
GREAT INDIAN
DEVELOPER
SUMMIT
Enterprise Message Bus

Michael André Pearce
Enterprise Systems Architecture

- Messaging
- Distributed Grids
- Real time Big Data
- Open Source Evangelist
  - Apache ActiveMQ – Committer
  - Registry (Hortonworks) - Committer
  - Apache Kafka – Contributor

GREAT INDIAN DEVELOPER SUMMIT™

OUR KEY NUMBERS

£491.1 MILLION

Net global trading revenue of £491.1 million (May 2017)

14

Sales offices in 14 countries across five continents

£2.1 BILLION

FTSE 250 company with market capitalisation of £2.1 billion (31 May 2017)

185,800

185,800 clients worldwide (May 2017)

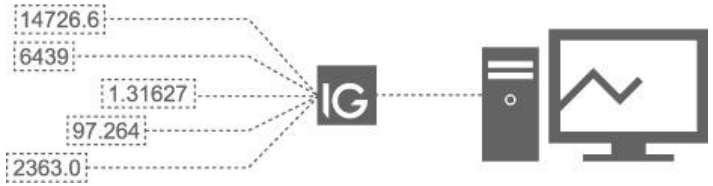
1500

Over 1500 staff around the world (May 2017)

GREAT INDIAN DEVELOPER SUMMIT™

Platforms

labs.ig.com
REST trading and streaming APIs



Enterprise Message Bus

1. Goal
2. Solution
3. Apache ActiveMQ Artemis
4. Apache Kafka

"With modern **Technology** data has become a commodity more than ever before.

The biggest hurdle for any organisation is how to get the data you already have to somewhere where you can start to build value from that data."

