REAL-TIME ANALYTICS WITH OPEN SOURCE TECHNOLOGIES

KAFKA · HADOOP · STORM · DRUID

FANGJIN YANG · GIAN MERLINO

SOFTWARE ENGINEERS @ METAMARKETS

METAMARKETS

OVERVIEW

PROBLEM	DEALING WITH EVENT DATA
MOTIVATION	EVOLUTION OF A “REAL-TIME” STACK
ARCHITECTURE	THE “RAD”-STACK
NEXT STEPS	TRY IT OUT FOR YOURSELF

THE PROBLEM

THE PROBLEM

- Arbitrary and interactive exploration of time series data
  - Online advertising
  - System/application metrics
  - Network traffic monitoring
  - Activity stream analysis
- Multi-tenancy: lots of concurrent users
- Scalability: 10+ TB/day, ad-hoc queries on trillions of events
- Recency matters! Real-time analysis

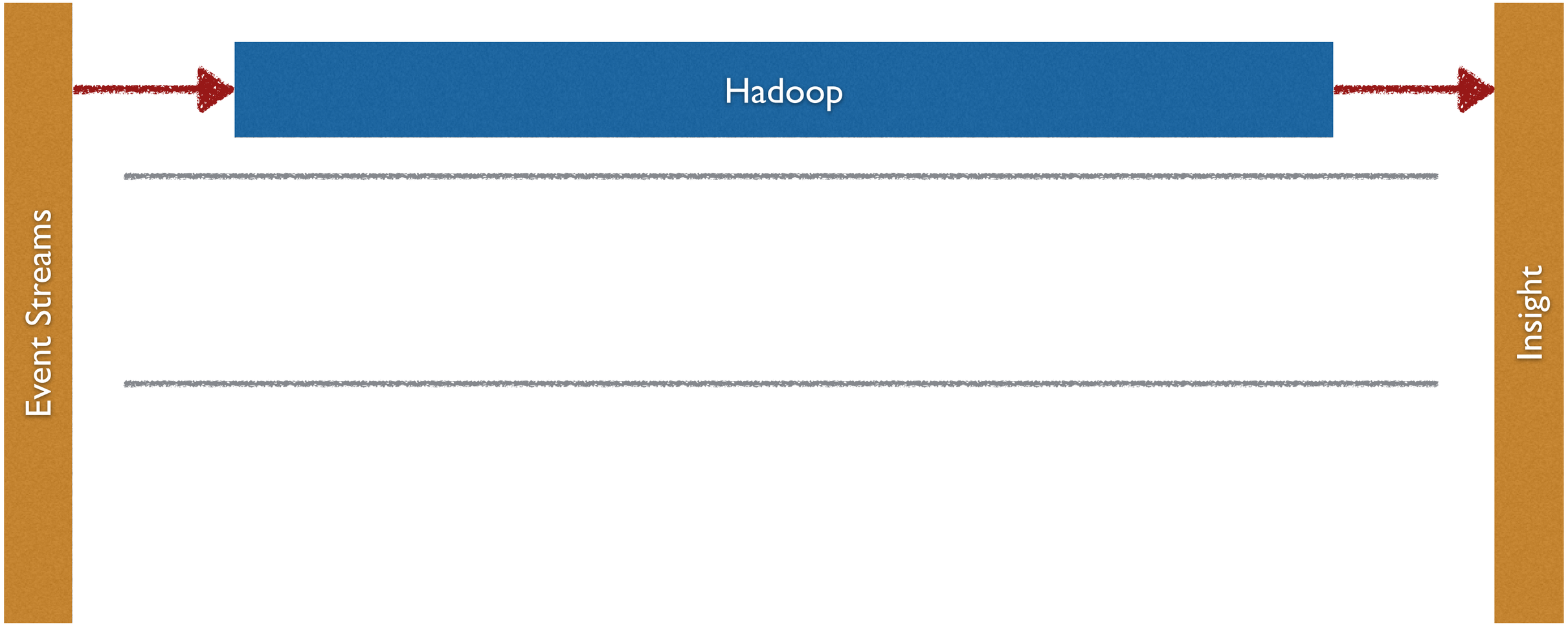
DEMO

IN CASE THE INTERNET DIDN'T WORK
PRETEND YOU SAW SOMETHING COOL

FINDING A SOLUTION

- Load all your data into Hadoop. Query it. Done!
- Good job guys, let's go home

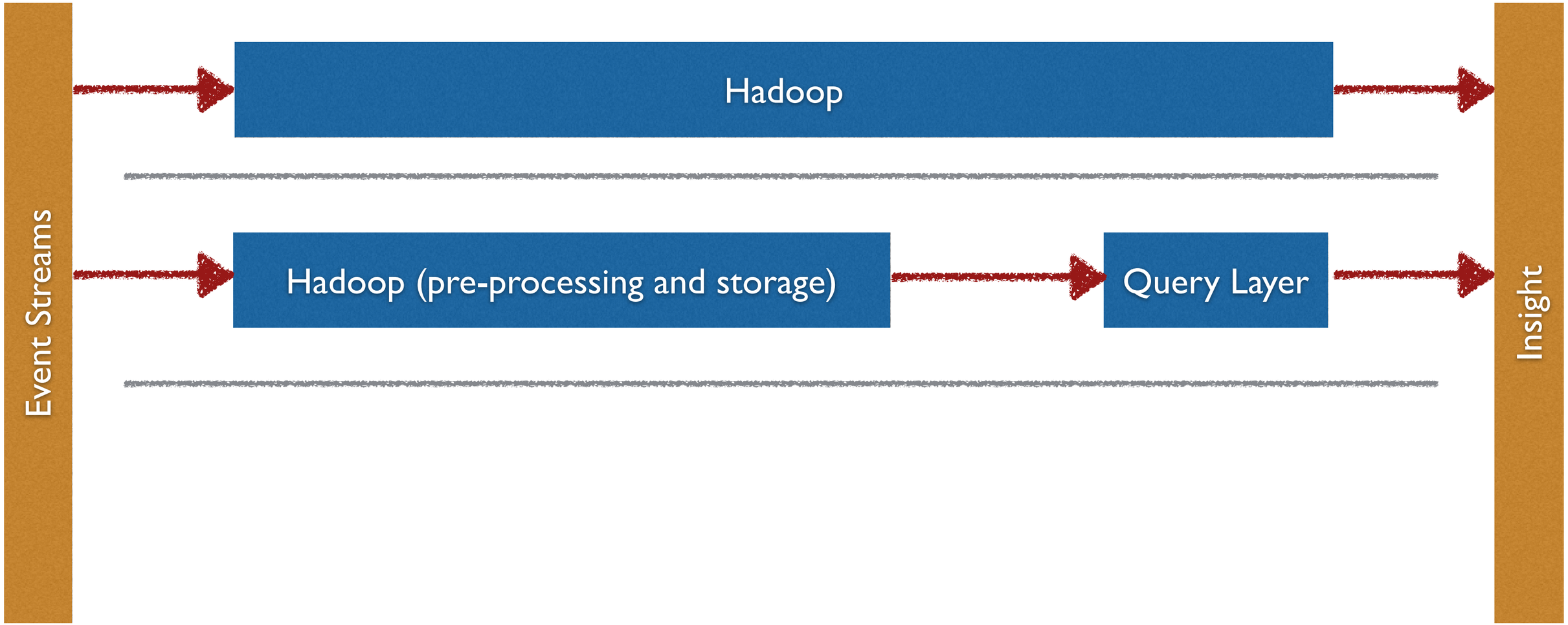
FINDING A SOLUTION



PROBLEMS WITH THE NAIVE SOLUTION

- MapReduce can handle almost every distributed computing problem
- MapReduce over your raw data is flexible but slow
- Hadoop is not optimized for query latency
- To optimize queries, we need a query layer

