Title: Resilient Enterprise Messaging

Presenters: Scott Cranton & Scott McCarty

Red Hat JBoss A-MQ

A small-footprint, high-performance, open source **messaging platform**

RED HAT JBOSS A-MQ

Development and tooling

Develop, test, debug, refine, deploy

JBoss Developer Studio including JBoss Fuse IDE

Reliable messaging

JMS/STOMP/NMS/MQTT, publishing-subscribe/point-2-point, store and forward

Apache ActiveMQ

Container

Life cycle management, resource management, dynamic deployment, security and provisioning

Apache Karaf + Fuse Fabric

Management and monitoring

System metrics, automated discovery, container status, automatic updates

JBoss Operations Network
+
JBoss Fabric Management Console

RED HAT ENTERPRISE LINUX

Windows, UNIX, and other Linux

What is JBoss A-MQ?

Open

Scalable

Reliable

*assuming you already know it does reliable messaging

JBoss A-MQ :: Open

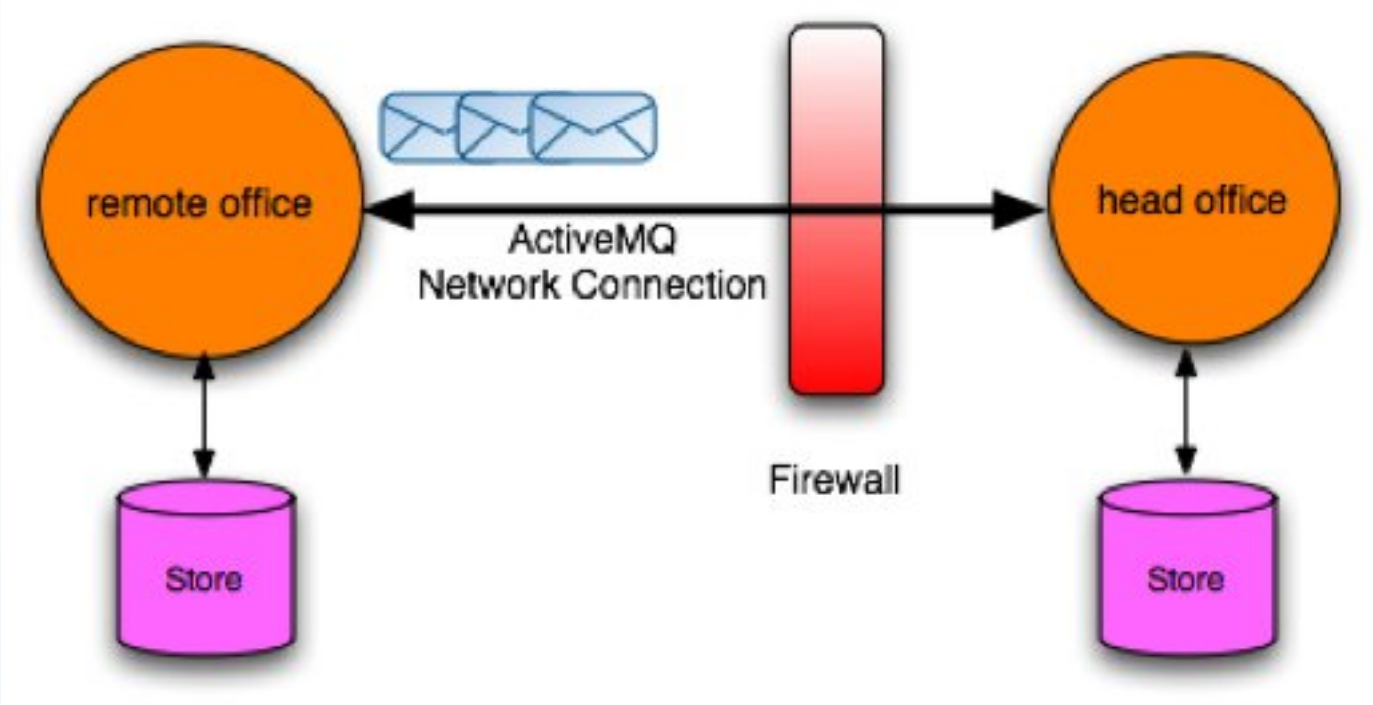
- Open Source
  - Apache Software License 2.0 licensed
  - Based on very popular Apache ActiveMQ
- Open Protocol
  - AMQP 1.0, MQTT, STOMP, OpenWire, …
- Open Language (Polyglot)
  - Client native languages: Java (JMS), C/C++, .NET
  - STOMP clients: Ruby, JavaScript, Perl, Python, PHP, ActionScript, ...

