CamelOne 2013
June 10-11 2013
Boston, MA

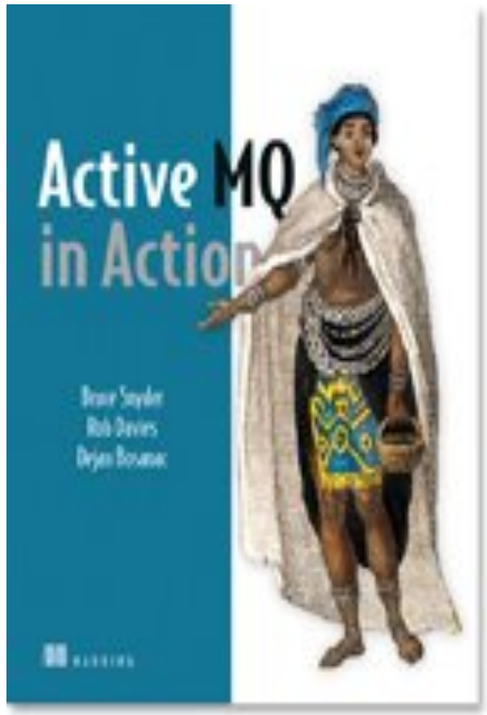
Connecting Applications Everywhere with ActiveMQ

Rob Davies
Technical Director, Fuse Engineering,
Red Hat Inc.

Rob Davies

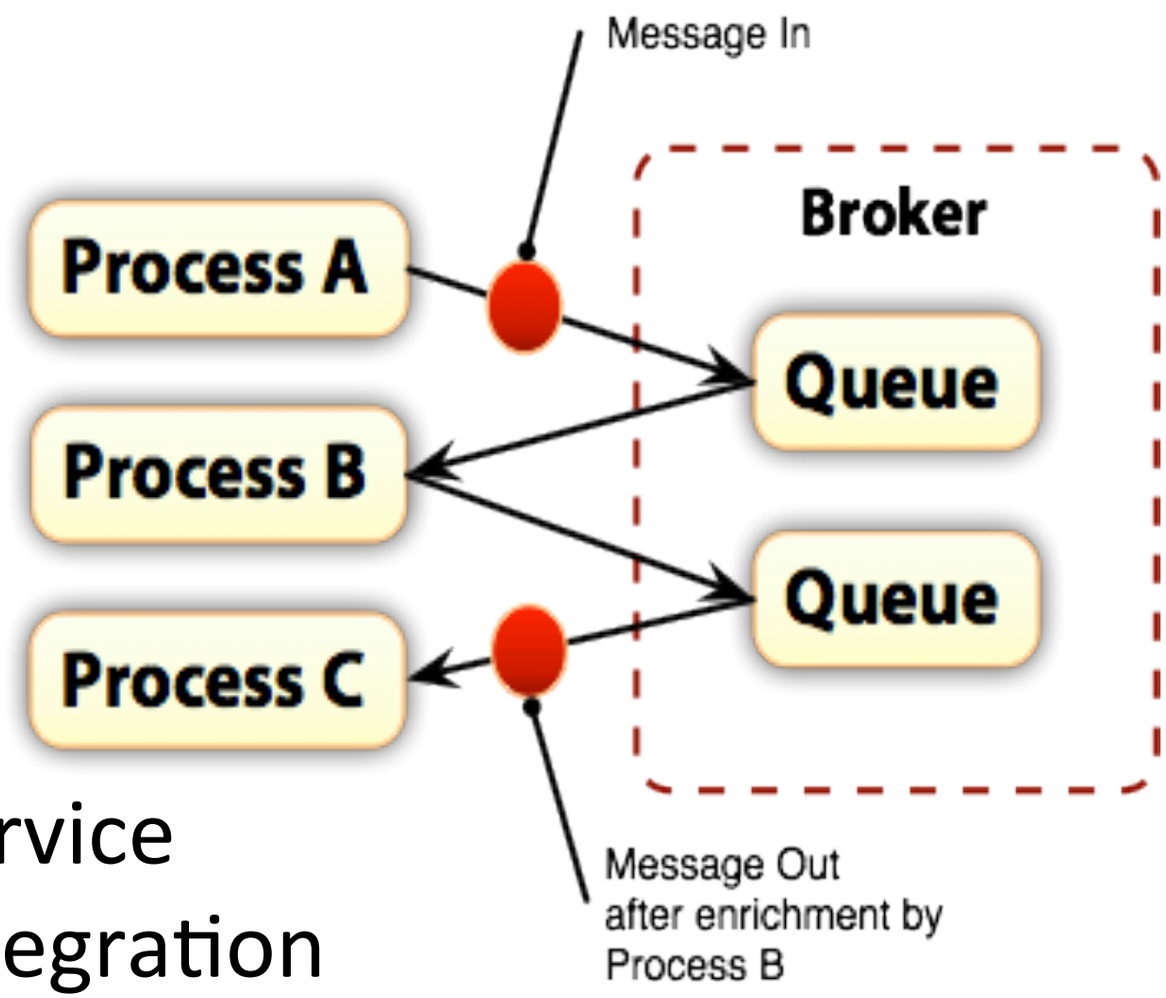
Technical Director, Red Hat -
http://www.redhat.com

- Previously CTO of FuseSource - #1 OS vendor for integration and messaging
- Software projects:
  - Apache ActiveMQ,
  - Apache Camel
  - Apache ServiceMix
- On Expert Group for JSR 343: JMS 2.0
- Co-author of ActiveMQ in Action:



Why use Message-Oriented Middleware?

- Robustness to change
- Time Independence
- Location Independence
- Hide Latency
- Scalability
- Event driven
- Simplicity
- Configurable Quality of Service
- Platform and Language Integration
- Fault tolerant



What is Apache ActiveMQ ?

- Top Level Apache Software Foundation Project
- Wildly popular, high performance, reliable message broker
- Connects to nearly everything
  - Native Java, C/C++, .Net,
  - AMQP 1.0, MQTT 3.1, STOMP (1.0-1.2) and OpenWire
  - STOMP enables Ruby, JS, Perl, Python, PHP, ActionScript …
- Embedded and standalone deployment options

CamelOne

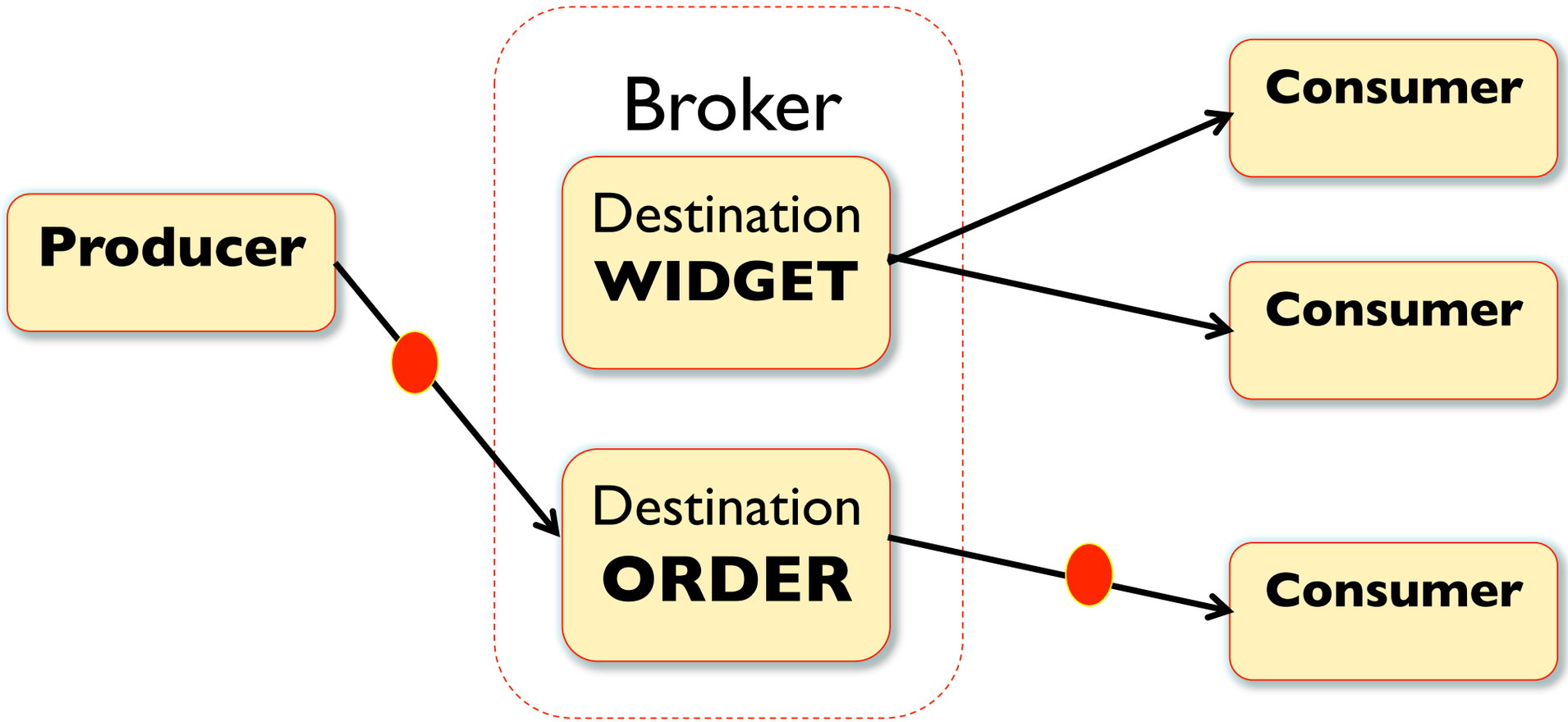
Messaging: The Basics

CamelOne 2013

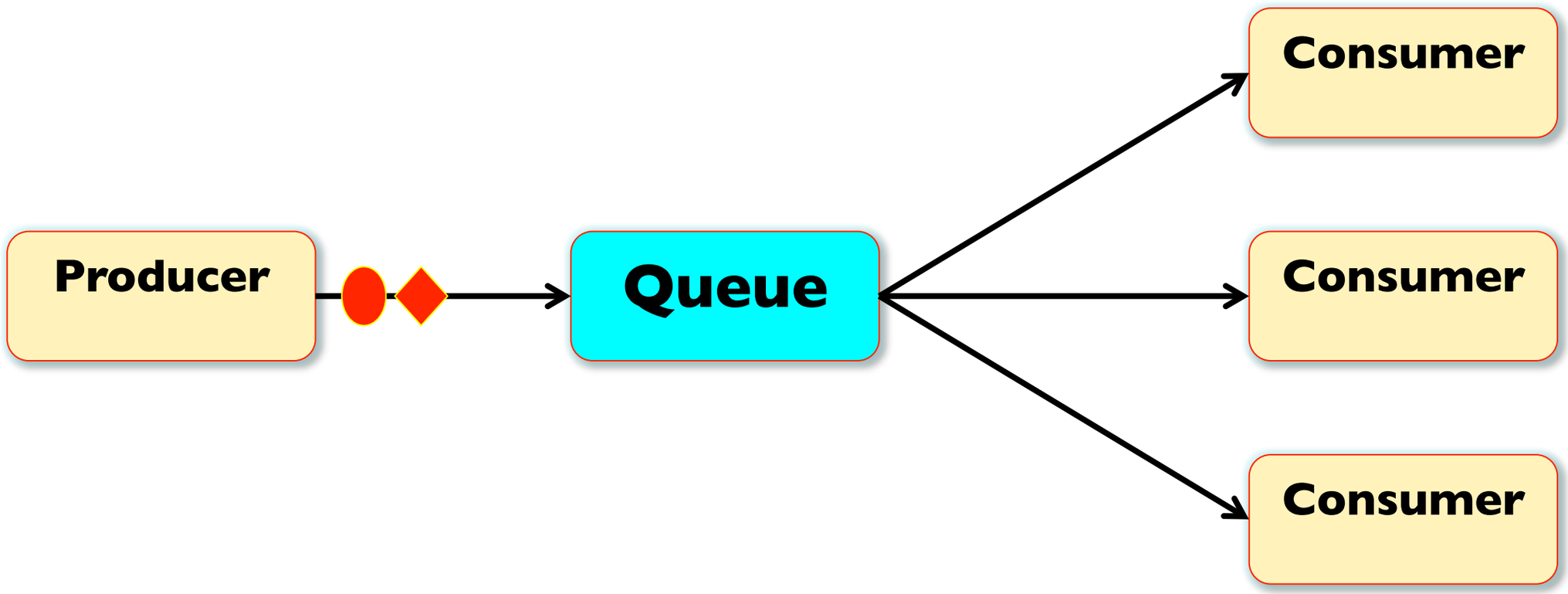
Message Channels and Routing

- Message Channels
  - Named communication between interested parties
  - JMS calls them ‘Destinations’
- Can “tune-in” to multiple channels using wildcards
- Can fine-tune message consumption with selectors
- Can route a message based on content