FINDING A SOLUTION

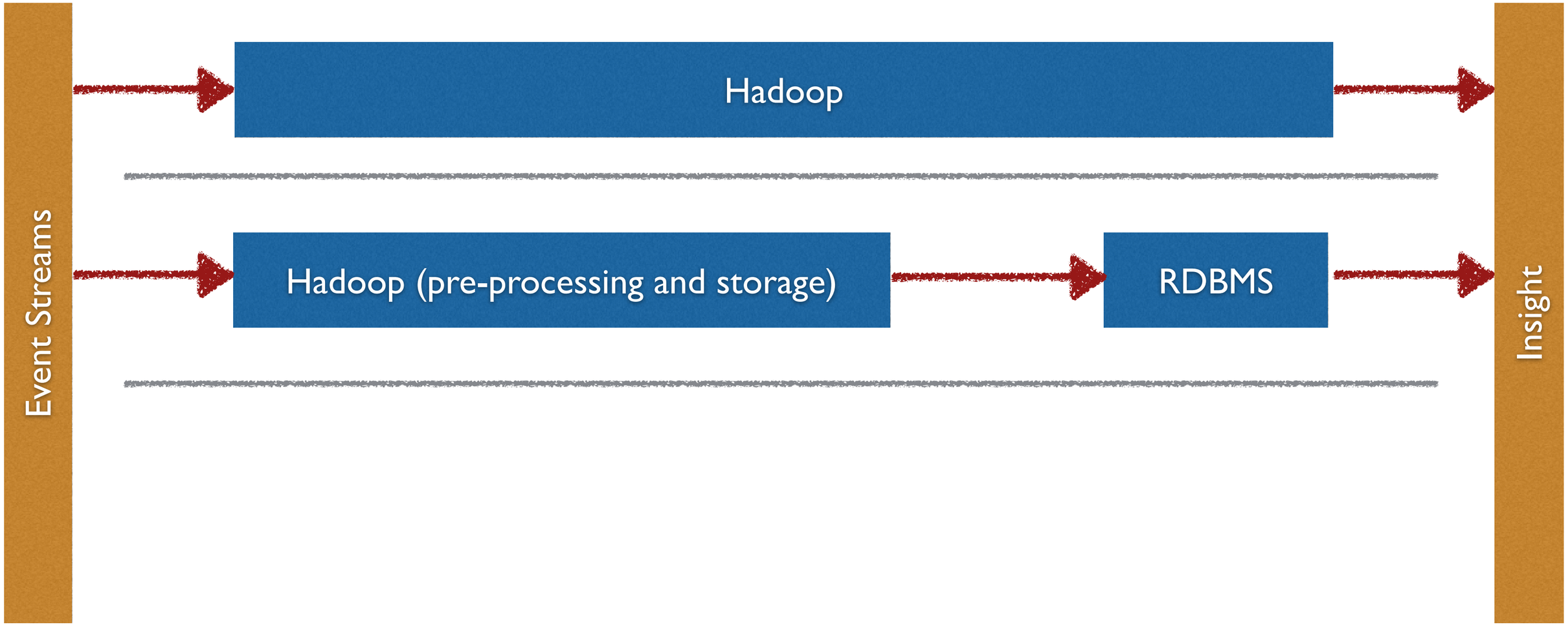


A FASTER QUERY LAYER

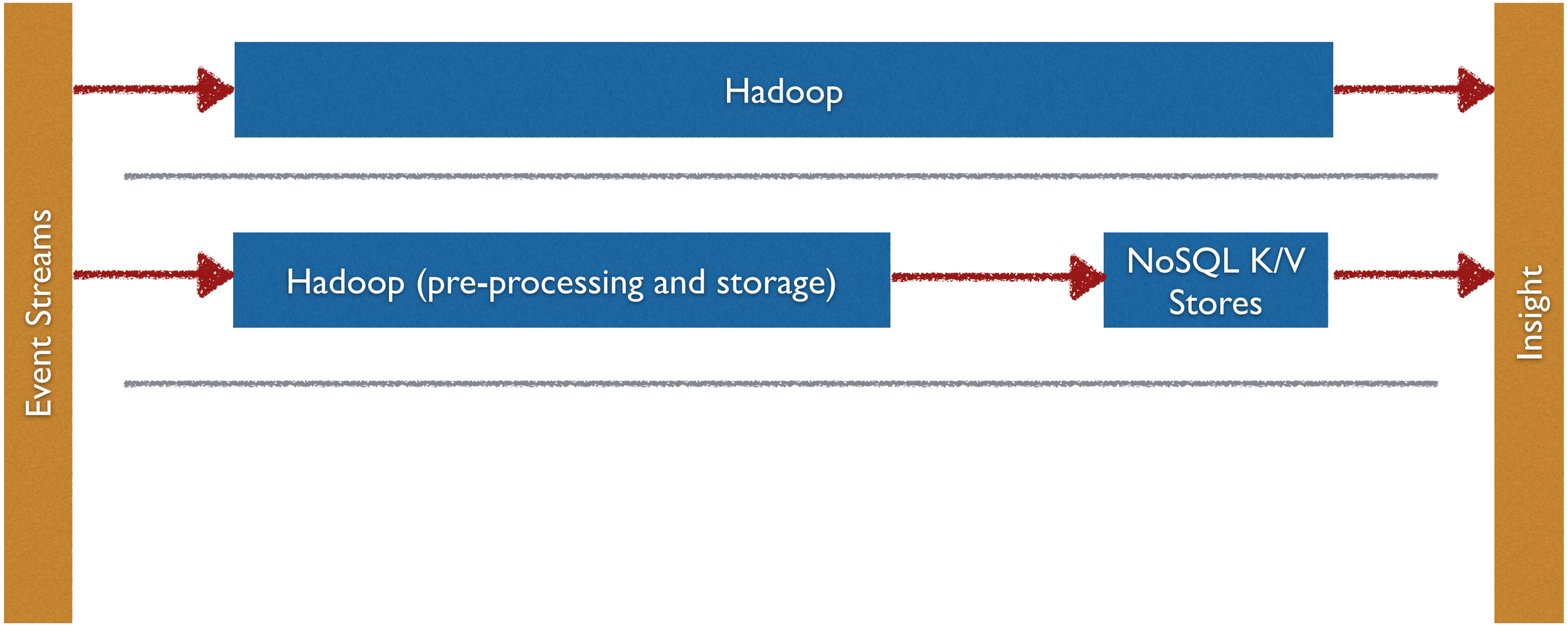
MAKE QUERIES FASTER

- What types of queries to optimize for?
  - Revenue over time broken down by demographic
  - Top publishers by clicks over the last month
  - Number of unique visitors broken down by any dimension
  - Not dumping the entire dataset