JBoss A-MQ :: Scalable

- Vertical Scaling
  - Only limited by I/O, compute, and I/O resources, …
- Horizontal Scaling
  - Network of Brokers (Clustering / Federation)
  - Fuse Fabric (https://github.com/jboss-fuse)
    - Central configuration management and provisioning
    - Client-side discovery, load balancing, and failover

JBoss A-MQ :: Scalable

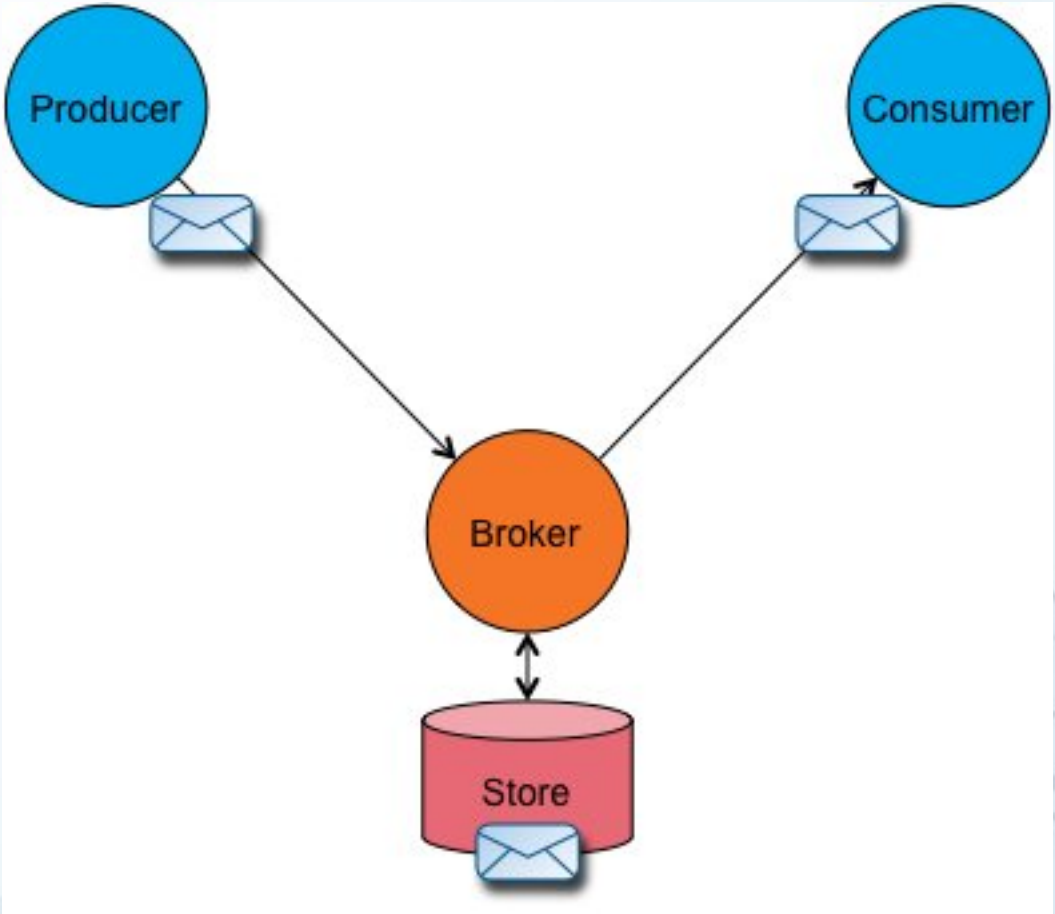