GREAT INDIAN DEVELOPER SUMMIT™

Goal

An unified platform flowing

transactional **Messages** into

process-able **Streams** into

Long lived query-able data

Solution

One single technology alone is insufficient

But lots of **building blocks** exist

Need to compose them simply and efficiently

Building Blocks

Data Messaging

RedHat AMQ-7
Apache ActiveMQ Artemis

Data Streaming

Apache Kafka

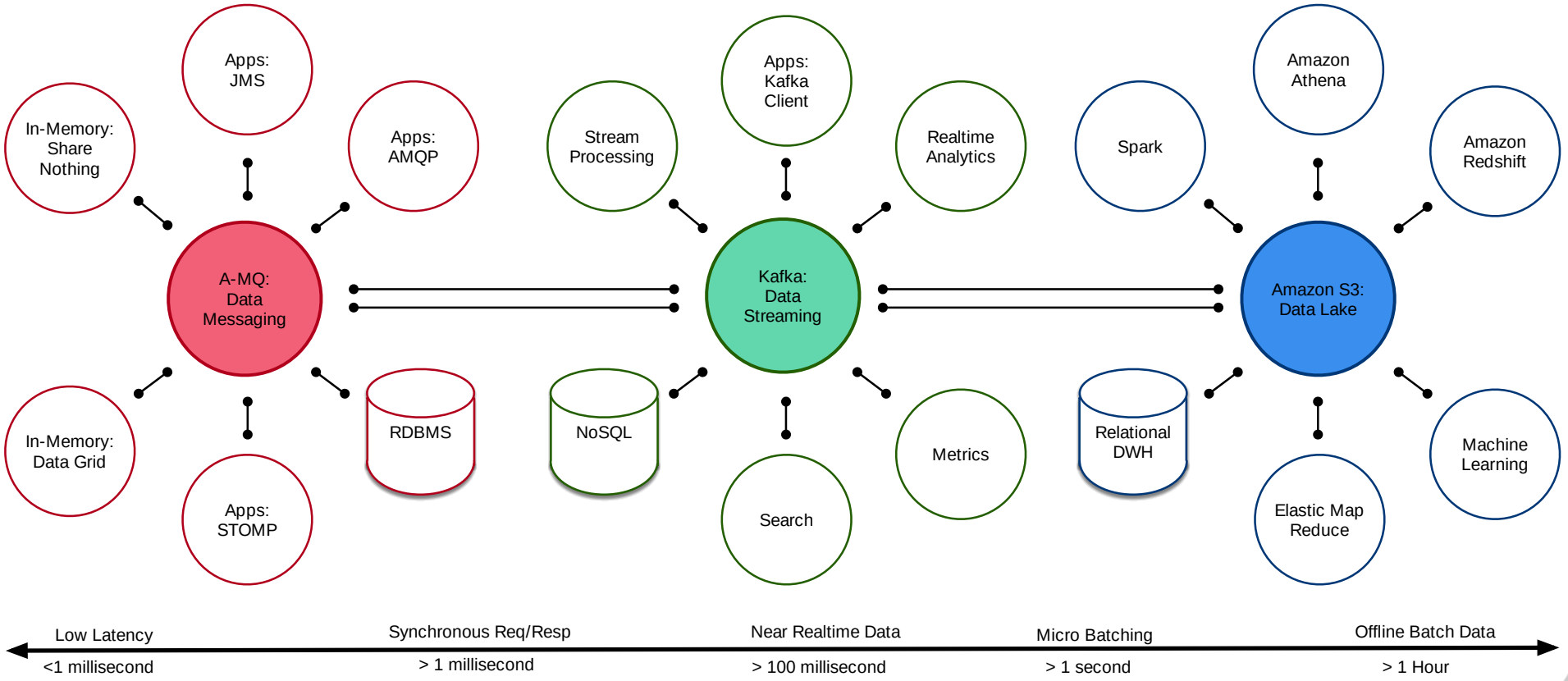
Data Lake

Amazon S3

Data Schema

Hortonworks Schema Registry

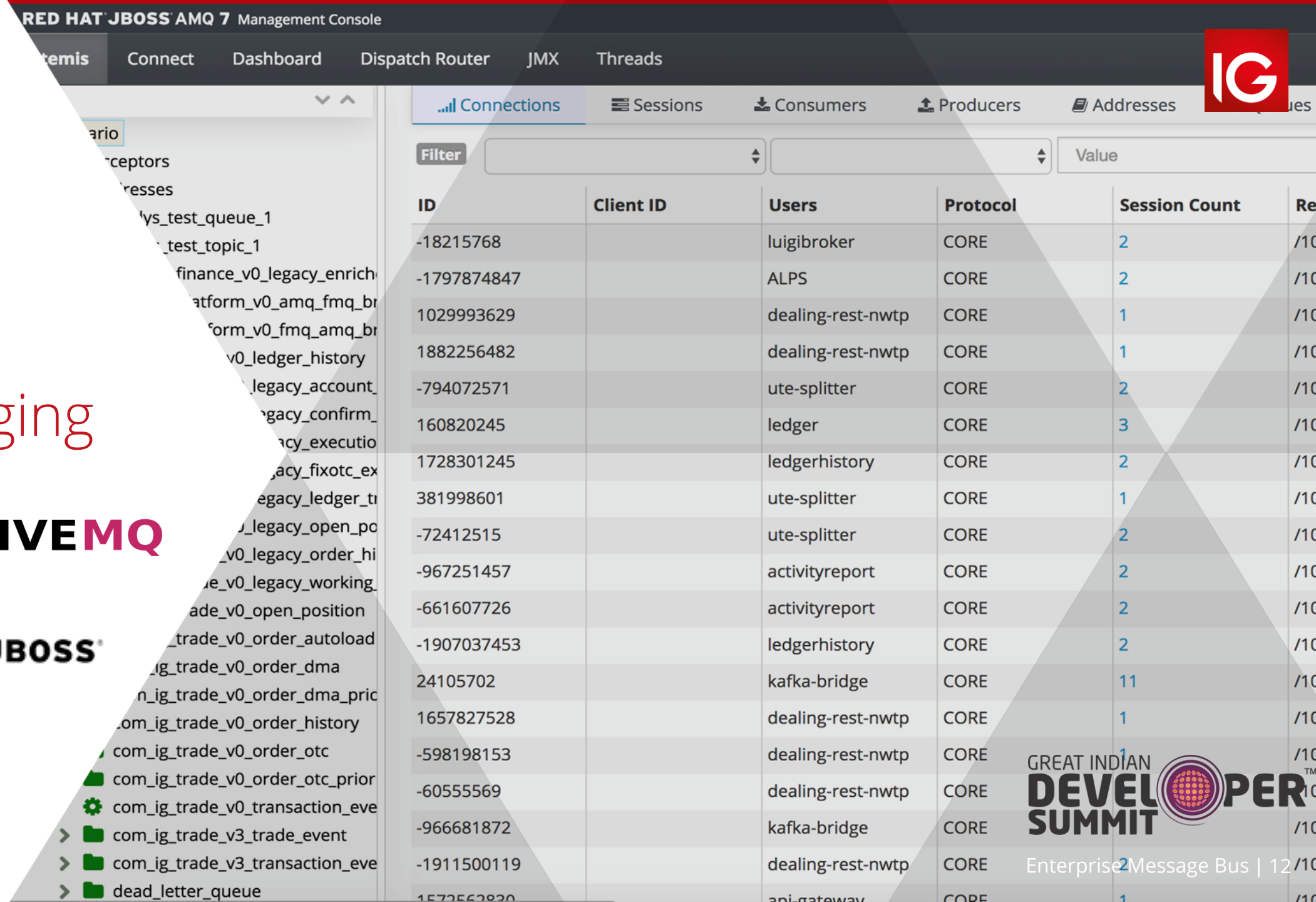
Building Blocks



Data Messaging

APACHE ACTIVEMQ

RED HAT JBOSS AMQ



GREAT INDIAN DEVELOPER SUMMIT

Red Hat AMQ

Power of Open Source

Apache Qpid

AMQP wire protocol implementation, Interconnect (Dispatch Router), JMS, reactive APIs, multiple language clients

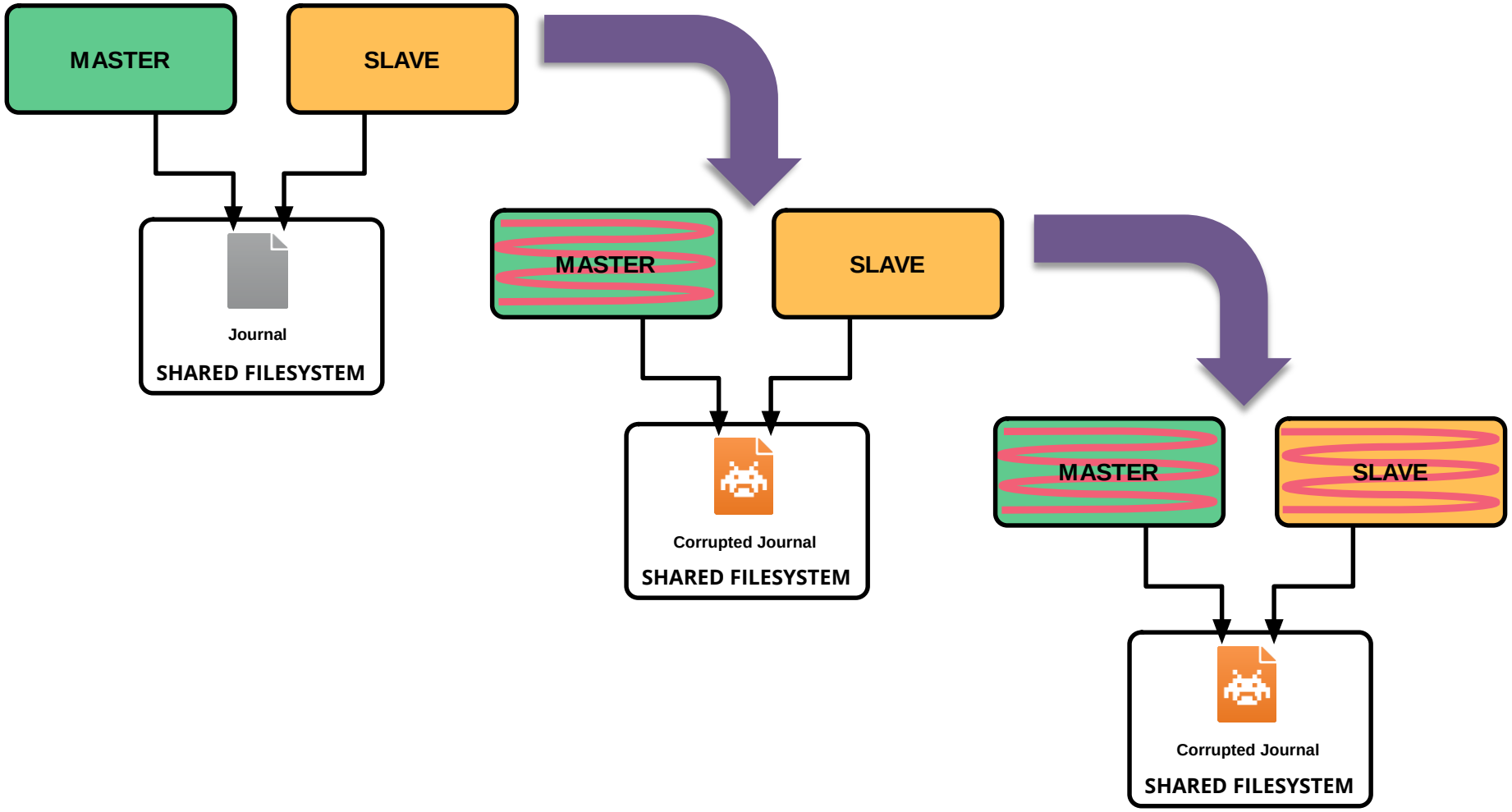
Apache ActiveMQ Artemis

Async core, fast persistence, high availability, scalability, JMS, AMQP support, MQTT support