  - Not examining individual events

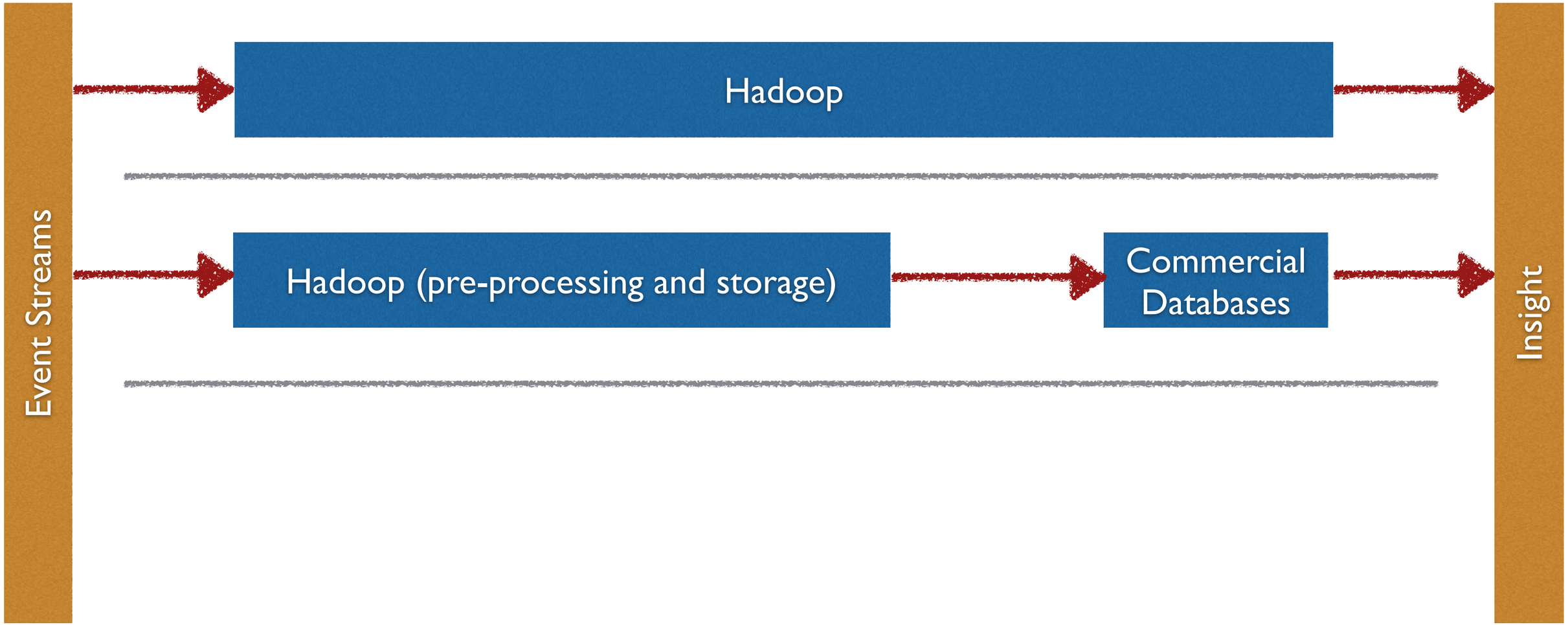
FINDING A SOLUTION



FINDING A SOLUTION



FINDING A SOLUTION



DRUID AS A QUERY LAYER

DRUID

- Open sourced in Oct. 2012
- Growing Community
  - 45+ contributors from many different organizations
- Designed for low latency ingestion and aggregation
  - Optimized for the types of queries we were trying to make

