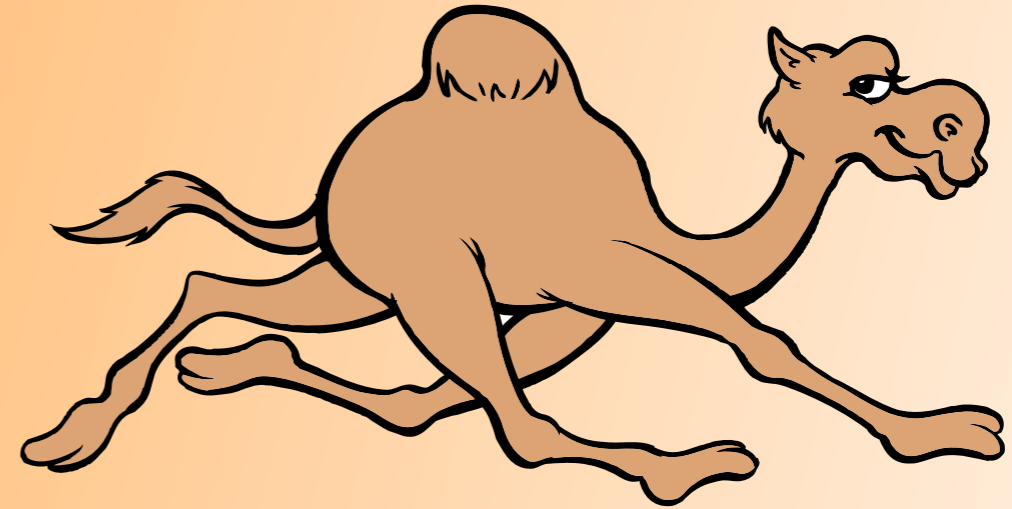


CamelOne 2013

June 10-11 2013

Boston, MA



Apache ActiveMQ:

The right choice for the Enterprise

By Matt Pavlovich



Agenda

- Background
- Apache ActiveMQ Advantages
- Deployment Options
- Typical Use Cases
- Migration Strategies
- Q&A



Speaker



- Founding Partner of Media Driver
- Leading Solutions Partner and Systems Integrator for Apache Integration products
 - Apache:ActiveMQ, Camel, CXF, ServiceMix
- 12+ Core Consultants, Project Management and Solution Delivery teams
- One week to multi-year engagements



ActiveMQ Advantages

ActiveMQ Advantages

- Introduction of Enterprise Messaging
- Developer Friendly
- Open Source and Commercially Supported
- Deployment options
- Technology Roadmap





Messaging Intro

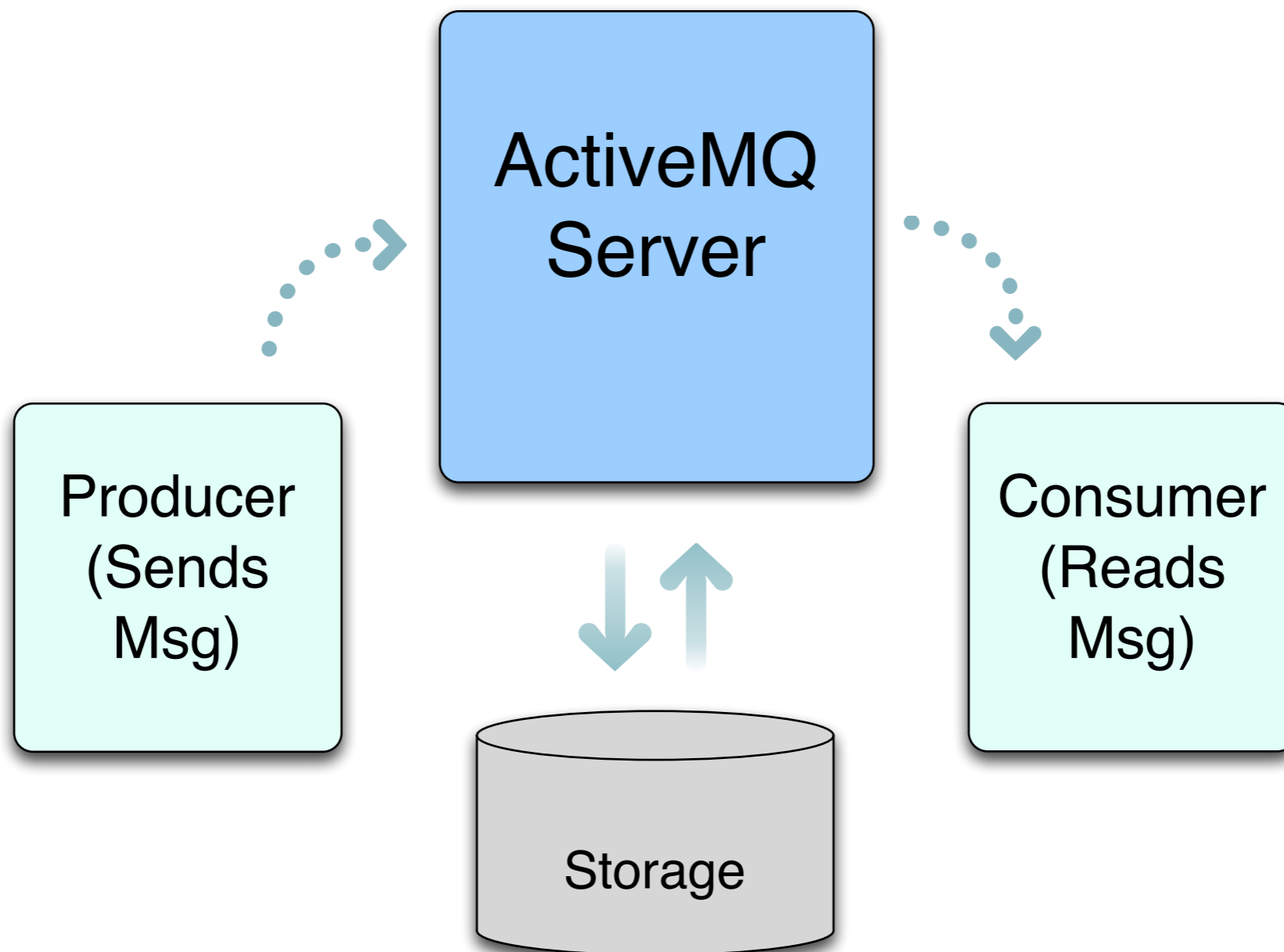
- Apache ActiveMQ is an Open Source messaging product
- Client-Server-Client design
- Standards compliant
- Simple administration
- Fast!

Simplifies the work of programmers.



