LinkedIn derived data platform

The unified platform for batch and streaming data set

Yan Yan

Staff Software Engineer

Derived data platform

Today's agenda

11:30	Introduction
11:35	Venice: LinkedIn derived data platform
11:40	Architecture
12:00	Go faster and faster
12:05	Lesson we learned
12:15	Conclusion

Introduction

Primary & Derived Data, Data Lifecycle, Lambda

Kinds of Data

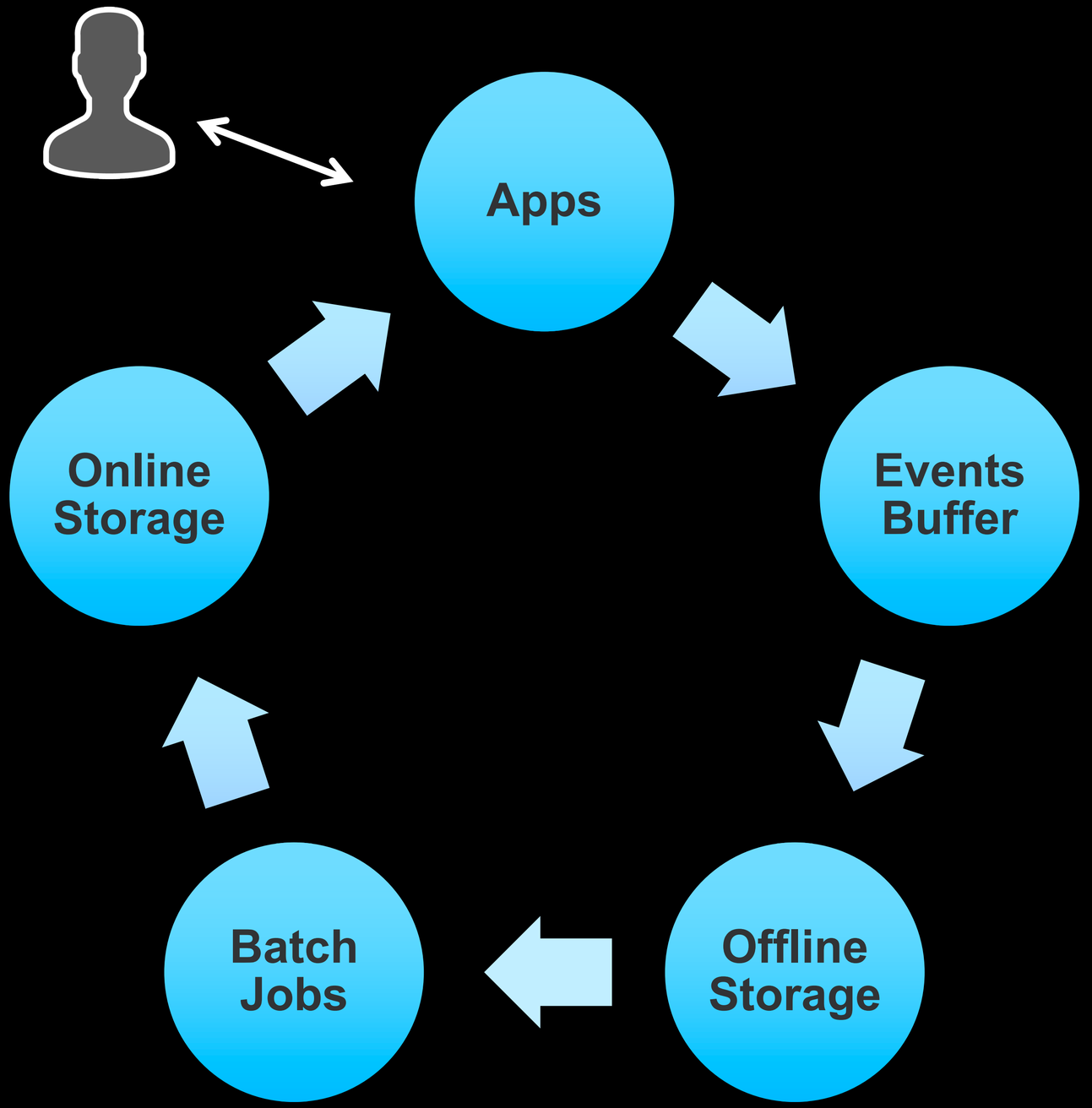
Primary Data

- Source of Truth
- Example use case:
  - Profile
- Example systems:
  - SQL
  - Document Stores
  - K-V Stores

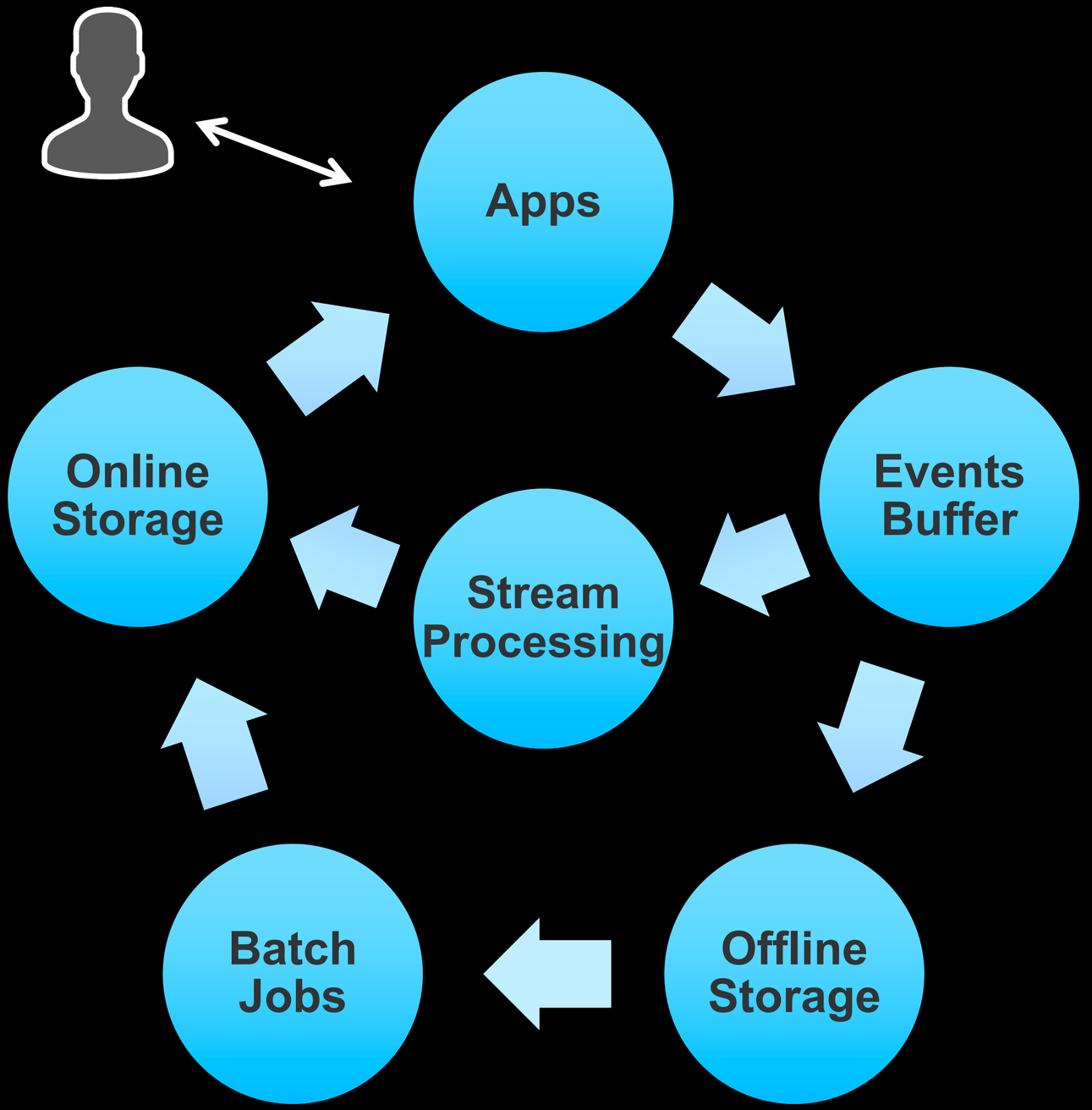
Derived Data

- Derived from computing primary data
- Example use case:
  - People You May Know
- Example systems:
  - Search Indices
  - File system
  - K-V Stores