RAW DATA

timestamp	publisher	advertiser	gender	country	click	price
2011-01-01T01:01:35Z	bieberfever.com	google.com	Male	USA	0	0.65
2011-01-01T01:03:63Z	bieberfever.com	google.com	Male	USA	0	0.62
2011-01-01T01:04:51Z	bieberfever.com	google.com	Male	USA	1	0.45
...						
2011-01-01T01:00:00Z	ultratrimfast.com	google.com	Female	UK	0	0.87
2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Female	UK	0	0.99
2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Female	UK	1	1.53

ROLLUP DATA

timestamp	publisher	advertiser	gender	country	impressions	clicks	revenue
2011-01-01T01:00:00Z	ultratrimfast.com	google.com	Male	USA	1800	25	15.70
2011-01-01T01:00:00Z	bieberfever.com	google.com	Male	USA	2912	42	29.18
2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Male	UK	1953	17	17.31
2011-01-01T02:00:00Z	bieberfever.com	google.com	Male	UK	3194	170	34.01

- Truncate timestamps
- GroupBy over string columns (dimensions)
- Aggregate numeric columns (metrics)

PARTITION DATA

timestamp	publisher	advertiser	gender	country	impressions	clicks	revenue
2011-01-01T01:00:00Z	ultratrimfast.com	google.com	Male	USA	1800	25	15.70
2011-01-01T01:00:00Z	bieberfever.com	google.com	Male	USA	2912	42	29.18

Segment 2011-01-01T01/2011-01-01T02

2011-01-01T02:00:00Z	ultratrimfast.com	google.com	Male	UK	1953	17	17.31
2011-01-01T02:00:00Z	bieberfever.com	google.com	Male	UK	3194	170	34.01

Segment 2011-01-01T02/2011-01-01T03

- Shard data by time
- Immutable chunks of data called “segments”

IMMUTABLE SEGMENTS

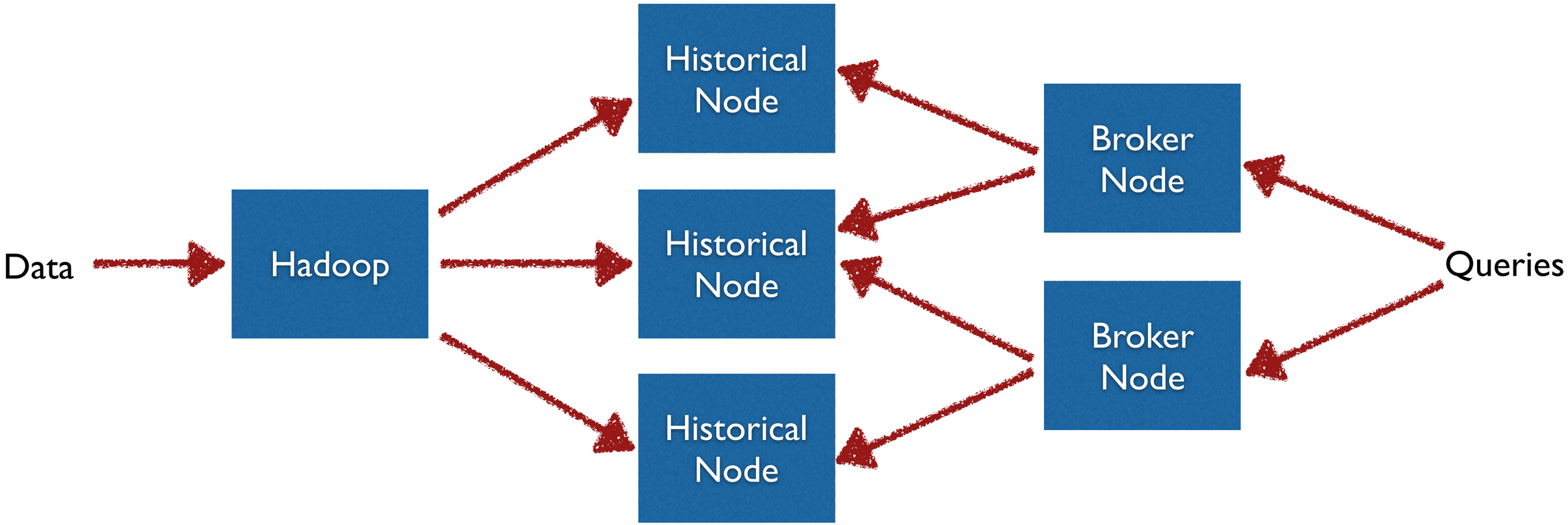
- Fundamental storage unit in Druid
- Read consistency
- One thread scans one segment
- Multiple threads can access same underlying data
- Segment sizes -> computation completes in ms
- Simplifies distribution & replication

COLUMN ORIENTATION

timestamp	publisher	advertiser	gender	country	impressions	clicks	revenue
2011-01-01T01:00:00Z	ultratrimfast.com	google.com	Male	USA	1800	25	15.70
2011-01-01T01:00:00Z	bieberfever.com	google.com	Male	USA	2912	42	29.18

- Scan/load only what you need
- Compression!
- Indexes!