Messaging Intro

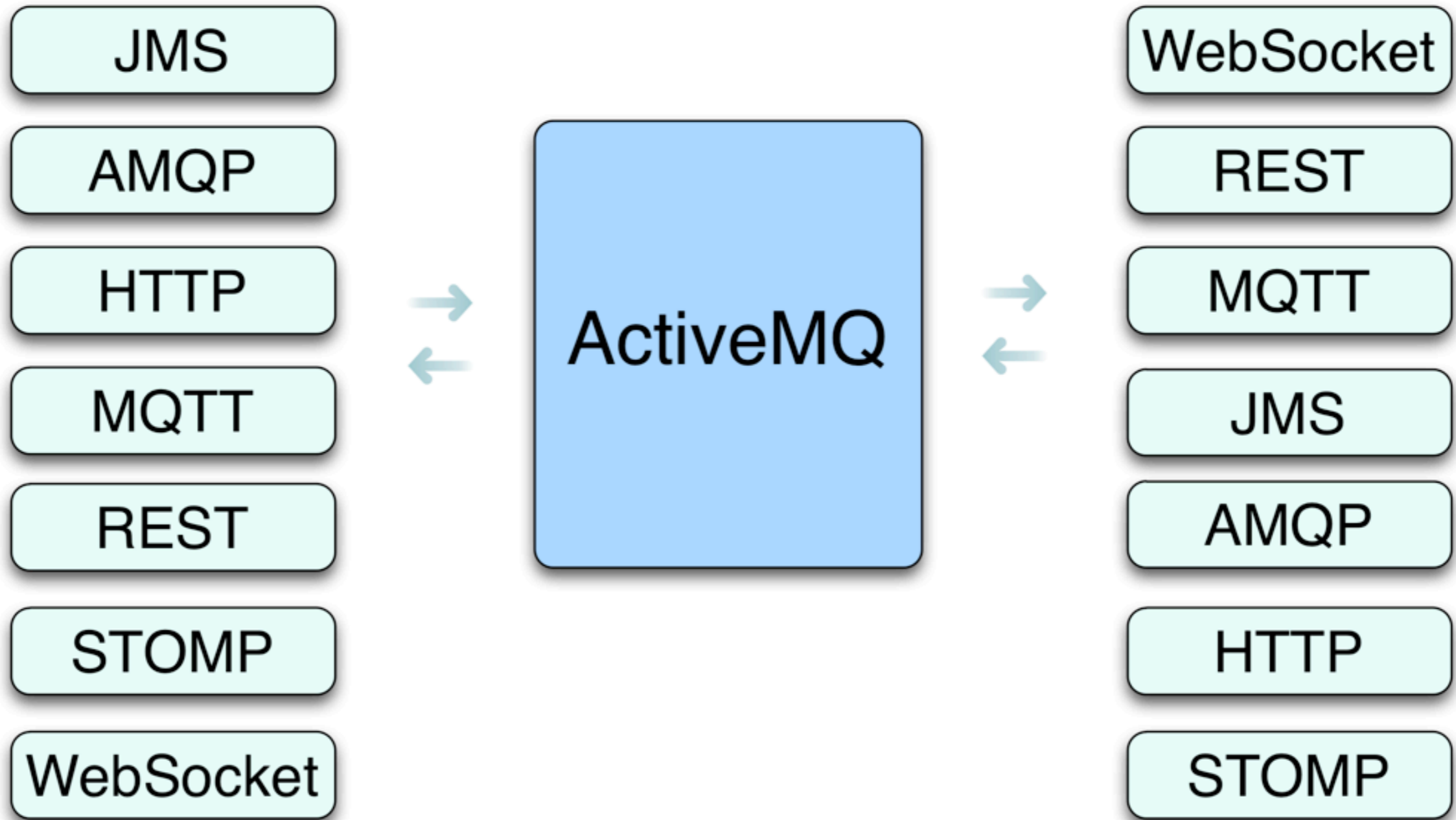
- An Enterprise messaging system serves a number of critical functions and saves developers from having to reimplement the same features over and over.
- Reliable delivery of messages
 - *Fire and forget*
- Once-and-only once message delivery - or - Publish and Subscribe option
- Message order maintained
 - *First in First Out (FIFO)*





Messaging Intro

- ActiveMQ is much more than JMS
- Supports many modern communication protocols
- Supports many deployment models
- Provides clustering and high availability options required by today's Enterprises



*ActiveMQ is an Enterprise Class **messaging** system that supports many standard communication protocols.*



Messaging Intro

- ActiveMQ is standards based and provides non-intrusive enhancements to work around limitations of those standards
- Creative solutions to work around limitations without negatively impacting developers and administrators
- Virtual Destinations, Master-Slave model, Message Groups, etc.



Developer Friendly

Developer Friendly

*Unit test without
a server!*

- Up and running in minutes
 - Easy to write code.
- Supports Java, C, C++, C#, JRubyScalaGroovyPHPython as well
- ActiveMQ supports many communication protocols
 - MQTT, AMQP, Stomp, SSL, Websockets, and REST.



Developer Friendly

- ActiveMQ's flexibility allows it to be deployed in many different ways
- ActiveMQ client library handles failover and connection pooling
- Fantastic integration with Camel, CXF and ServiceMix.
- *Code is not strictly tied to deployment architecture*





Administrator Friendly



Administrator Friendly

- Up and running in minutes *Straight forward*
- Includes examples to test infrastructure
- Supports JMX for monitoring
- One XML configuration file for the broker
 - Easy to create standard builds
 - Same config file structure used in various deployment models



Developer Friendliness == Organization Friendliness

ActiveMQ ties cross-platform solutions together across the organization.



Short learning curve



Multiple Protocol Support



Multiple Language Support



Simple configuration



Easy to standardize builds



Cost Structure

Hard Cost Structure

- Open Source
- Free to try, use and extend forever.
- Short learning curve, easy to for developers and administrators to get started.
- Scale ActiveMQ with your business



Commercial Options

- A-La-Carte support options that align with your organization
- 24/7 Product Support
- Consulting and Services
- On-going Solution Support





Soft Costs

- Organizations limit utilization of messaging, due to soft costs associated with developing against the commercial messaging products-- IBM Websphere MQ Series, Tibco EMS, etc
- ActiveMQ allows you to more freely deploy instances to leverage best architecture and design
- ActiveMQ allows you to “use more queues” without fear of increased administration overhead



Deployment Options



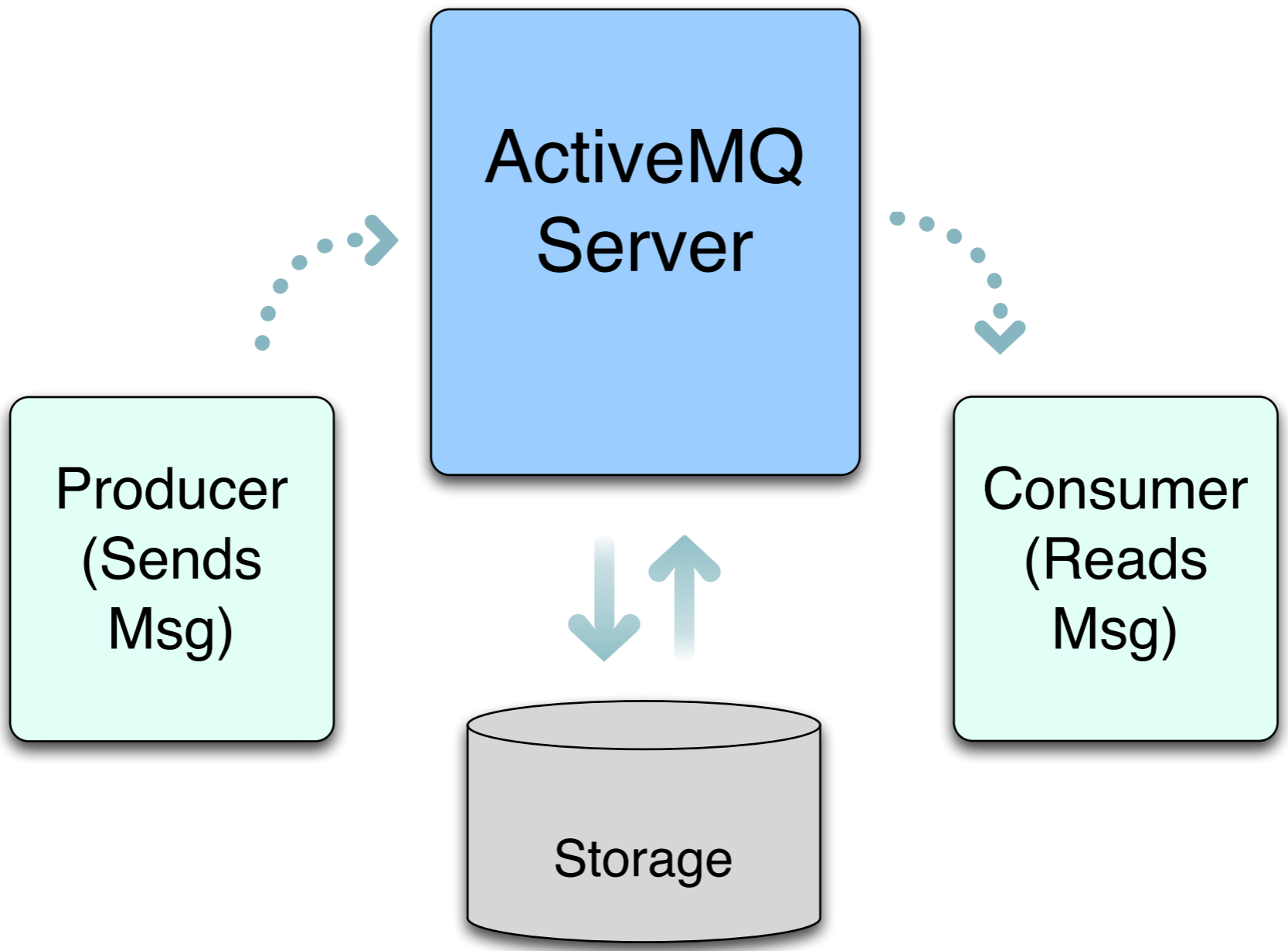
Deployment

ActiveMQ's flexibility allows for many deployment options.