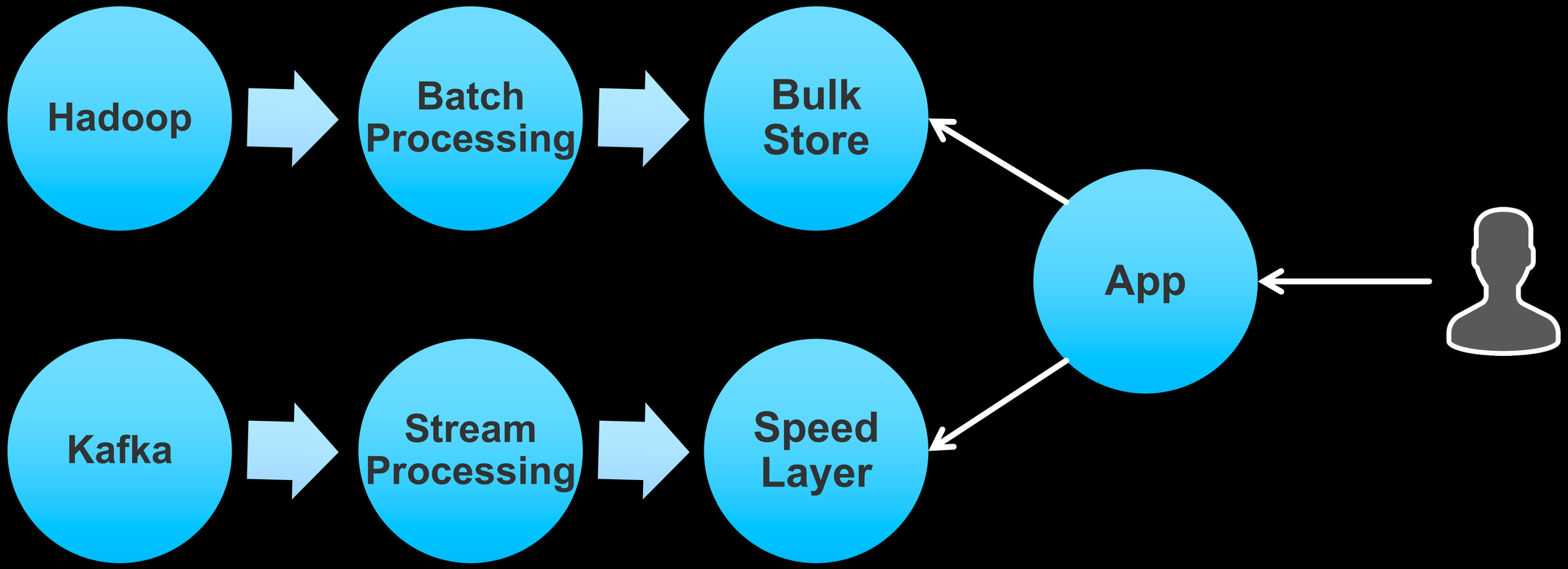
Derived Data Lifecycle



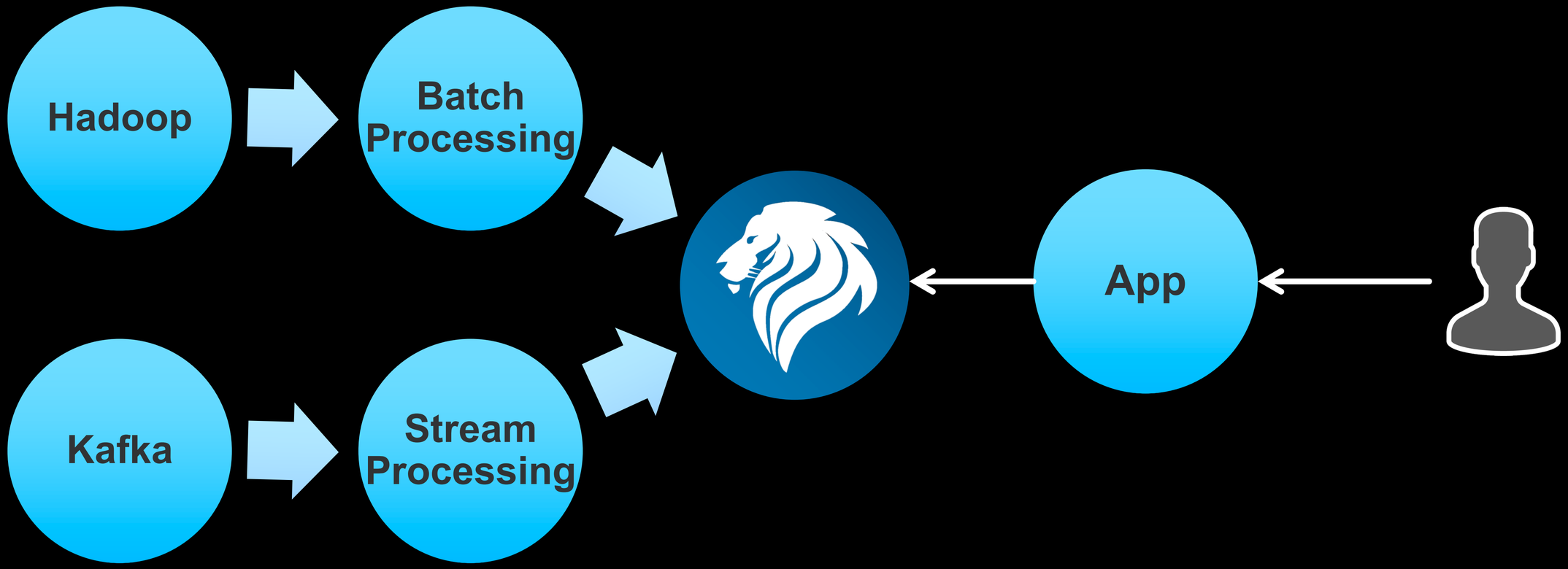
Derived Data Lifecycle Today



Lambda Architecture



Lambda Architecture, v2



Venice: Derived data platform

Design goals, Features, Scale, Trade off

Venice

Design Goals

- To replace Voldemort Read-Only
  - Drop-in replacement
  - More efficient
  - More resilient
  - More operable
- To enable new use cases
  - Nearline derived data
  - Single view of both types of data

Venice

Features

- High through ingestion from Hadoop and Samza
- Avro schema evolution
- Dynamic cluster management
  - Fully automatic replica placement
  - Cluster expansion
  - Self-healing
  - Rack-awareness

Venice

Scale

- Large scale
  - Multi-Datacenter
  - Multi-Cluster
- Run “as a service”
  - Self-service onboarding
  - Each cluster is multi-tenant
  - Isolation

Venice

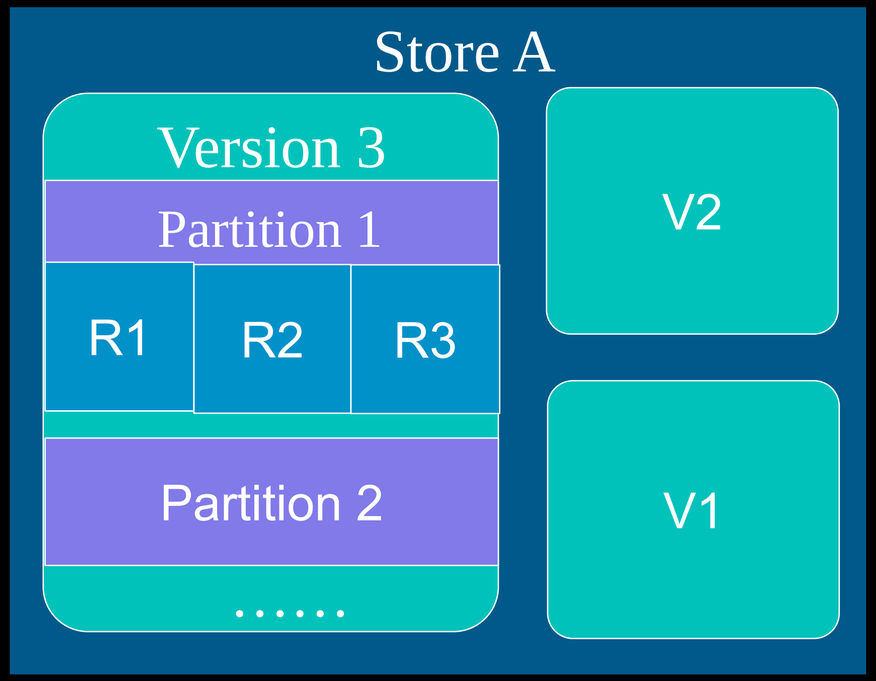
Tradeoffs

- All writes go through Kafka
  - Scalable
  - Burst tolerant
  - Source of truth
  - No native “read your writes” semantics

Architecture

Data model, Components, Batch mode, Global replication

Venice Data Model



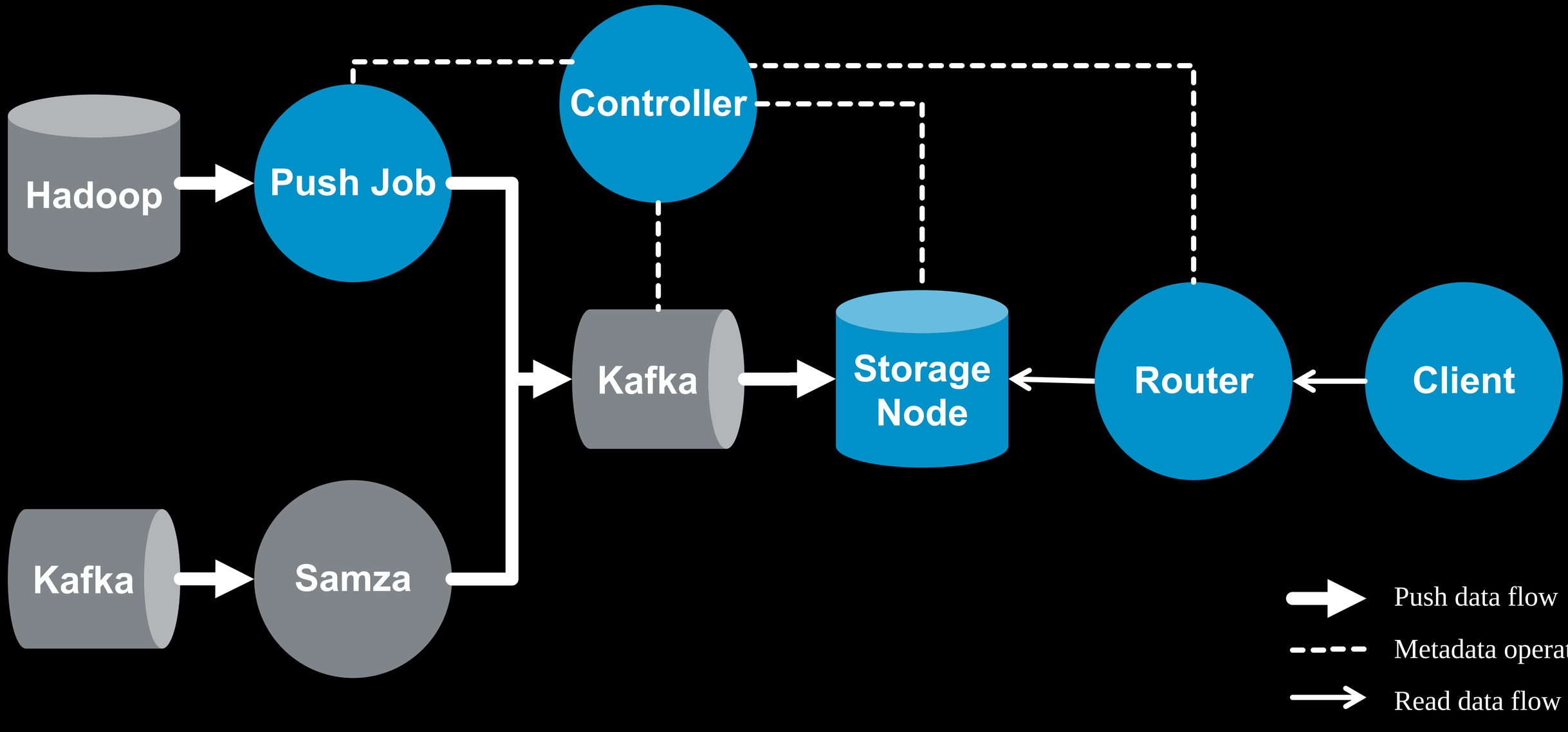
- Store
- Version
- Partition
- Replica
- Record
  - Avro