ARCHITECTURE (EARLY DAYS)

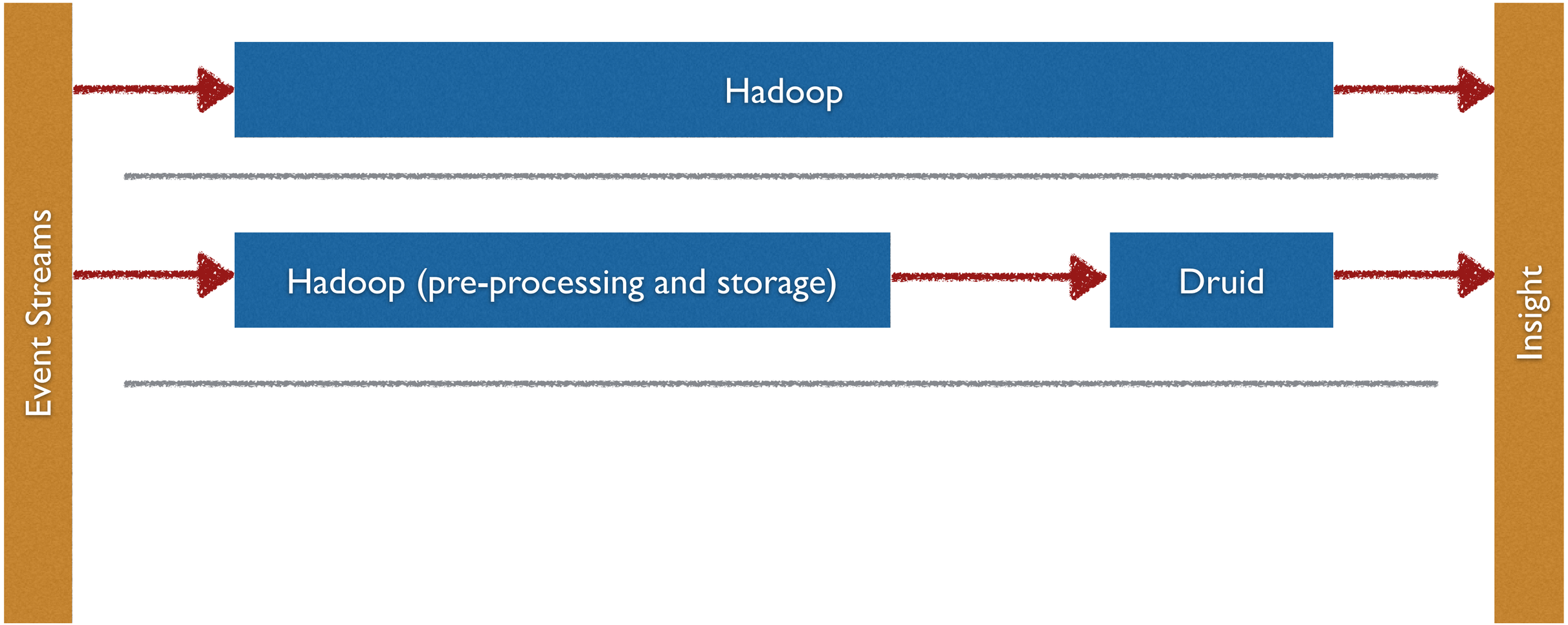


MORE PROBLEMS

- We've solved the query problem
  - Druid gave us arbitrary data exploration & fast queries
  - 90th/95th/99th percentile queries on >100TB data within 1s/2s/10s
- But what about data freshness?
  - Batch loading is slow!
  - We want "real-time"
  - Alerts, operational monitoring, etc.

A FASTER DATA PIPELINE

THE STORY SO FAR



THE STORY SO FAR

- Clients uploaded data to S3
- We used Hadoop + Pig to clean it up, transform, and join it
- We loaded the result into Druid
- Typical turnaround time: 2–8 hours