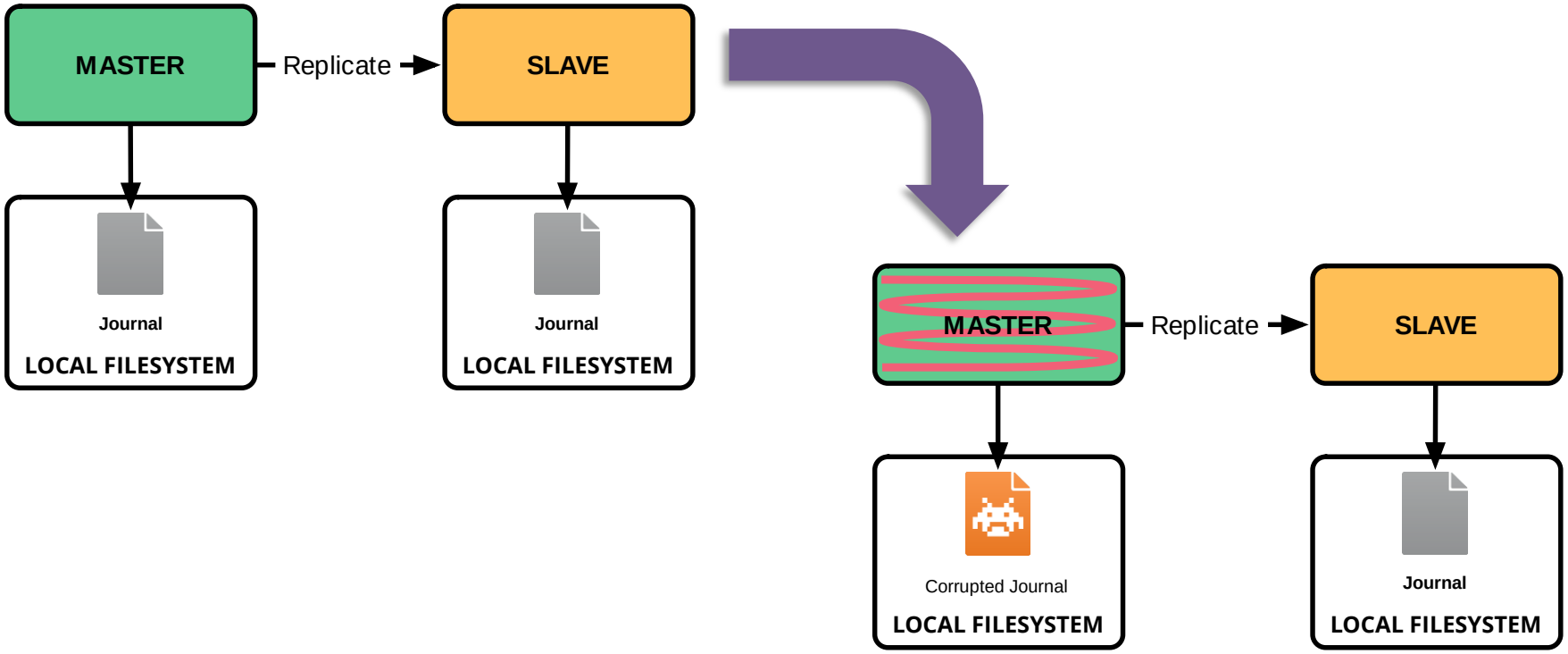
Hawtio

Real-time web console for interacting with the broker and real-time inspection of the router

