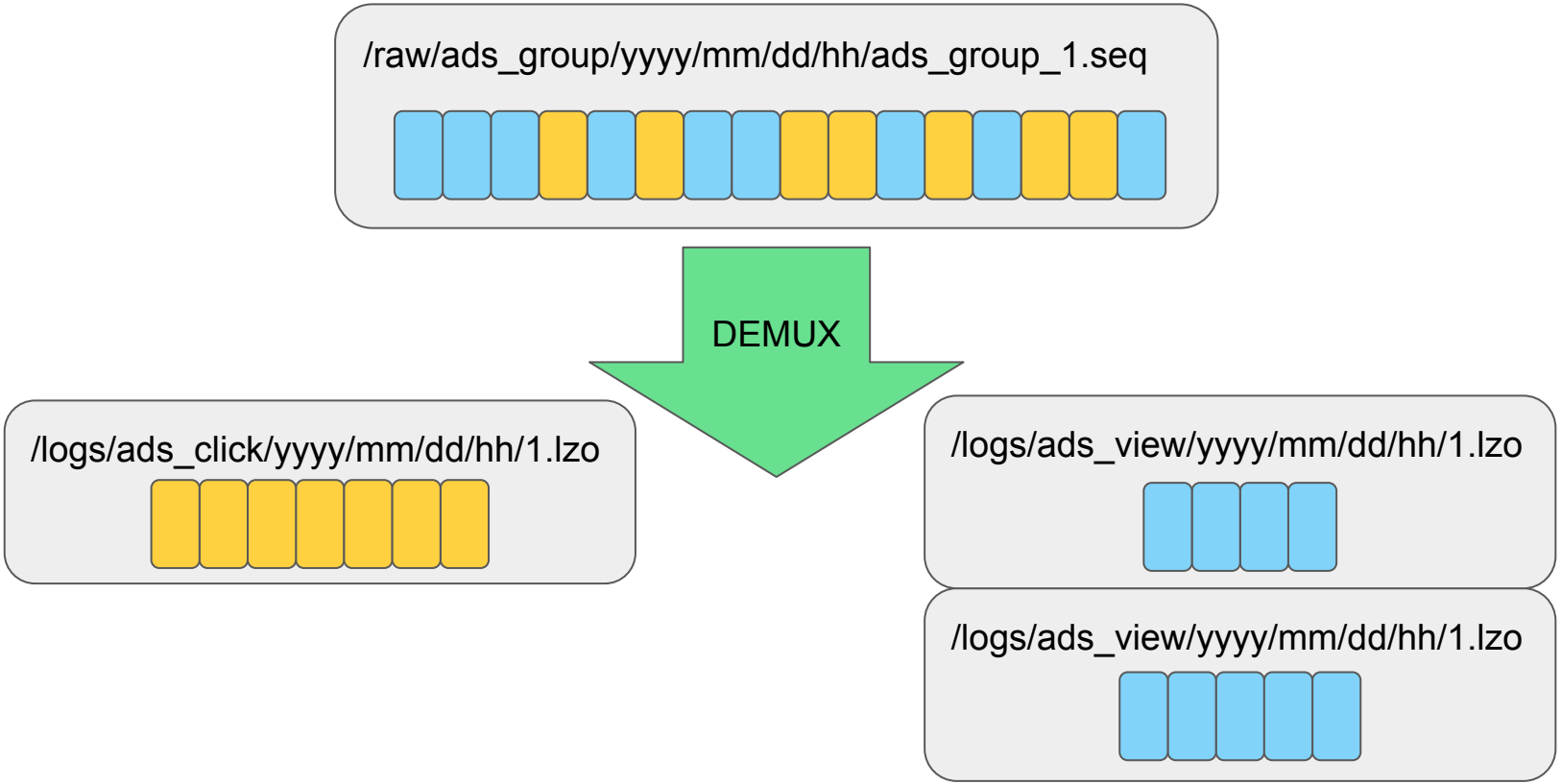
Log Demux Visual



Log Demux Visual



Log Demux Visual



Log Processor Daemon

- One log processor daemon per RT Hadoop cluster, where Flume aggregates logs
- Primarily responsible for demuxing category groups out of the Flume sequence files
- The daemon schedules Tez jobs every hour for every category group in a thread pool
- Daemon atomically presents processed category instances so partial data can't be read
- Processing proceeds according to criticality of data or "tiers"

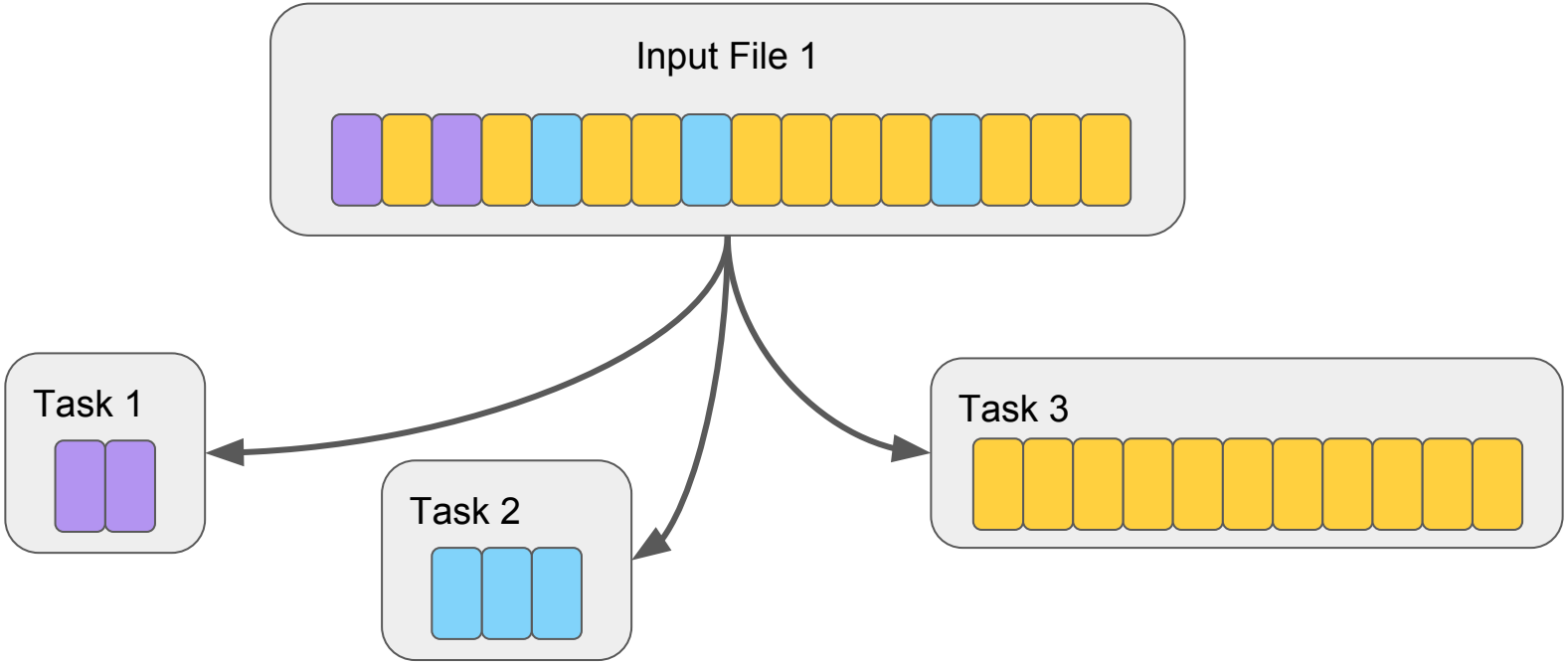
Why Tez?