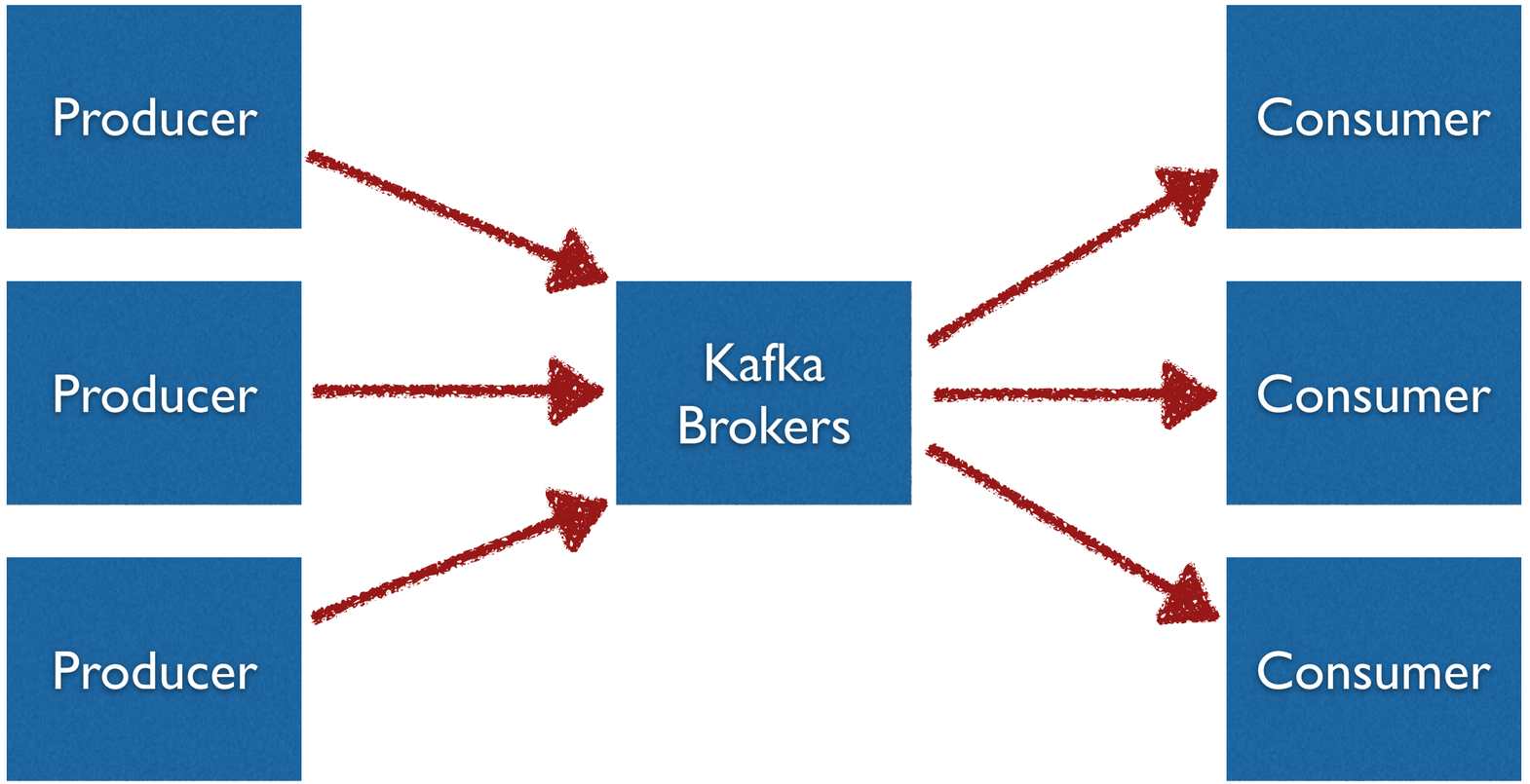
INGESTION DELAYS

- Let's build a streaming version of this data pipeline
- Three obstacles
  - Acquiring raw data
  - Processing the data
  - Loading processed data into Druid

FAST DELIVERY WITH KAFKA

- High throughput event delivery service
- Straightforward, reliable design
- Buffers incoming data to give consumers time to process it
- We can place an HTTP API in front of this

FAST DELIVERY WITH KAFKA



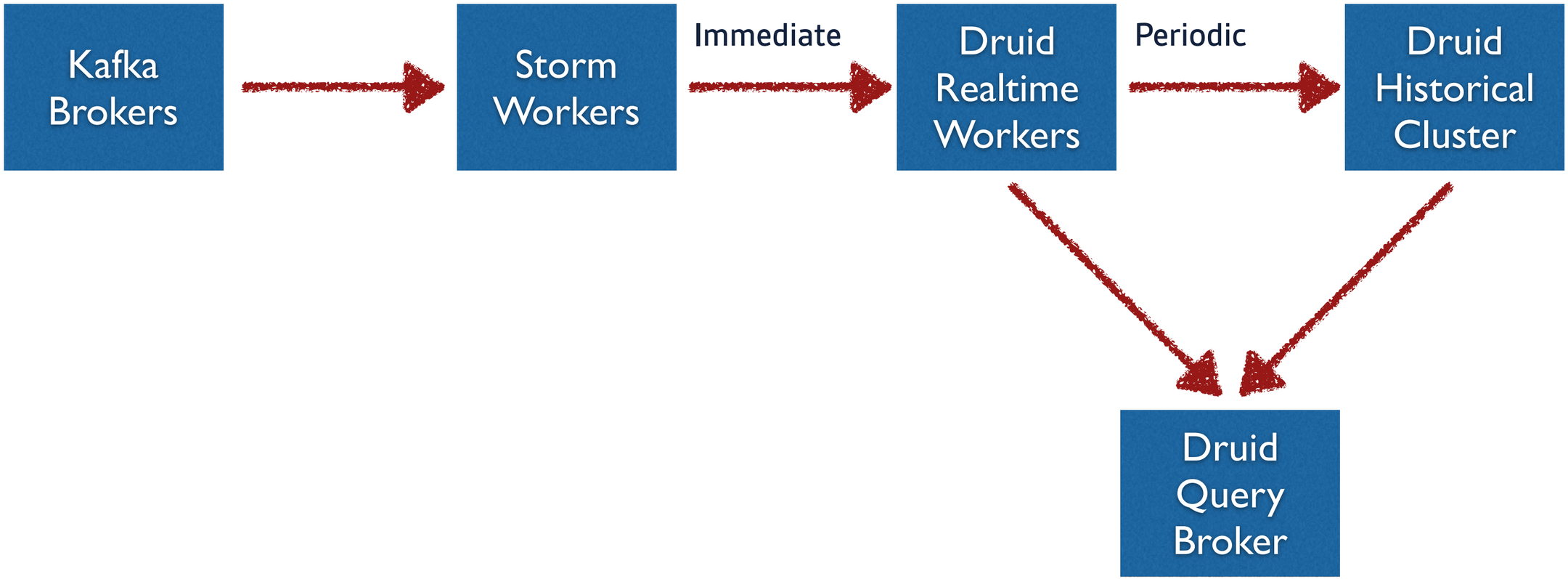
FAST PROCESSING WITH STORM

- Storm is a stream processor— one event at a time
- We can already process our data using Hadoop MapReduce
- Let's translate that to streams
  - "Load" operations stream data from Kafka
  - "Map" operations are already stream-friendly
  - "Reduce" operations can be windowed with partitioned state