Network of Brokers



JBoss A-MQ :: Reliable

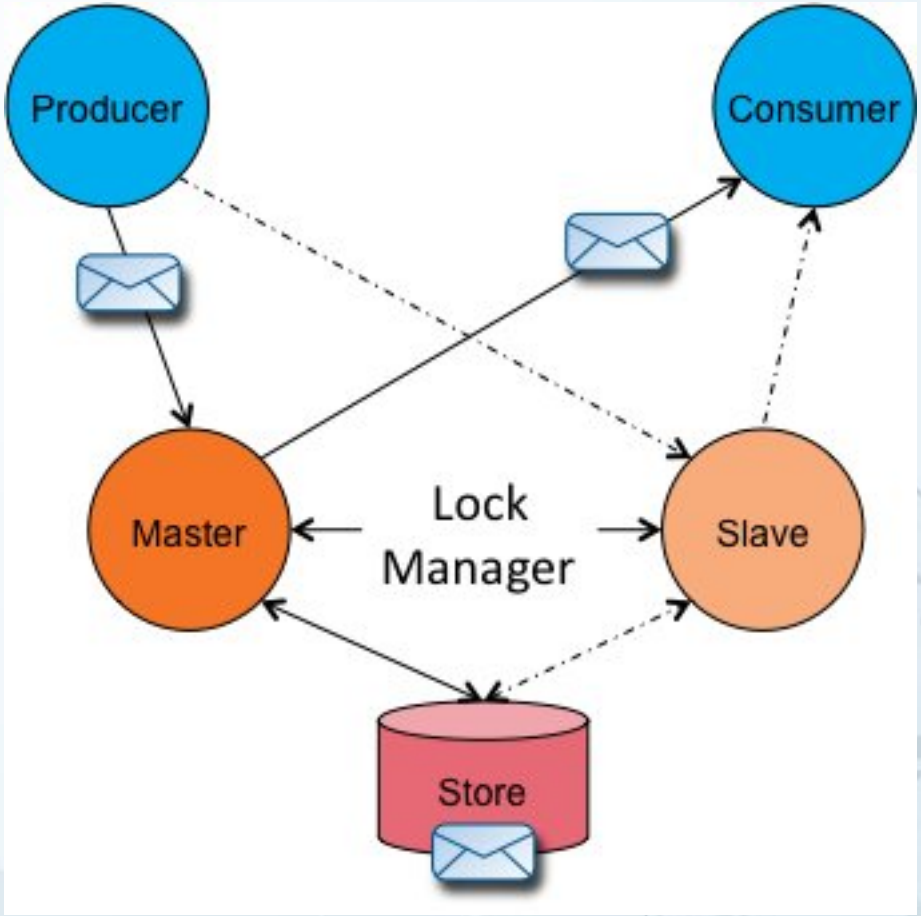
- Persistent Messaging
  - Store on File system or RDBMS
  - Survive restart, and process failure
- Master / Slave
  - Lock manager
  - Shared Storage - SAN/GFS2 or NFS v4 or RDBMS
  - Replicated - Block or RDBMS replication
- Managed
  - Red Hat Cluster Suite
  - Shared or Replicated storage