Architecture

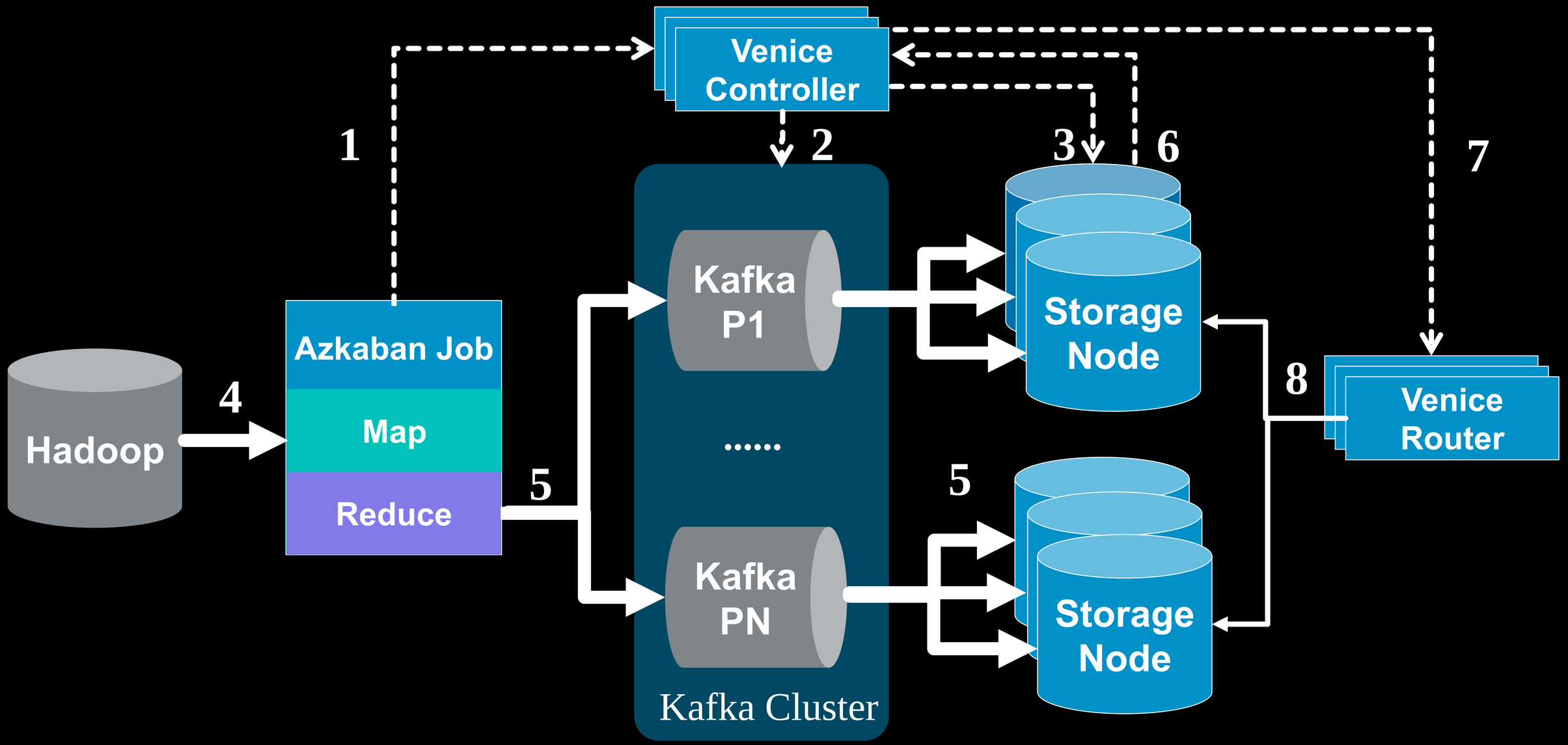
Components

- Server Processes
  - Storage Node
  - Router
  - Controller
- Libraries
  - Client
  - Hadoop to Venice Push Job
  - Samza System Producer