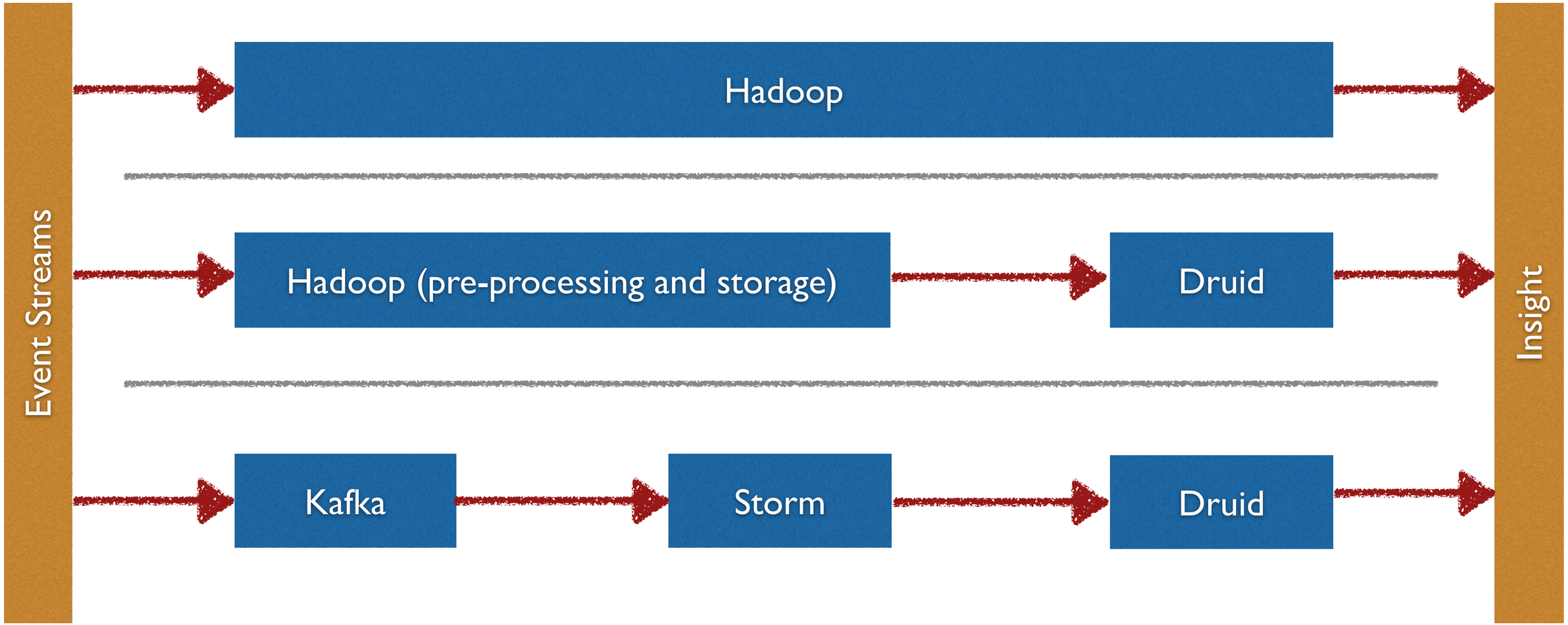
FAST LOADING WITH DRUID

- We have an indexing system
- We have a serving system that runs queries on data
- We can serve queries while building indexes!
- Real-time indexing workers do this

FAST LOADING WITH DRUID



THE STORY SO FAR



WHAT WE GAINED

- Druid queries reflect new events within seconds
- Systems are fully decoupled
- Brief processing delays during maintenance
  - Because we need to restart Storm topologies
  - But query performance and availability are not affected

WHAT WE GAVE UP

- Stream processing isn't perfect
- Difficult to handle corrections of existing data
- Windows may be too small for fully accurate operations
- Hadoop was actually good at these things

THE RETURN OF HADOOP

- An open-source “lambda architecture”
- Batch re-processing runs for all data older than a few hours
- Batch segments replace real-time segments in Druid
- Query broker merges results from both systems

