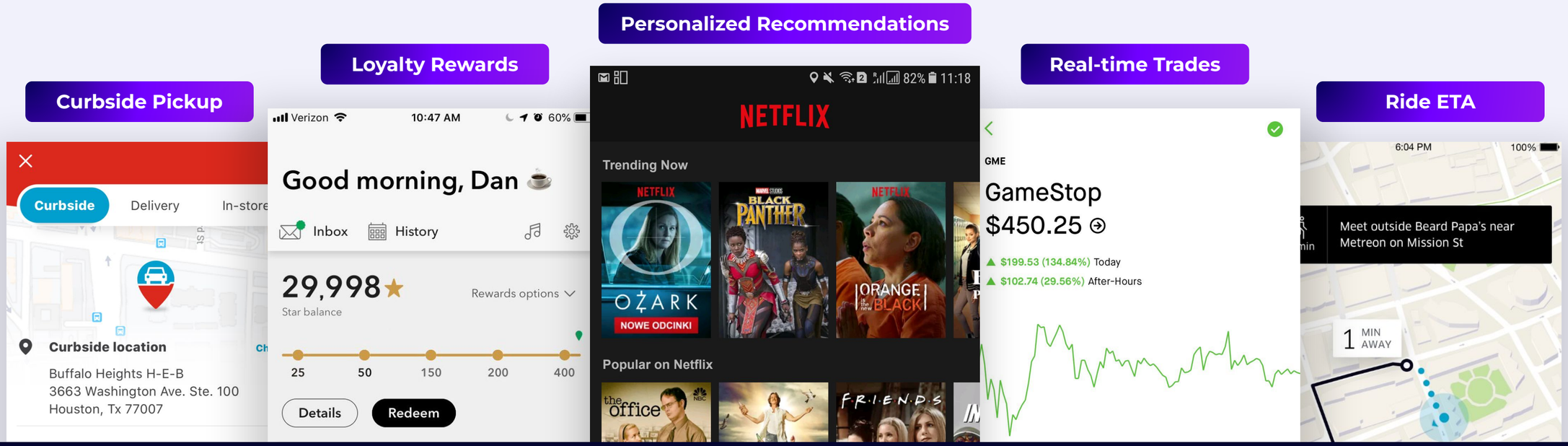
Reinventing Kafka in the Data Streaming Era

Jun Rao, co-founder @ Confluent

Apache Kafka has ushered in the data streaming era...