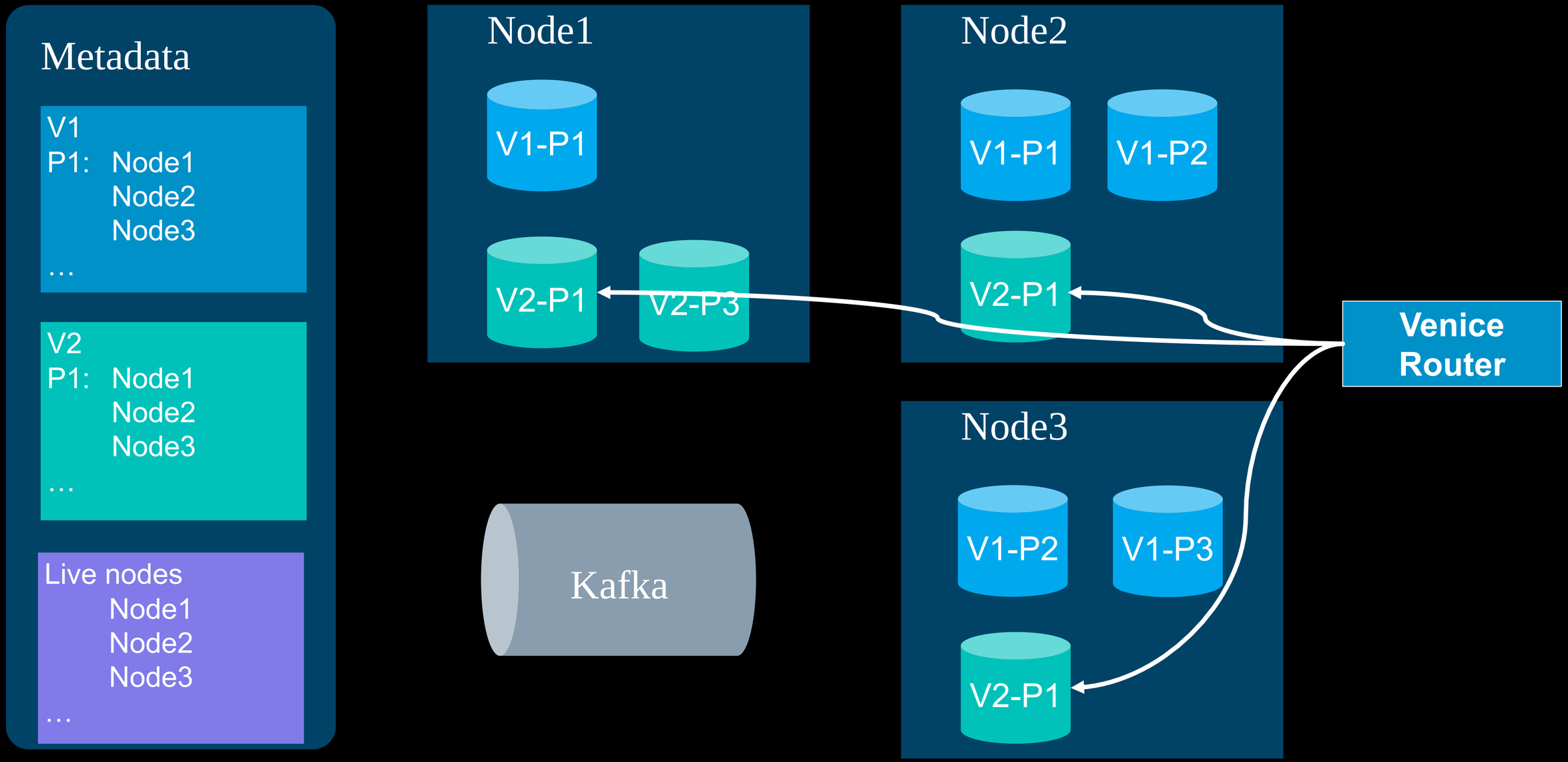
Venice Components



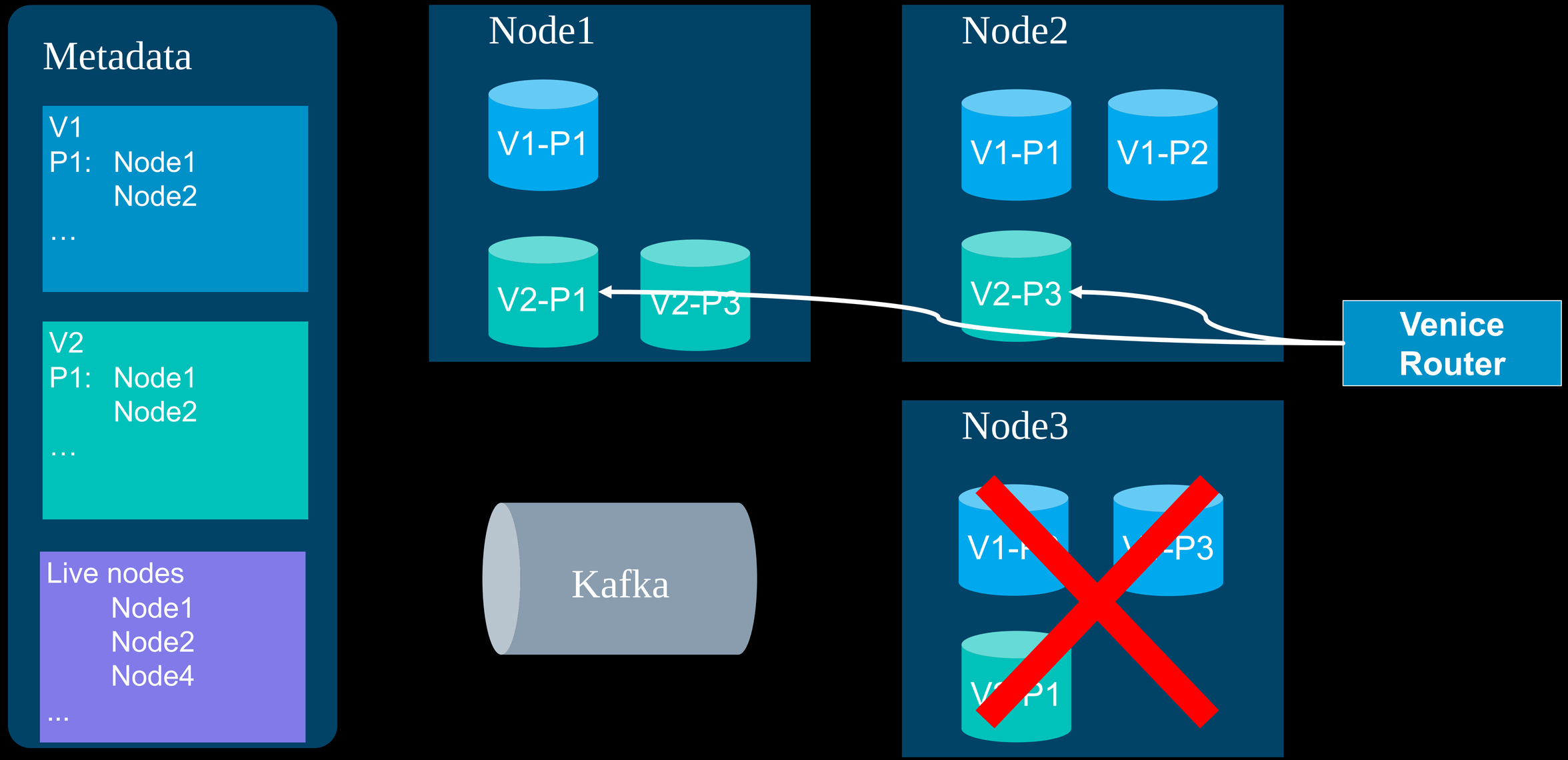
Venice Batch Mode



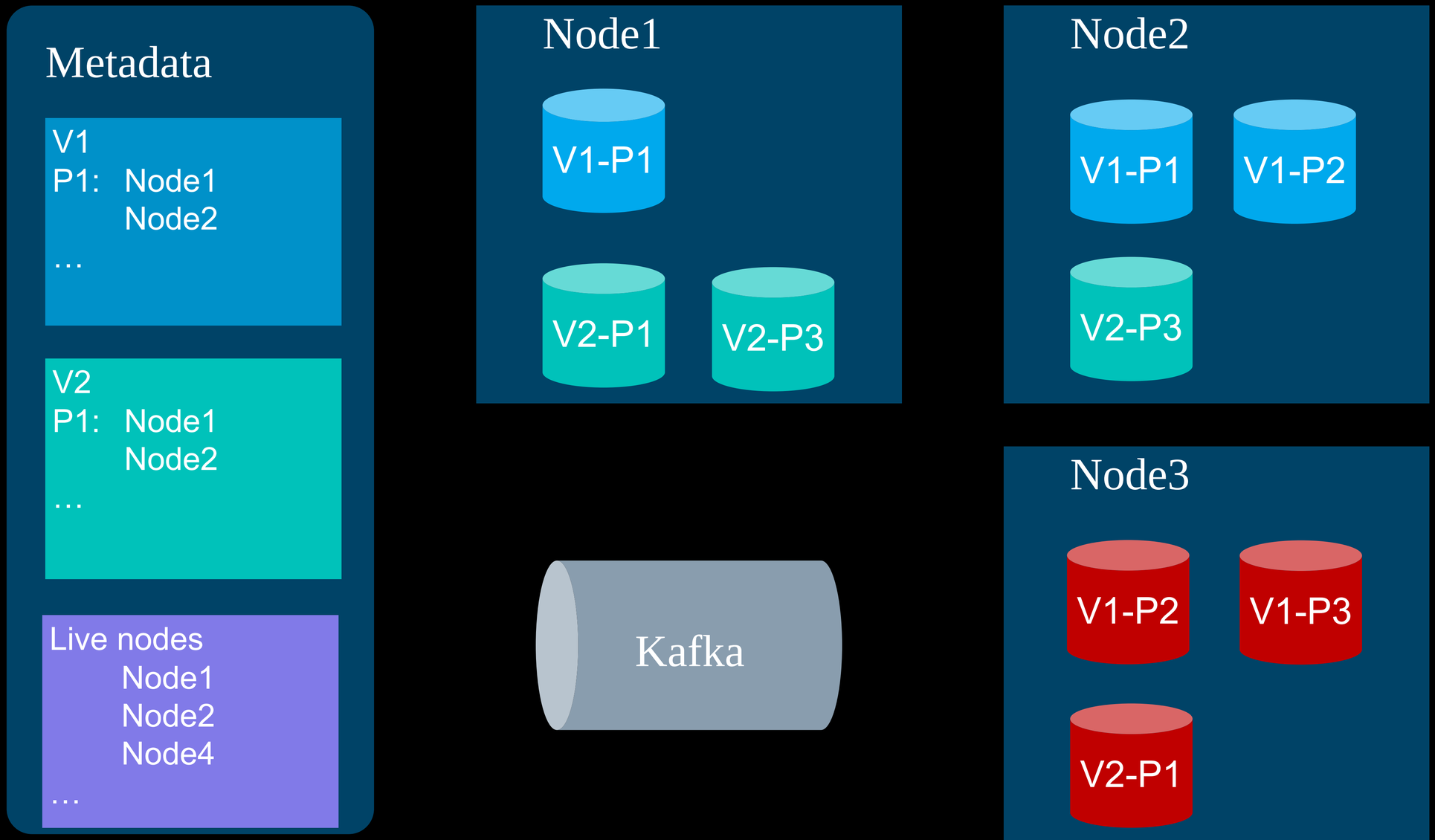
Data Layout


Metadata
V1
P1: Node1
Node2
Node3
…
V2
P1: Node1
Node2
Node3
…
Live nodes
Node1
Node2
Node3
…
Node1
V1-P1
V2-P1
V2-P3
Node2
V1-P1
V1-P2
V2-P1
Node3
V1-P2
V1-P3
V2-P1
Kafka
Venice Router