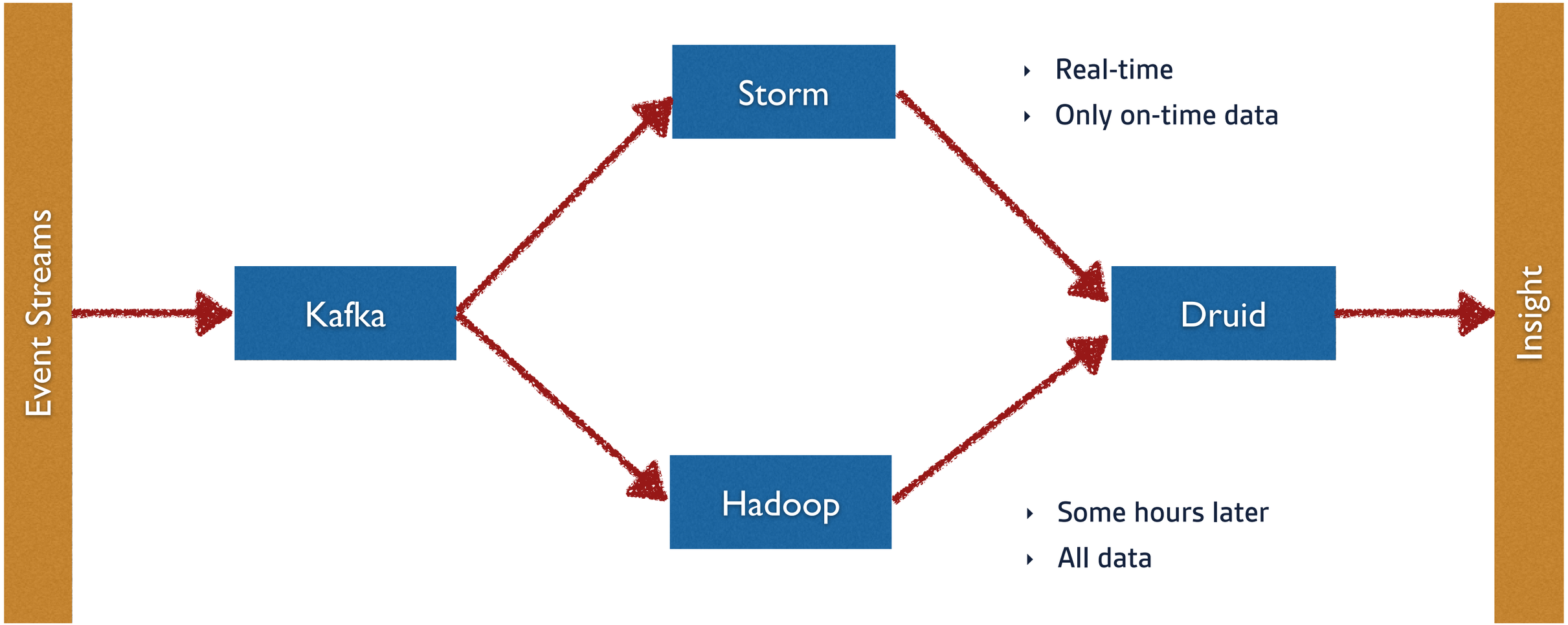
“Fixed up,” immutable, historical data

—by Hadoop

Realtime data

—by Storm & Realtime Druid

THE STACK

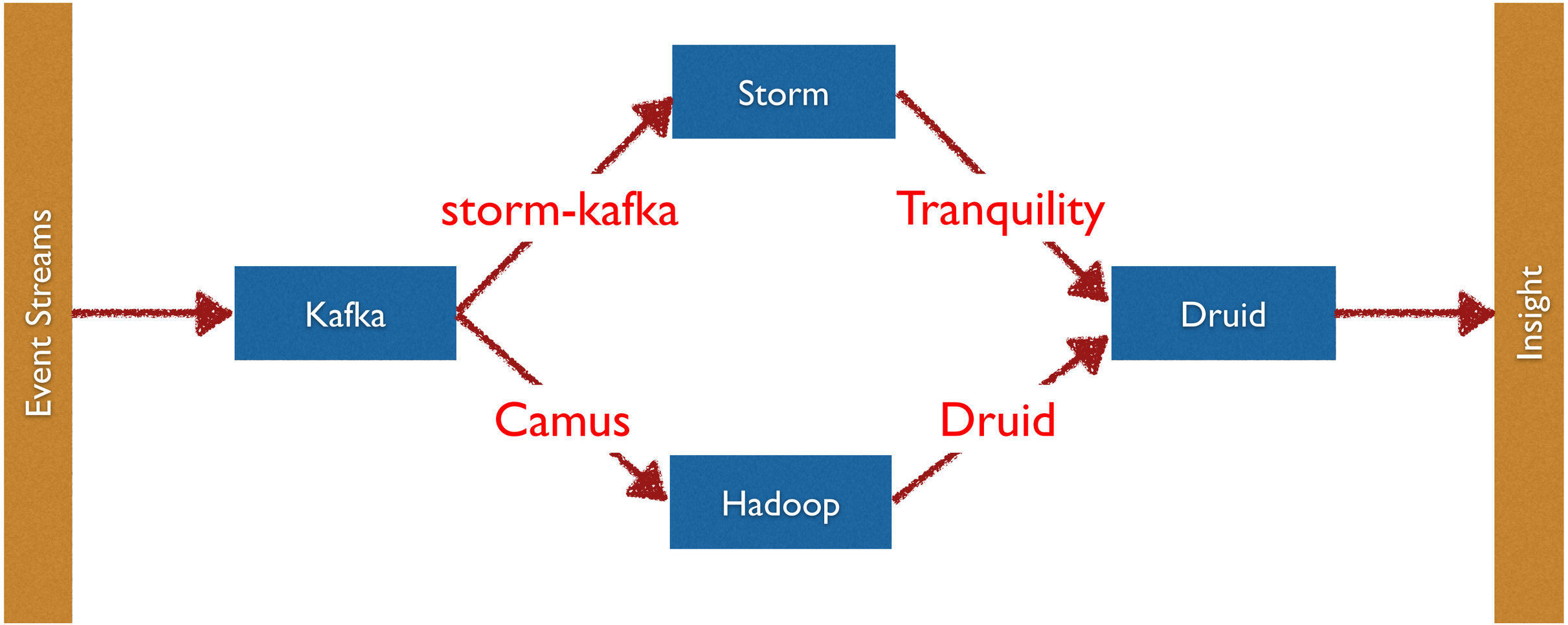

Event Streams
Kafka
Storm
Hadoop
Druid
Insight
Real-time
Only on-time data
Some hours later
All data

DO TRY THIS AT HOME

CORNERSTONES

- Druid - druid.io - @druidio
- Storm - storm.incubator.apache.org - @stormprocessor
- Hadoop - hadoop.apache.org
- Kafka - kafka.apache.org - @apachekafka

GLUE


Event Streams
Kafka
Storm
storm-kafka
Tranquility
Camus
Druid
Hadoop
Druid
Insight

GET RADICAL

- Queries answered quickly, on fresh data
- Kafka provides fast, reliable event transport
- Storm and Hadoop clean and prepare data for Druid
- Druid handles queries and manages the serving layer
- “Real-time Analytics Data Stack”
  - ...a.k.a. RAD Stack
- https://metamarkets.com/2014/building-a-data-pipeline/

THANK YOU

@DRUIDIO @METAMARKETS