Corrupted Journal – Shared Filesystem

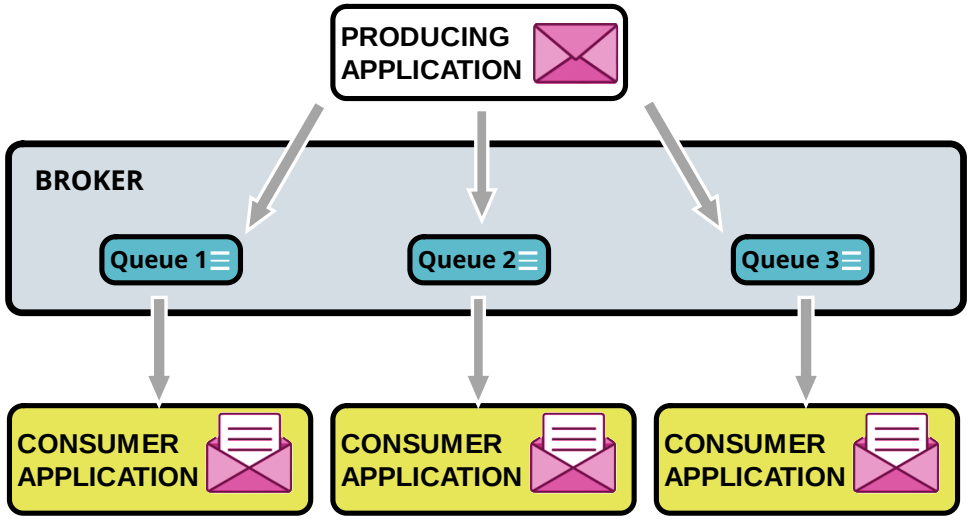


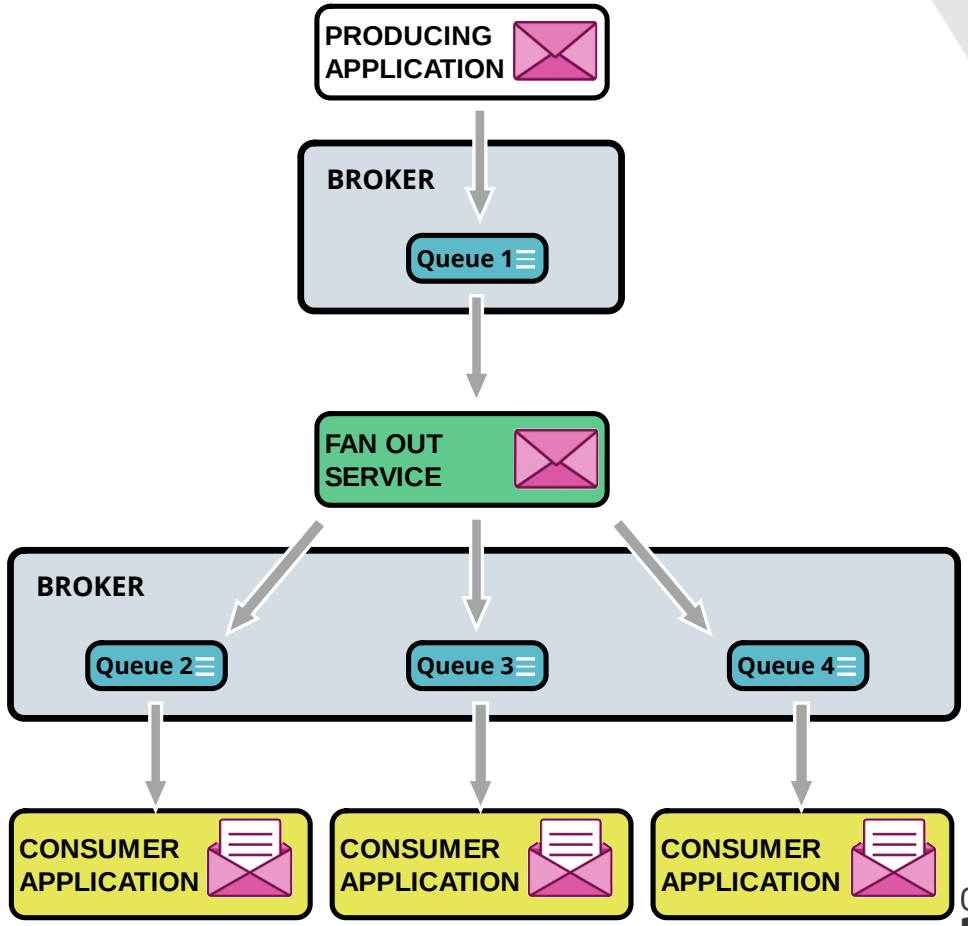
Corrupted Journal - Replication



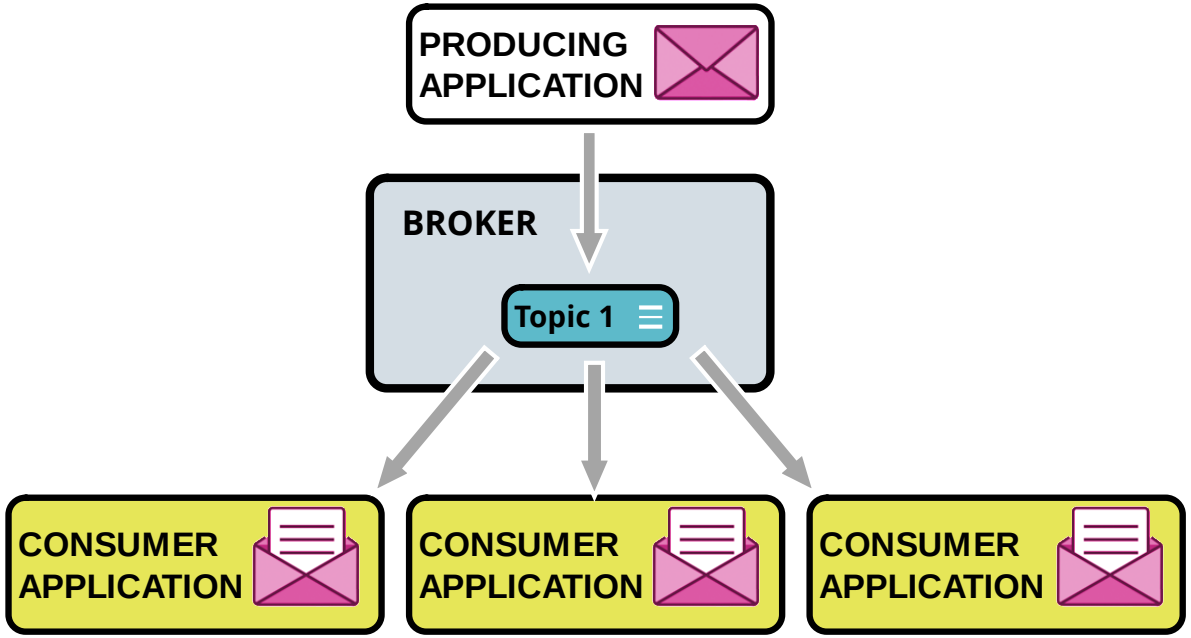
- ✓ Apache ActiveMQ Artemis
- ✓ Apache Kafka

Event Data Publishing with Queues





Event Data Publishing with Topics



- ✓ Apache ActiveMQ Artemis
- ✓ Apache Kafka

Why not Apache Kafka?

Transaction Processing Requirements

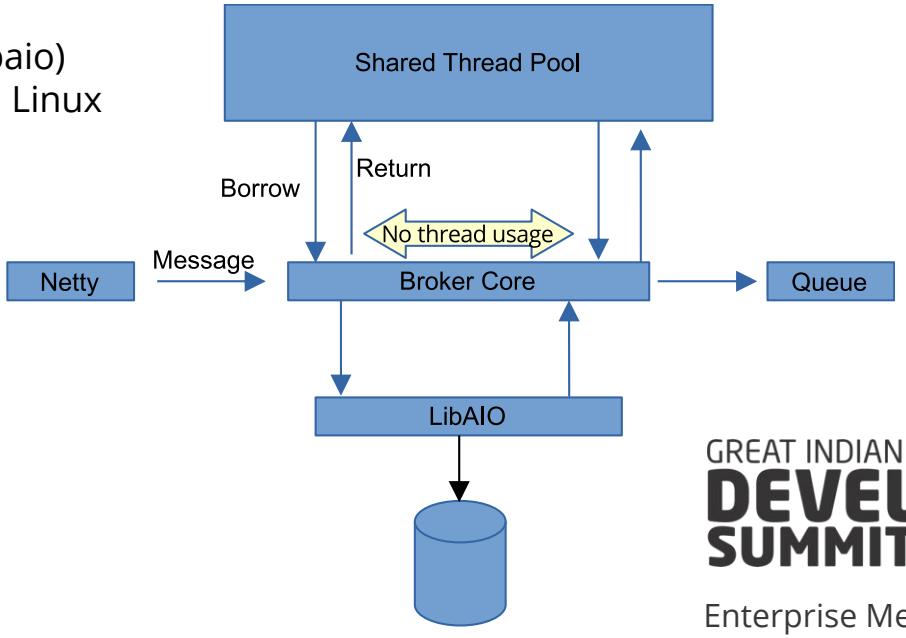
1. JMS 2.0 Supported
2. Transactions Supported
3. Single Message Processing
4. AMQP Standardized Protocol
5. Real-time <10 milliseconds
6. Disk persistence
7. Admin Management Console

Apache ActiveMQ Artemis
Features

- Pure Java, high-performance message broker
- Multi-protocol: AMQP 1.0, MQTT, STOMP, OpenWire, Artemis Core
- Polyglot: Java JMS 2.0, C++, .NET, Python, JavaScript (inc. Node.js)
- Flexible persistence: high performance journal or JDBC
- Support for large messages
- Flexible clustering
- High availability
  - shared nothing replication or shared SAN
- Open Source Community - Apache Foundation Project