- Stand alone broker
- Embedded in other Java runtimes
- Included with Apache ServiceMix for ESB
- Failover is easy
- Clusters are even easier

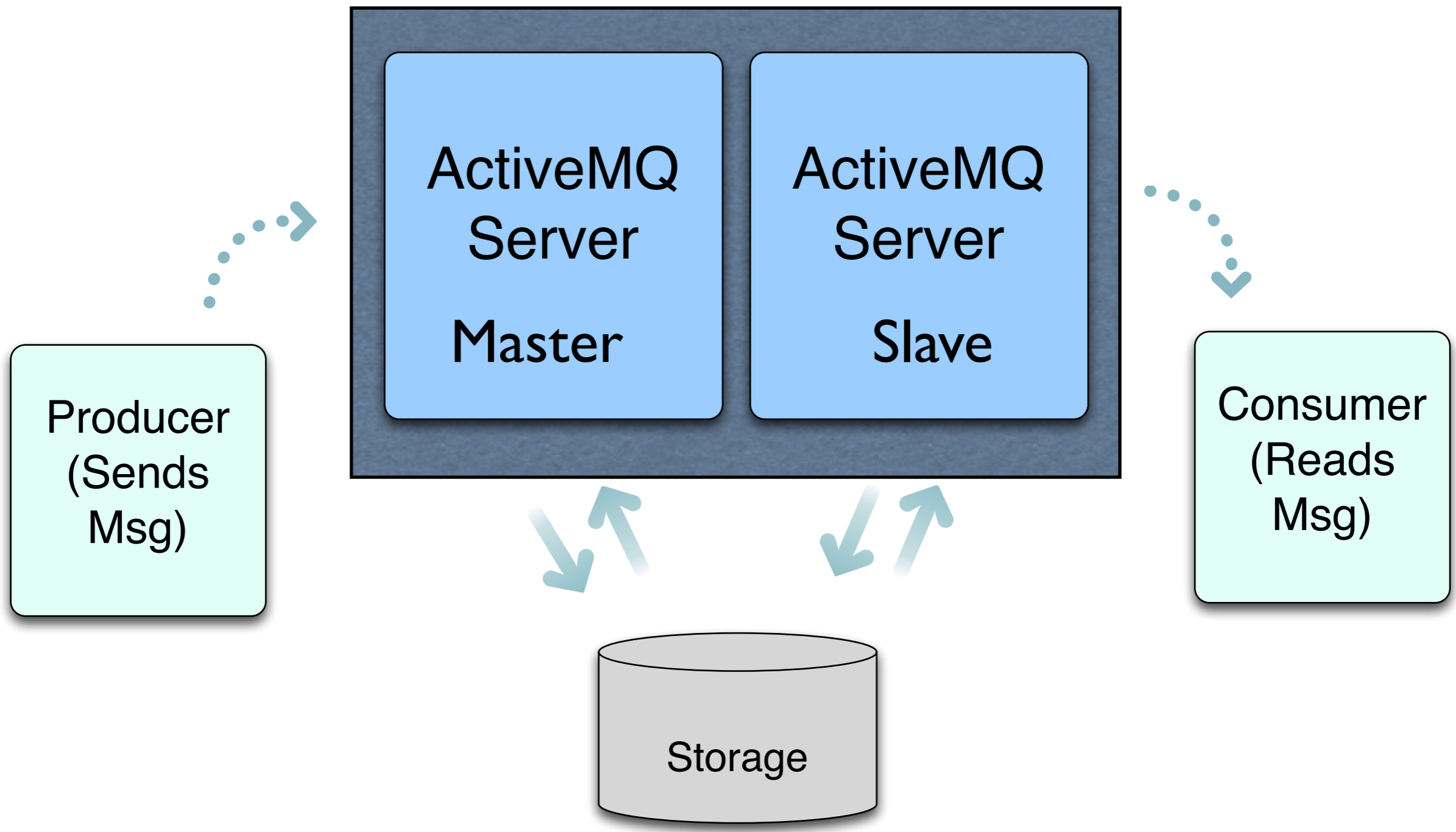


Single Server



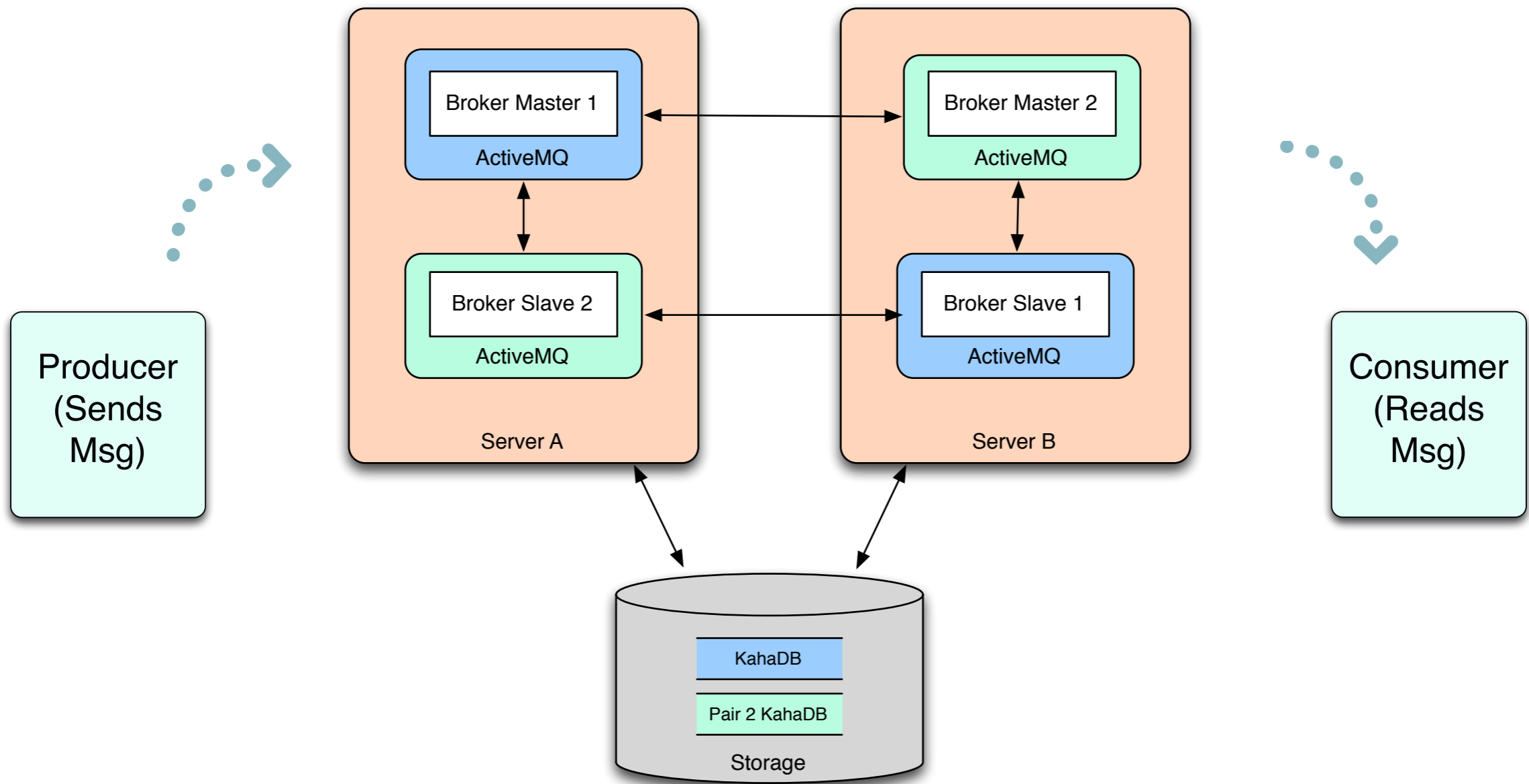


Master - Slave

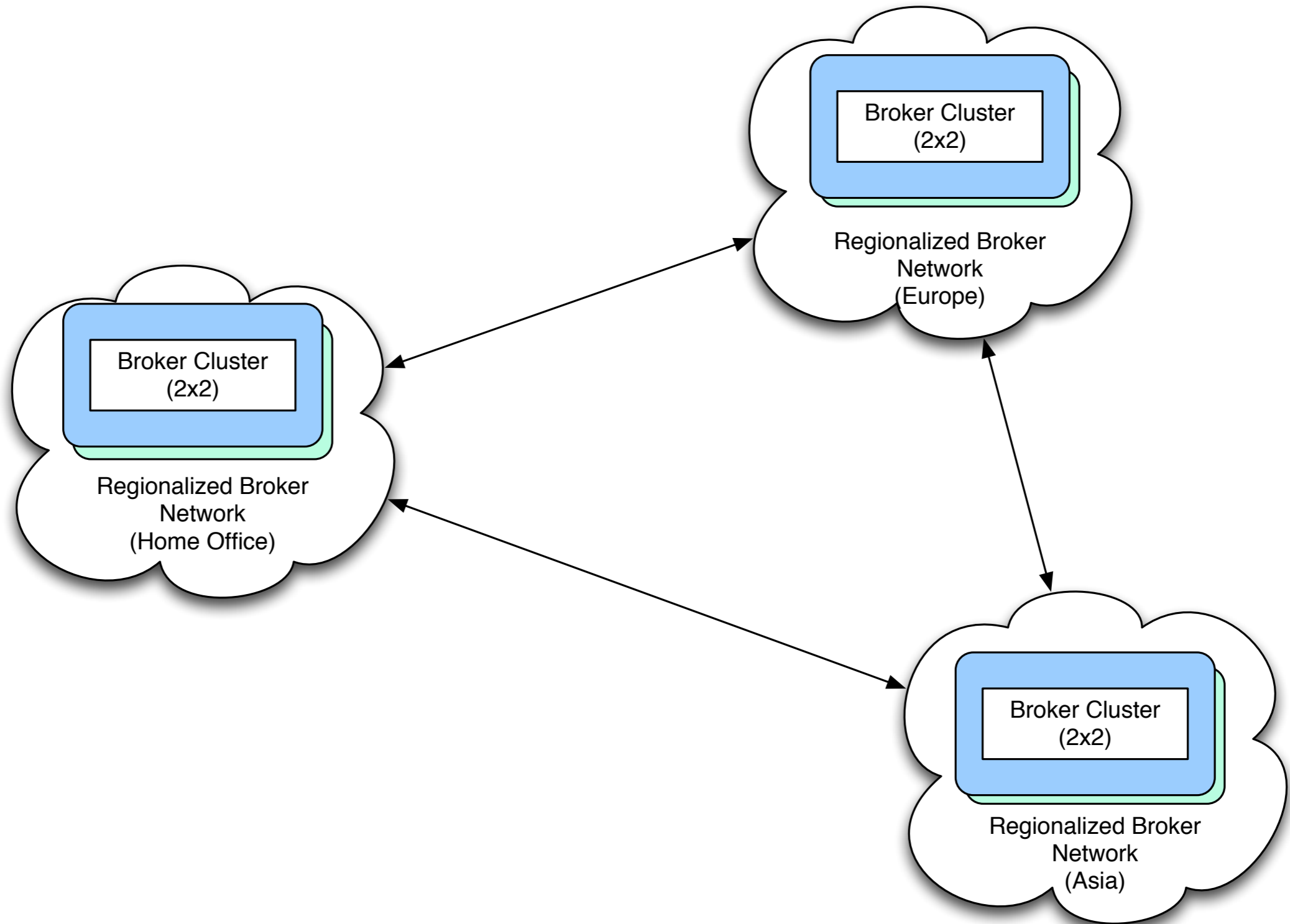




Clustered and Redundant



Globally Distributed











First Choice

- ActiveMQ usually starts out as a cost savings option for a specific project
- Success-based growth within organizations
- Developers and Administrators appreciate how lightweight and easy it is to get projects “live” with ActiveMQ