- Some categories are significantly larger than other categories (KBs v TBs)
- MapReduce demux? Each reducer handles a single category
- Streaming demux? Each spout or channel handles a single category
- Massive skew in partitioning by category causes long running tasks which slows down job completion time
- Relatively well understood fault tolerance semantics similar to MapReduce, Spark, etc

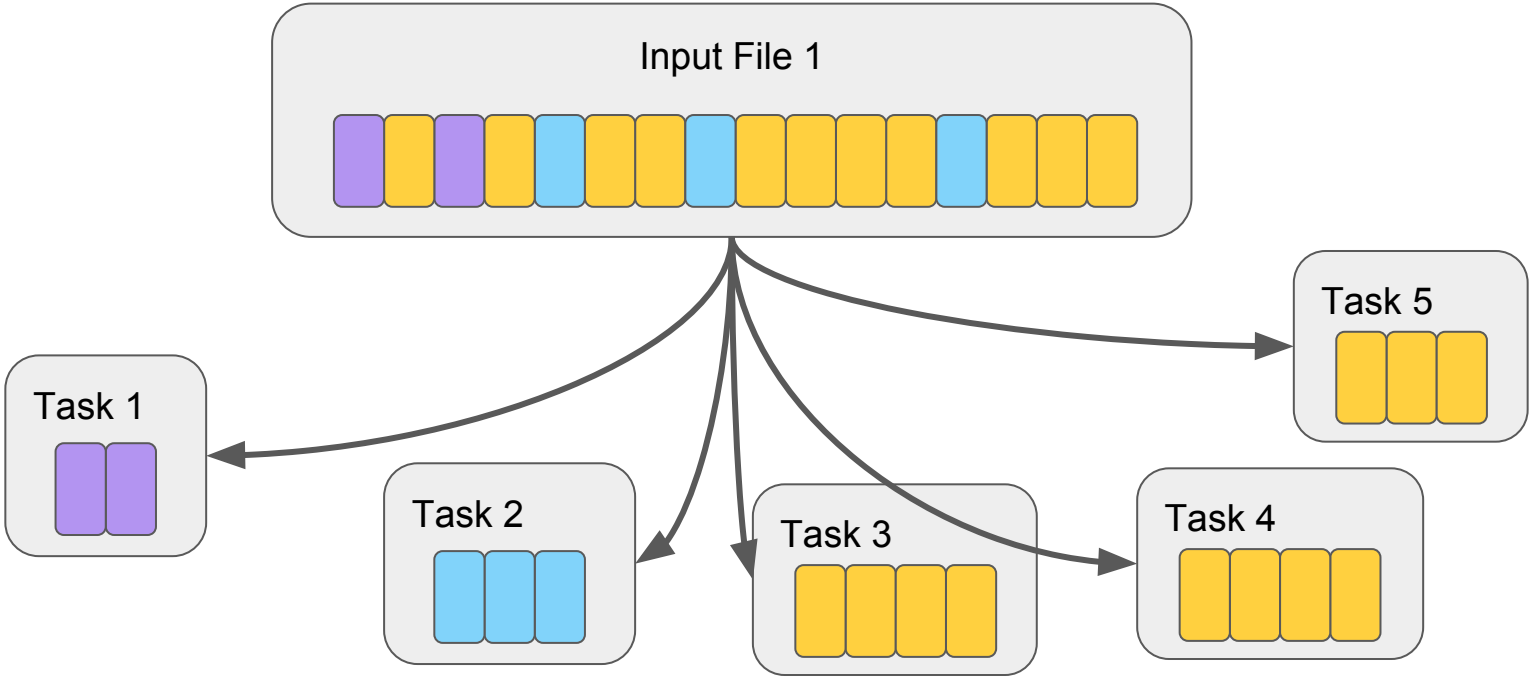
Dynamic Partitioning

- Tez's dynamic hash partitioner adjusts partitions at runtime if necessary, allowing large partitions to be further partitioned so multiple tasks process events for a single category one task
  - More info at TEZ-3209.
  - Thanks to team member Ming Ma for the contribution!
- Easier horizontal scaling simultaneously providing more predictable processing times