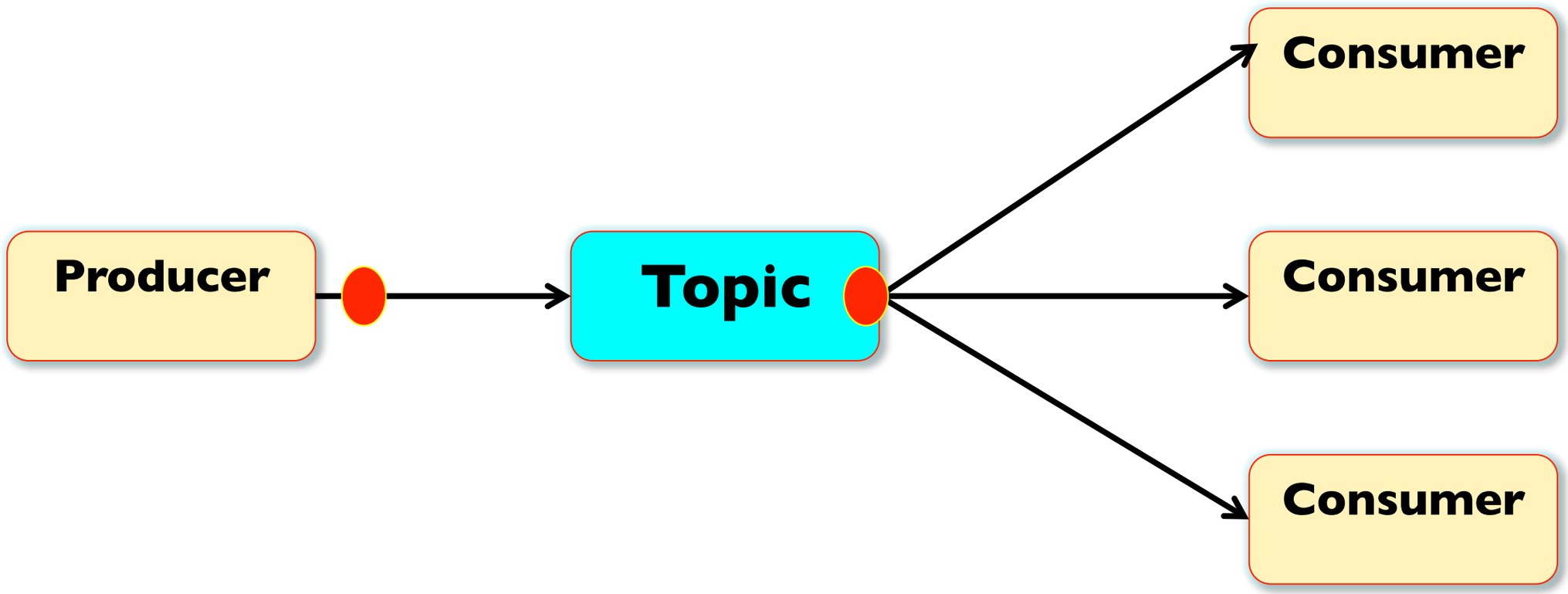
Message Channels = JMS Destinations



Point-toPoint Channel: JMS Queues

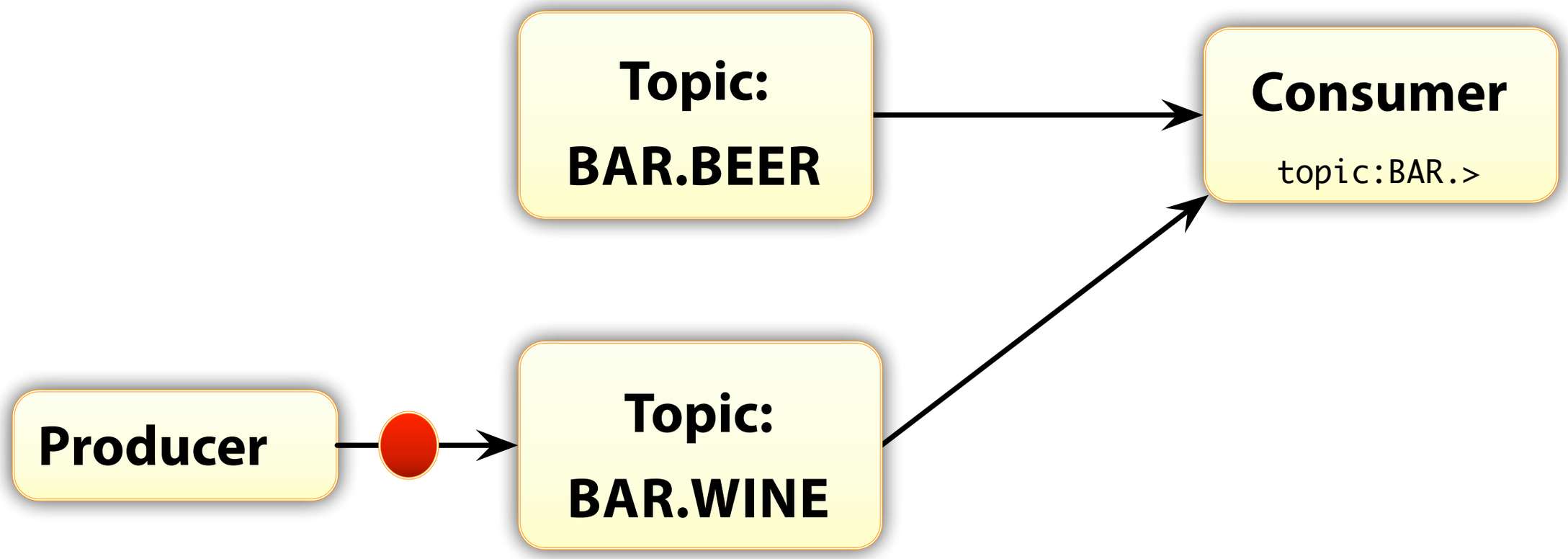


Publish/Subscribe Channel: JMS Topics

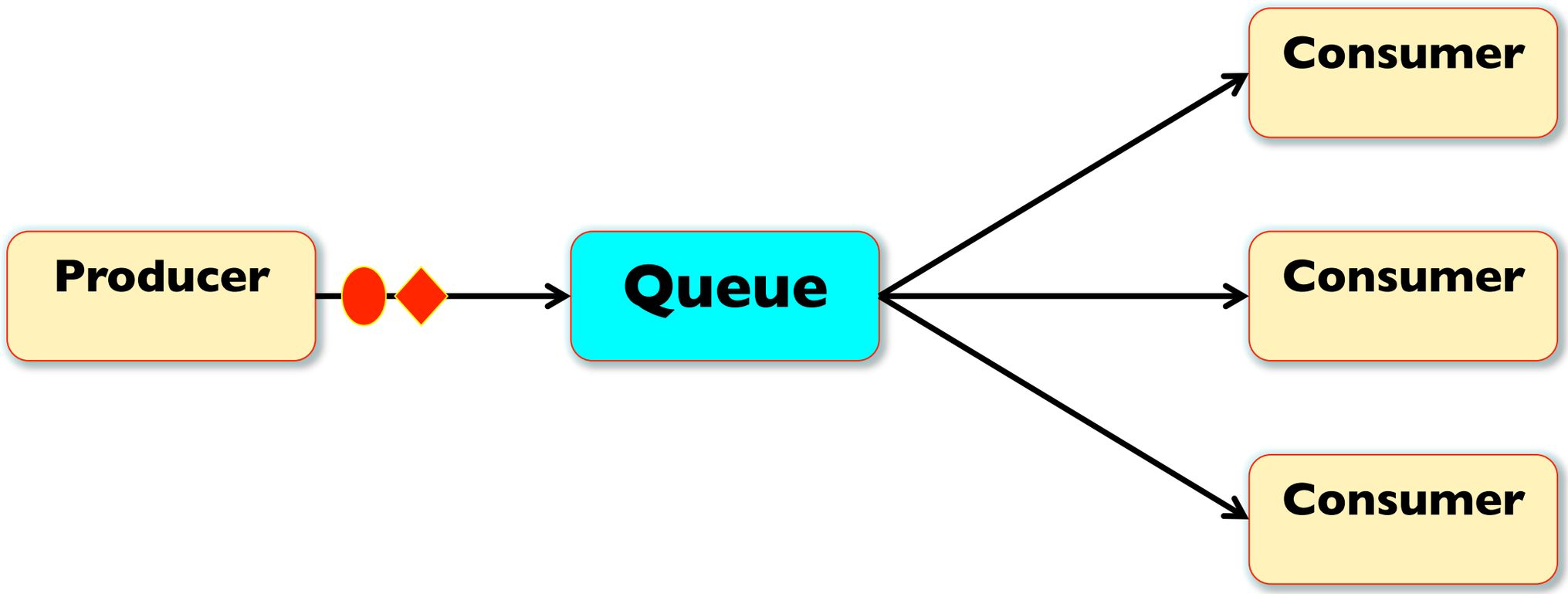


Message Routing : Selectors

Message Routing : Destination Wildcards



Exclusive Consumers



Message Groups

Like Parallel Exclusive Consumers

- Guaranteed ordering of related messages across a Queue
- But – load balancing of messages across multiple consumers
- All messages with the same **JMSXGroupID** go to the same consumer
- How you group messages is down to the application's producer
- To explicitly close a group, set the **JMSXGroupSeq** to -1

Message Groups code:

```
//Starting a Group ...

Message message = session.createTextMessage("<foo>hi from Devoxx</foo>");

message.setStringProperty("JMSXGroupID", ”RHT_NYSE");

producer.send(message);

//Close a Group ...

Message message = session.createTextMessage("<foo>bye from Devoxx</foo>");
message.setStringProperty("JMSXGroupID", ”RHT_NYSE");
message.setStringProperty("JMSXGroupSeq", -1);
producer.send(message);
```

Deploying ActiveMQ

ActiveMQ Can run standalone or embedded

- As a standalone, or part of a highly available message broker cluster
- Embedded – easy to use an entire broker in JUnit tests (no need to Mock)
- In Tomcat, deployed as a war
- In JEE Server – either co-locate – or use client with JCA
  - ActiveMQ distributions contain a rar for this purpose