- ActiveMQ becoming the First Choice for many organizations



ActiveMQ Deployed Today

-  Low memory
-  Big memory
-  Mobile device
-  Clustered
-  Hardware appliance
-  Cloud

ActiveMQ can be used in just about any computing scenario.



Typical Use Cases

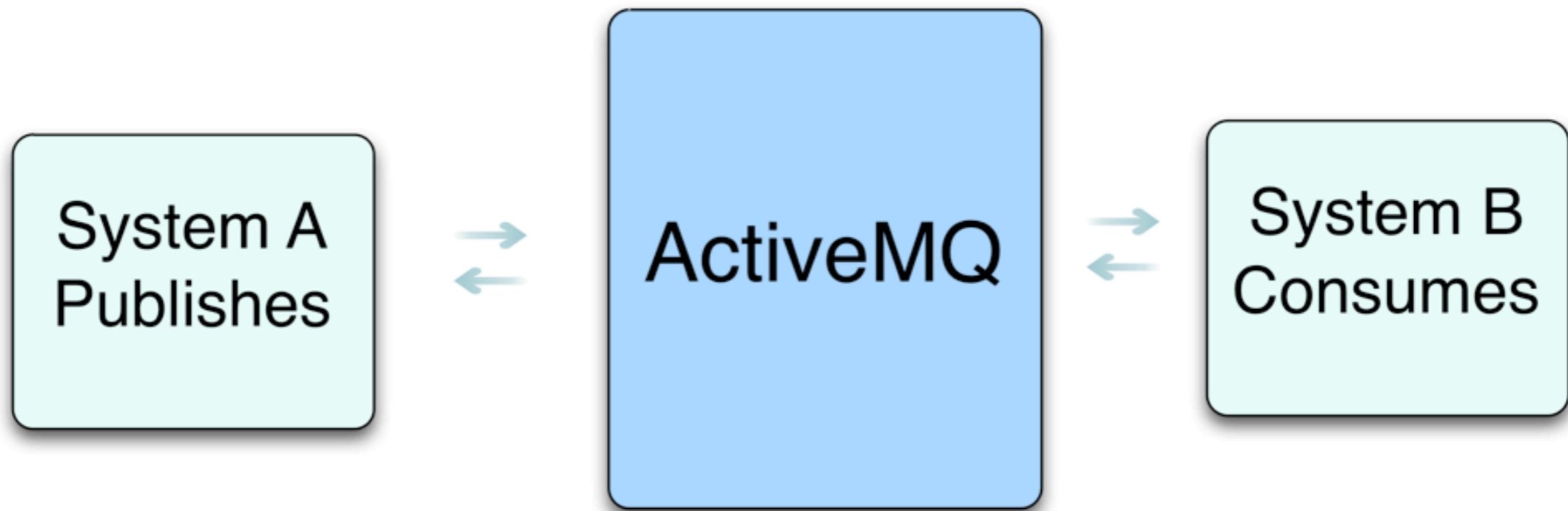


Typical Use Cases

- ActiveMQ's solves many Enterprise computing needs.
- Messaging is a core capability of Service Oriented Architecture
- Decouple applications and systems
- Reliably move data between remote location
- Manage access and traffic to backend systems