APACHE kafka®

Created by the founders of Confluent while at LinkedIn

>70%
of the Fortune 500

>100,000+
Organizations

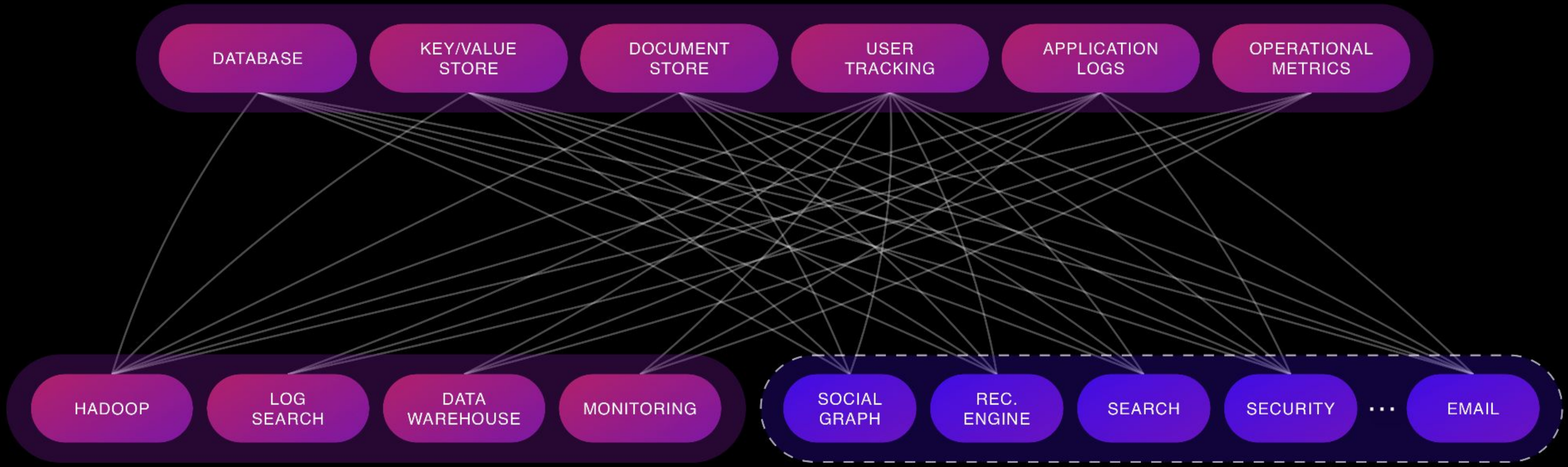
>41,000
Kafka Meetup Attendees

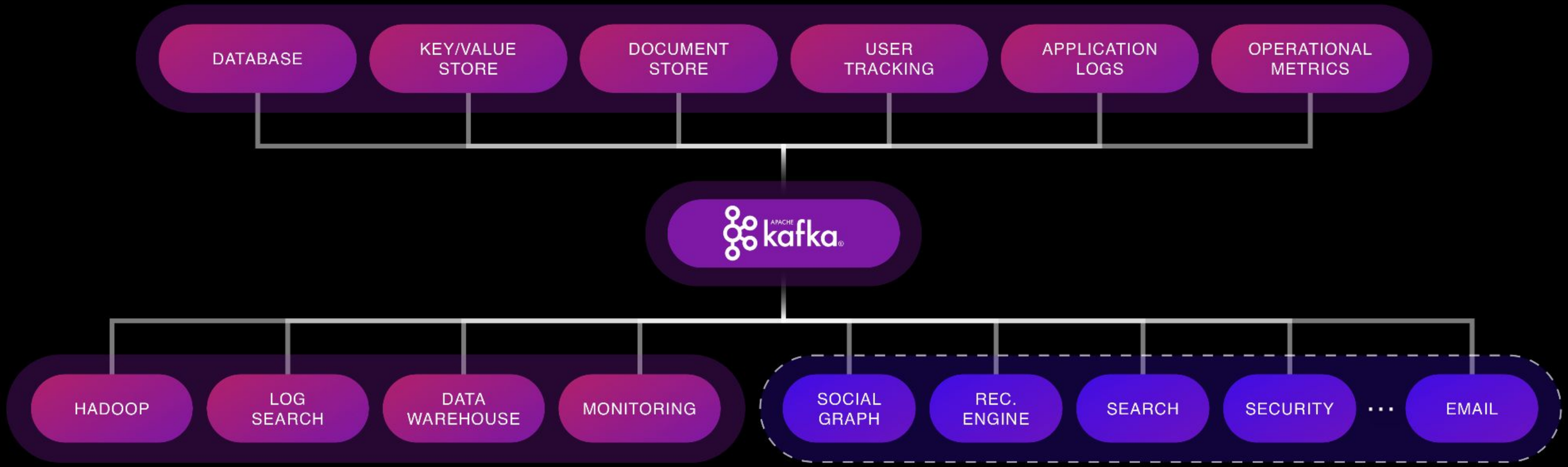
>200
Global Meetup Groups

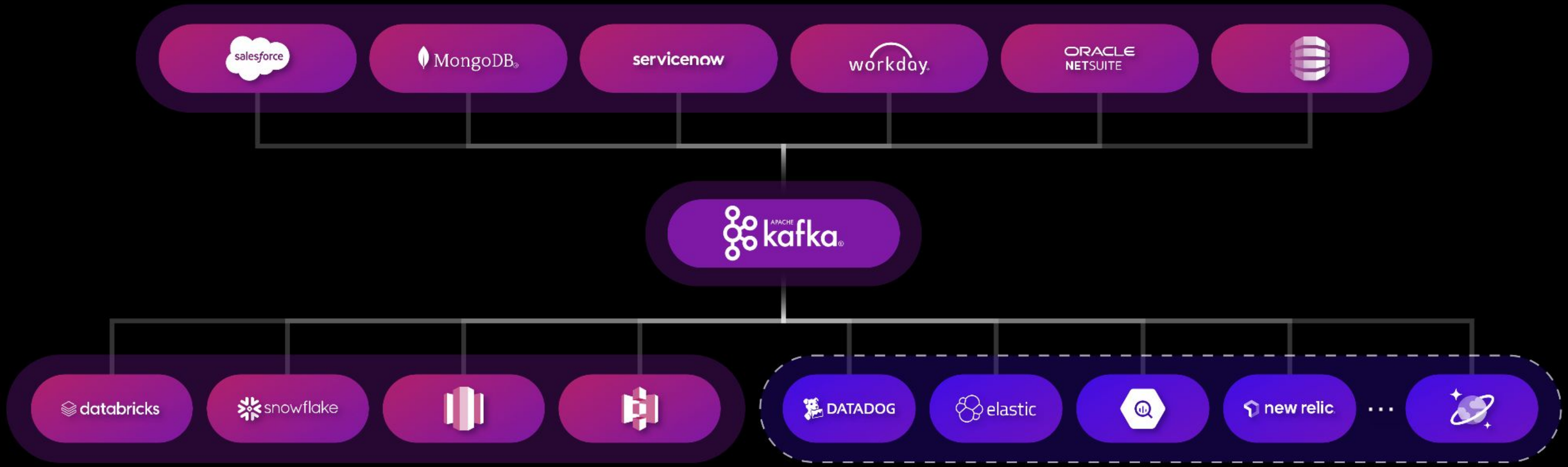
>750
Kafka Improvement Proposals (KIPs)