CamelOne

ActiveMQ: Message Storage

CamelOne 2013

Message Delivery Mode

- **Messages can be persisted – allowing for decoupled applications**
- For Queues, any message delivered as **PERSISTENT** must be stored on long term storage ***before*** being delivered to a consumer
- For Topics, a message delivered as **PERSISTENT** will only be stored if there exists a durable subscriber
- **NON-PERSISTENT** messages may also be stored on disk if the memory limits of the broker have been reached

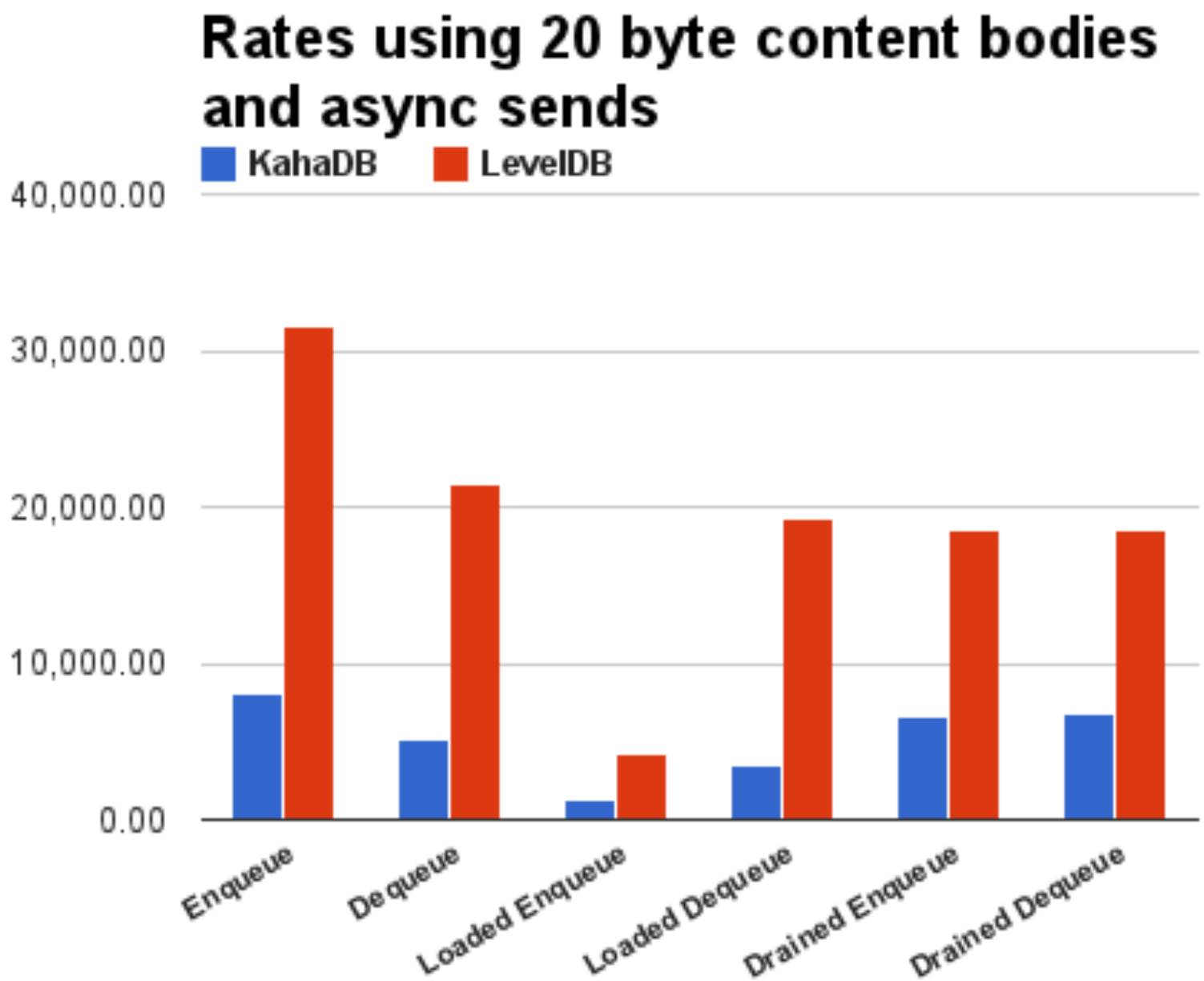
Message stores supported by ActiveMQ

- SQL (JDBC) ← Slow but popular
- SQL (JDBC) with journal
- AMQ Message Store ← Don't use (deprecated)
- Kaha
- KahaDB ← The "current" default
- LevelDB ← The "best yet"

LevelDB Store vs KahaDB

- Fewer index entries per message than KahaDB
- Faster recovery when a broker restarts
- LevelDB index out-perform Btree index at sequential access .
- LevelDB indexes support concurrent read access.
- Pauseless data log file garbage collection cycles.
- Fewer IOPS to load stored messages.
- It exposes it's status via JMX for monitoring

ActiveMQ: LevelDB Store



CamelOne

ActiveMQ: Protocols

OpenWire

http://activemq.apache.org/openwire.html

Advantages:

- Fast – optimized for ActiveMQ
- client failover
- automatic reconnect
- Client load balancing
- Flow control
- Many advanced features

Disadvantages:

- Not a recognized standard
- Only Java, C/C++/.Net

MQTT http://mqtt.org

Advantages:

- M2M/"Internet of Things" transport
- Proposed as an OASIS standard
- Extremely light weight
- Growing support from vendors and OS products
  - WebSphereMQ
  - ActiveMQ + Apollo
  - Mosquitto
  - RabbitMQ

Disadvantages:

- 3.1 does not support Queues
- Advanced features not standard
  - Flow control
  - Failover etc.

AMQP – see www.amqp.org

Advantages:

- AMQP 1.0 OASIS standard
- Proposed as an OASIS standard
- Commoditizes the Broker

Disadvantages:

- One size doesn't really fit all
- Currently no plans for IBM or Tibco to adopt it

STOMP – http://stomp.github.com

Advantages:

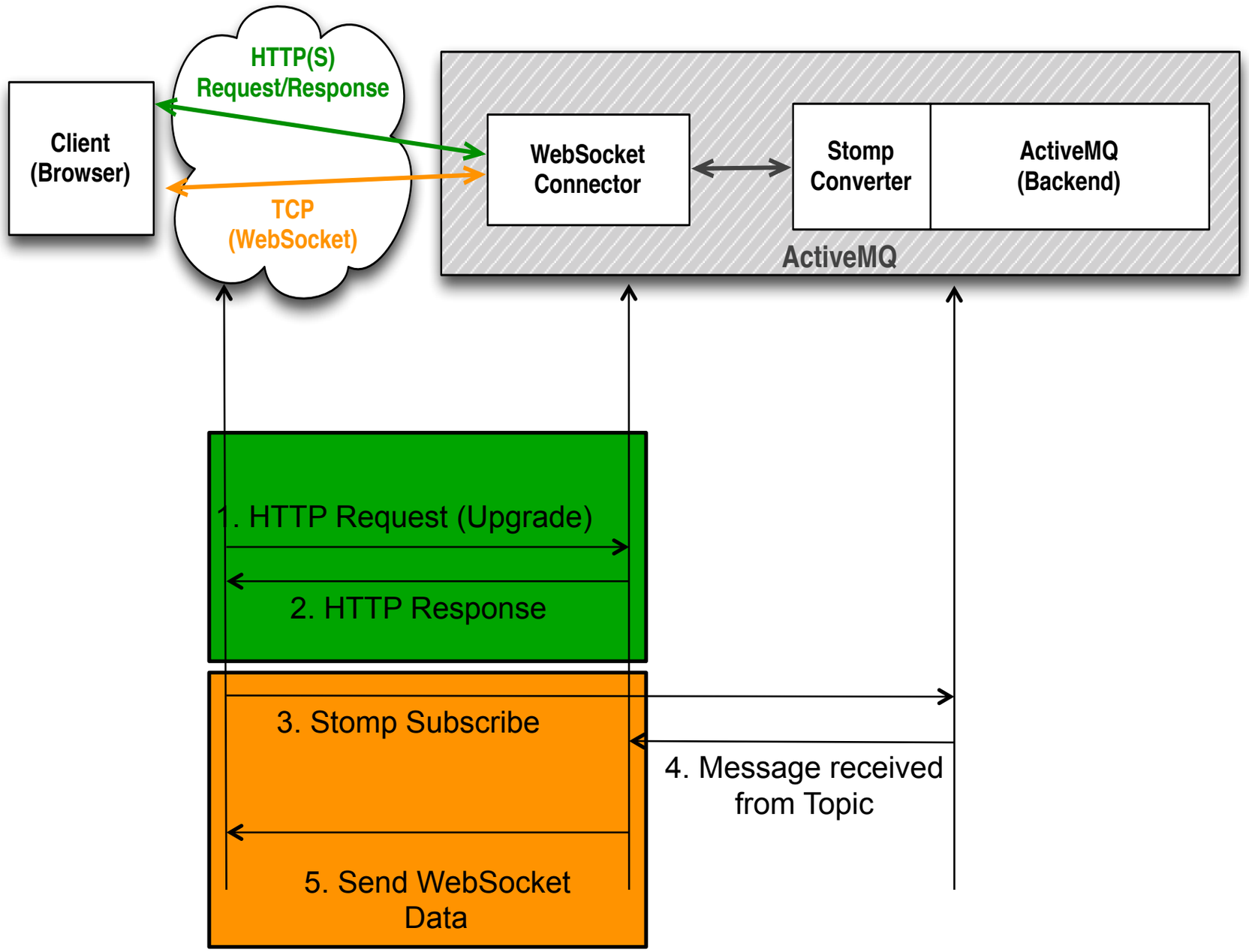
- Easy to use text based protocol
- Can use telnet as a client
- Defacto standard:
  - ActiveMQ + Apollo
  - HornetQ
  - RabbitMQ
  - PocoMQ
  - StompServer
  - OpenMQ
  - Many more …

Disadvantages:

- Not as fast as binary formats

But there's more: WebSockets

STOMP is a natural wire protocol for WebSockets



CamelOne

ActiveMQ: Management

Command line:

- Activemq
- activemq-admin
  - activemq-admin stop
  - activemq-admin query
  - activemq-admin bstat
  - activemq-admin browse
  - activemq-admin list

Runs the broker

Manages the broker

ActiveMQ WebConsole

http://localhost:8161/admin

ActiveMQ

The Apache Software Foundation
http://www.apache.org/

Home | Queues | Topics | Subscribers | Connections | Send | Support

Topic Name [ ] Create

Topics

Name	Number Of Consumers	Messages Sent	Messages Received	Operations
ActiveMQ.Advisory.Connection	0	2	0	Send To Delete
foo.bar	1	1	0	Send To Delete
ActiveMQ.Advisory.Topic	0	2	0	Send To Delete
ActiveMQ.Advisory.Consumer.Queue.ID:nc-rmerritt...	0	1	0	Send To Delete
ActiveMQ.Agent	1	0	0	Send To Delete
ActiveMQ.Advisory.TempQueue	1	1	0	Send To Delete
ActiveMQ.Advisory.Consumer.Topic.foo.bar	0	1	0	Send To Delete
ActiveMQ.Advisory.Consumer.Topic.ActiveMQ.Agent	0	1	0	Send To Delete

Queue Views
- Graph
- XML

Useful Links
- Documentation
- FAQ
- Downloads
- Forums

Copyright 2005-2007 The Apache Software Foundation. (printable version)

Introducing hawtio …

hawtio

localhost:8080/hawtio/#/activemq/browseQueue?tab=messaging&nid=root-org.apache.activemq-broker1-Queue-my.queue

hawtio | Integration | Messaging | Dashboard | Jetty | JMX | Logs | Wiki | Preferences | Help