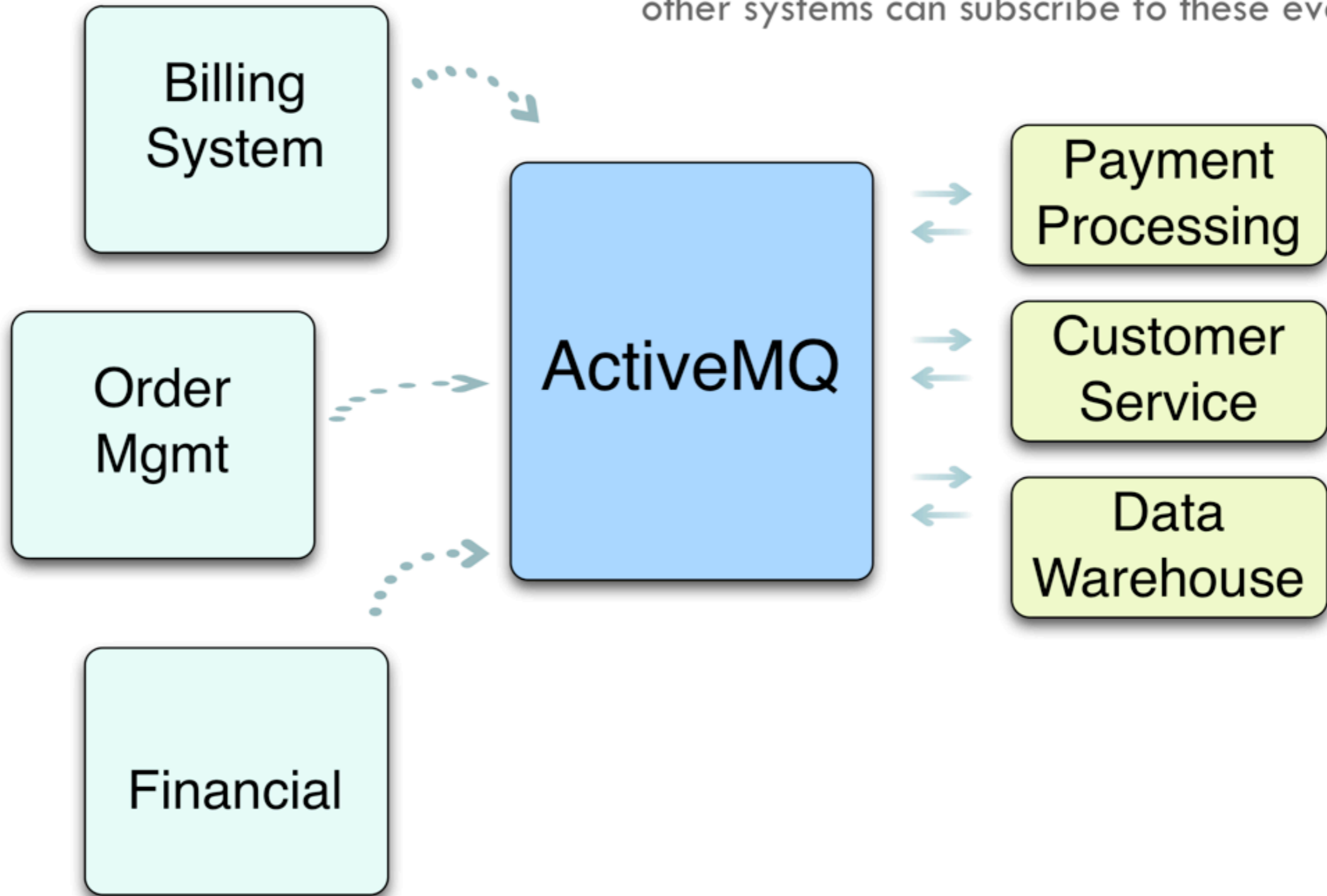


System A is required to send data to System B, but does not want to implement reliable delivery, message persistence, idempotence and ordering.



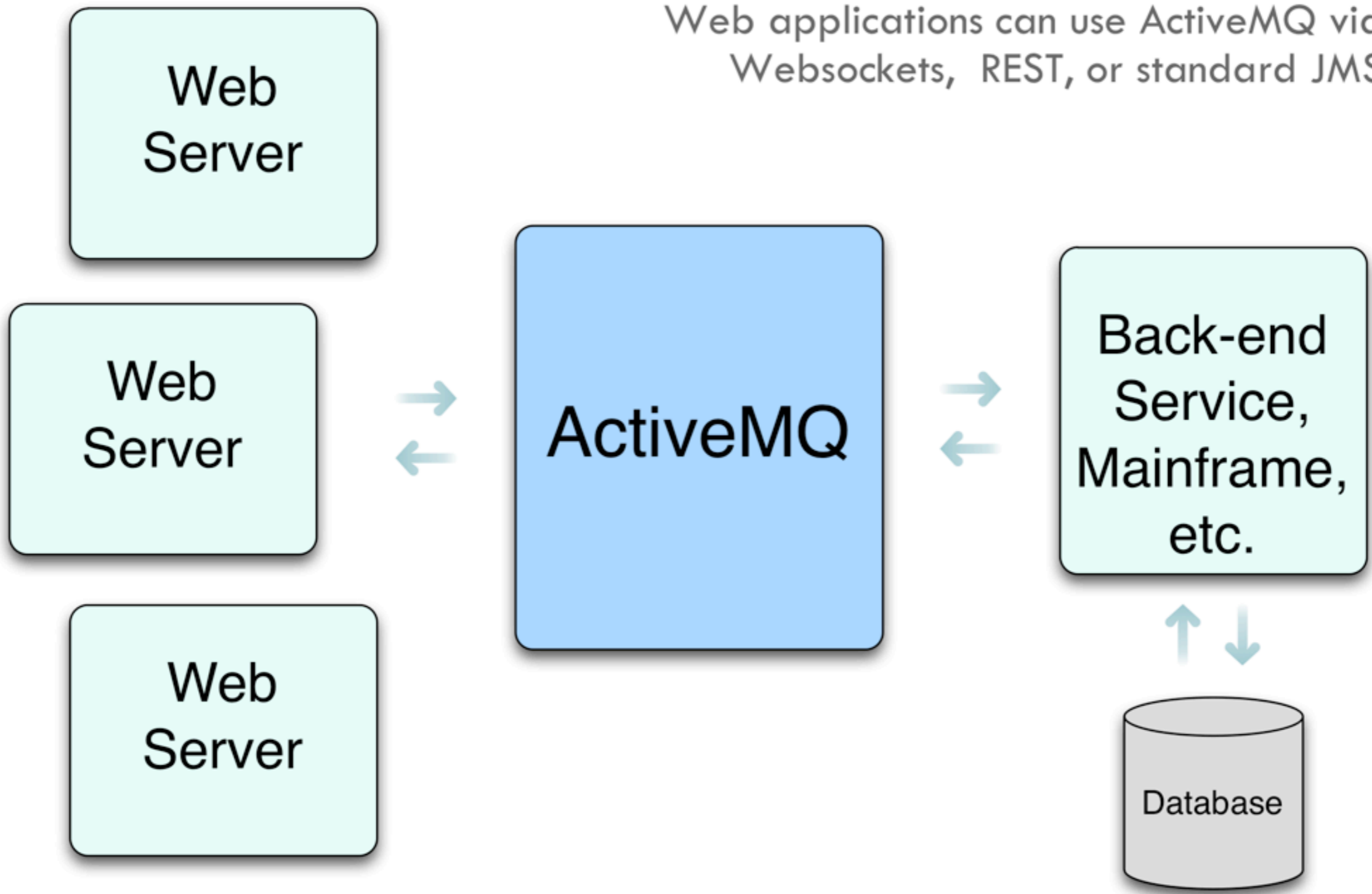


Various applications send events to ActiveMQ, and other systems can subscribe to these events



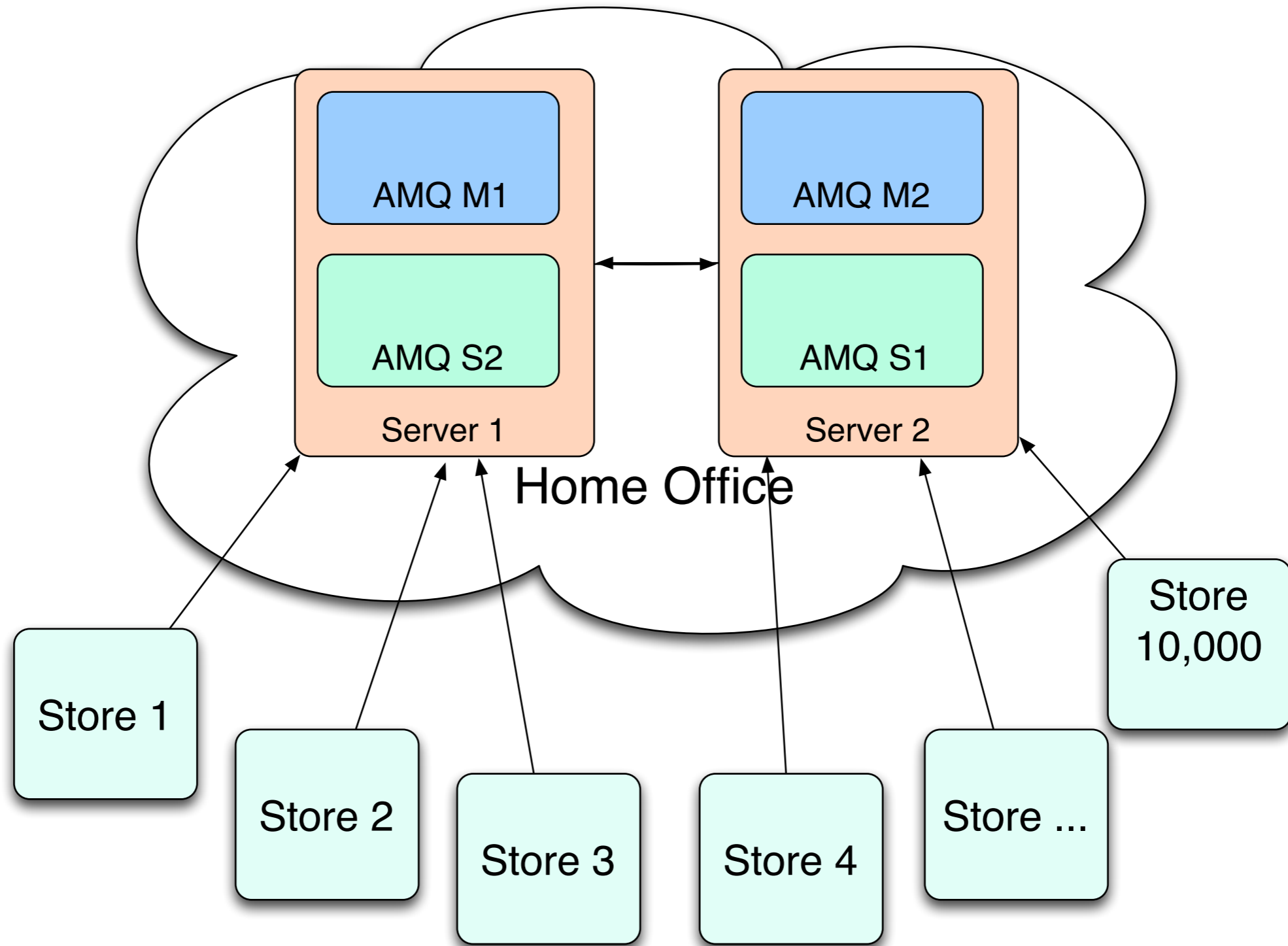


Web applications can use ActiveMQ via Websockets, REST, or standard JMS





Retail Scenario



ActiveMQ Industries



Retail, retail, retail.