>12,000
Jiras for Apache Kafka

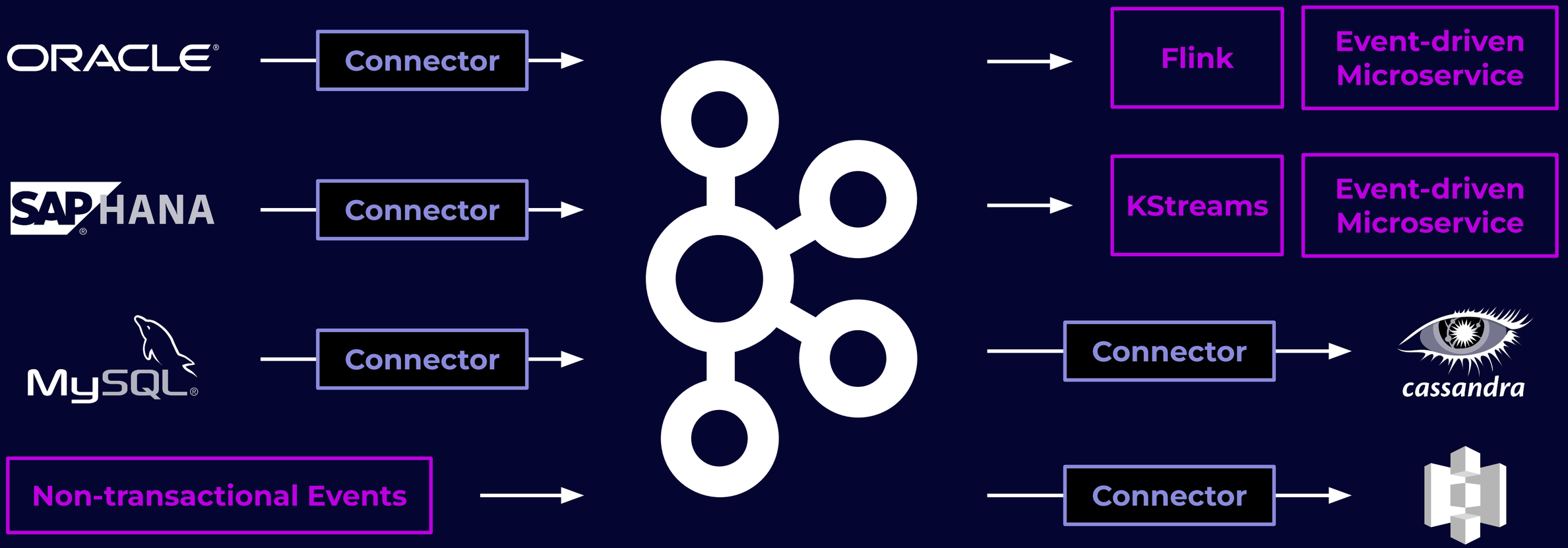
>32,000
Stack Overflow Questions


DATABASE
KEY/VALUE STORE
DOCUMENT STORE
USER TRACKING
APPLICATION LOGS
OPERATIONAL METRICS
HADOOP
LOG SEARCH
DATA WAREHOUSE
MONITORING
SOCIAL GRAPH
REC. ENGINE
SEARCH
SECURITY
• • •
EMAIL


DATABASE
KEY/VALUE STORE
DOCUMENT STORE
USER TRACKING
APPLICATION LOGS
OPERATIONAL METRICS
APACHE kafka
HADOOP
LOG SEARCH
DATA WAREHOUSE
MONITORING
SOCIAL GRAPH
REC. ENGINE
SEARCH
SECURITY
• • •
EMAIL


salesforce
MongoDB.
servicenow
workday.
ORACLE NETSUITE
APACHE kafka
databricks
snowflake
DATADOG
elastic
new relic
• • •

Defacto Data Streaming Platform With Open API/Protocol



Continuous Innovation in This Open Platform

Apache Kafka 3.6 (Sep 2023)

KIP-405: Tiered Storage

KIP-866: ZooKeeper to KRaft migrations (GA)

KIP-833: Delegate Tokens in KRaft

Apache Kafka 3.7 (Jan 2024)

KIP-858: JBOD in KRaft

KIP-650: Enhanced Raft Semantics

KIP-714: Client Metrics and Observability

KIP-853: KRaft Voter Changes

Apache Kafka 4.0 (Apr 2024)