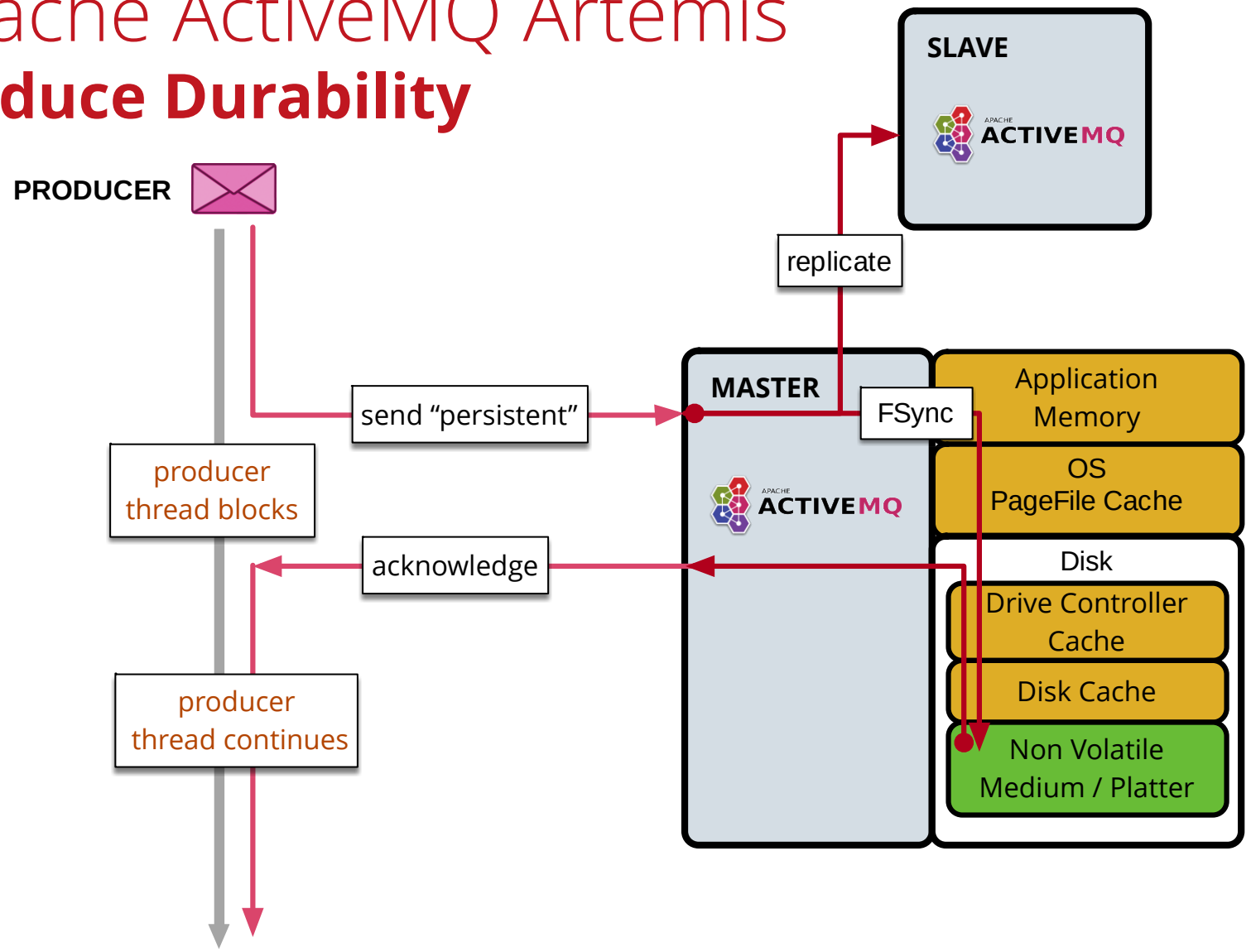
Apache ActiveMQ Artemis
Internal Architecture

- Fully asynchronous (non-blocking) internal architecture
  - Developed using reactive patterns
  - Netty IO
- Thread pooling
  - Predictable thread usage (not 1 thread per client/queue/*)
  - Configurable thread pools
- High performance journal
  - Custom implementation using Linux asynchronous I/O (JNI to libaio)
  - Automatically switches to Java NIO implementation when not on Linux
  - Compaction