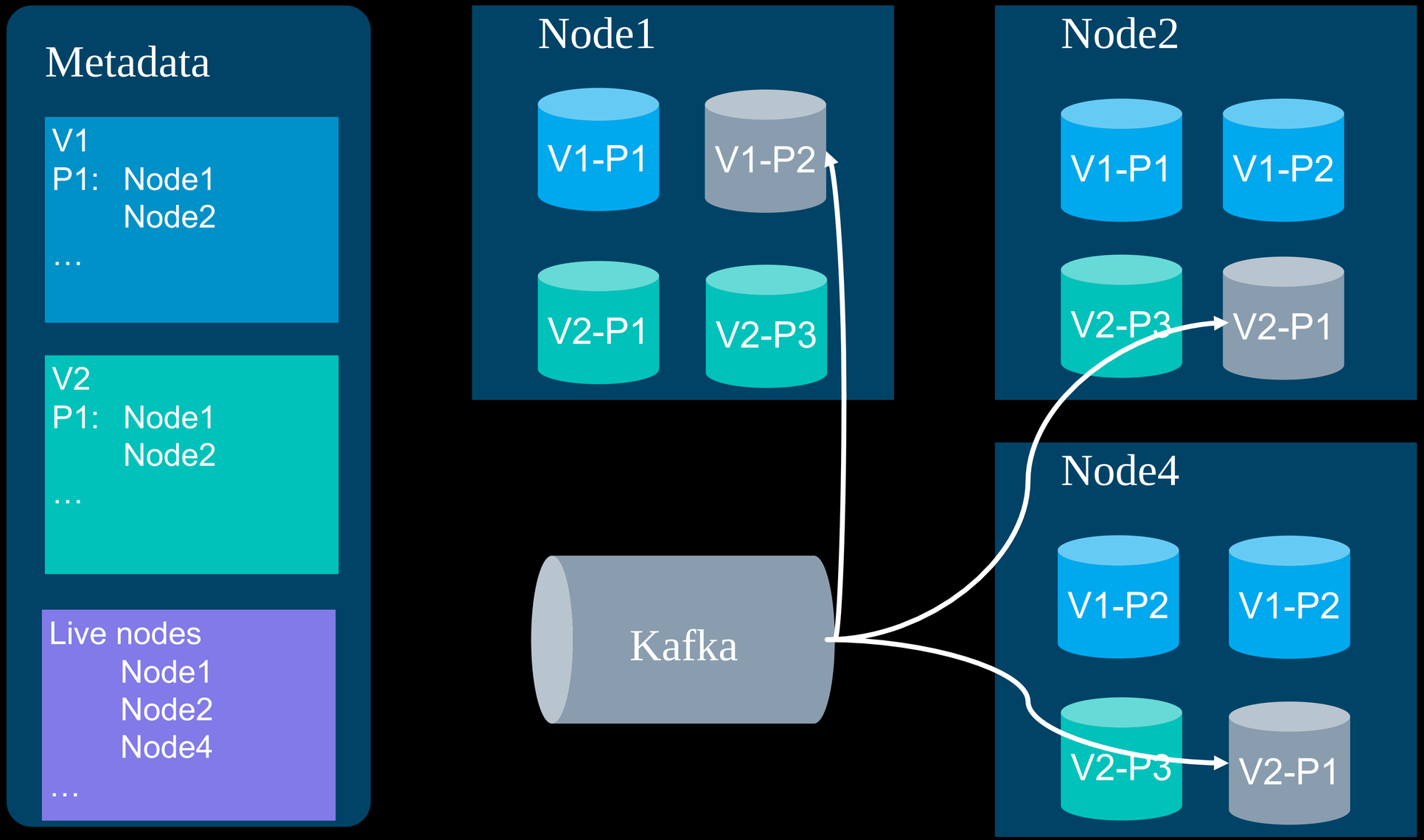
Node failover



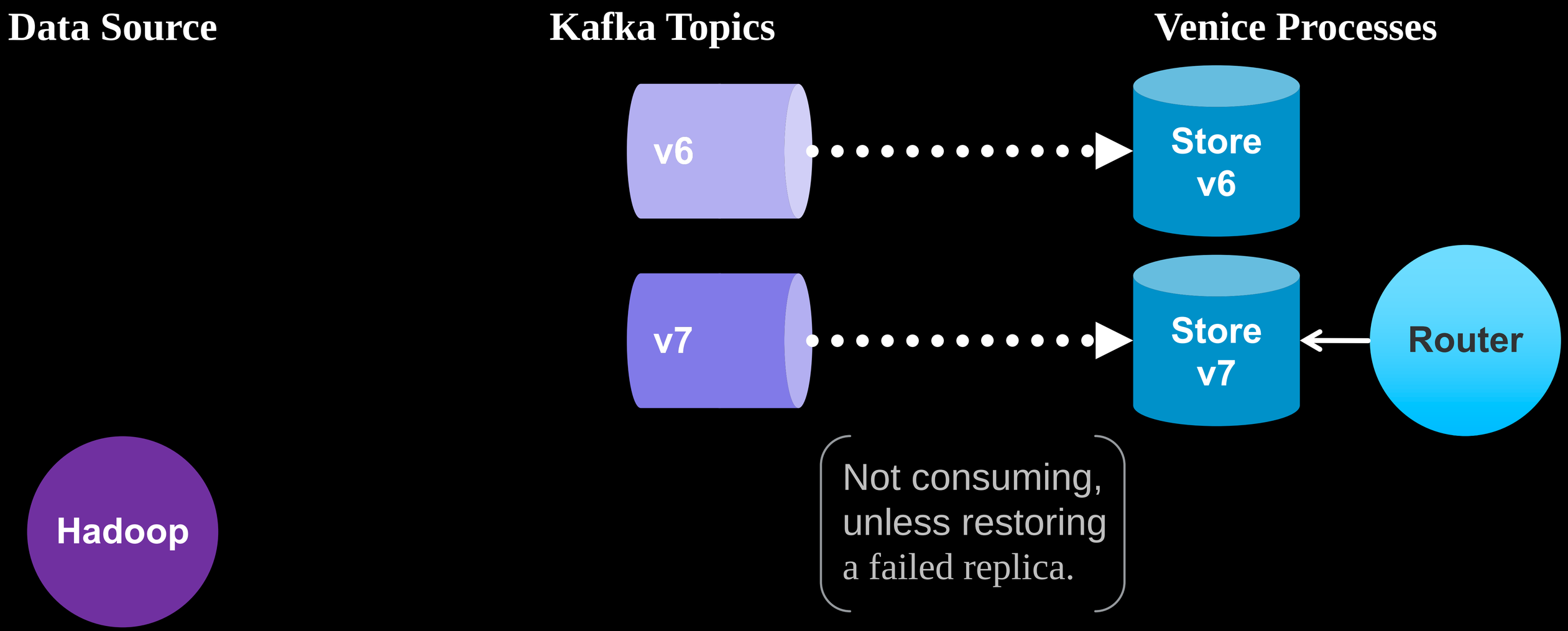
Node failover



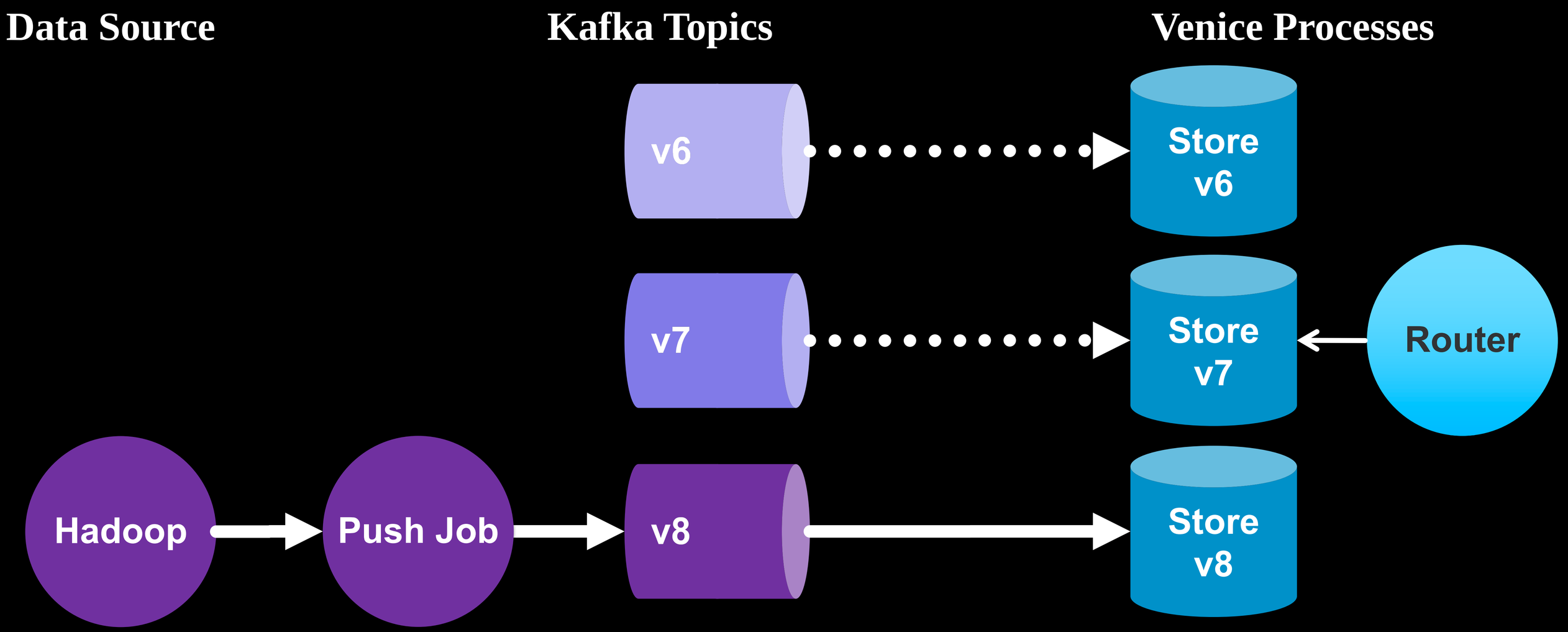
Node failover



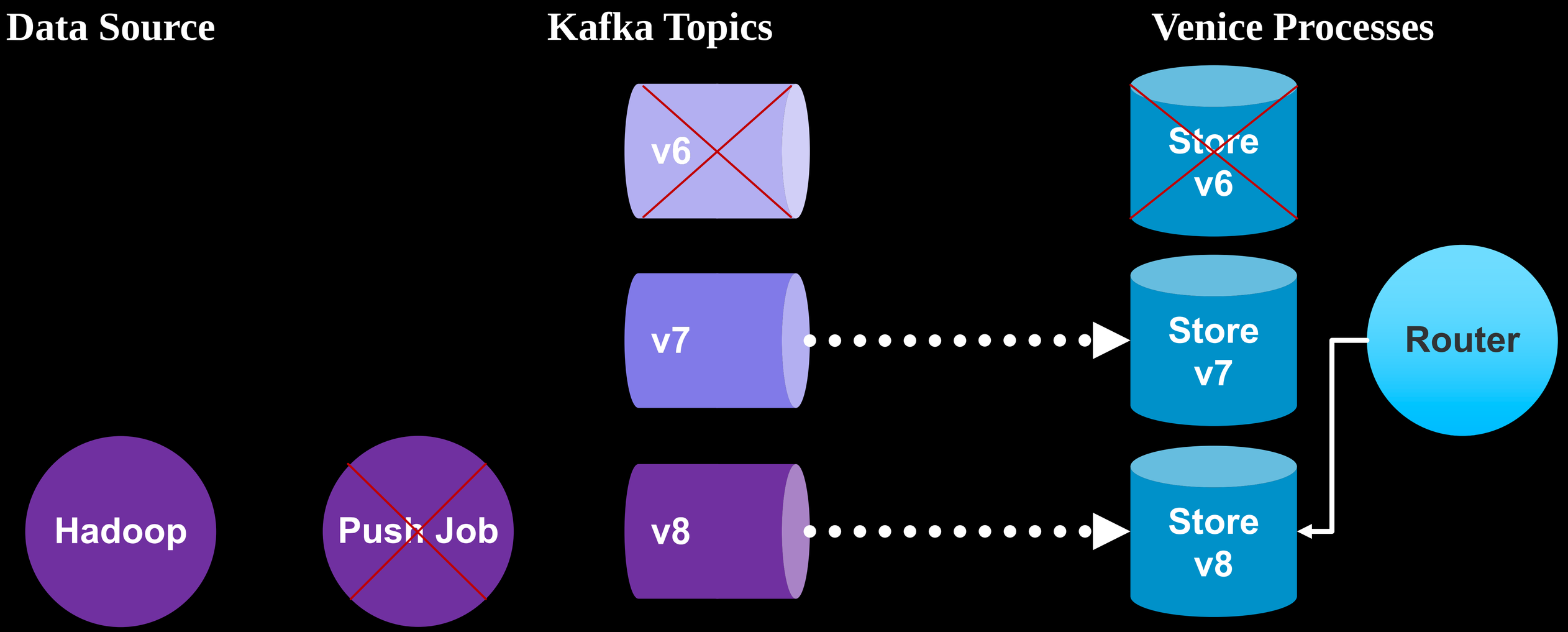
Step 1/3: Steady State, In-between Bulkloads