ZK Removed

KIP-848: New Consumer Group Protocol

Future Ideas

KIP-905: Broker Interceptors

KIP-932: Queues for Kafka

KIP-939: Support Partition in 2PC

Topic Directories and Hierarchies

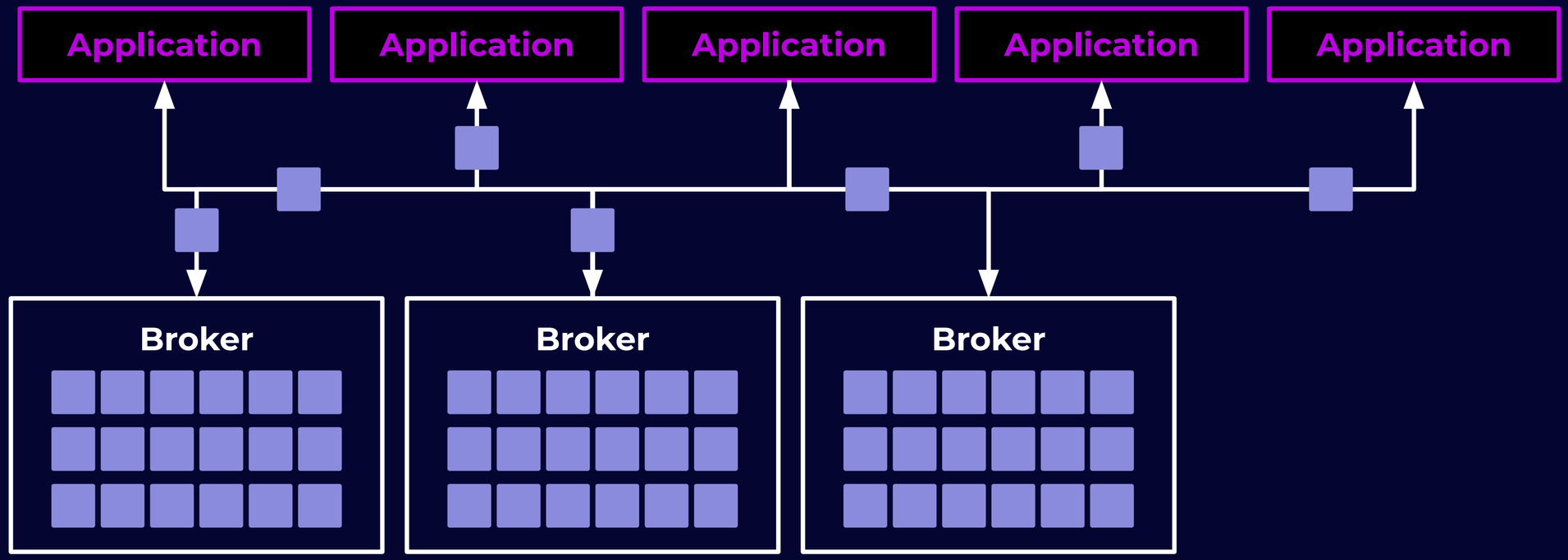
Autoscaling or Partition-less Topics

Simplified Protocol, Better Clients


Kafka as a Service


PUTTING KAFKA IN THE CLOUD...
ISN’T JUST PUTTING KAFKA IN THE CLOUD.