- broker1
  - Broker
  - Connection
  - Connector
  - LevelDBStore
  - NetworkBridge
  - NetworkConnector
  - Producer
  - Queue
    - browse.me
    - my.queue
    - personnel.records
  - Subscription
  - Topic
- broker2

Browse | Send | Diagram | Delete Queue | Attributes | Chart | Operations

Search:

Message ID	Correlation ID	Timestamp	Delivery Mode	Reply To	Redelivered	Priority	Group ID	Expiration	Type	Destin
ID:stracmac.local-51461-1361198034483-10:1:1:1:1		2013-02-18T14:37:37Z	NON-PERSISTENT		false	0	0	0		queue

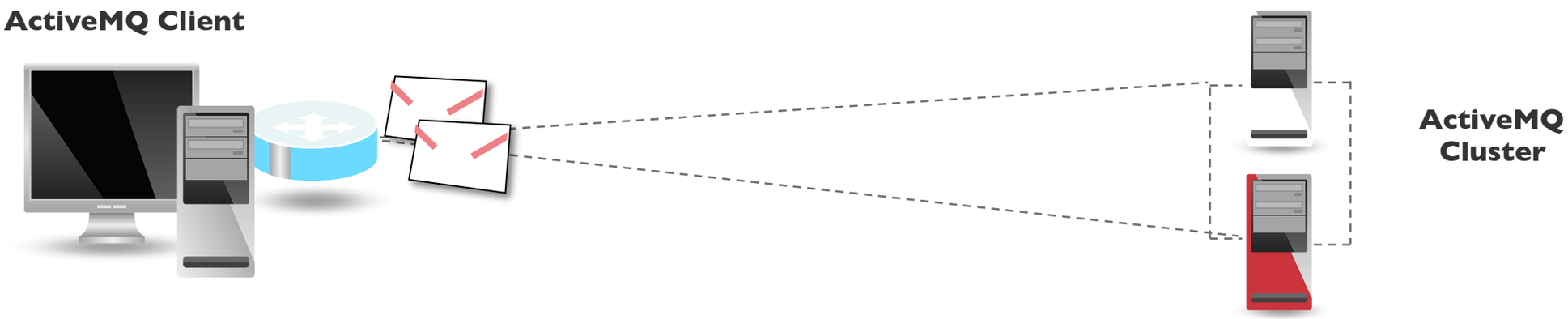
Headers

```
1 <hello>world!</hello>
```

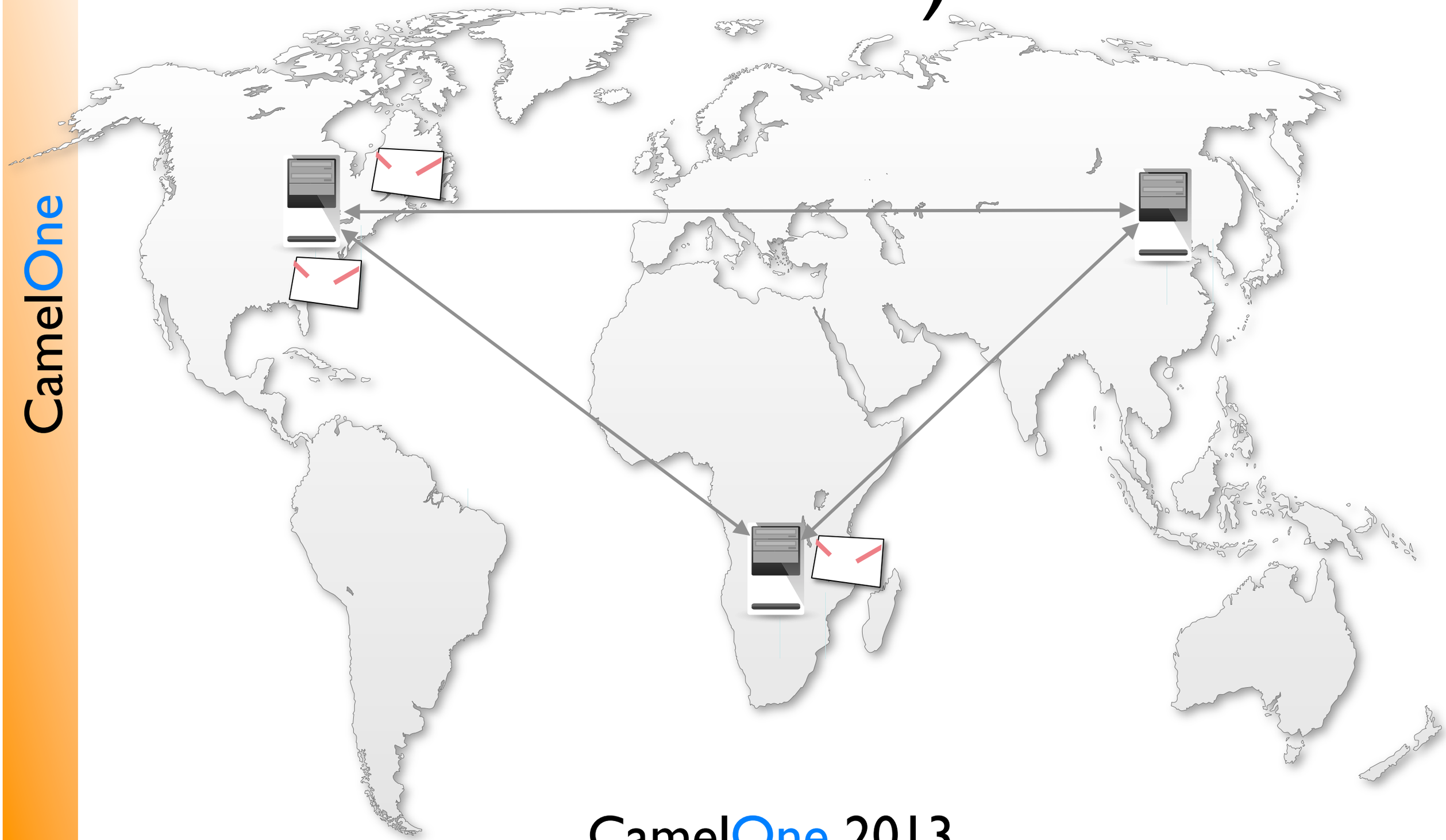
ActiveMQ: Enterprise Features

High Availability

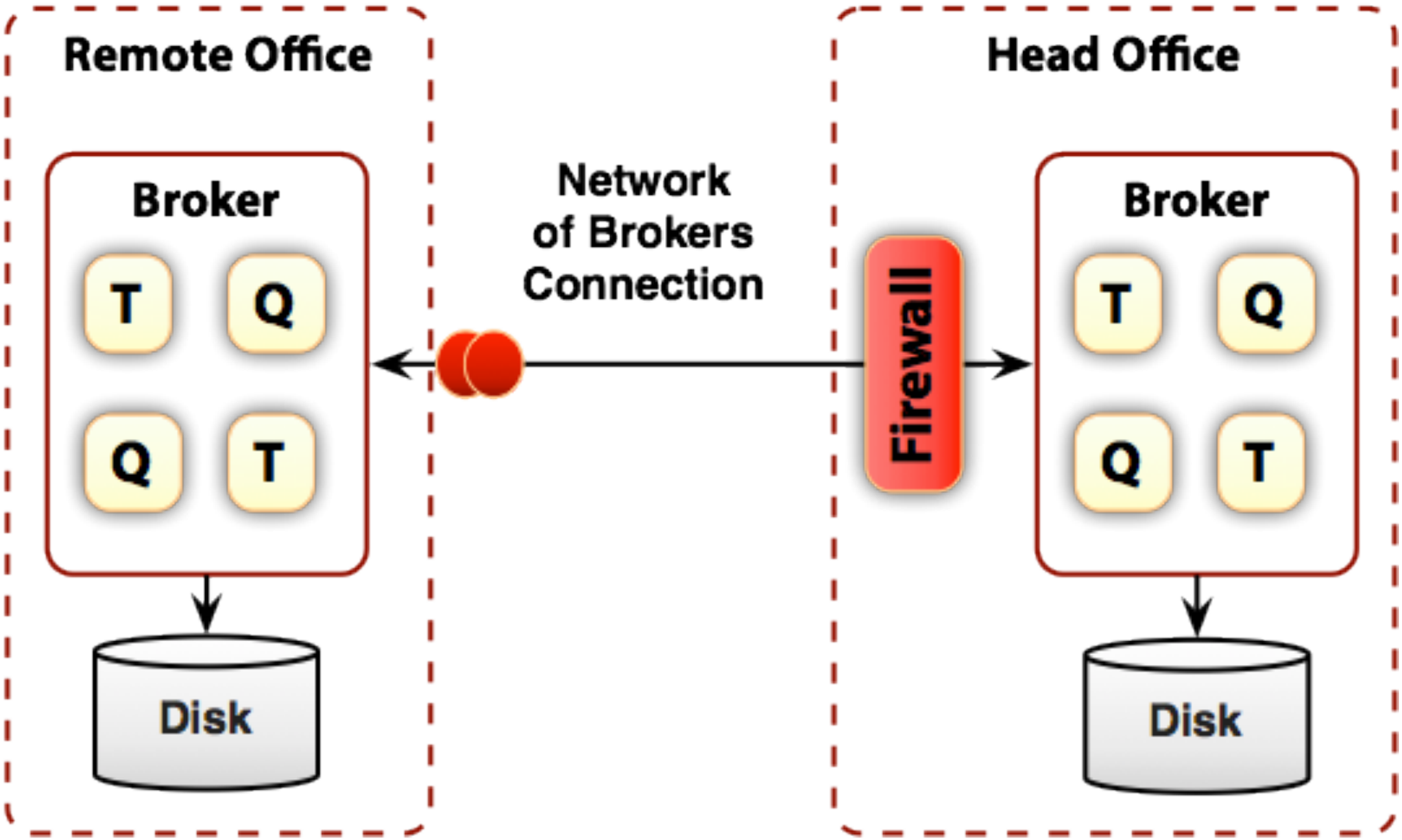
Supported by both Java and C++ OpenWire clients