JBoss A-MQ :: Reliable



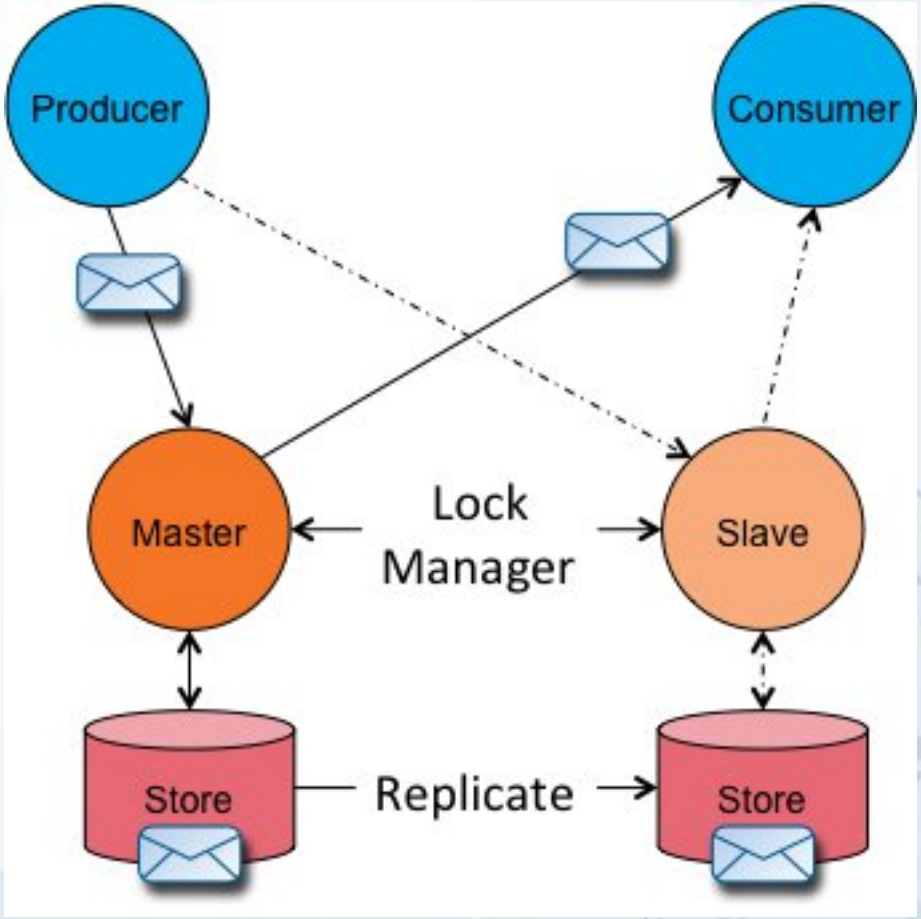
JBoss A-MQ :: Reliable

Master / Slave – Shared Storage

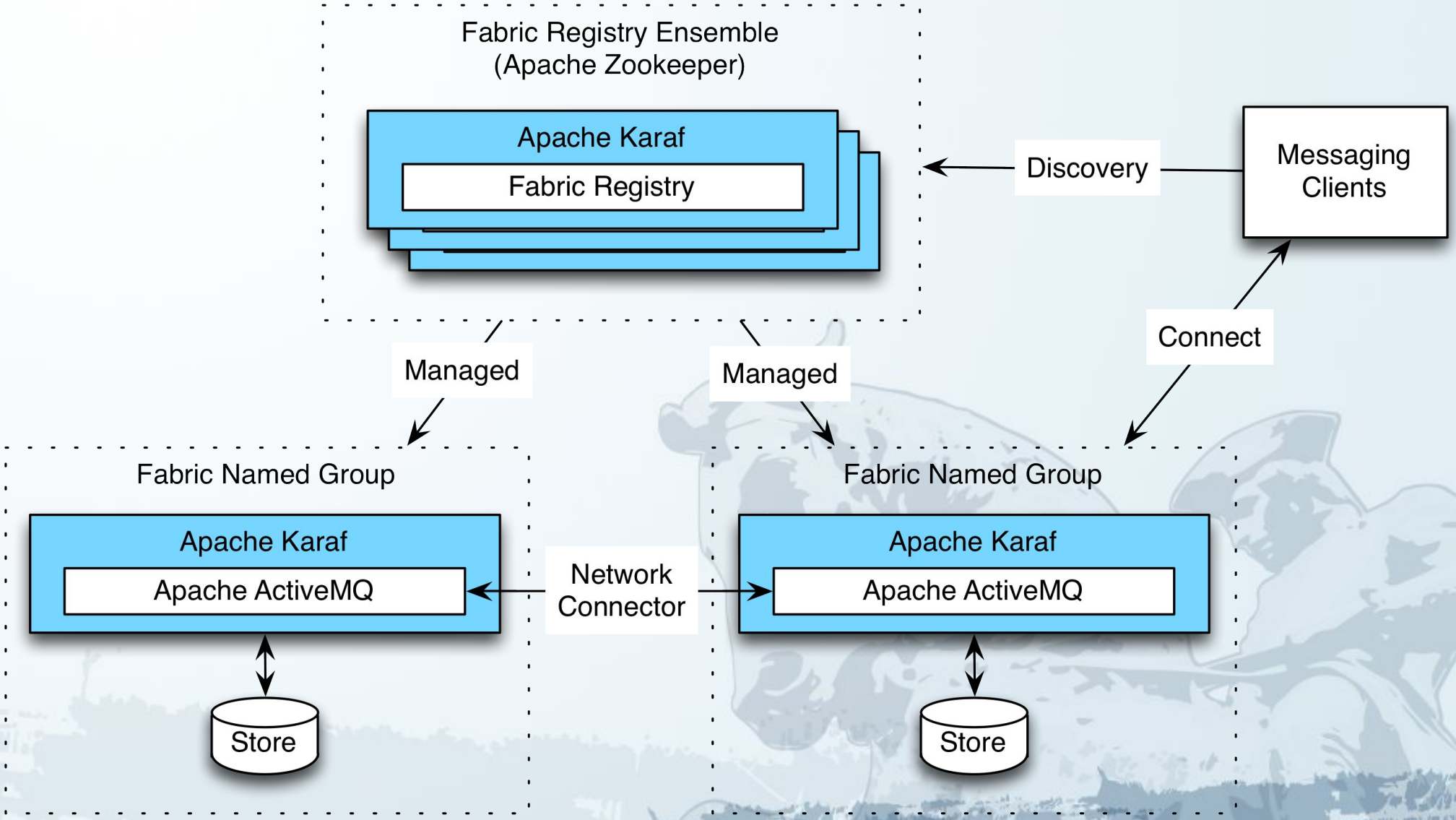


JBoss A-MQ :: Reliable