Health Care and Pharmaceuticals



Government, Utility and Telecommunications



Fashion



E-commerce, Internet and Technology



Travel and Hospitality





Roadmap



Roadmap

- Active development community with a feature rich roadmap
 - Making development and administrations even easier
- Blazing the trail for support of advanced technology and greater performance
 - LevelDB, AMQP, Apache Karaf and more.
- Moving past commercial options in terms of performance, manageability, Total Cost of Ownership and strategic technology direction



Migration Strategies

Migration Strategies

- How does an organization make the switch to ActiveMQ?
- New development
- Drop-in Replacement
- Hybrid Integration



New Development

- All new development starts on ActiveMQ
- Reduce hard cost by stopping new commercial messaging instances
- Ideal for mid-sized organizations that can turn over applications in short period of time 2-3 years and plan on EOL existing systems,
 - Internet and Technology companies.

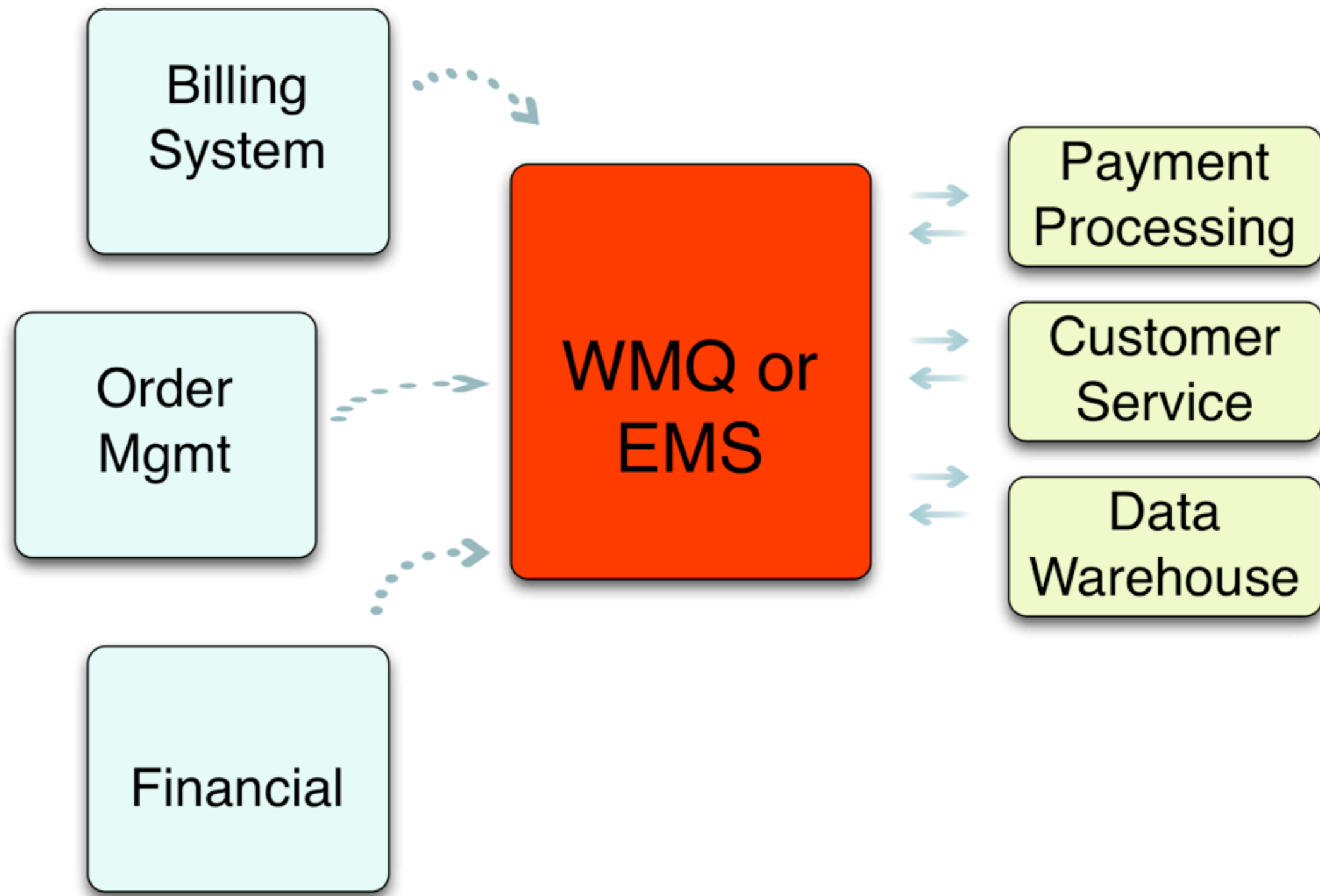


Drop-in Replacement

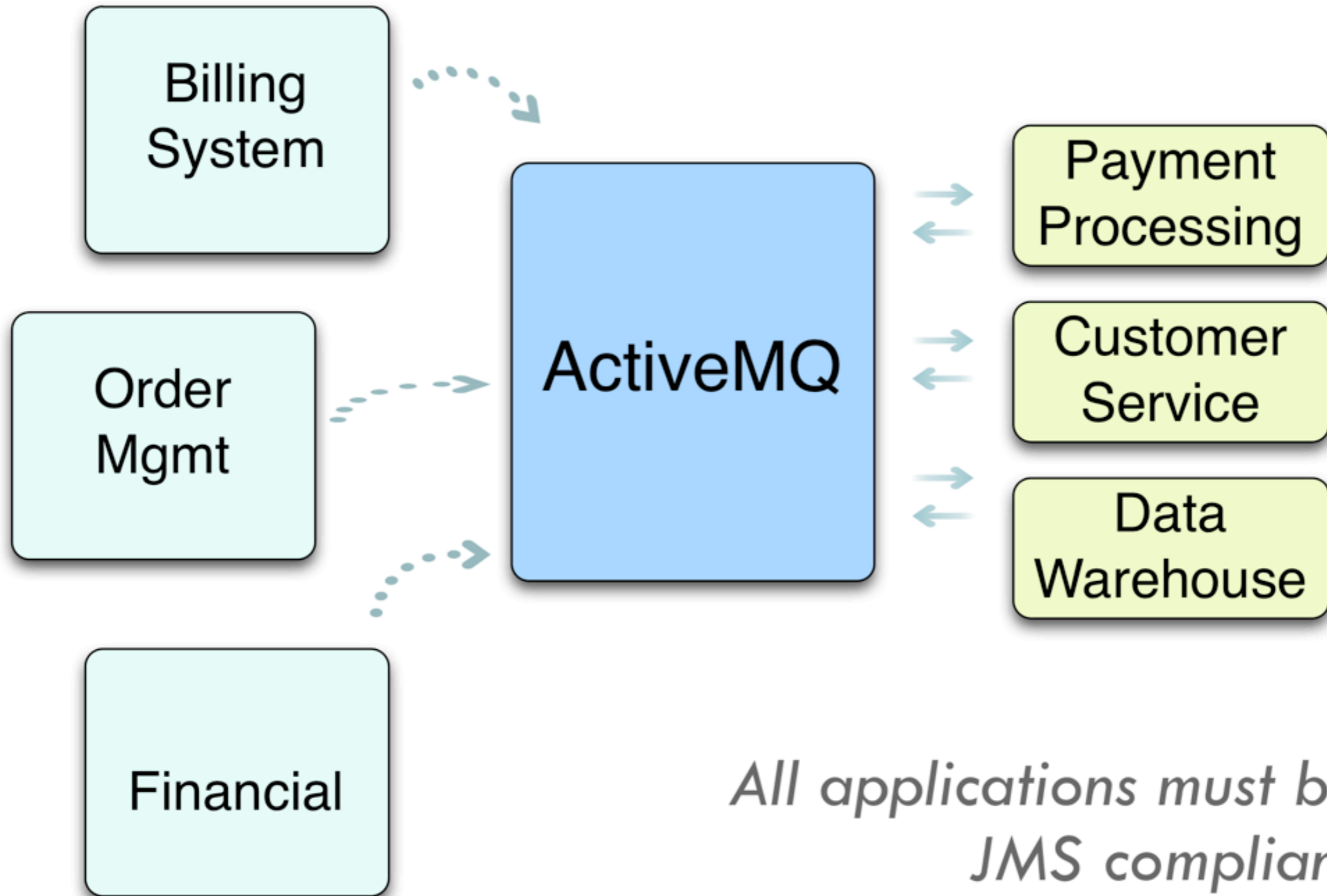
- Replace commercial messaging with ActiveMQ
- Eliminate hard costs from commercial messaging
- Ideal for any sized organization that has fairly uniform application stack
 - JMS-based messaging with limited use of extensions
- Straight-forward rollback