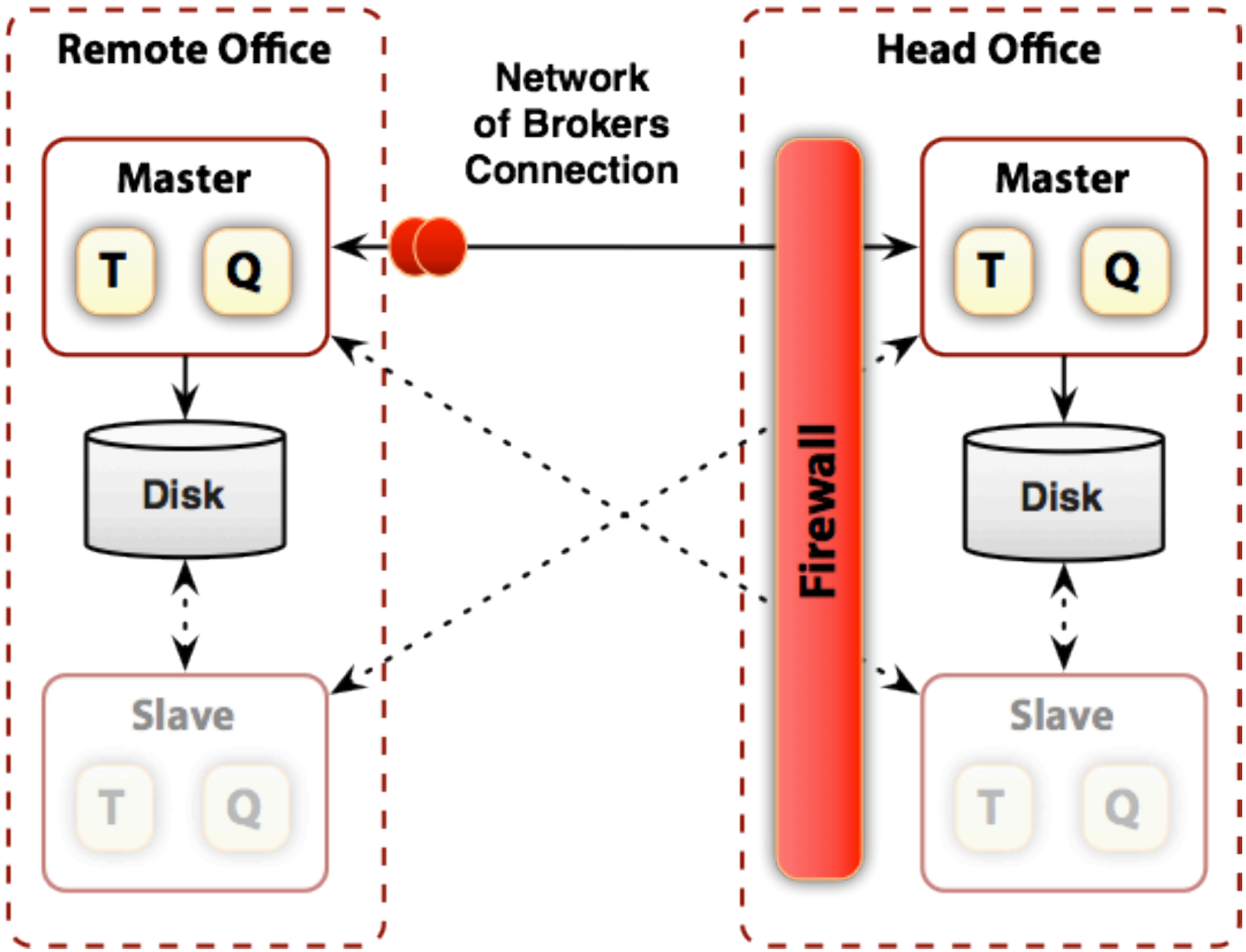
Broker Networks (Store and Forward)



Network of Brokers : Geographically Dispersed



Network of Brokers : Network with Master/Slave



Scaling Networks of Brokers

- Brokers share routing information across networks
- All destinations are considered Global
- This is really convenient ☺
- Though it starts to get problematic with 1000's of brokers ☹
- However – we can do filtering to shape the traffic across networks ☺
- But this involves a lot of manual configuration ☹ crikey!

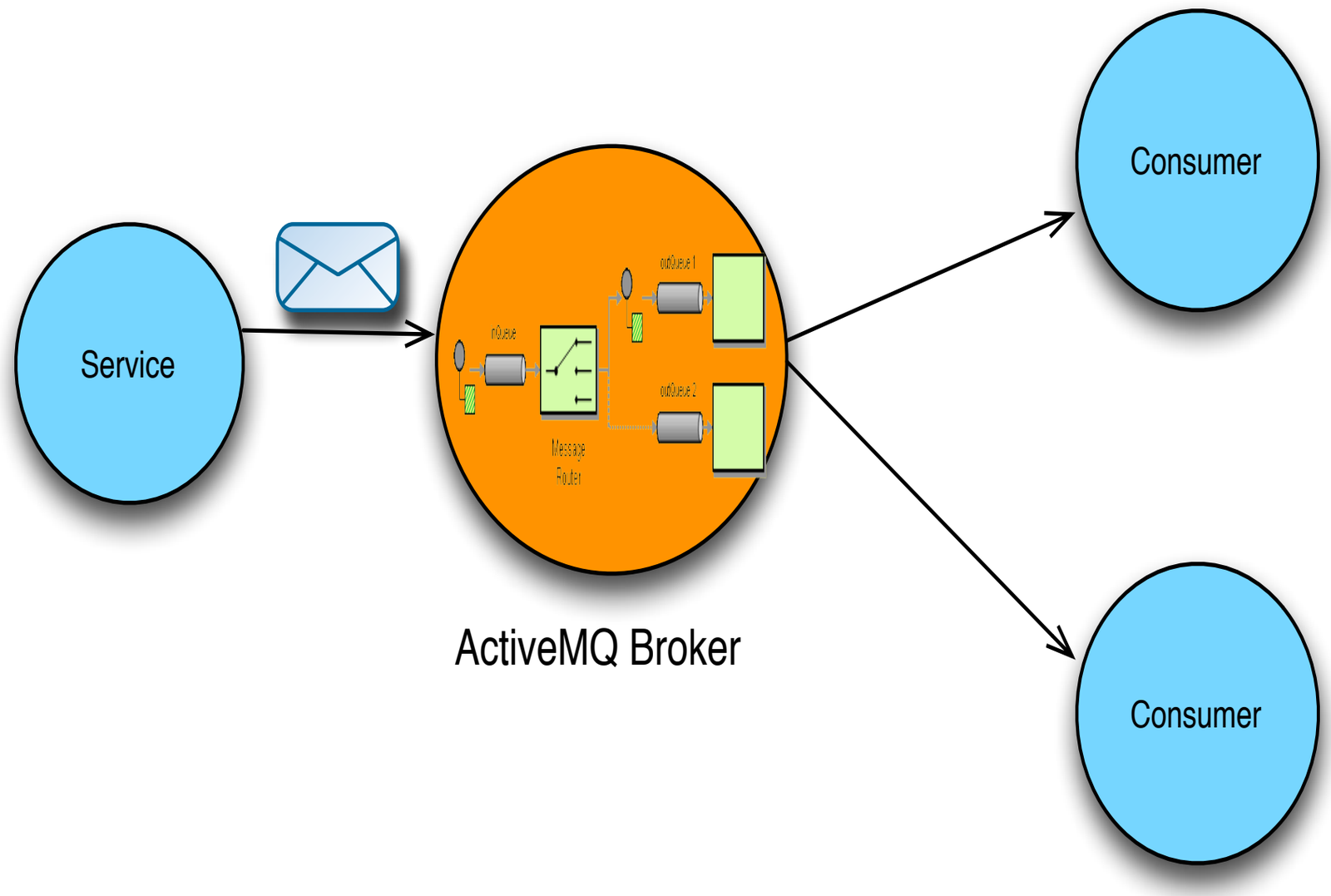
Scaling Networks of Brokers

- Brokers share routing information across networks
- All destinations are considered Global
- This is really convenient
- Though it starts to get problematic with 1000's of brokers
- However – we can do filtering to shape the traffic across networks
- But this involves a lot of manual configuration

Time to talk about Apache Camel ...

ActiveMQ with embedded Camel

Flexible and is faster ☺



ActiveMQ with embedded Camel

Import Camel into ActiveMQ broker config:

```
<beans>
  <broker brokerName="testBroker"  xmlns="http://activemq.apache.org/schema/core">
    <transportConnectors>
      <transportConnector uri="tcp://localhost:61616"/>
    </transportConnectors>
  </broker>
  <import resource="camel.xml"/>
</beans>
```

Setup Camel Context in the usual way

```
<beans>
…
    <camelContext id="camel" xmlns="http://camel.apache.org/schema/spring">
        <route>
            <description>Example Camel Route</description>
            <from uri="activemq:topic:audio"/>
            <setHeader headerName="JMS_AMQP_MESSAGE_FORMAT">
                <constant>0</constant>
            </setHeader>
            <to uri="activemq:queue:audio"/>
        </route>
    </camelContext>
…
</beans>
```

Scaling Networks – use Apache Camel

- Allows for non-chatty networks to be established
- Routing information can be externalized to the broker
- Successfully used in production for 1000's of brokers today

CamelOne

More problems with Large Deployments

CamelOne 2013

Problems – Deploying and maintenance

- Main problems
  - Installing brokers on multiple hosts
    - ssh, untar, set directories and environment
  - Setting configuration manually for every broker
    - copying xml config, tweaking, testing
  - Updating configuration across cluster
  - Upgrading brokers

It's a tedious and error-prone process

Problems – Traditional best-practice tips

- Keep XML as a template and configure node-specific details through properties
- Keep configuration in SVC system (git, svn, ...)
- Keep configuration separate from installation for easier upgrades

Deployment with Fuse Fabric moves it to the next level

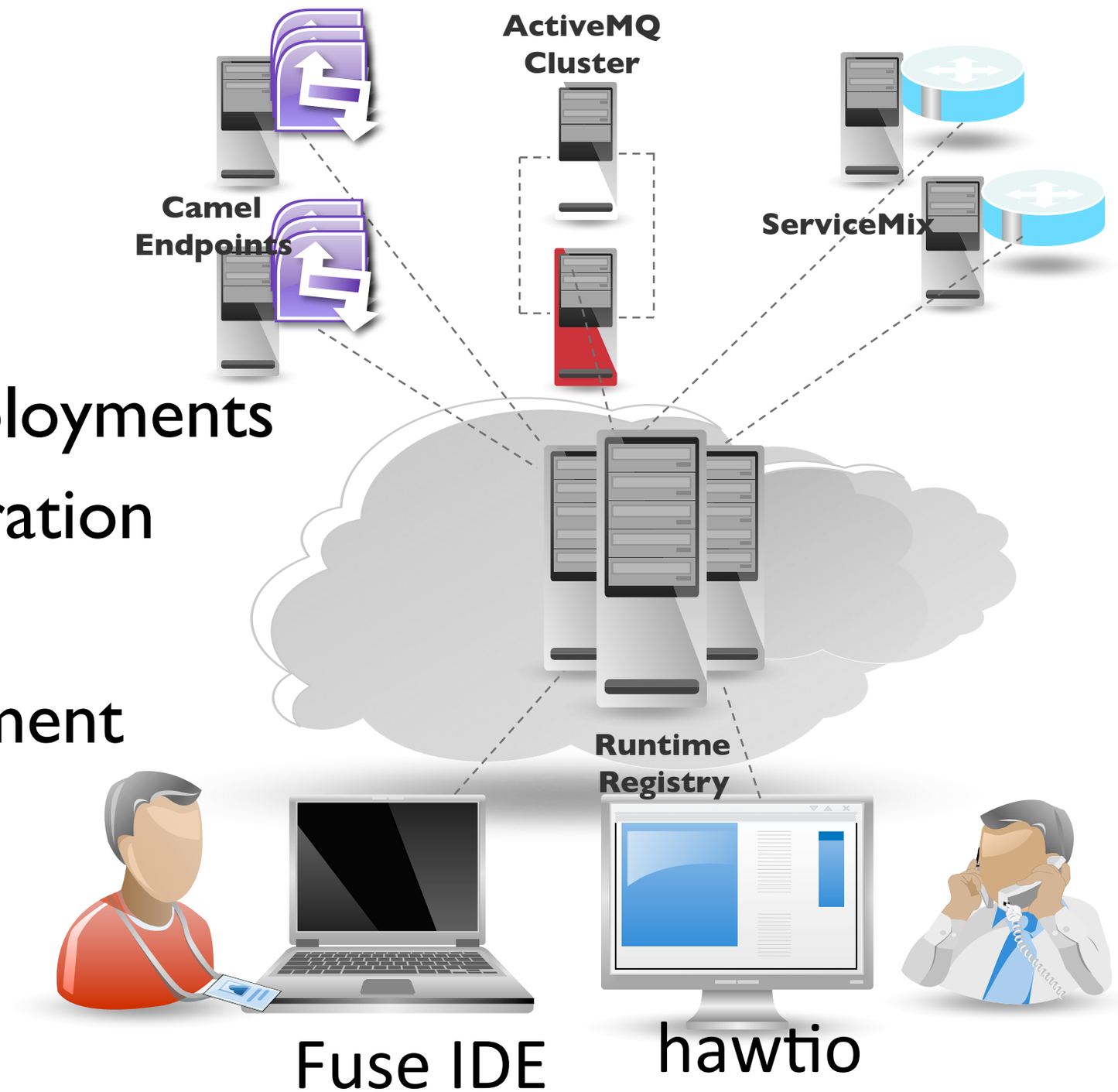
Problems - Clients

- Topology is very “static”
- Clients need to be aware of topology
- Clients need to know broker locations
- Changes are not easy as clients need to be updated
- Adding new resources (brokers) requires client updates
- Not suitable for “cloud” deployments

Fuse Fabric makes deployments more “elastic”

Fuse Fabric to the rescue!