Step 2/3: Offline Bulkload Into New Store-Version

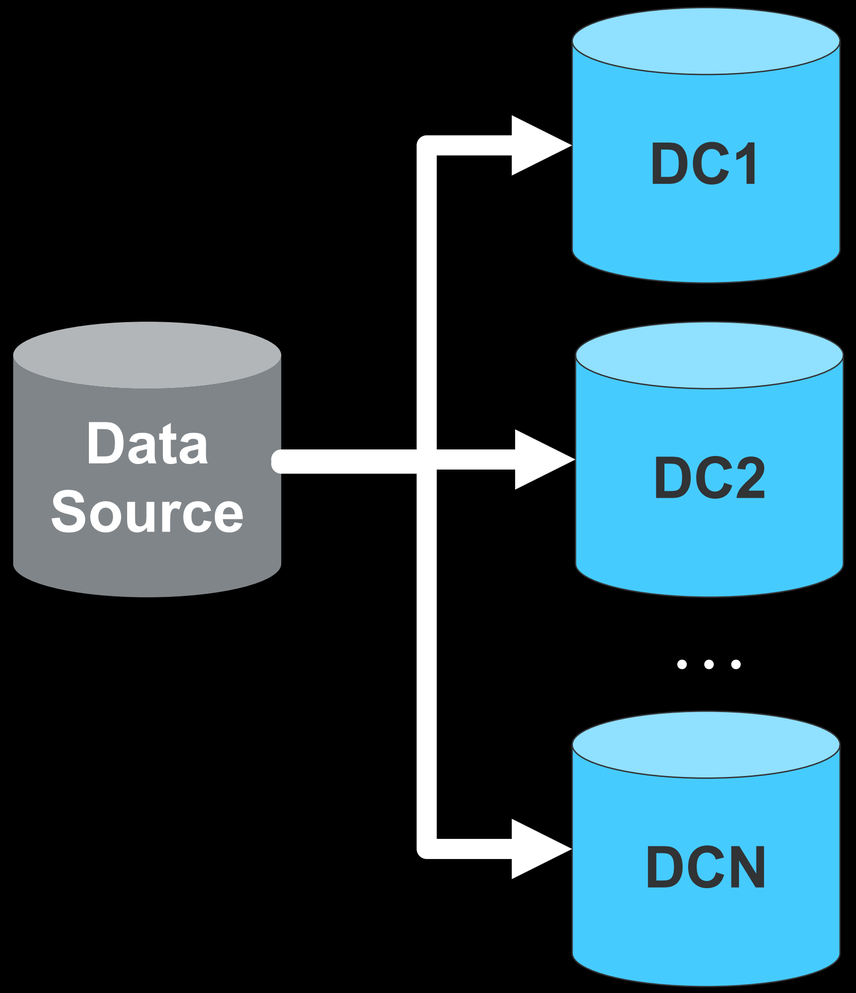


Step 3/3: Bulkload Finished, Router Swaps to New Version

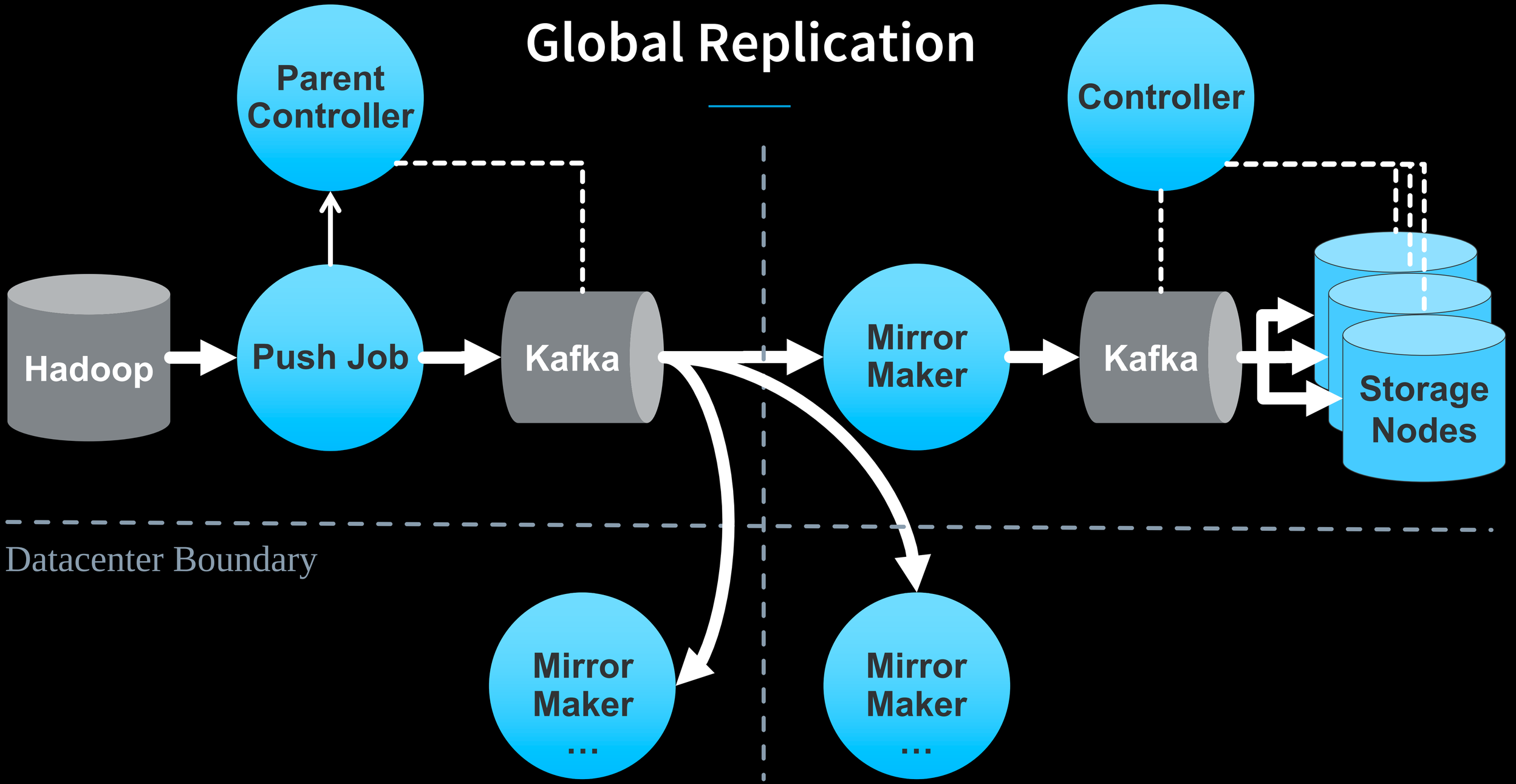


Architecture

Global Replication



Global Replication



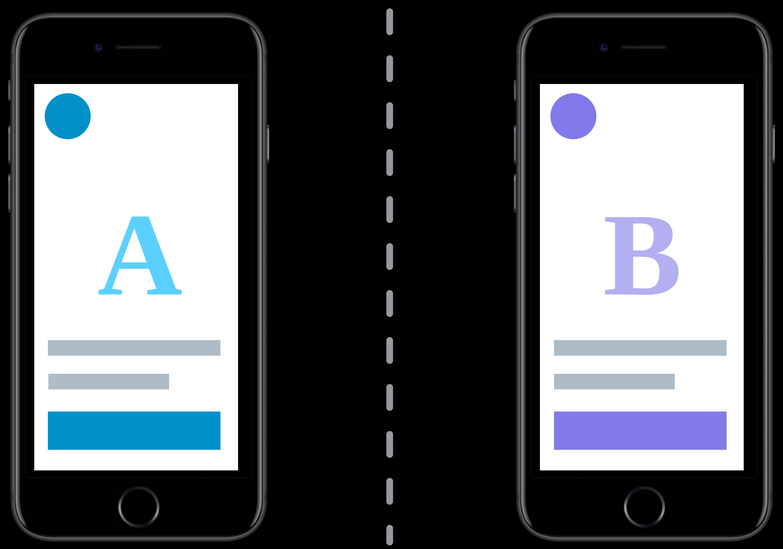
Architecture

Metadata Replication

- Admin operations performed on parent
  - Store creation/deletion
  - Schema evolution
  - Quota changes, etc.
- Metadata replicated via “admin topic”
  - Resilient to transient DC failures

Go faster and faster

hybrid mode



A/B Testing Platform

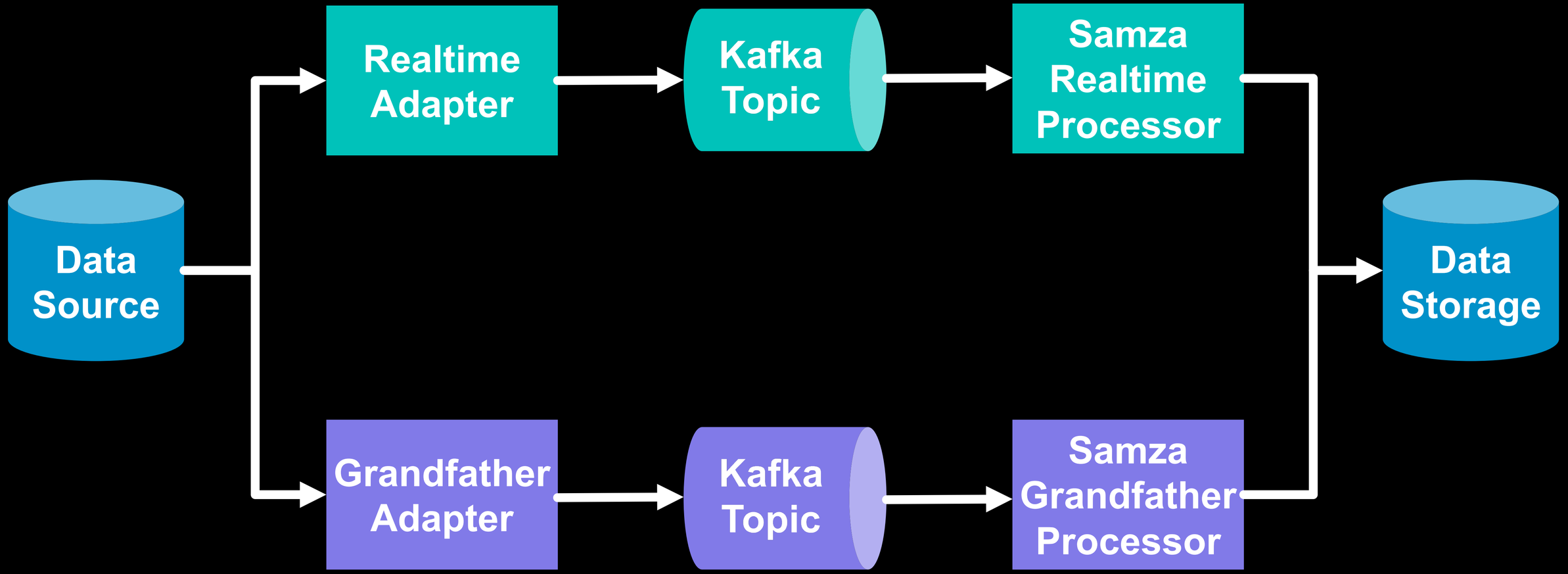
- Experiment
- Ramping up
- Custom selector
- Faster is better

Standardization

Overview

- Member profile – title, position, sills
- Job description – title, seniority, location
- Article topic tagging etc.