Drop-in Replacement



Drop-in Replacement



All applications must be JMS compliant

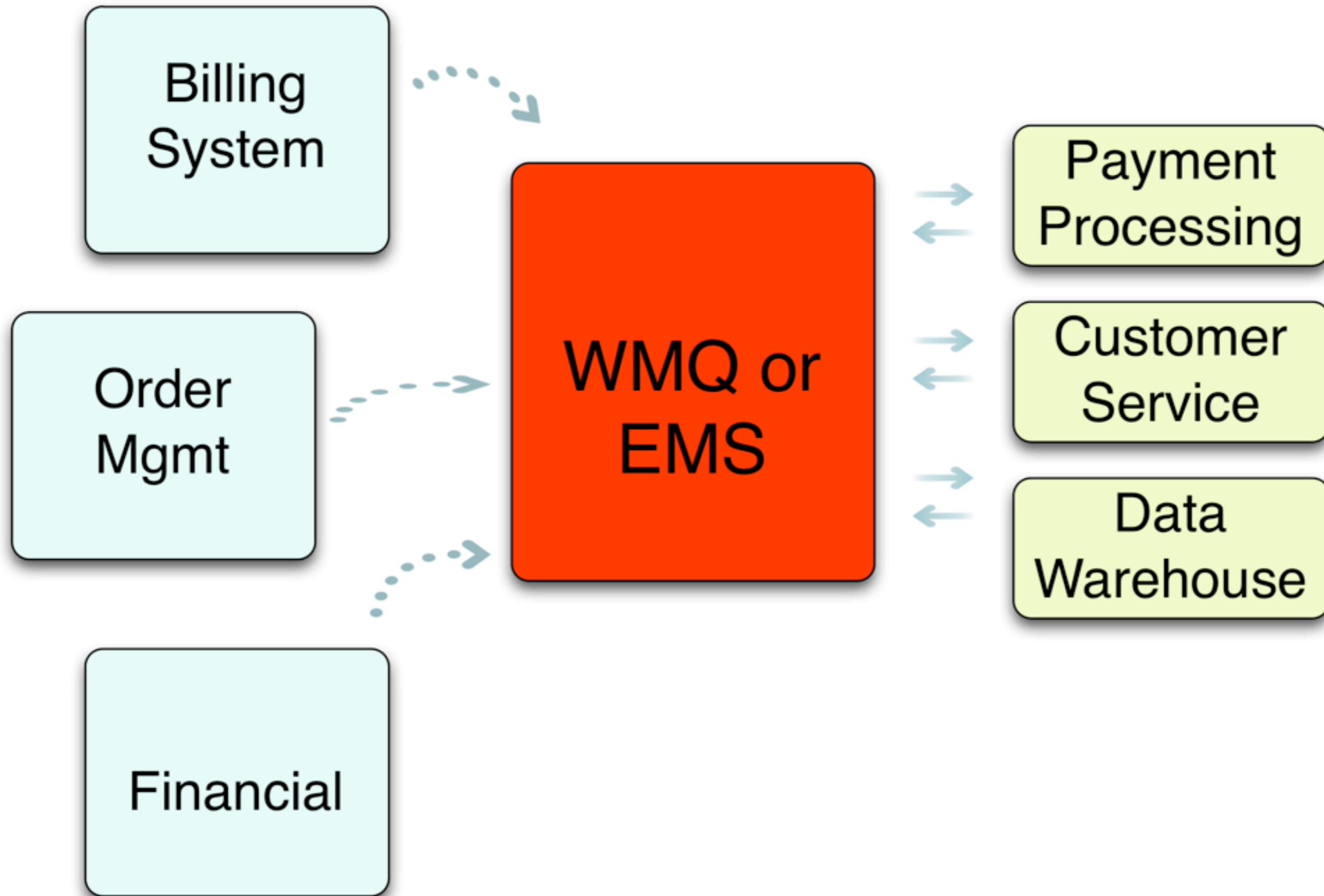


Hybrid Integration

- The exciting option!
- Dynamically integration ActiveMQ with existing IBM WMQ or Tibco EMS platforms
- Leverage Apache Camel
- Dynamic rollout, rollback and gradual migration of applications
- Ideal for large messaging implementation