Typical Partitioning Visual

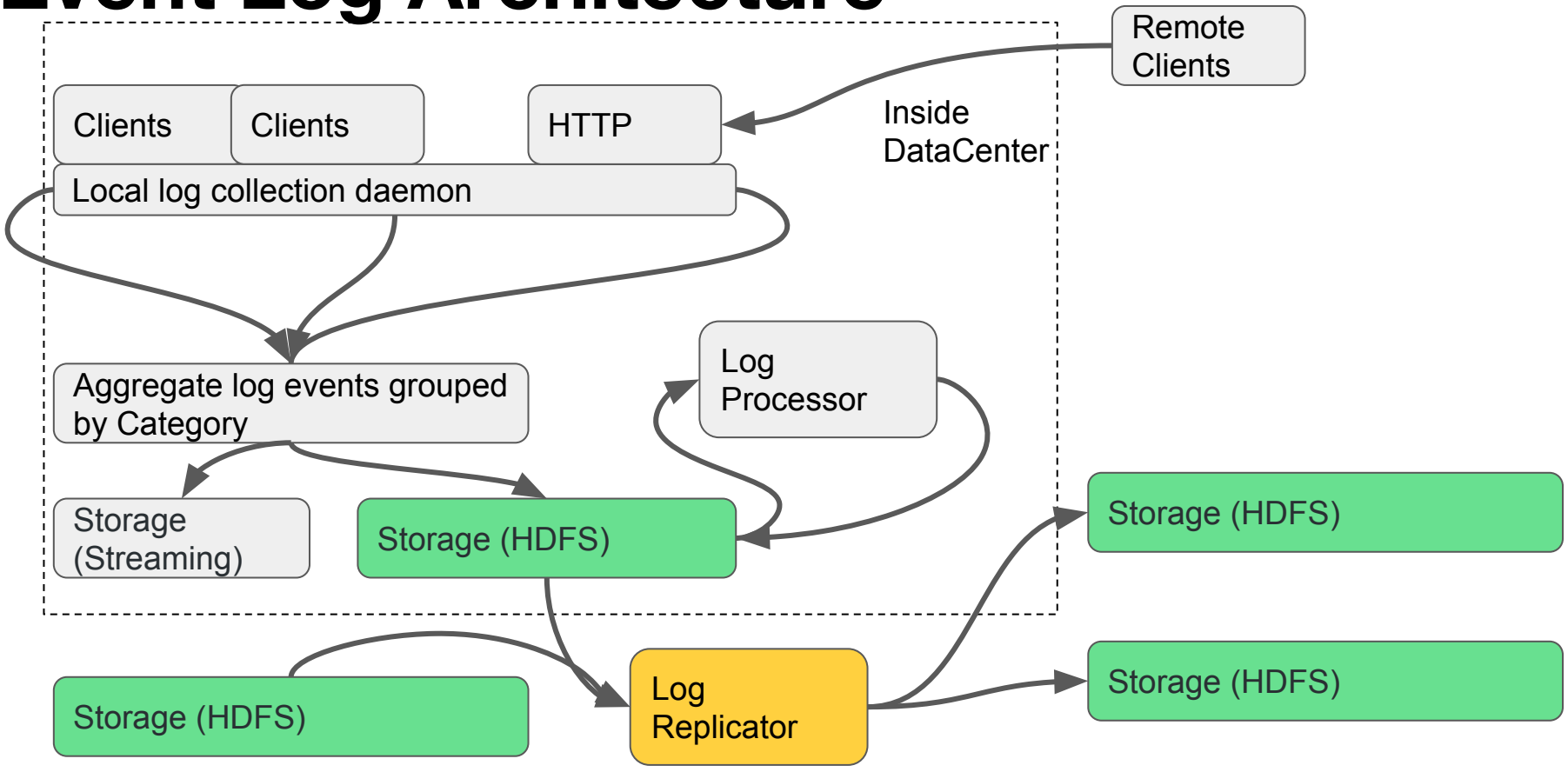


Hash Partitioning Visual




Replication

Event Log Architecture



Log Replication Stats

Replicating Trillion Events per Day

>24k

Copy Jobs per Day

Across all analytics clusters

>1PB

PB of Data per Day

Replicated to analytics clusters

~10

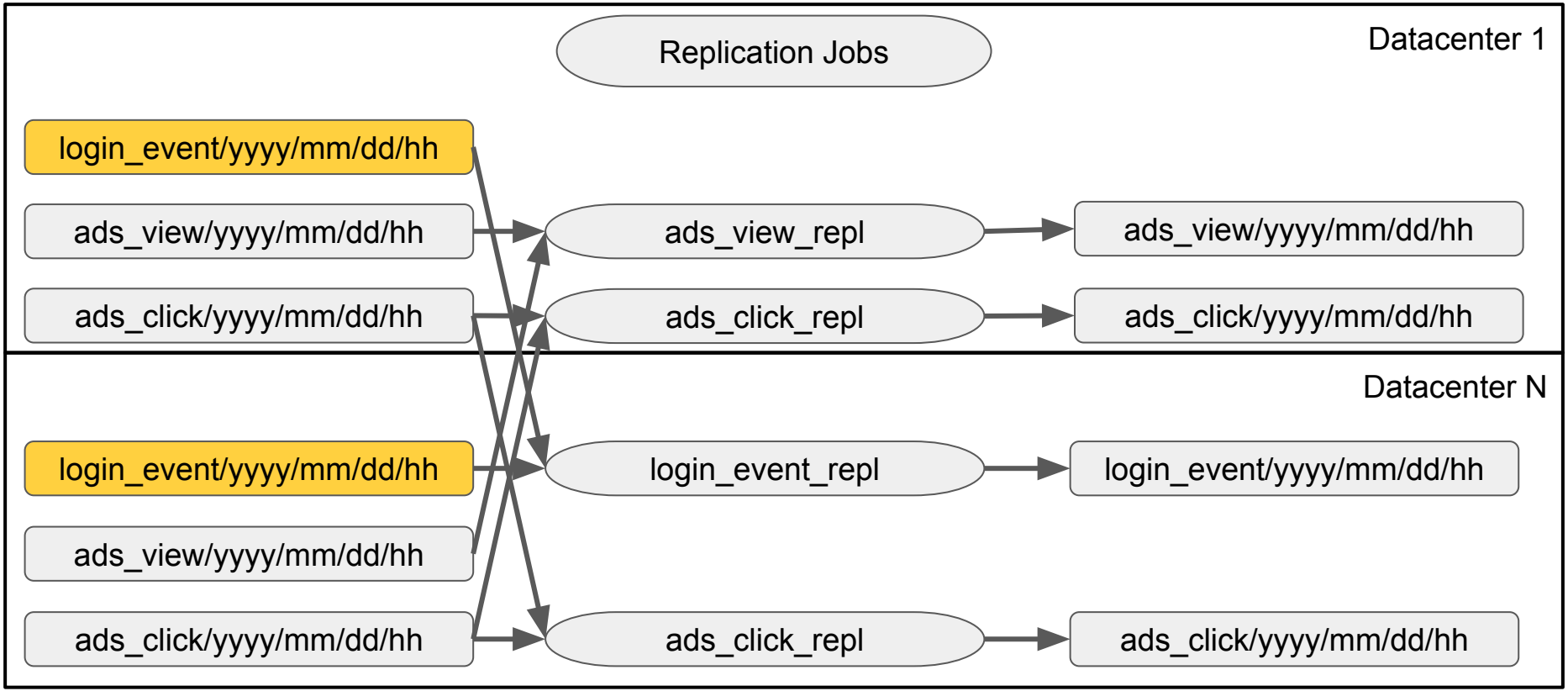
Analytics Clusters

Log Replication Needs

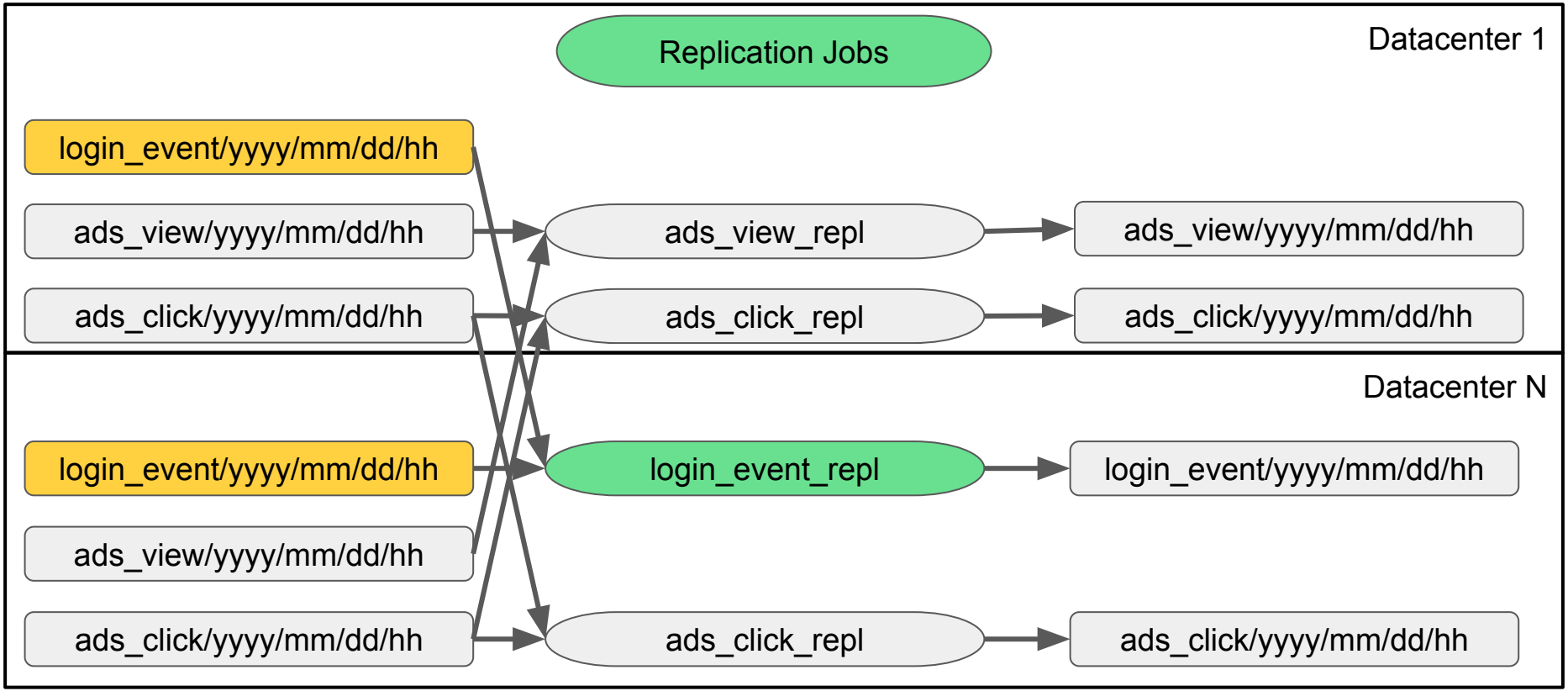
Processing Trillion Events per Day

- Collocate logs with compute and disk capacity for analytics teams
  - Cross-data center reads are incredibly expensive
  - Cross-rack reads within data center are still expensive
- Critical data set copies and backups in case of data center failures