Standardization Processing



Hybrid Mode

Overview

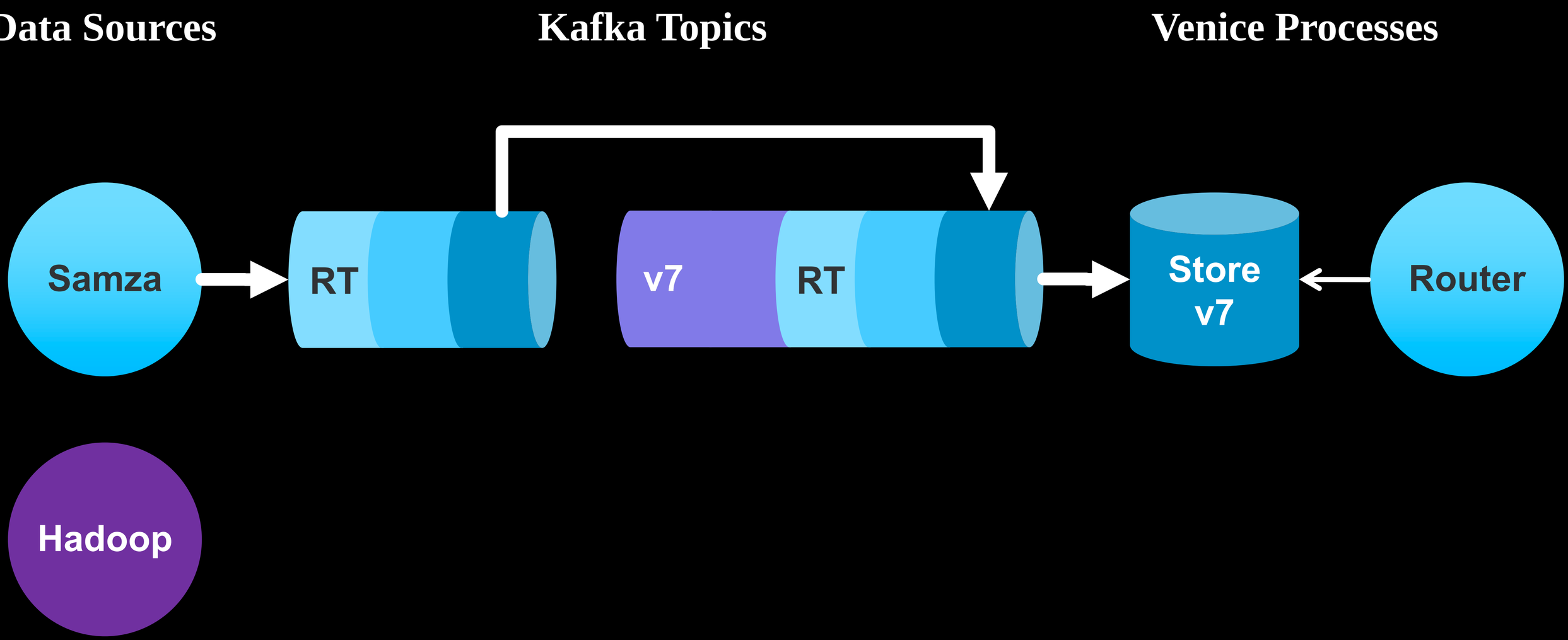
- Hybrid mode aims to
  - Merge batch and streaming data
  - Minimize application complexity
  - Multi-version support

Hybrid Store

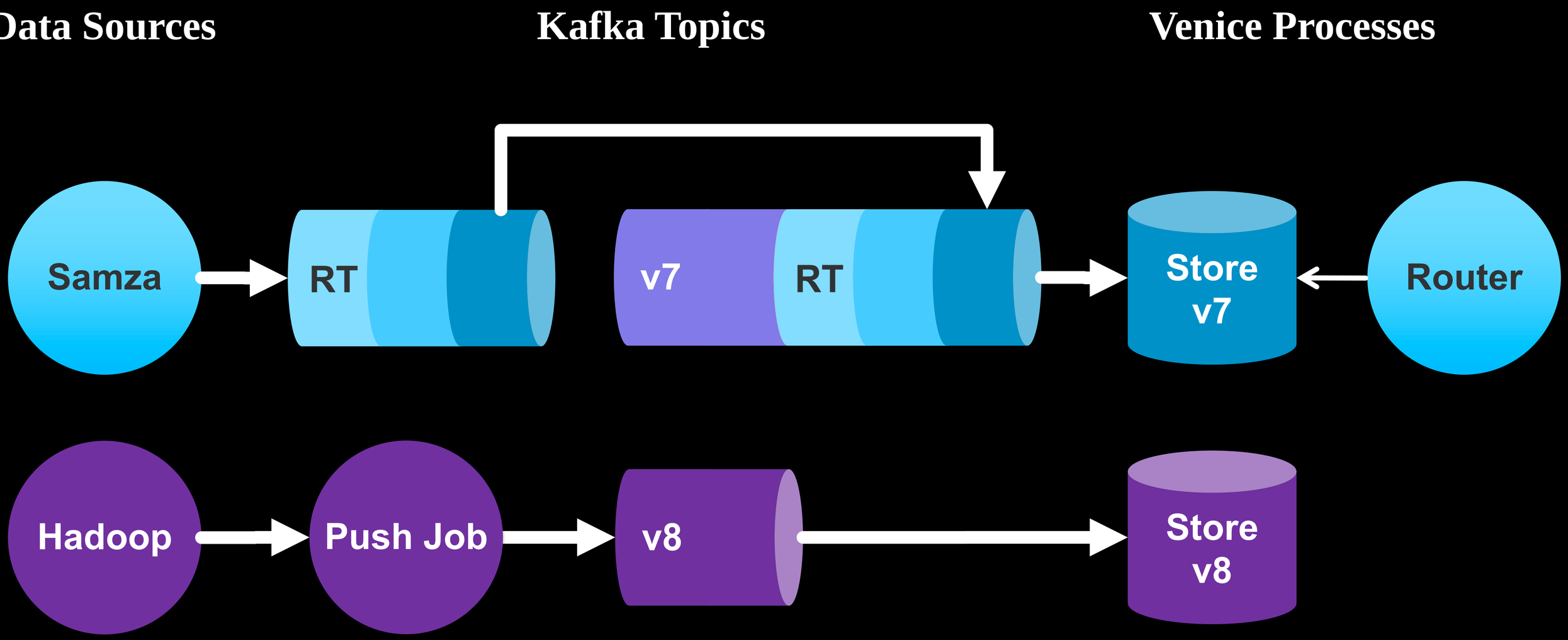
Data Merge

- Write-time merge
- All writes go through Kafka
  - Hadoop writes into store-version topics
  - Samza writes into a Real-Time Buffer topic (RTB)
  - The RTB gets replayed into store-version topics

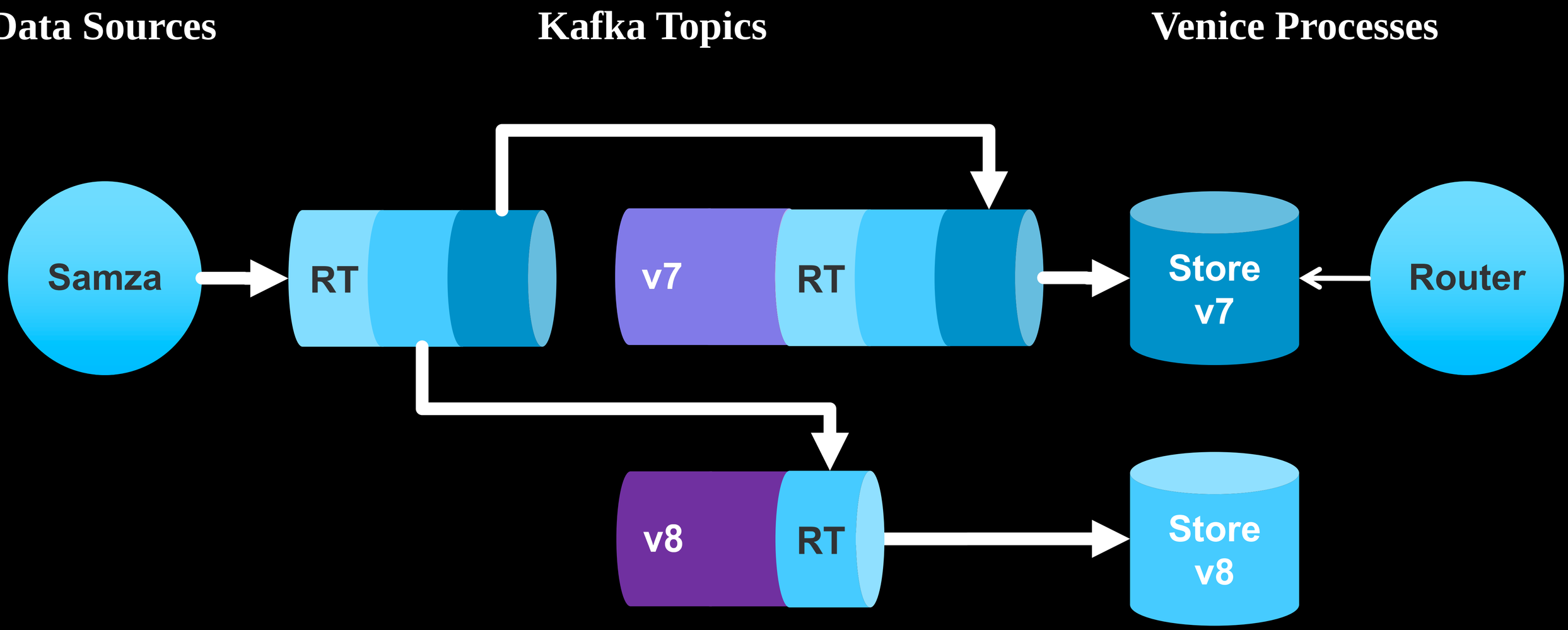
Step 1/4: Steady State, In-between Bulkloads



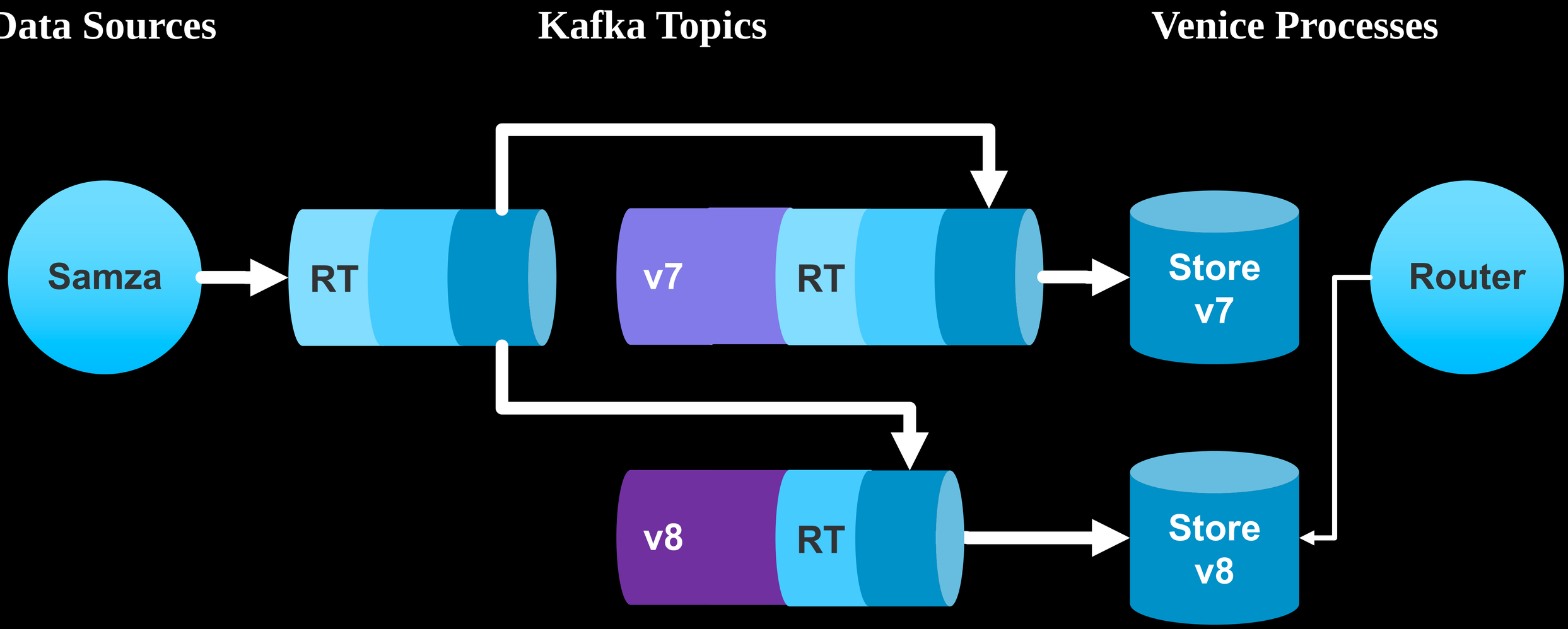
Step 2/4: Offline Bulkload Into New Store-Version