Master / Slave – Replicated



Fuse Fabric

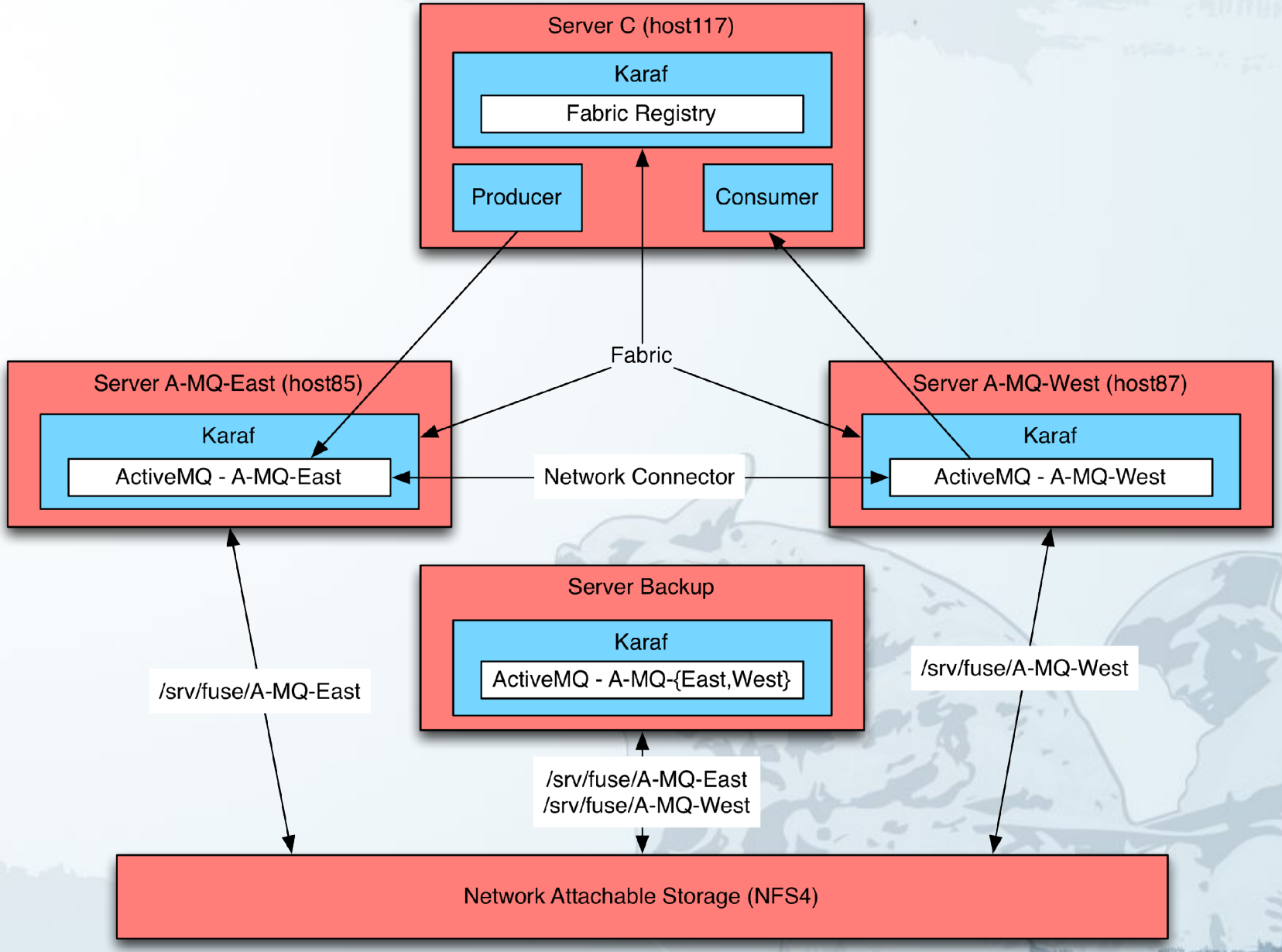

Fabric Registry Ensemble
(Apache Zookeeper)
Apache Karaf
Fabric Registry
Discovery
Messaging Clients
Connect
Managed
Managed
Fabric Named Group
Apache Karaf
Apache ActiveMQ
Store
Network Connector
Fabric Named Group
Apache Karaf
Apache ActiveMQ
Store

What is Red Hat High Availability?