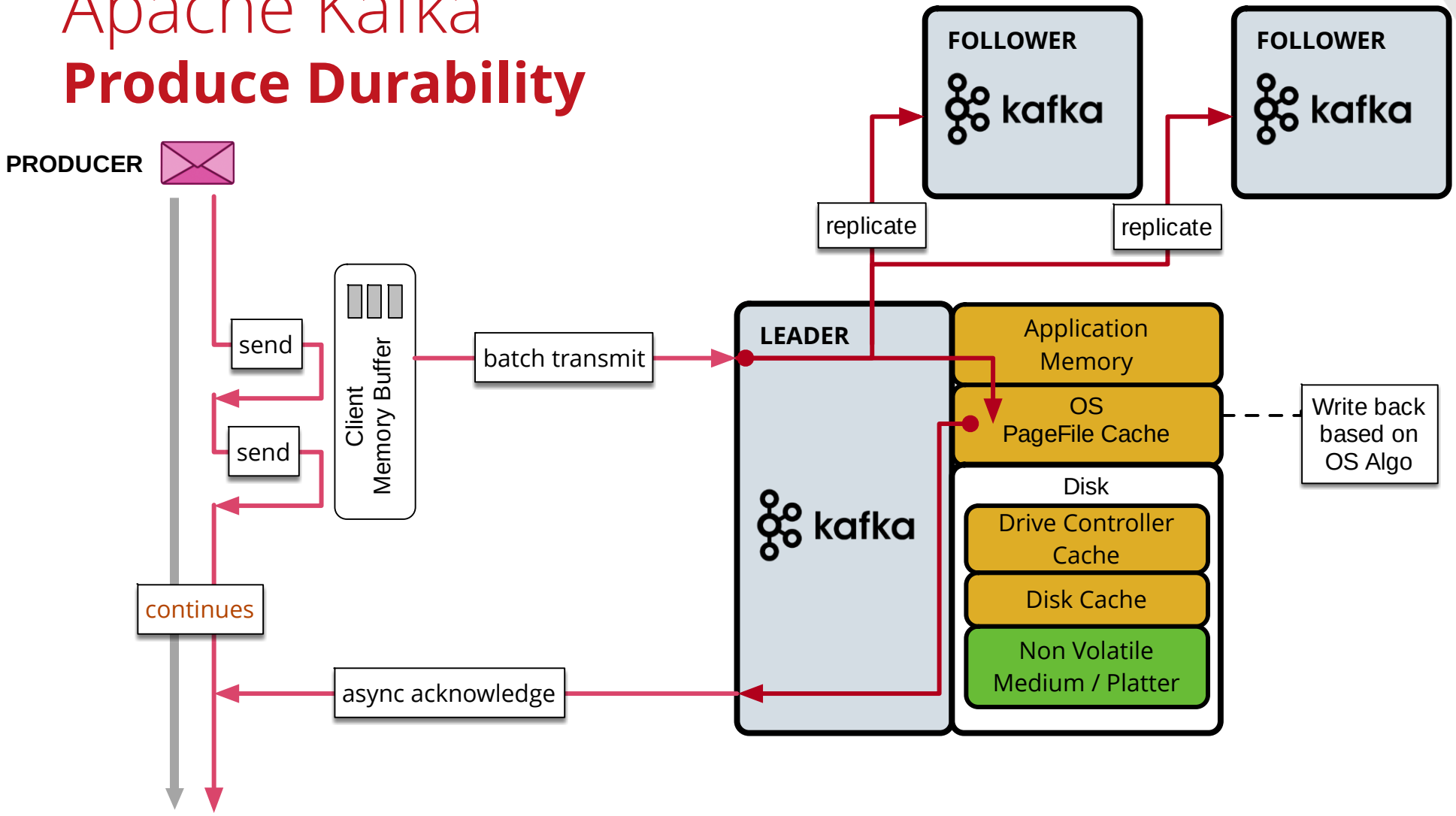


Apache ActiveMQ Artemis

Produce Durability

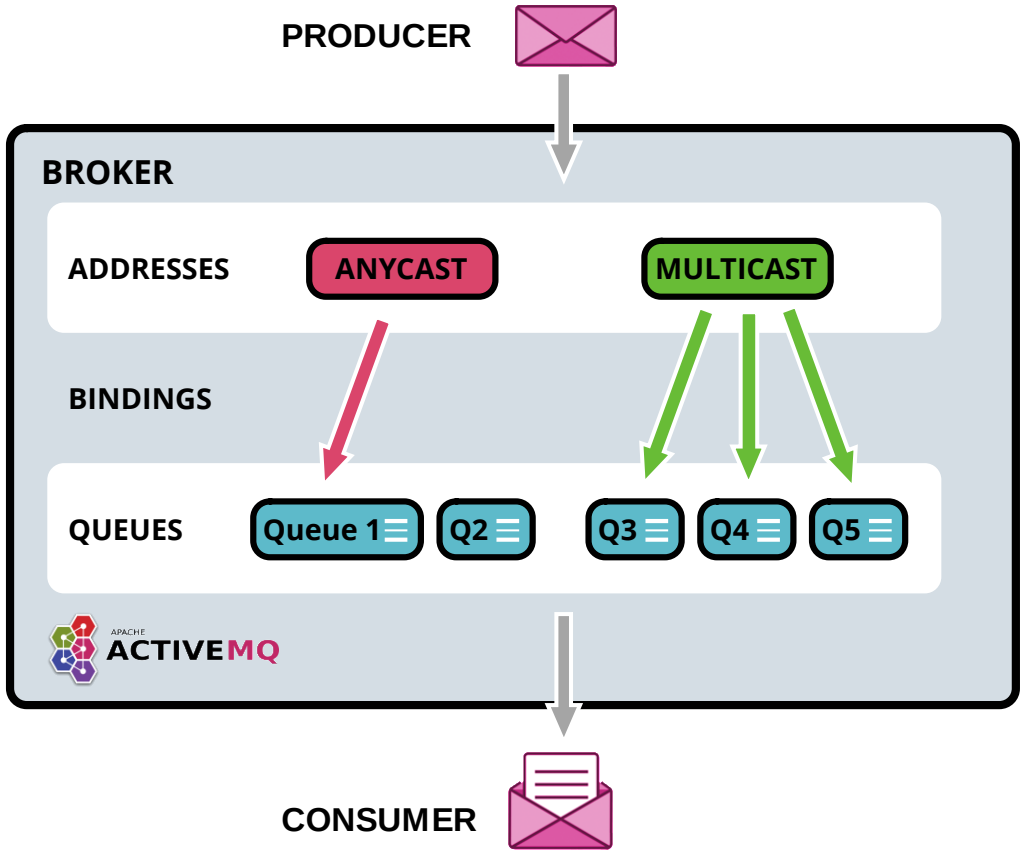


Apache Kafka
Produce Durability



Apache ActiveMQ Artemis

Core Address Model



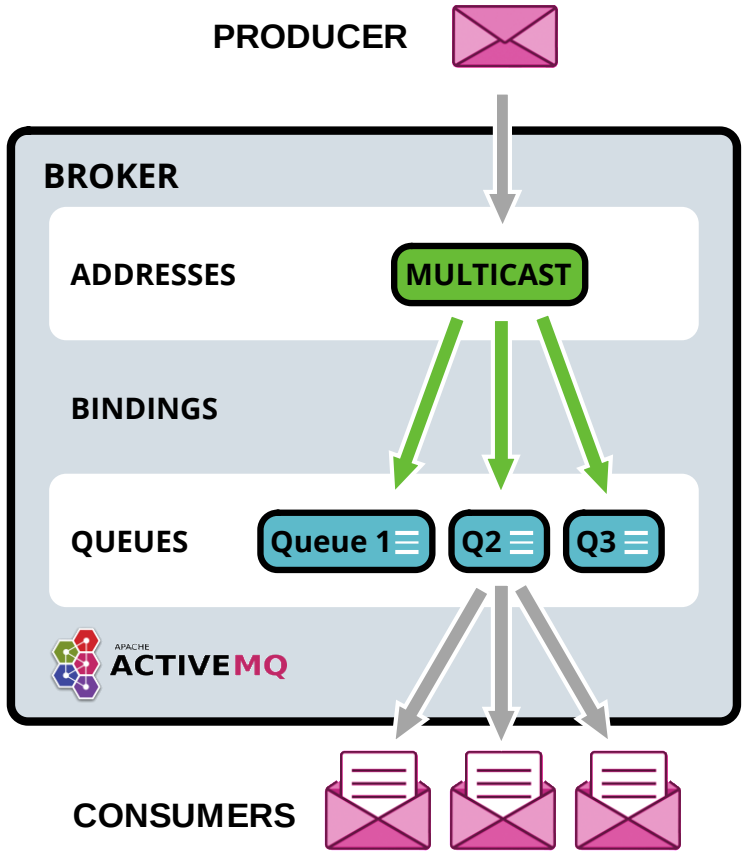
Type

- Anycast
- Multicast

Features