Fuse Fabric – Key Features



- Supports Hybrid deployments
- Distributed Configuration
- Runtime registry
- Centralized Management

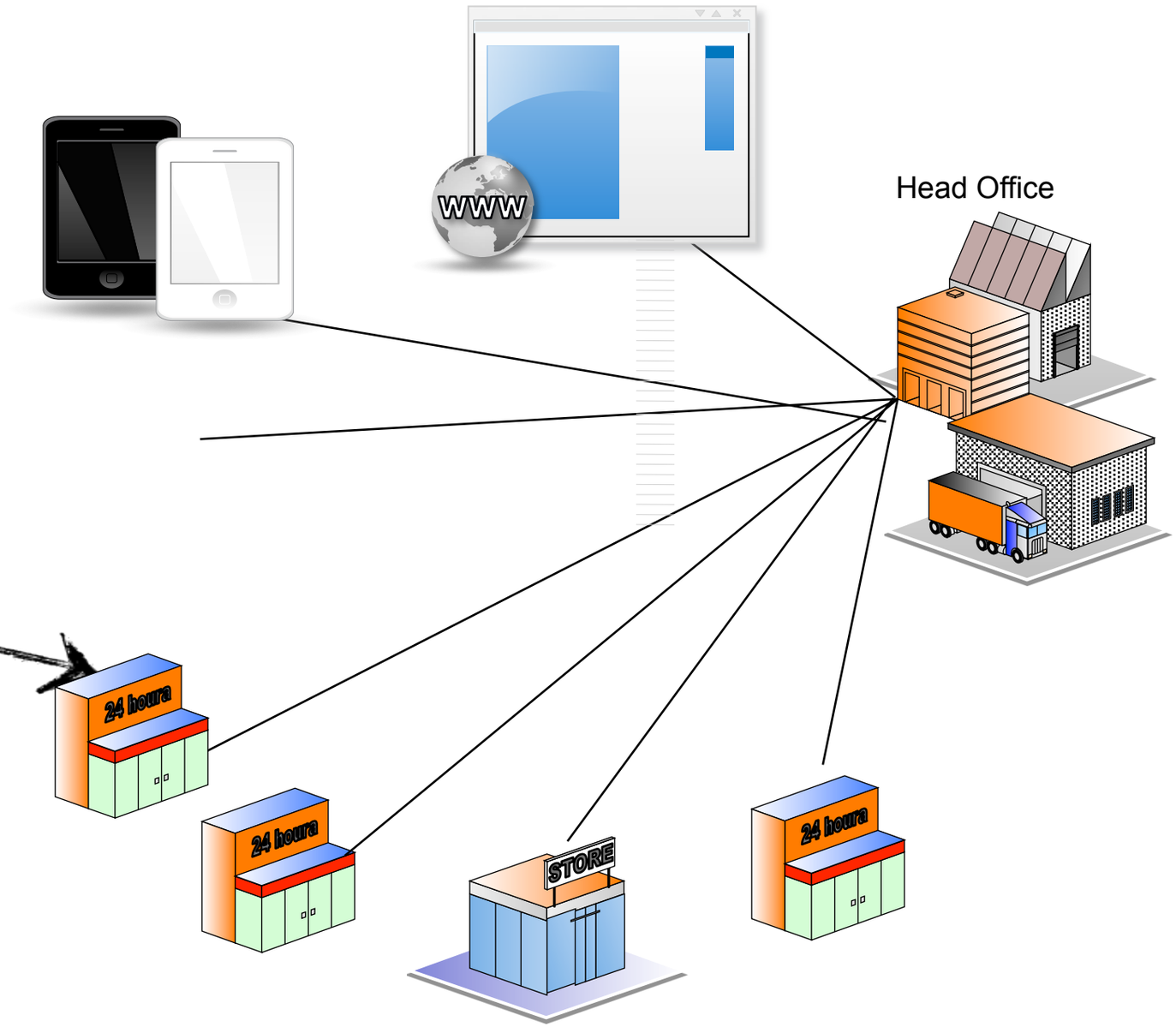
FuseMQ features

FuseMQ = ActiveMQ deployed in Apache Karaf

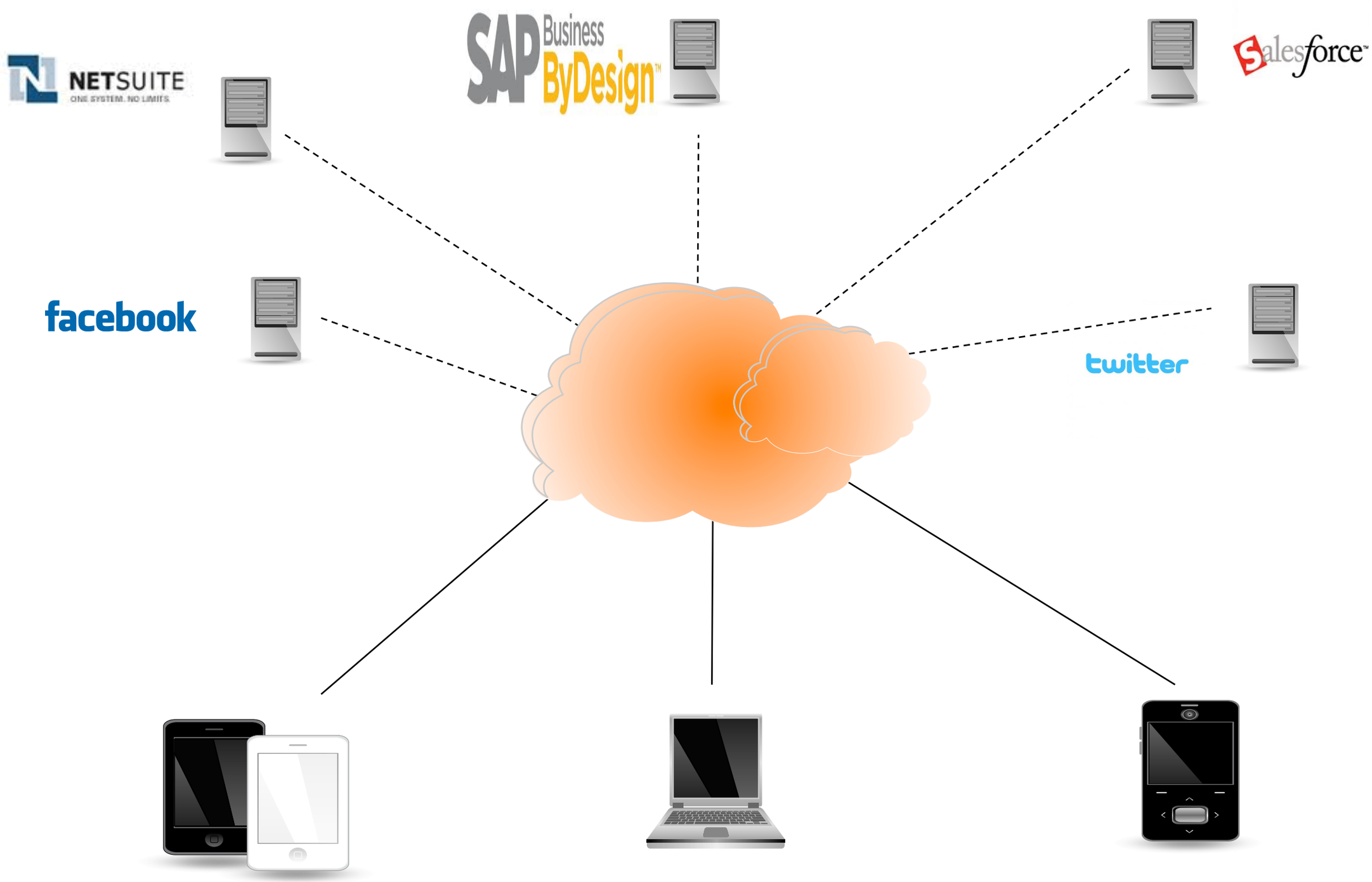
- mq-base profile
  - Defines OSGi features and bundles to be installed
  - Defines basic broker settings
- mq-create command
  - Helper command for creating brokers
  - Creates an new profile based on mq-base
  - Optionally creates new containers
  - Assigns the profile to containers (essentially starts the broker)

Why is Integration and Messaging Important ?

Enterprises Need to know Everything!



There are more things to Integrate with!



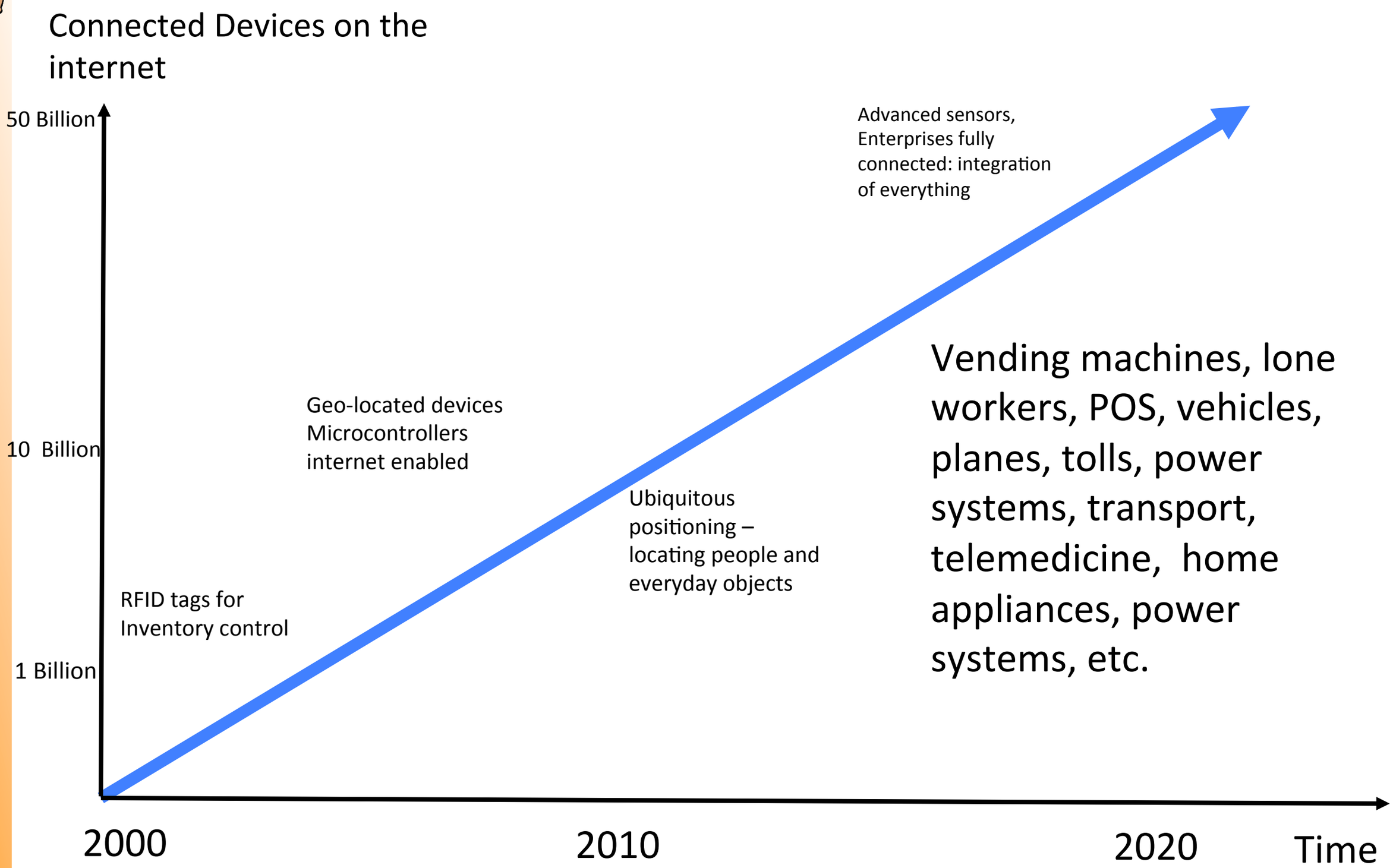
The Internet of Things

- Uniquely identifiable objects and their virtual representations in an internet-like structure
- Originally defined by Kevin Ashton, co-founder of Auto-ID center at MIT
- Today its meaning has evolved to encompass a world where physical objects are integrated into the information network, and where physical objects participate in business processes.