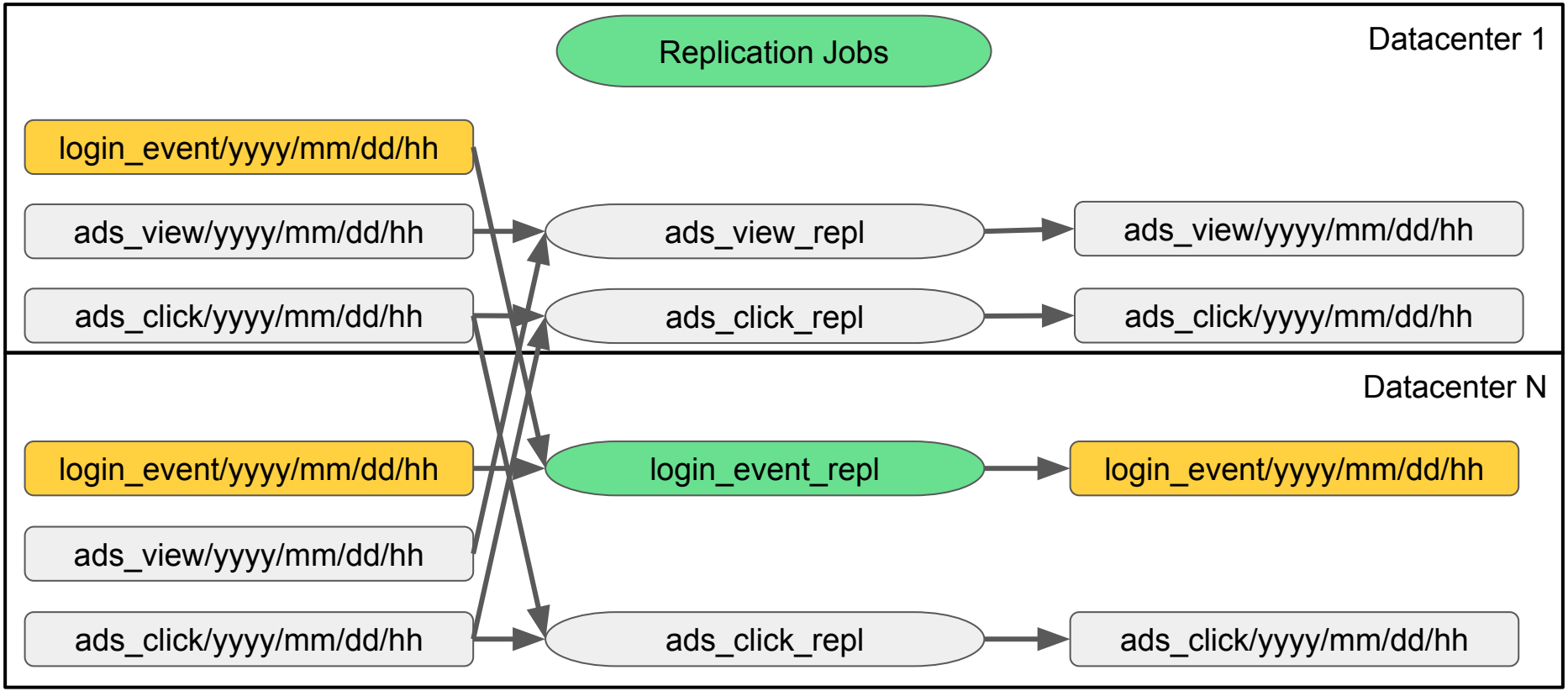
Log Replication Visual



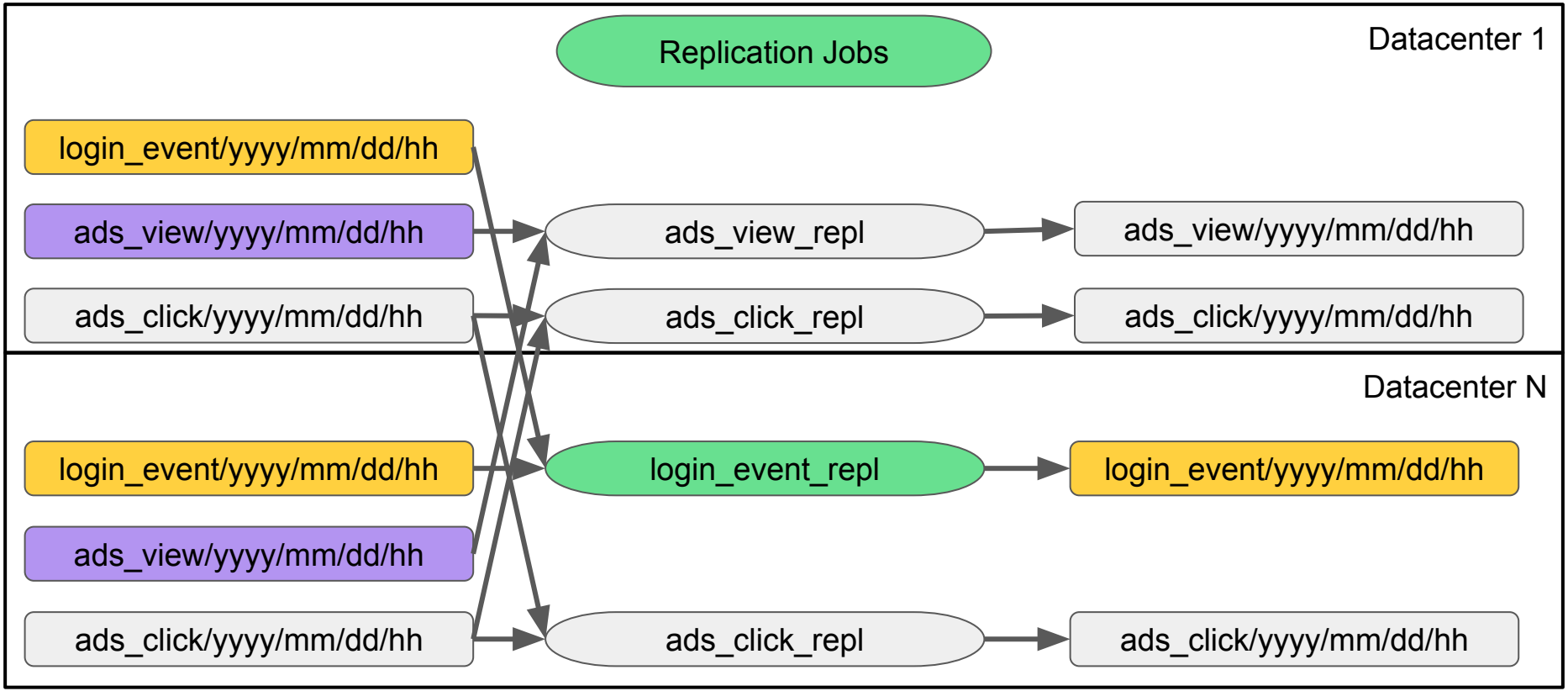
Log Replication Visual



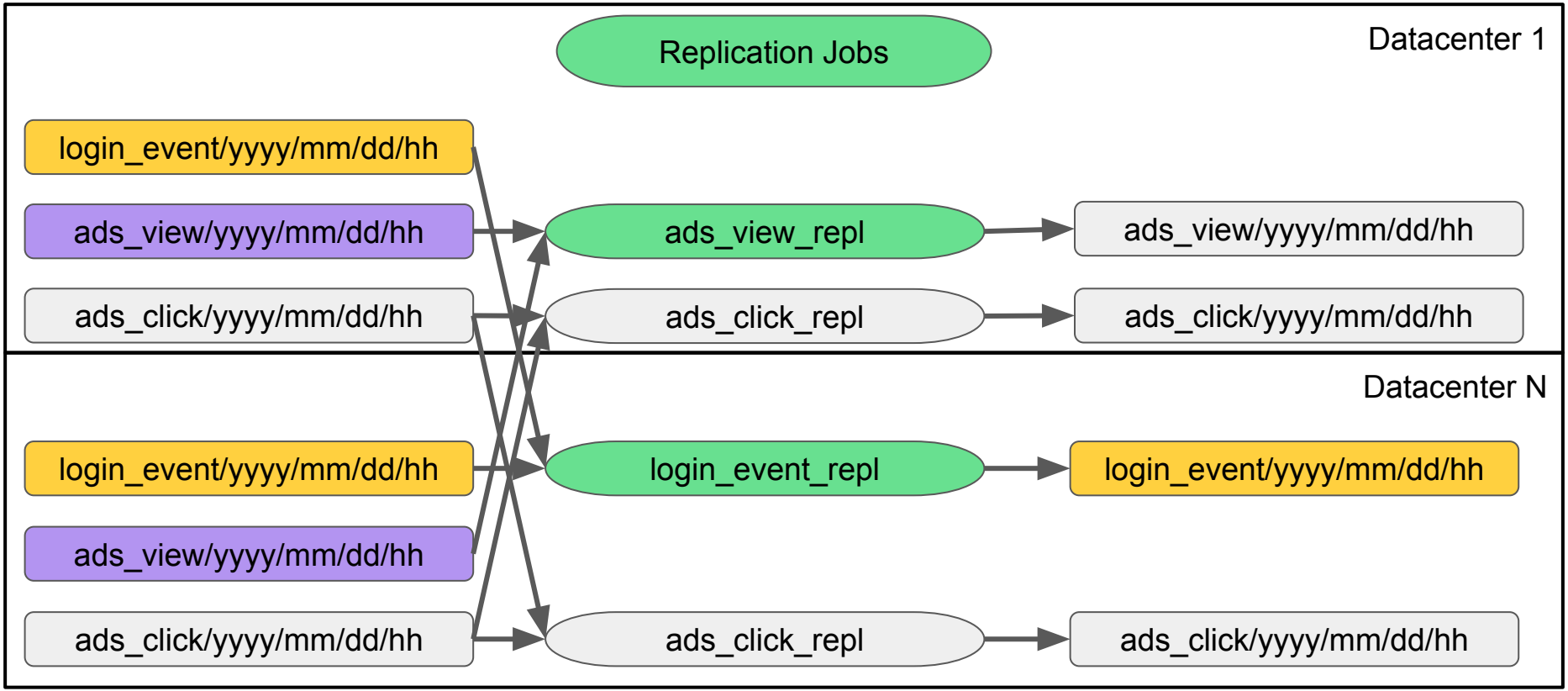
Log Replication Visual



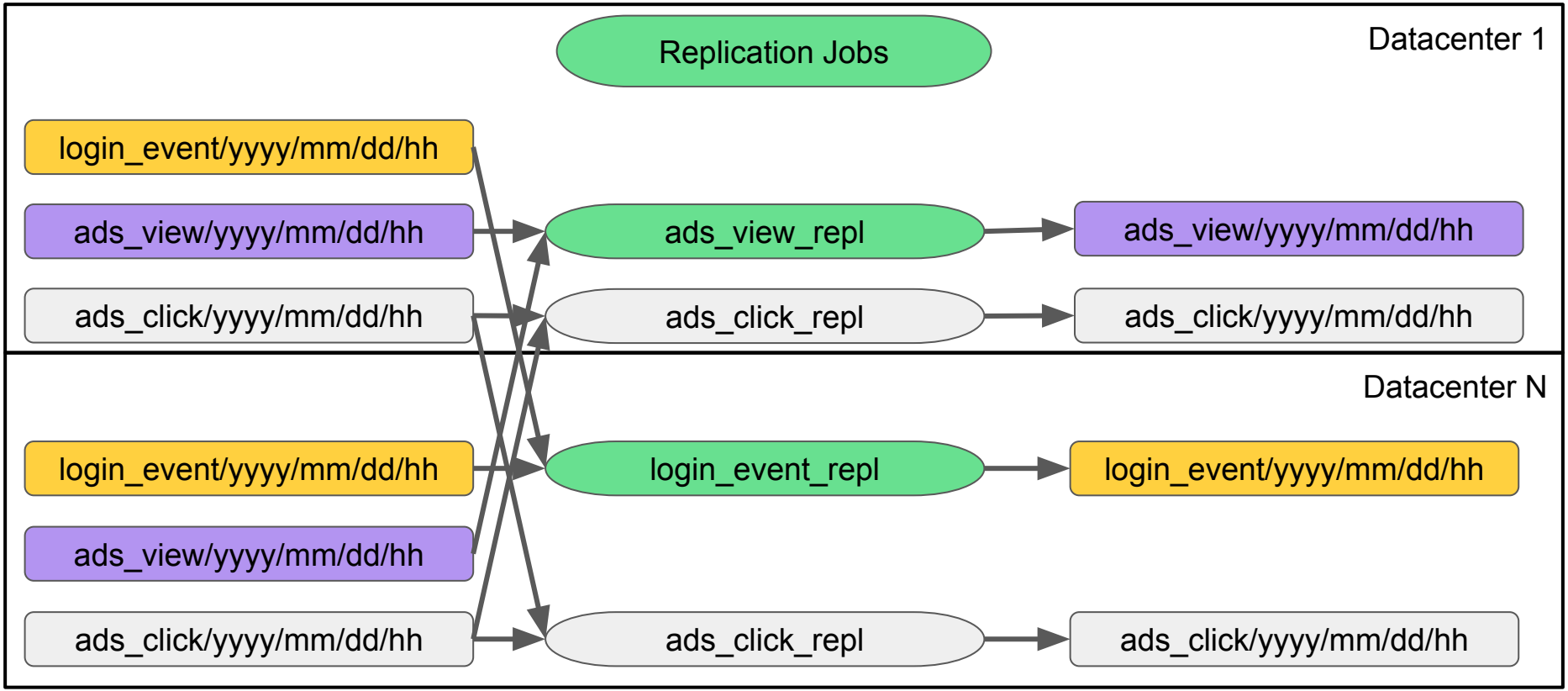
Log Replication Visual



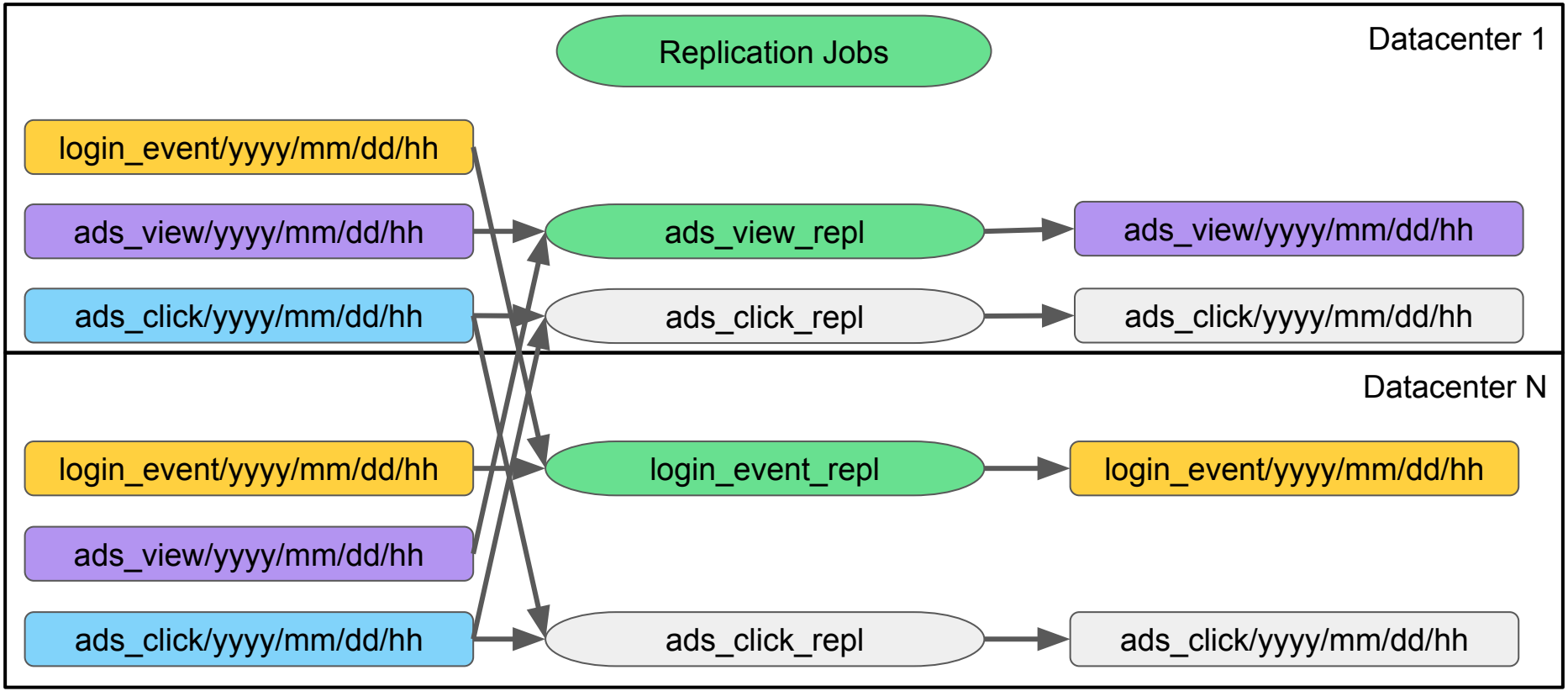
Log Replication Visual