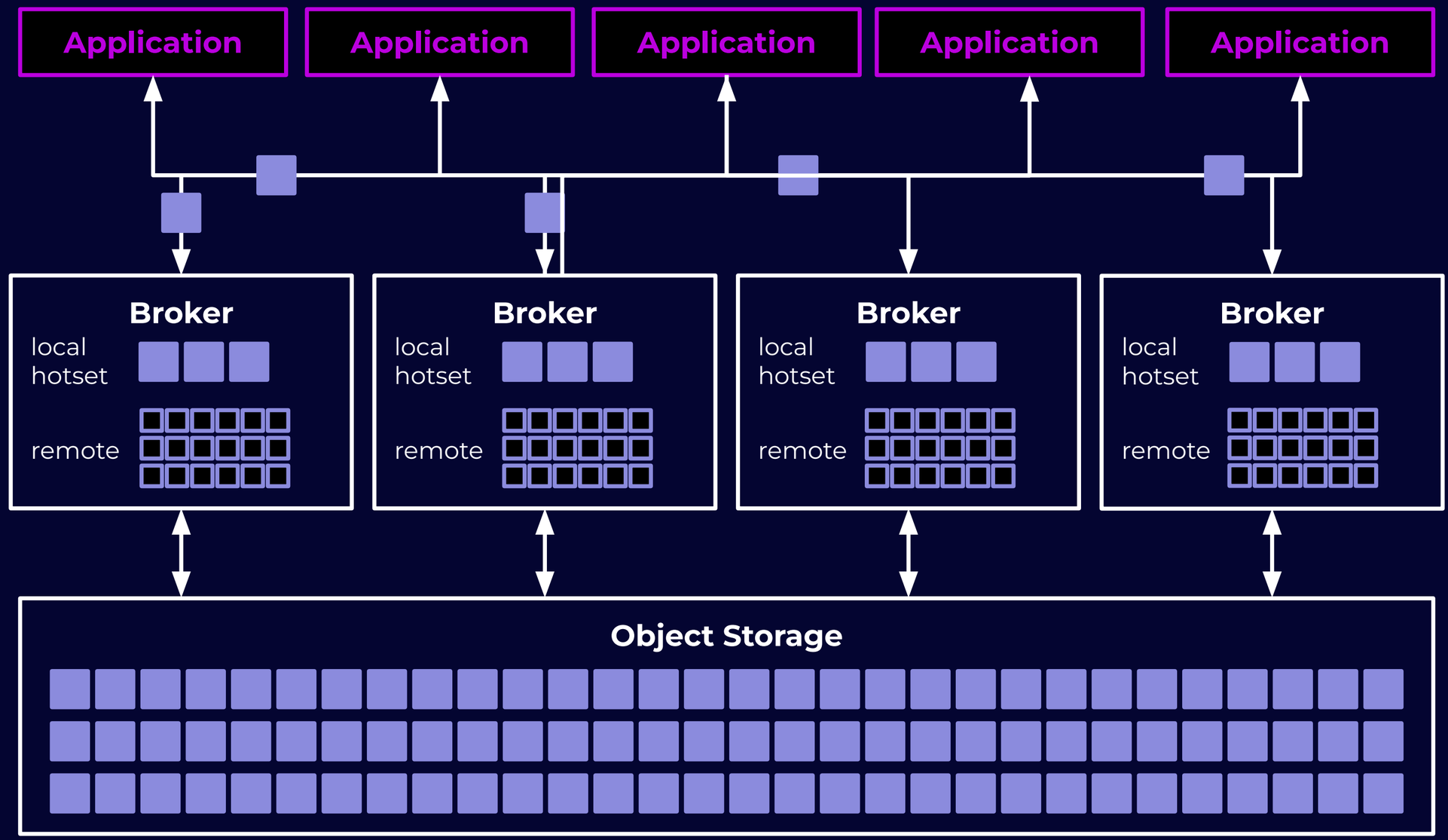
How Kafka Stores Data



Cost

Elasticity

KIP-405 Tied Storage

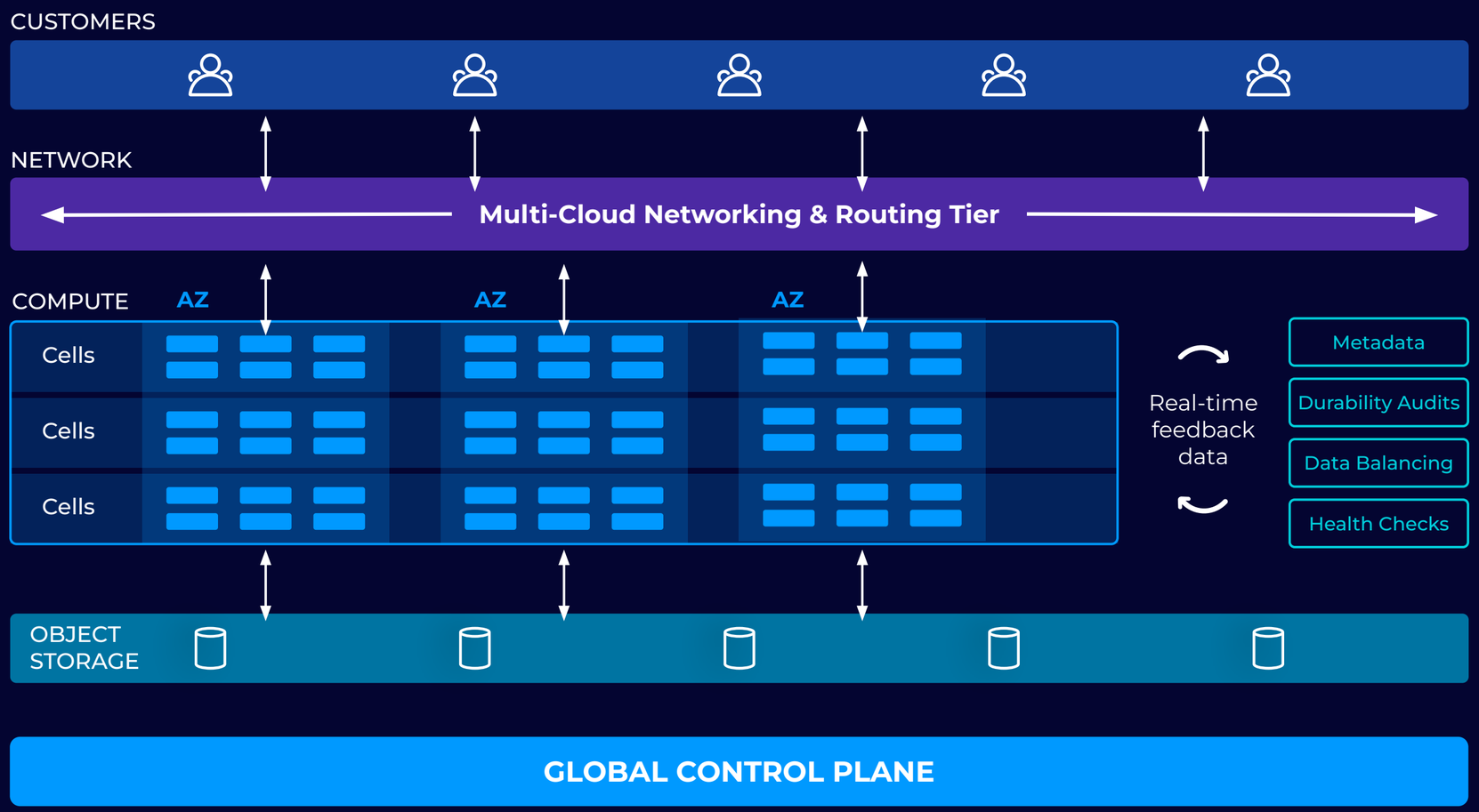


Cost Efficiency

Improved Elasticity

Kora: The Cloud-Native Apache Kafka® Engine

- Best industrial paper in VLDB 2023
- Kora feature
  - Tiered storage
  - Transparent maintenance
  - Auto balancing
  - Degradation handling
  - Disaggregation
  - Isolation
  - Optimized network
  - Multi-AZ cross region replication
  - Self validation
- Benefits
  - Quality of service
  - Lower cost