Step 3/4: Bulkload Finished, Start Buffer Replay



Step 4/4: Replay Caught Up, Router Swaps to New Version

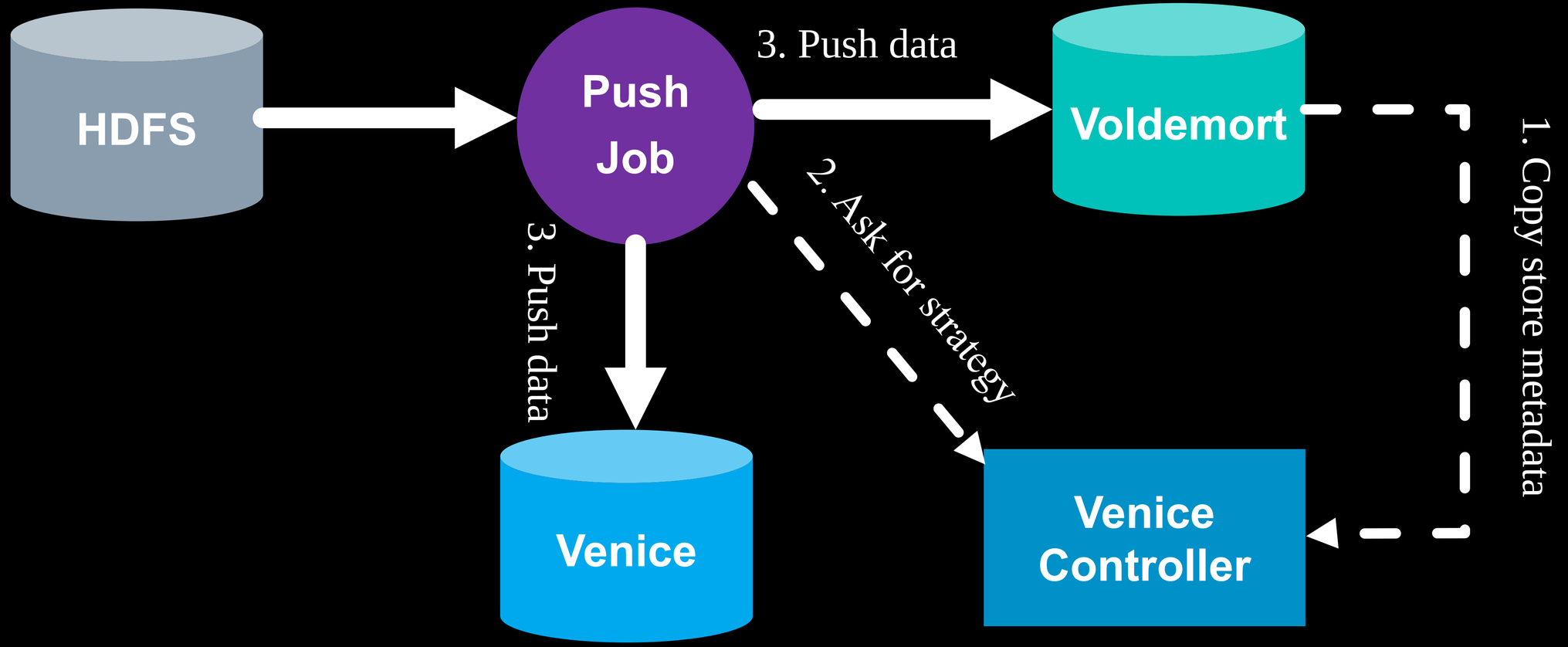


Lessons we learned

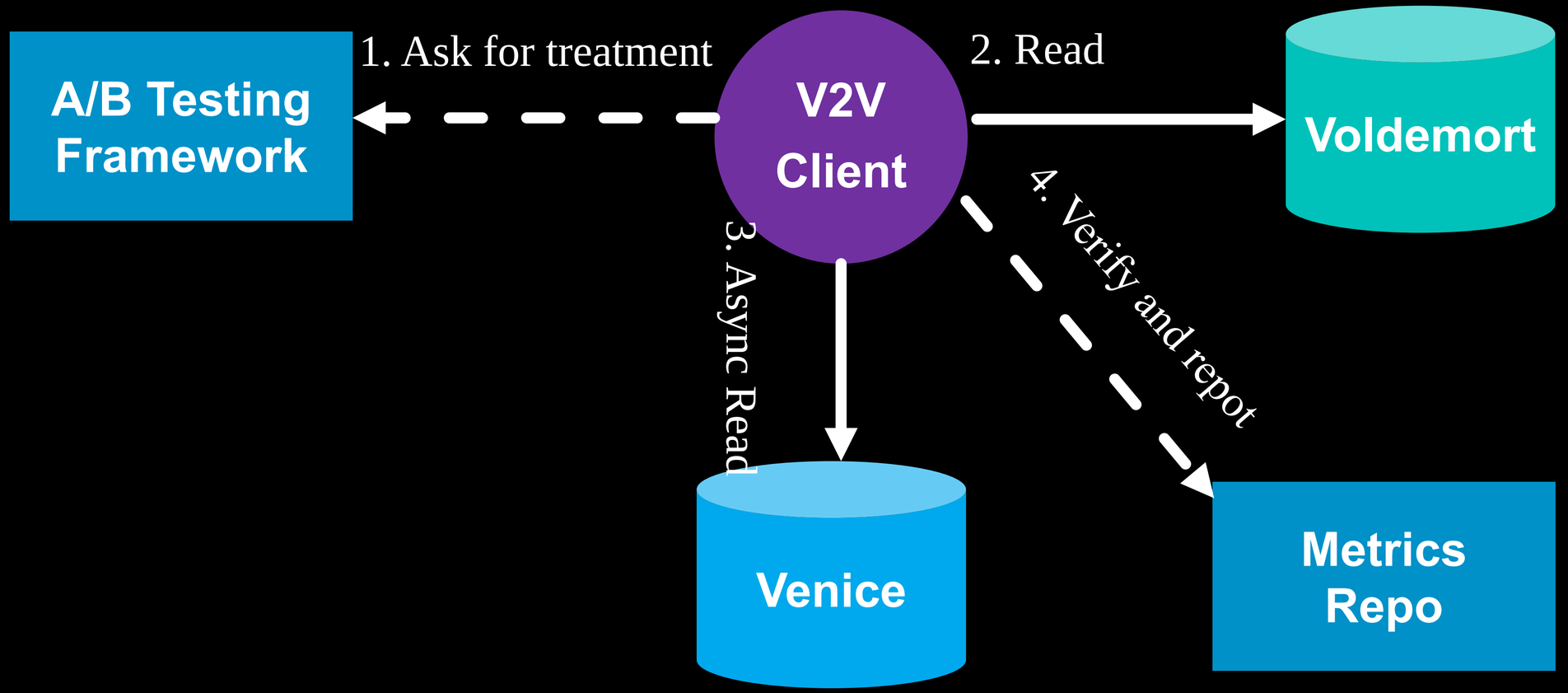
Migration, Latency

Migration is painful

Seamless migration - Dual writes



Seamless migration – Read verify

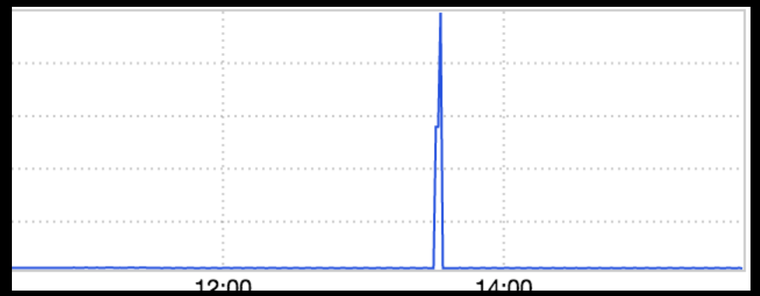


Seamless Migration

Zero code change

- Accelerate the migration
  - Migrated 500 stores in 4 month
  - No code change
  - No extra config on client side
  - Auto cluster discovery

Latency spike would kill the application



Latency spikes

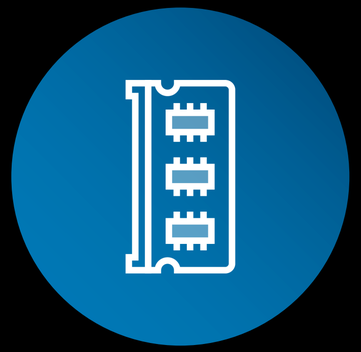
Impact

- Latency sensitive applications
  - A/B testing platform – adds latency to all traffic
  - Ads – lose money

Latency spikes

Sticky routing + Long tail retry

- Sticky routing
  - Try best to send same keys to the same replica
  - Memory efficiency
- Long tail retry
  - Retry once the time cost of the first choice replica exceed its SLA



Latency spike

Cache + Throttling

- Cache
  - Router off heap cache
- Extra health check router->storage node
- Throttling
  - Storage node GC caused by push
  - Throttling on bandwidth usage
  - Throttling on records consumed

Conclusion

Product status, summary

Conclusion

Production Status

- Venice is running in production
  - Batch mode since late 2016
  - Hybrid mode since September 2017
  - Migrated most of Voldemort stores by end of Q1, 2018

Summary

Background

- Primary data vs derived data
- Data lifecycle
- Voldemort’s pain points

Venice

- Features
- Architecture
- Batch mode
- Global replication
- Failover

Faster

- Hybrid mode
- Seamless migration
- Latency improvements

Learn more: engineering.linkedin.com/blog