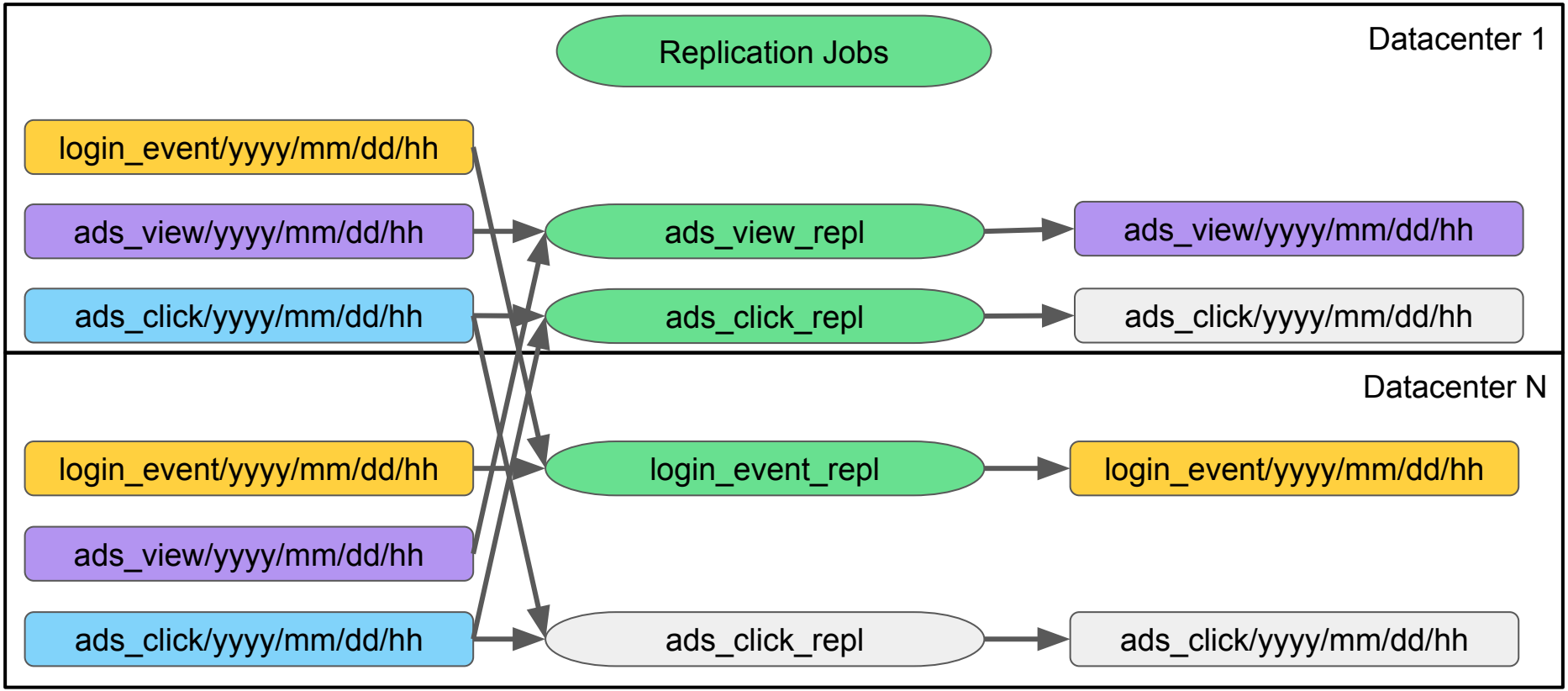
Log Replication Visual



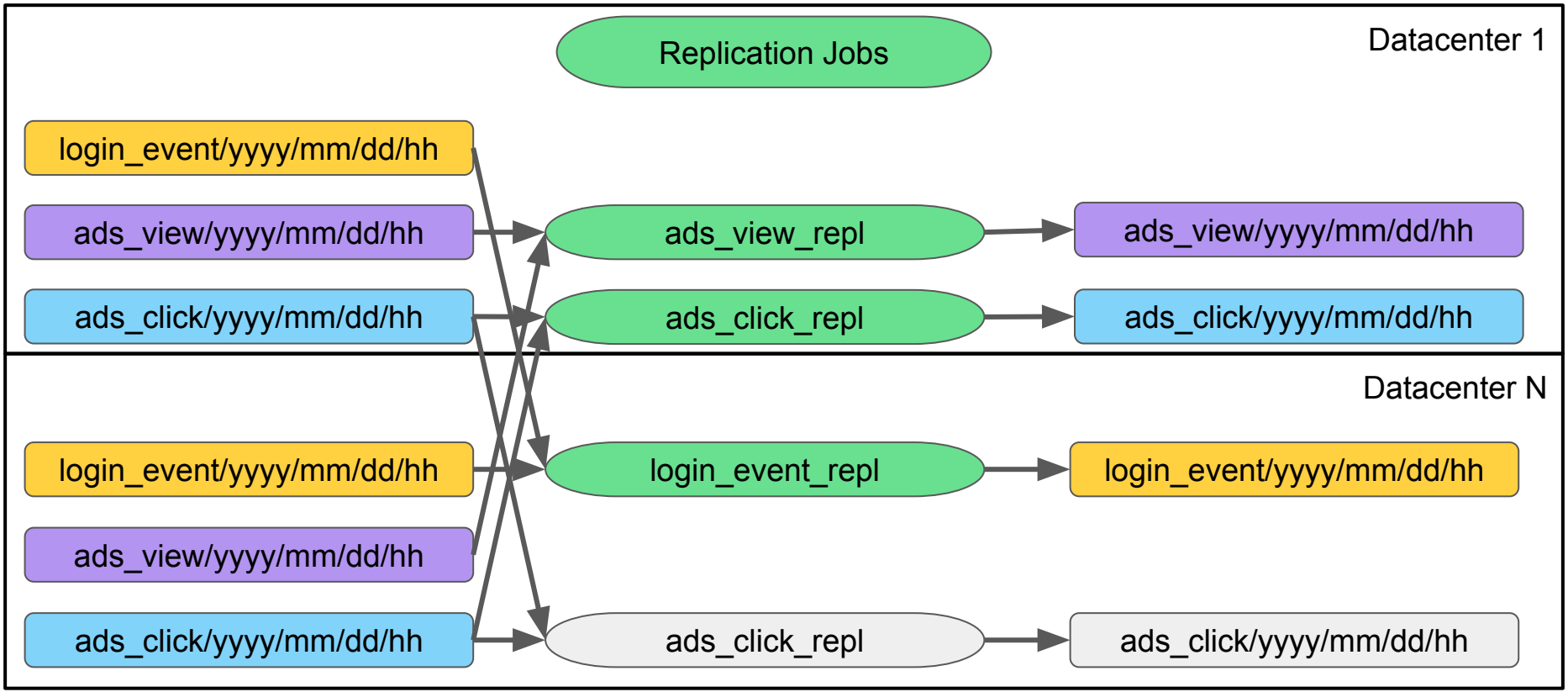
Log Replication Visual



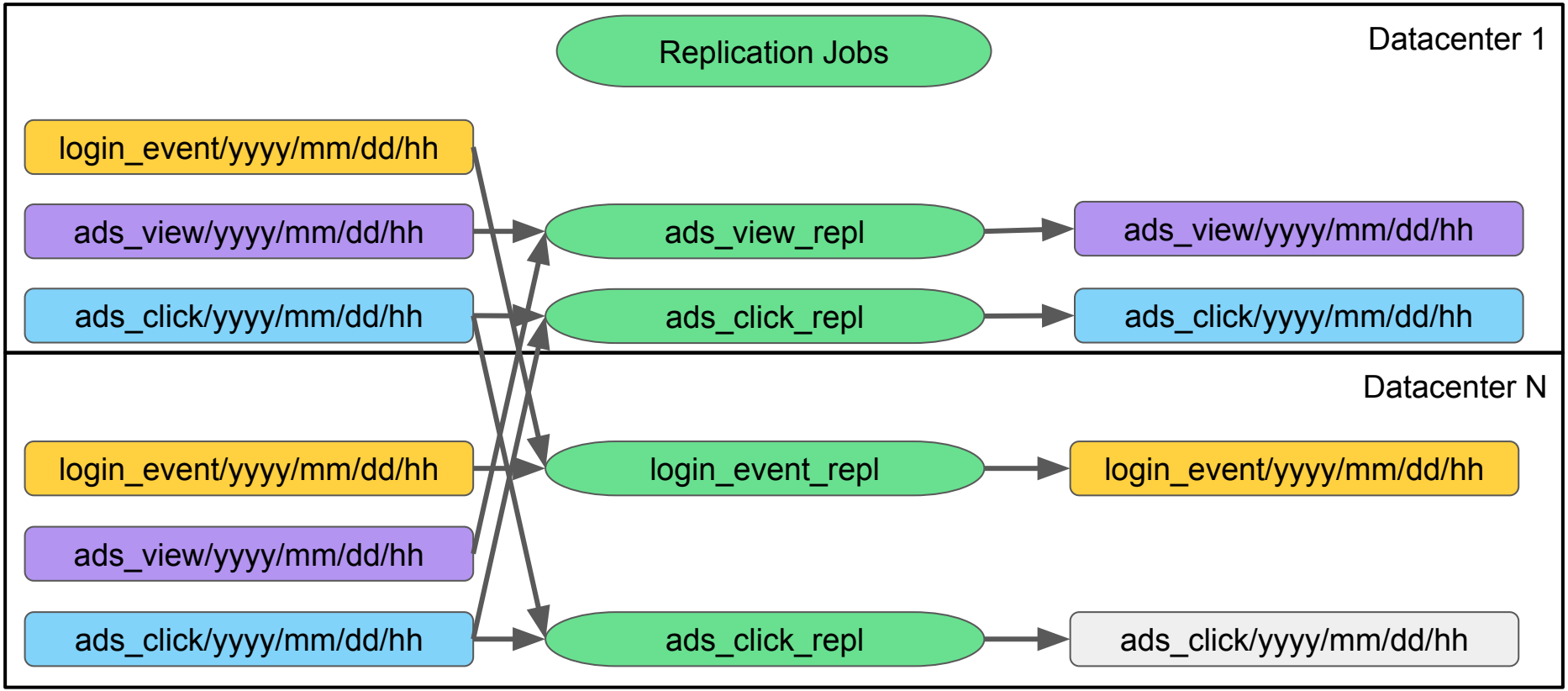
Log Replication Visual



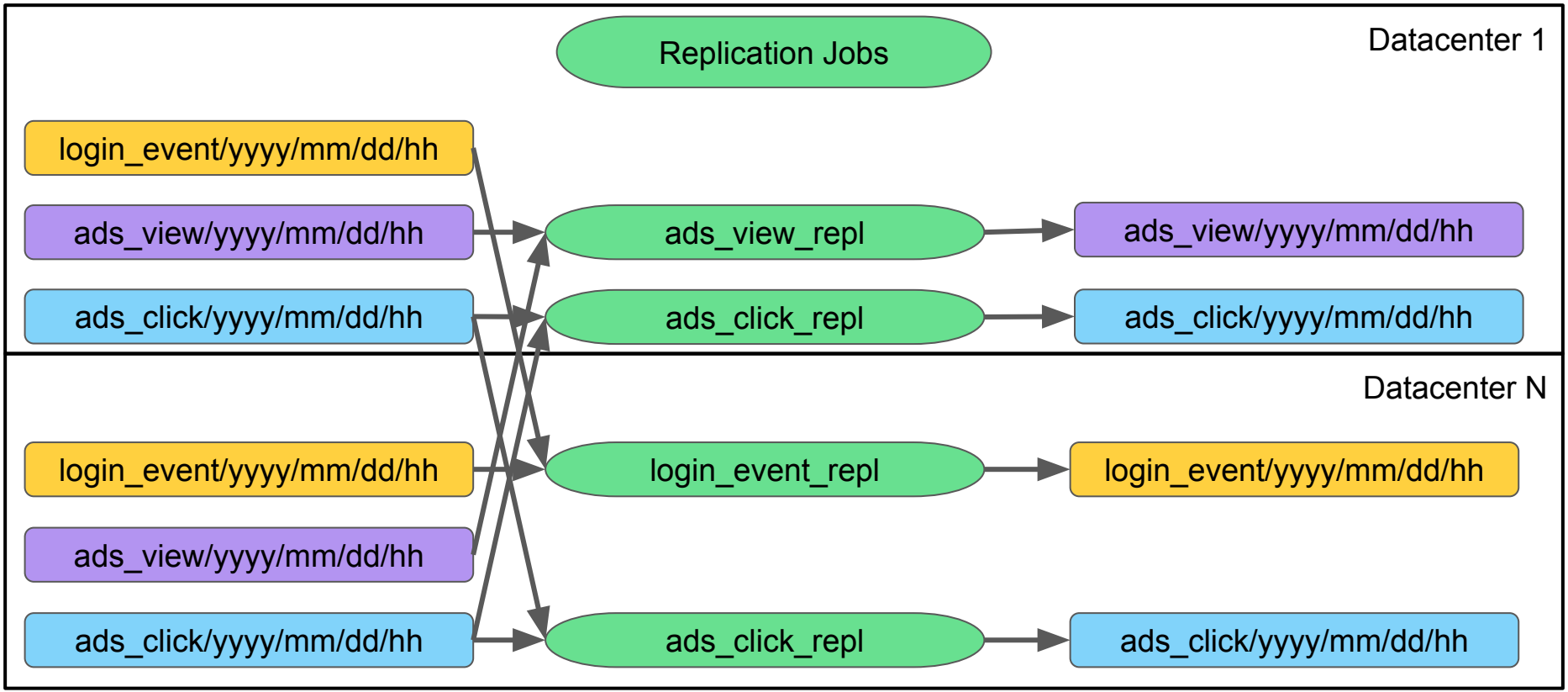
Log Replication Visual



Log Replication Visual



Log Replication Visual



Log Replication Steps

Distributing Trillion Events per Day

1 **Copy**
Logged data from all processing clusters to the target cluster.

2 **Merge**
Copied data into one directory.

3 **Present**
Data atomically by renaming it to an accessible location.

4 **Publish**
Metadata to the Data Abstraction Layer to notify analytics teams data is ready for consumption.

Log Replicator Daemon

- One log replicator daemon per DW, PROD, or COLD Hadoop cluster, where analytics users run queries and jobs
- Primarily responsible for copying category partitions out of the RT Hadoop clusters
- The daemons schedule Hadoop DistCp jobs every hour for every category
- Daemon atomically presents processed category instances so partial data can't be read
- Replication proceeds according to criticality of data or "tiers"

The Future

Future of Log Management

- Flume client improvement for tracing, SLA and throttling
- Flume client to support for message validation before logging
- Centralized configuration management
- Processing and replication every 10 minutes instead of every hour

Thank You

Questions?