Why Developers Choose Flink

Elastic Scalability

Flink is capable of supporting stream processing workloads at tremendous scale

Language Flexibility

Flink supports Java, Python, & SQL, enabling developers to work in their language of choice

Unified Processing

Flink supports stream processing, batch processing, and ad-hoc analytics through one technology

Flink is a top 5 Apache project and has a very active community

Common Pattern in Flink Job

Write State to Flink Store

Write Log to Kafka

Want atomicity of dual writes to support EoS

2 Phase Commit Refresher

- Prep phase:
  Coordinator asks each participant to prepare to commit

- Complete phase:
  Coordinator commits/aborts based on responses in prep phase

KIP-939 Support 2 Phase Commit

Kafka Already Has Txn Support

- Create Kafka producer
- Start Txn
- Send records to topic1-partition1
- Send records to topic1-partition2
- Commit/Abort Txn

Gaps in Kafka Txn for 2PC

Restarted producer auto aborts any ongoing Txn

Txn can be aborted after timeout

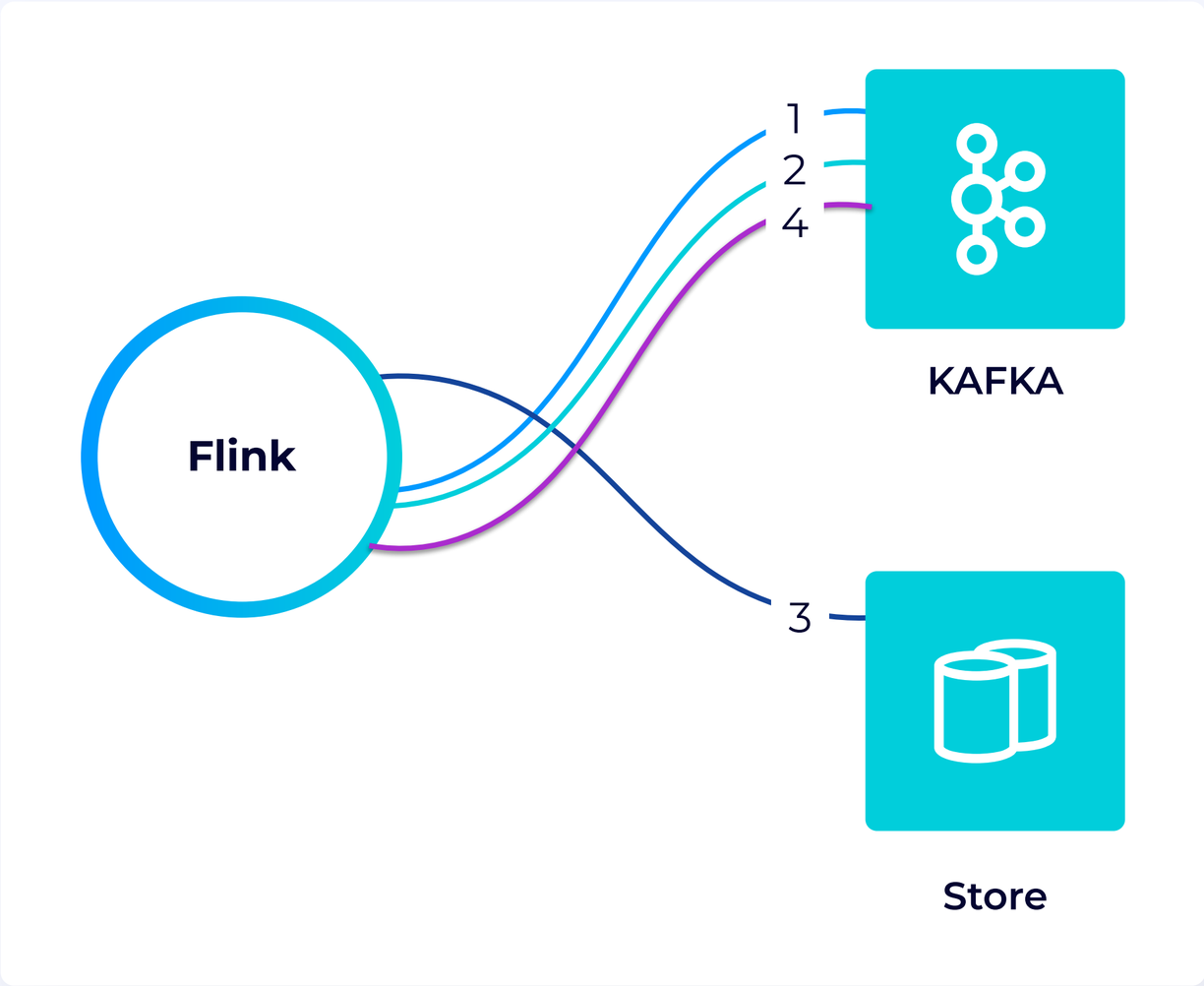
KIP-939 Support 2 Phase Commit

Restarted producer auto aborts any ongoing Txn

Txn can be aborted after timeout

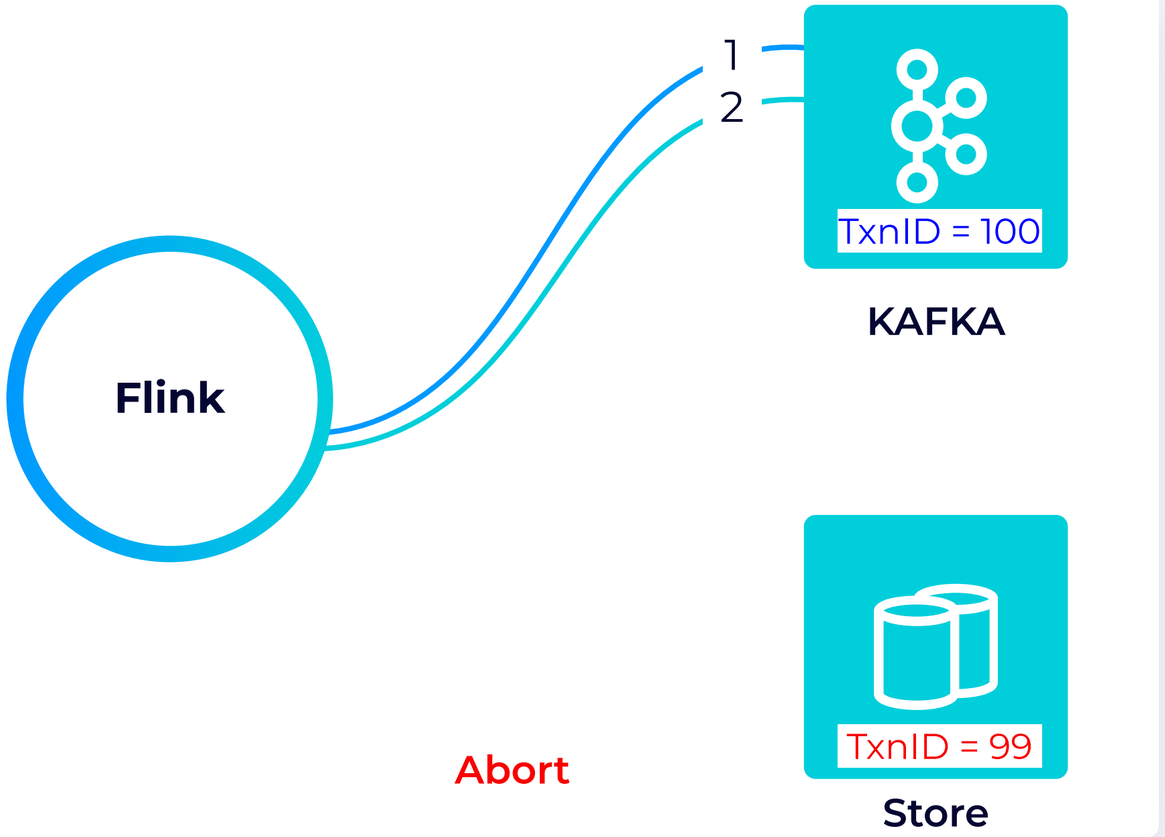
- Disable auto abort, if 2PC enabled
- TxnID prepareTxn()
- completeTxn(TxnID)