- Paging
- Direct routing
- Divert

Apache ActiveMQ Artemis
Core Queues



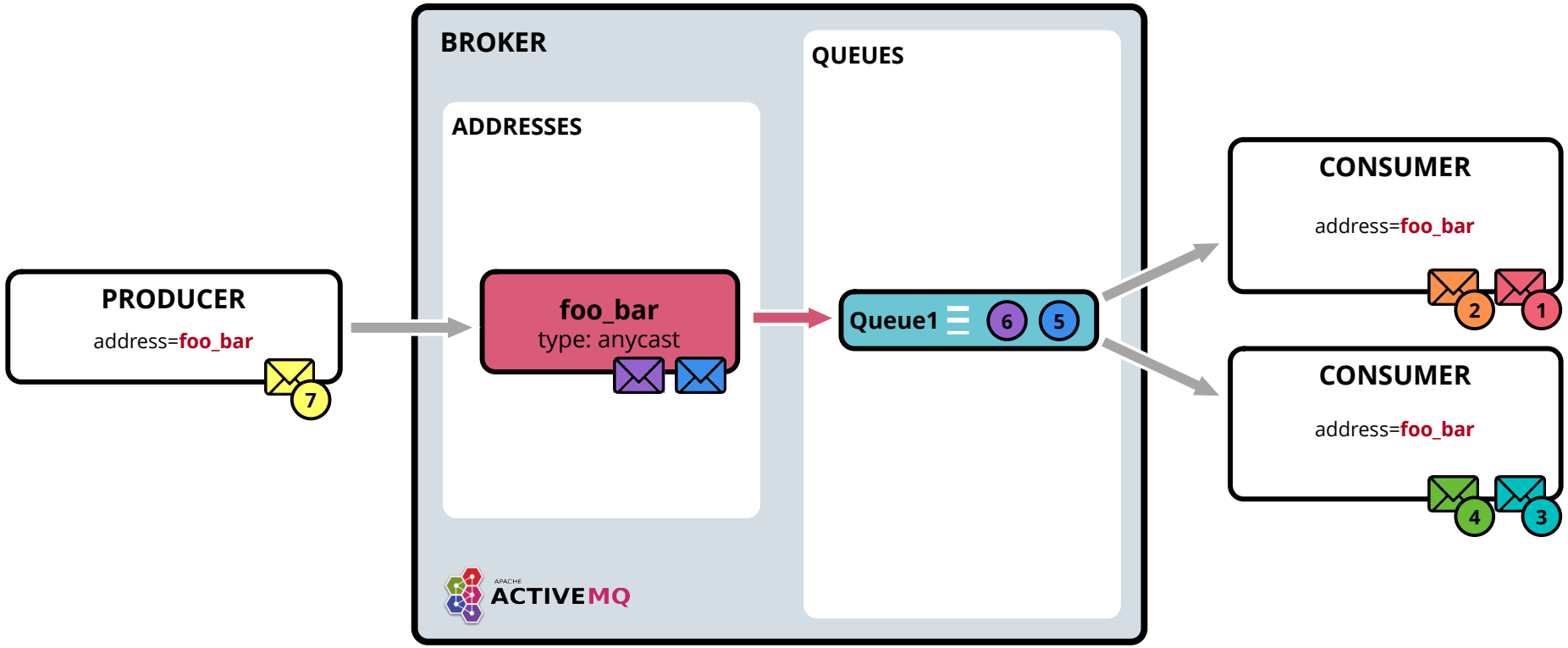
Dispatch

- Round Robin (Default)
- Exclusive
- Message Groups

Features

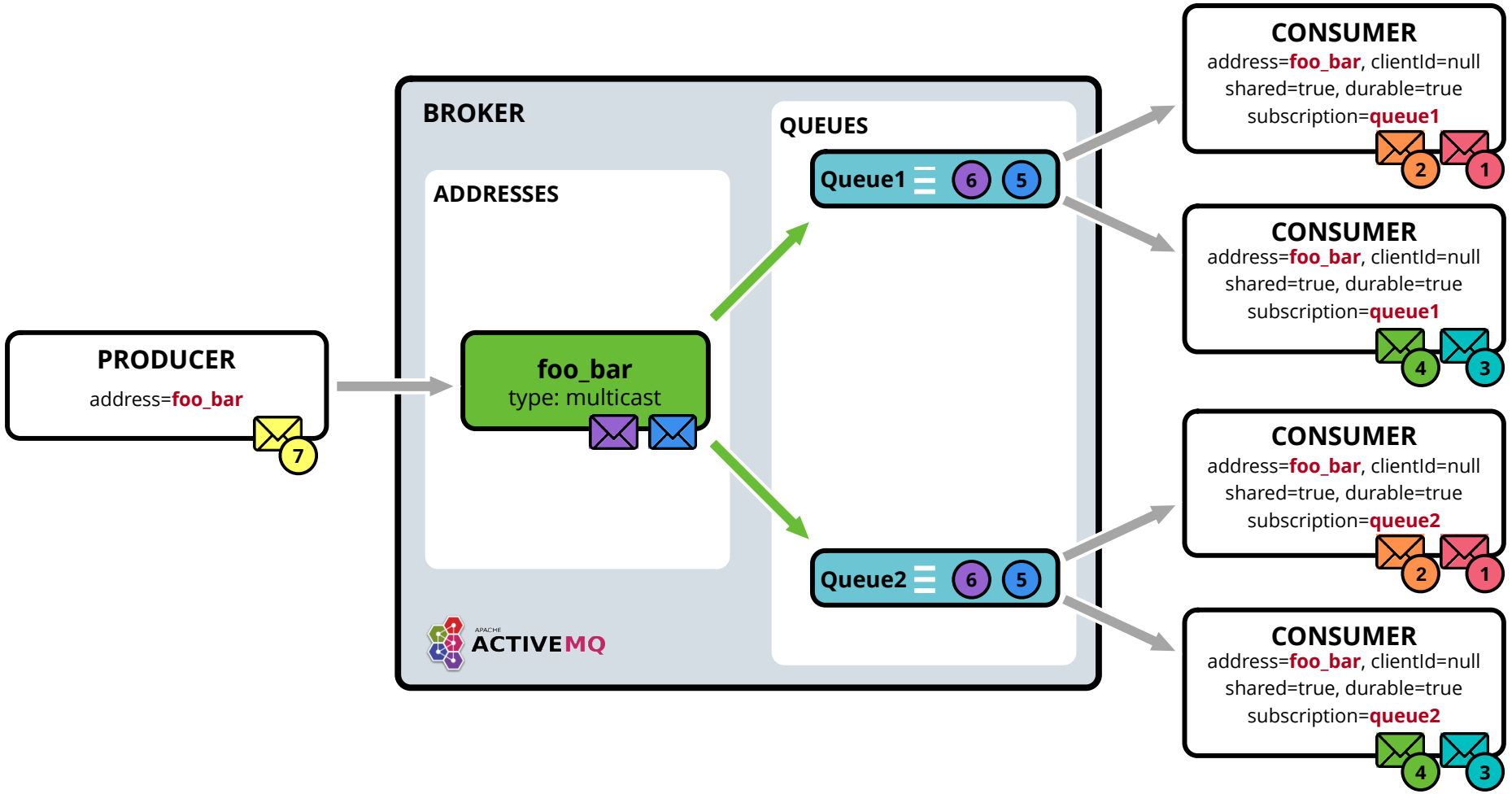
- Durable or Non Durable
- Filter / Selectors
- Last Value