Machine to Machine

- M2M involves collection and transmission of event level data (used to be called telemetry) from intelligent devices
- Machines can interact over the net – and/or attached networks – by wired and wireless networks
- Payloads are typically minimal (temperature, pressure, location, metering etc.)

Internet of Things



M2M Examples

- Smart Energy –

  Smart Grid uses smart meters for monitoring energy uses by premises for billing, substation and transmission line monitoring, energy production (renewables)

- Inventory Control –

  smart labels and RFID tags – connected scanners pass information upstream to servers for monitoring

- In-Vehicle Systems

  Maintenance information, global positioning, 'black-box' (speed and driving habits), wireless payments for tolls/gas

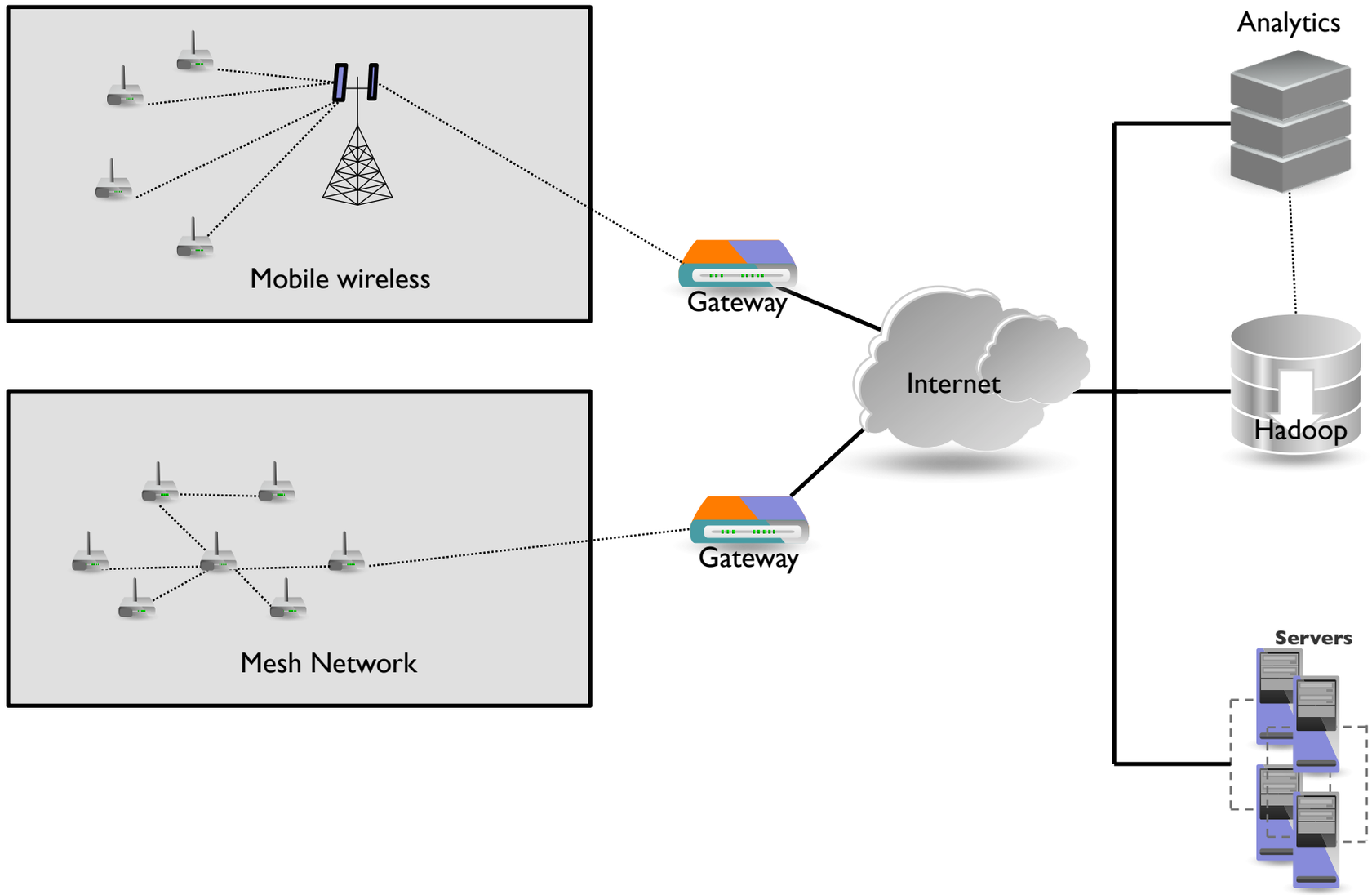
- Environmental Monitoring

  Weather stations (temperature, air-pressure), ocean monitors, earth movements, volcanic activity crop yields etc.

- Livestock monitoring

  Cow herds, monitored for optimum time for milking, sheep herds for location

M2M Topology



Power Consumption!

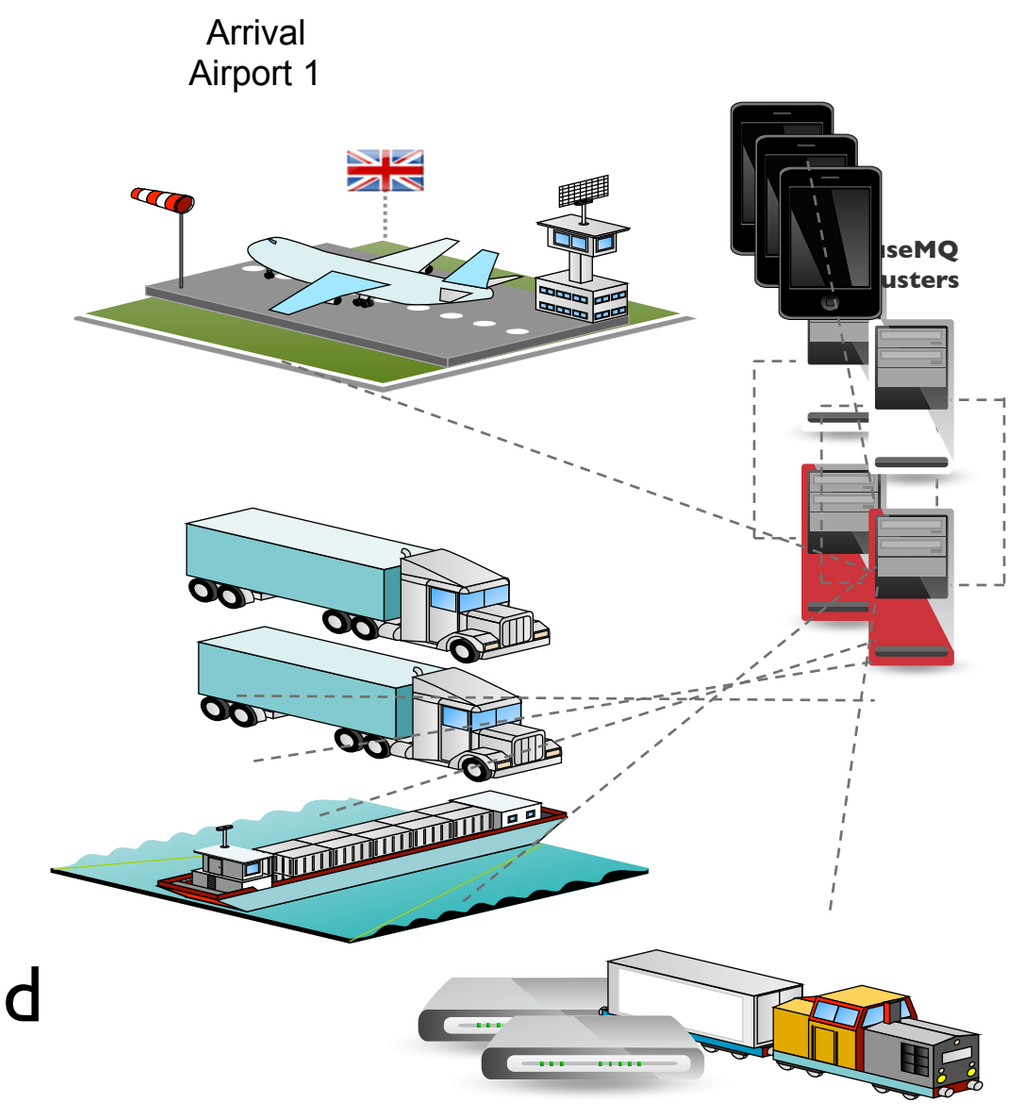
- Battery life for mobiles, power consumption for smart devices is an important consideration.
- Facebook use MQTT for real-time updates – efficiency was one of the things considered
- MQTT is more efficient at establishing connections, sends information more reliably, and consumes less power to transmit data than HTTP.

Deluge of Data

- Billions of devices send information will require storage and processing of terabytes or even petabytes of data.
- Big Data Ingestion: Processing of vast amounts of data business generates to make good decisions is only half the problem. Ingesting that data from all areas of the enterprise will require TBs of data to be ingested in highly reliable and scalable way.