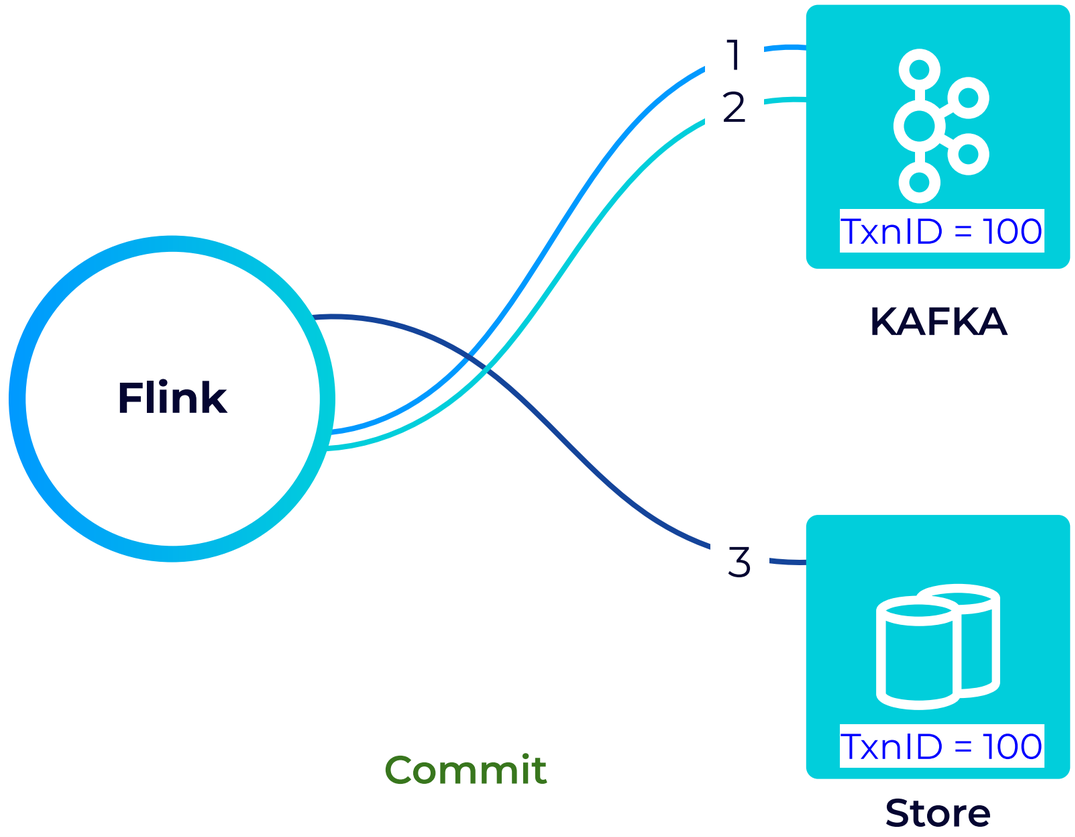
Normal Flow in Flink Application



1. Write log records to Kafka (with 2PC enabled)
2. TxnID = prepareTxn() in Kafka
3. Write state and TxnID to store atomically
4. commitTxn() in Kafka

What if Flink fails in between





Recovery process on Flink failure:

1. Retrieve last TxnID from store

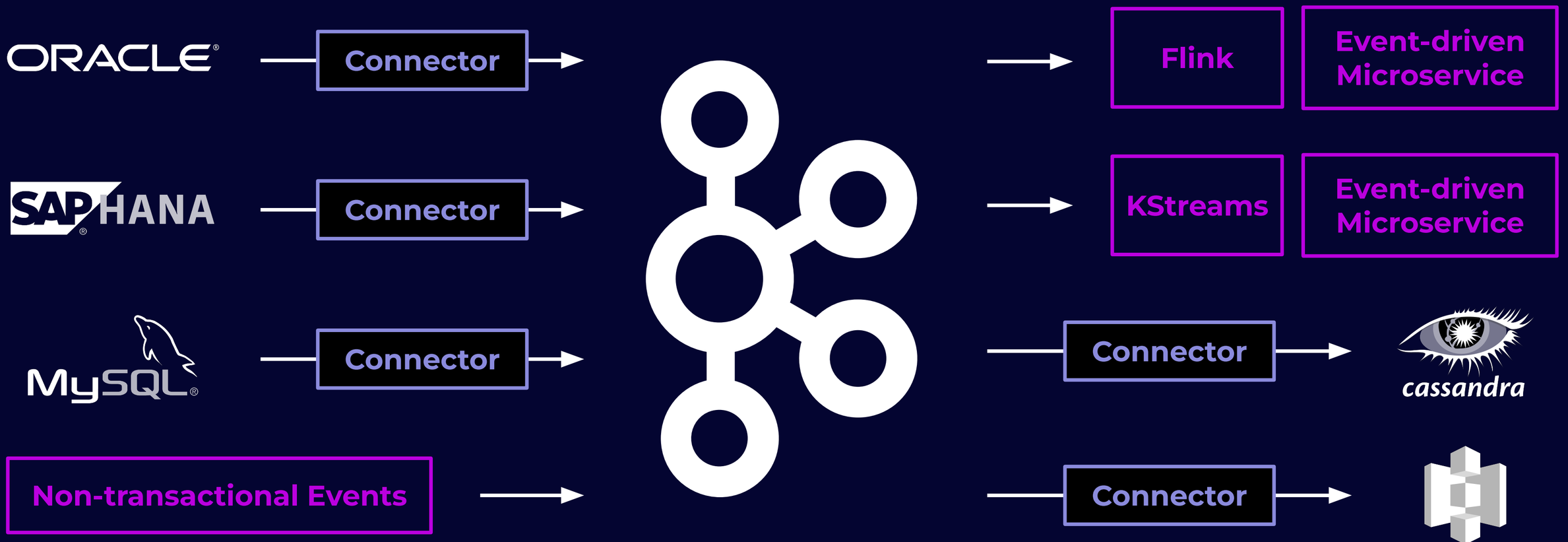
2. Call completeTxn(TxnID) in Kafka
   a. compare with last TxnID from broker
   b. If match, commit; otherwise, abort

FLIP-319 – Integrating with Kafka 2PC Transactions

Hardens EOS
Across Flink and Kafka

Simplifies Upgrading
Kafka Clients Used by Flink

Thank You


ORACLE
Connector
SAP HANA
Connector
MySQL
Connector
Non-transactional Events
Flink
Event-driven Microservice
KStreams
Event-driven Microservice
Connector
cassandra
Connector