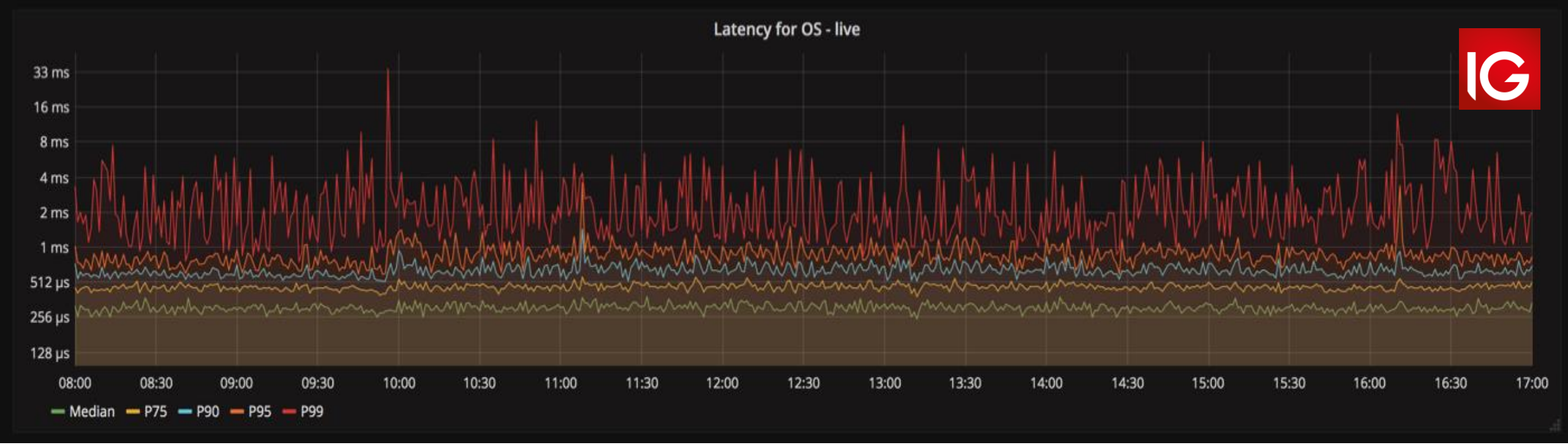
Apache ActiveMQ Artemis
JMS Queue



Apache ActiveMQ Artemis

JMS Topic - Shared Durable Subscriber

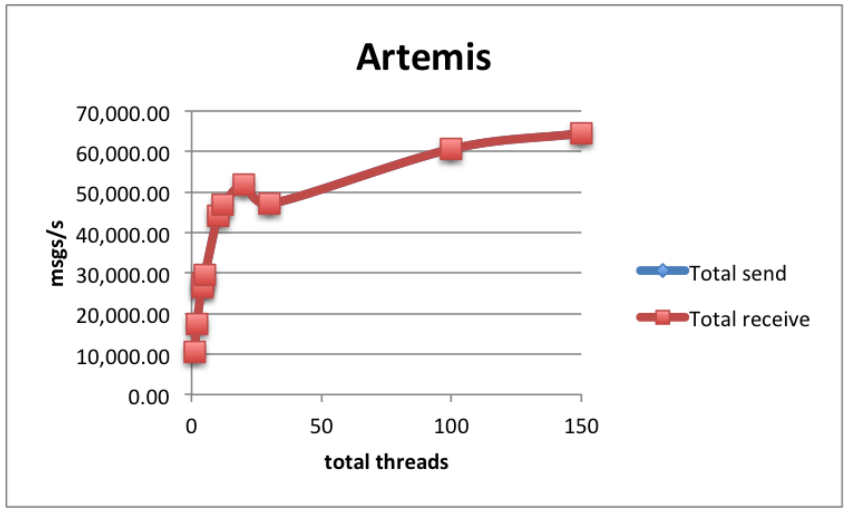


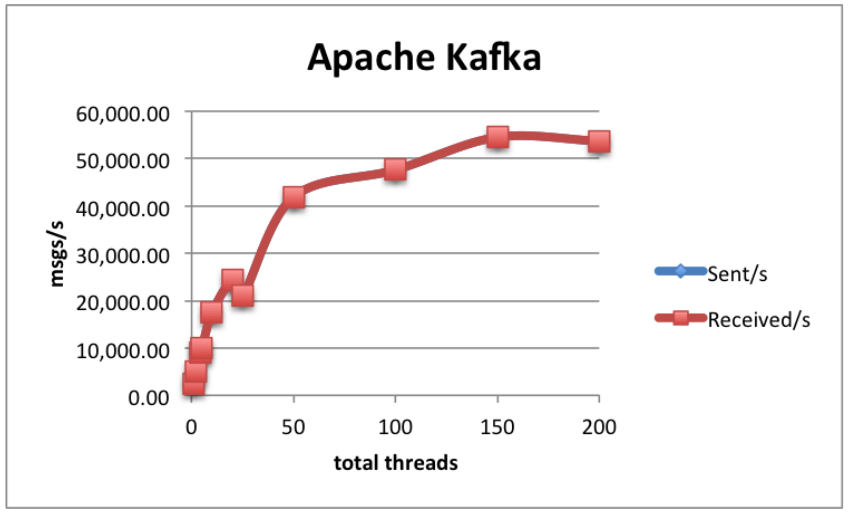


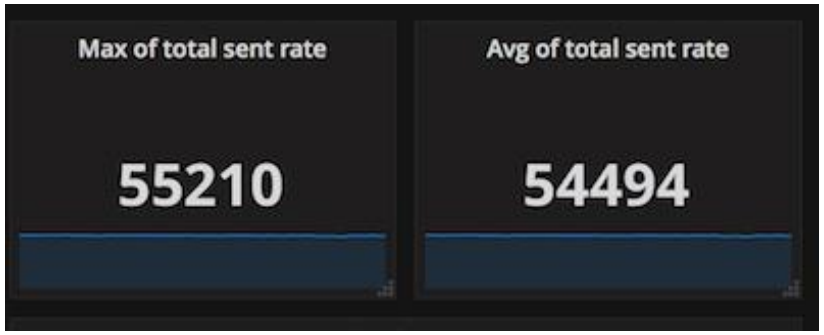
Latency – AMQ-7 Apache Artemis

- Core Trading Day
- Persistent with File Sync to Disk
- Replicated HA
- Standard Servers
  - 2 x 6 Core Intel Xeon
  - 4 x 32Gb Ram
  - Dedicated SSDs for Journal and Paging

Softwaremill Blog https://softwaremill.com/mqperf/







Use Apache ActiveMQ Artemis for

- individual message processing
- messages guaranteed to disk / non-volatile storage
- clients to use standard APIs (e.g. JMS)
- clients to use standardised wire protocols (e.g. AMQP, MQTT, STOMP)
- transactional sends and receives
- request-reply messaging
- selectors and filters
- Advanced messaging features
  - time-To-Live semantics (TTL)
  - scheduled delivery
  - Dead Letter Queue semantics (DLQ)
  - Etc.
- don’t want to implement broker functionality in your clients
  - e.g. partitioning, dispatching, coordination

Use Apache Kafka for

- messages in volume
- raw throughput
- batching (micro)
- sliding-window replay abilities
- simple K,V store (compacted topics)
- large numbers of subscribers for published events
- finely control the parallelism/scalability of consumers

Where do I think things are going?

Traditional JMS Brokers for instance ActiveMQ 5.x (Classic)
are the **SQL** of the messaging world

Brokers like Kafka, Pulsar
are the **NoSQL** of the messaging world

Next gen JMS Brokers such as ActiveMQ Artemis
are the **NewSQL** of the messaging world.

Final Thoughts

Apache Kafka and Apache ActiveMQ Artemis
are different brokers with different tradeoffs for different use cases.

There is no silver bullet.
You ***should*** co-exist different brokers in your architecture.

Data in transit is just as important as it is at rest,
standardise, **schema, secure** and **govern** it.

Thank You

@itsourcery

github.com/michaelandrepearce

@LifeatIG
iggroup.com/careers


GREAT INDIAN
DEVELOPER™
SUMMIT 2019
Conference : April 23-26, Bangalore
12th EDITION
#gids19
Register early and get the best discounts!
www.developersummit.com
@greatindiandev
bit.ly/gidslinkedin
facebook.com/gids19
bit.ly/saltmarchyoutube
flickr.com/photos/saltmarch/