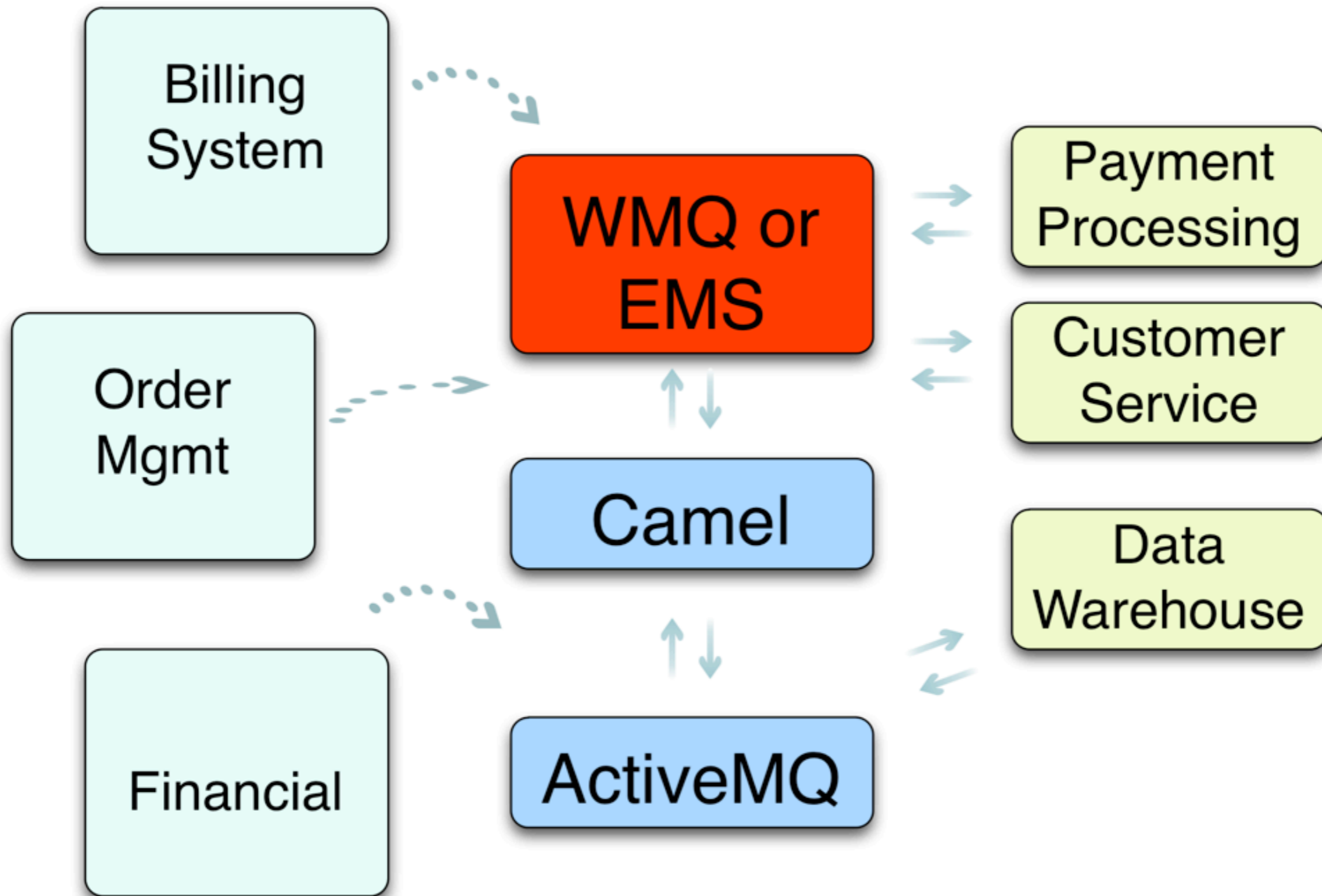


Hybrid Integration



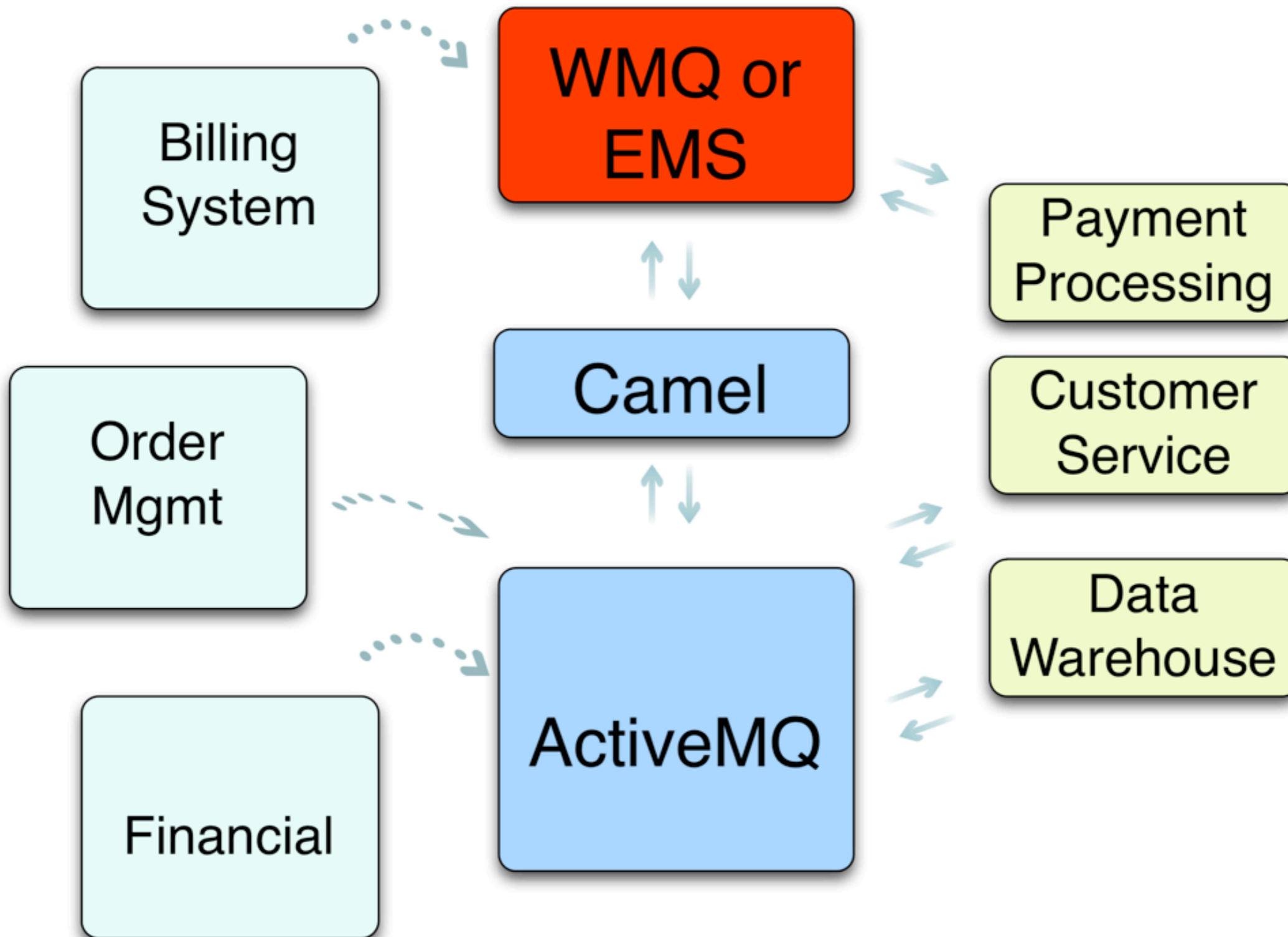


Hybrid Integration



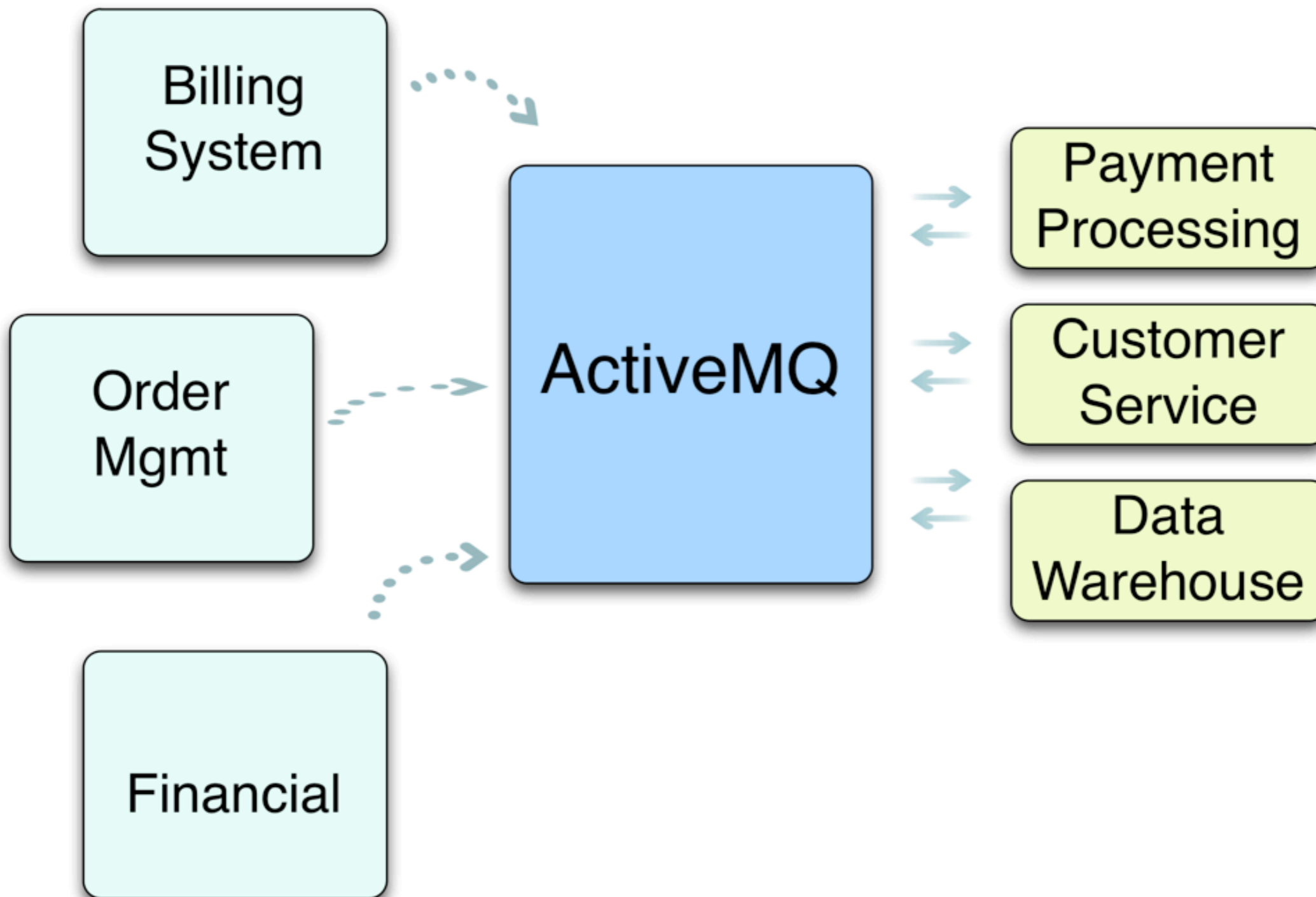


Hybrid Integration





Hybrid Integration





ActiveMQ



Open Source



Developer Friendly



Use it anywhere and everywhere



Migration Options for all scenarios



Q&A

- Thank you for attending this session and Camel One!
- More on ActiveMQ: <http://activemq.apache.org>
- More about speaker:
 - <http://mediadrivers.com>
 - <http://www.linkedin.com/in/mattpavlovich/>