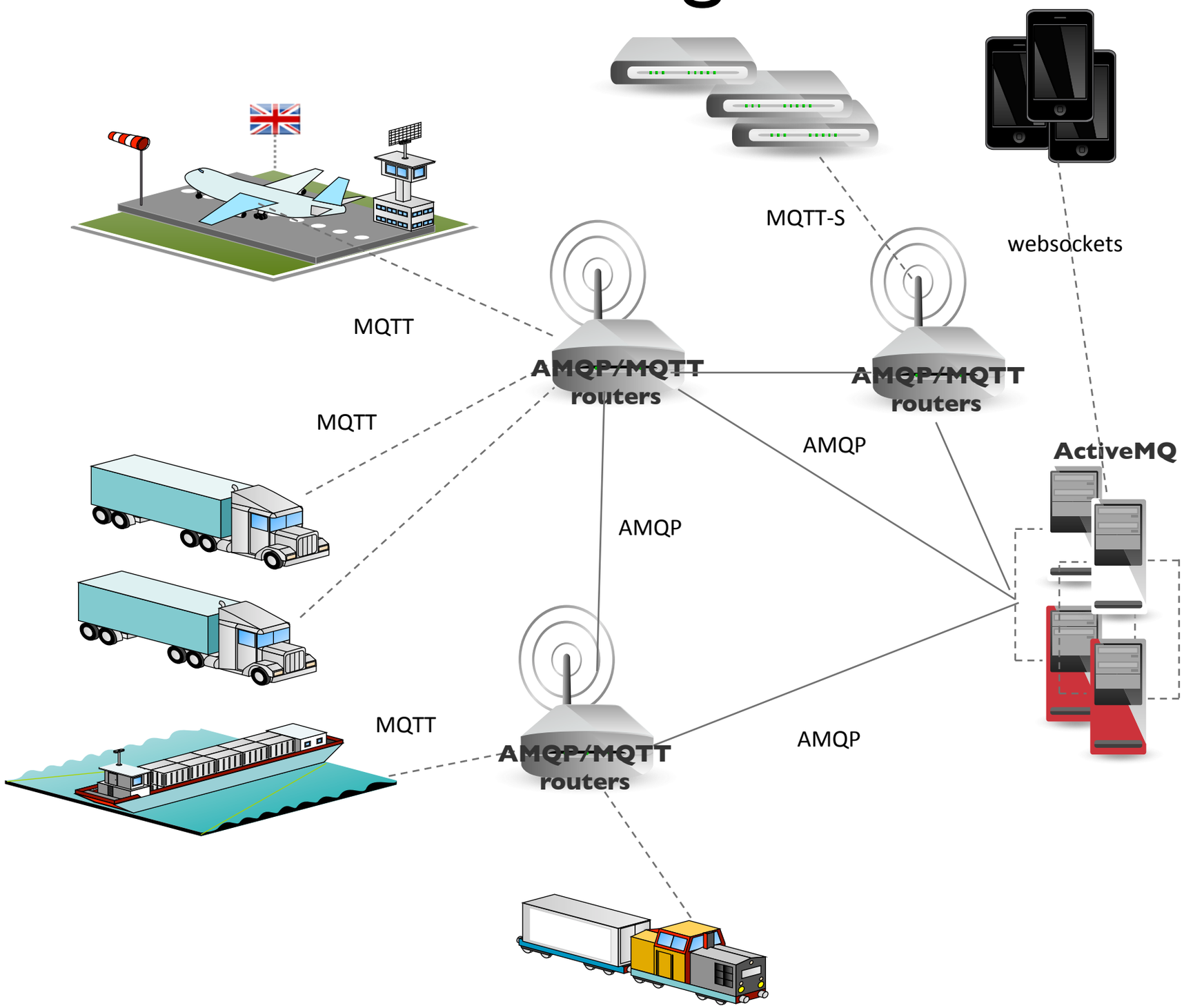
Need an Eco system of technologies

- Integration is getting harder
  - Traditional message brokers are key to connect "everything"
  - Multiple protocols need to be supported
- Camel can provide integration at the edges – and the data center
- Technologies like Fuse Fabric provide centralized provisioning and management
- Need to mix protocols, brokers and "message routers" to scale

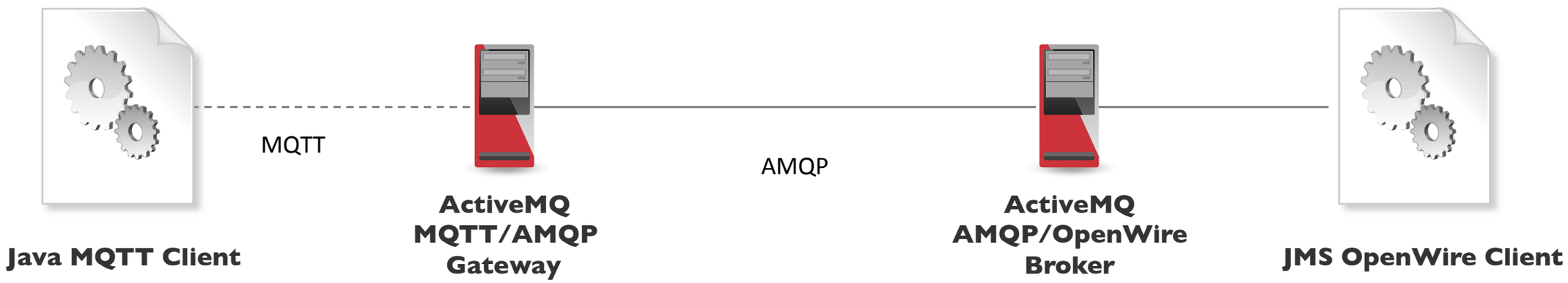


Demo Time!

Messaging through the Internet of Things



Simplified demo



What is Arduino ?

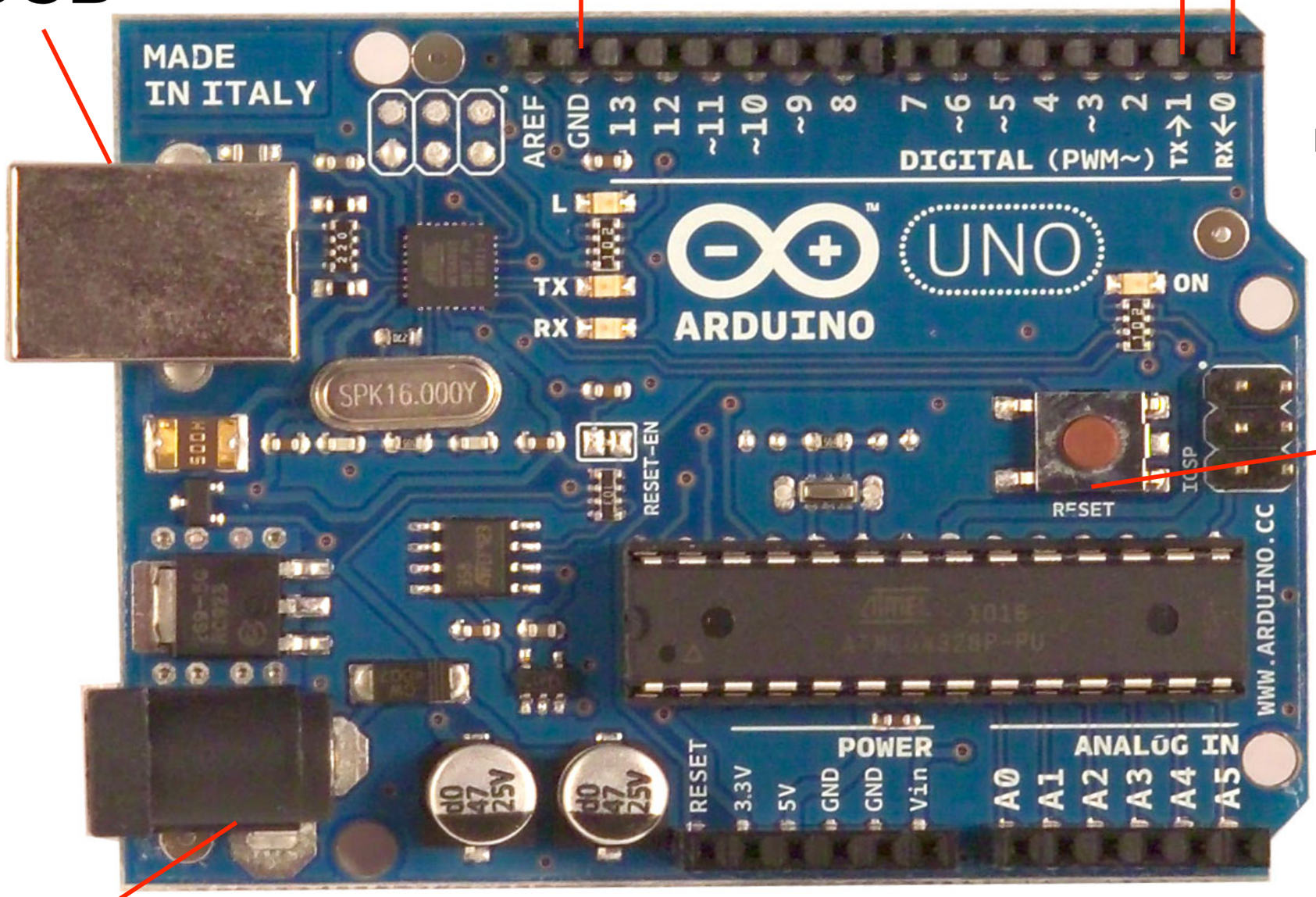
- Open Source Hardware microcontroller
- Cheap and easily available.
- Open Source Software.
- Widespread: many projects.
- Extra HW (shields) available (e.g. WiFi shield)

Digital Input/Output Pins
Pins with ~ are PWM
[Analog Output]

GRD

Transmitter/Receiver
Serial Connection

USB



Microcontroller ATmega328
Operating Voltage 5V
Input Voltage (recommended)7-12V
Input Voltage (limits)6-20V
Digital I/O Pins 14
(of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA

7-12 v

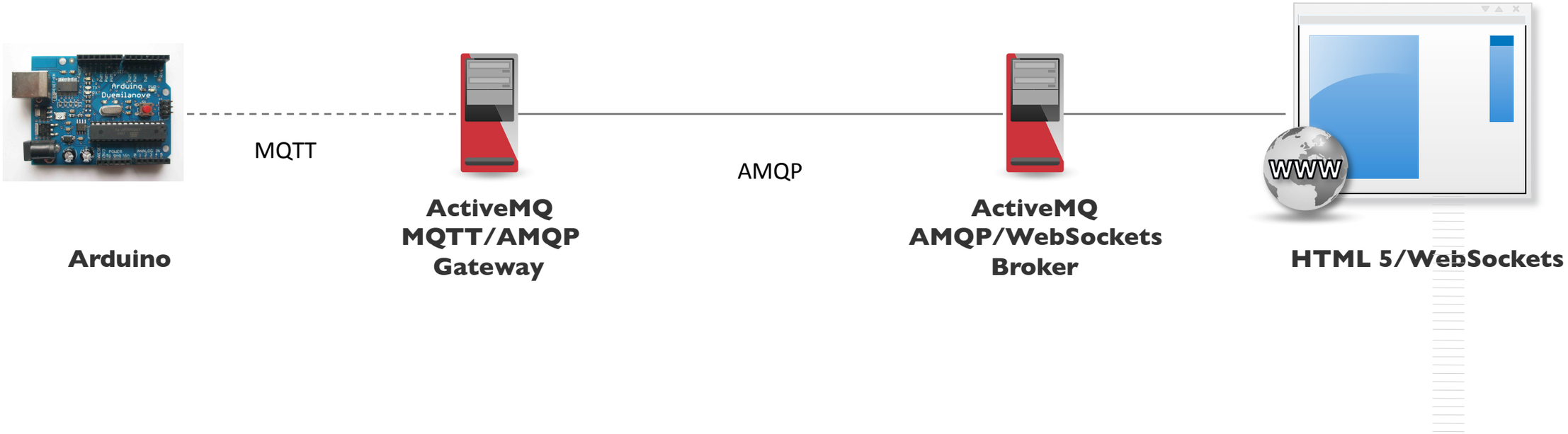
3 v GRD 5 v

Analog Input Pins

Arduino Programming

- 3 types of memory on Arduino
  - Flash memory (program space) – 32 Kb
  - SRAM – used by sketches – 2 Kb
  - EPROM – long term memory – 1 Kb
- Programming language based on Wiring: C/C++ library designed to make input/output easier. Programs are called sketches.
- Arduino has a simple IDE, available on Windows, Linux and Mac

Arduino MQTT to WebSockets



Useful Resources:

- http://activemq.apache.org
- http://camel.apache.org
- http://fuse.fusesource.org/fabric/docs/overview.html
- http://fuse.fusesource.org/mq/docs/mq-fabric.html
- https://github.com/fusesource/fuseide
- http://hawt.io
- http://www.arduino.cc/

Messaging and Integration

Configuration and provisioning

Developer tooling

Management

Arduino