Open

Scalable

Reliable

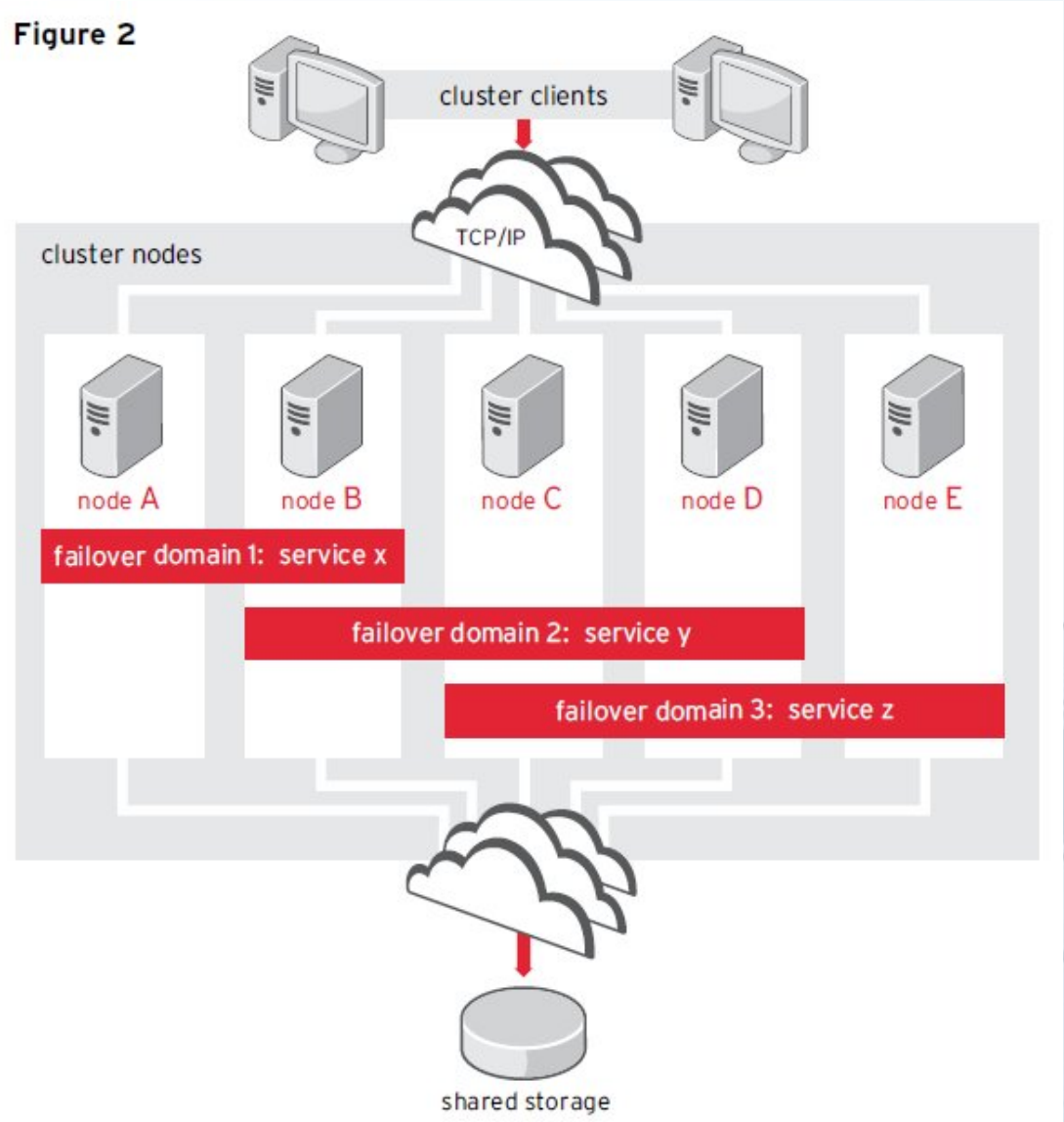

Server C (host117)
Karaf
Fabric Registry
Producer
Consumer
Fabric
Server A-MQ-East (host85)
Karaf
ActiveMQ - A-MQ-East
Network Connector
Server A-MQ-West (host87)
Karaf
ActiveMQ - A-MQ-West
Server Backup
Karaf
ActiveMQ - A-MQ-{East,West}
/srv/fuse/A-MQ-East
/srv/fuse/A-MQ-West
/srv/fuse/A-MQ-East
/srv/fuse/A-MQ-West
Network Attachable Storage (NFS4)

Key Benefits

- Fault tolerant design which utilizes N+1 versus 2X resources
- Administrators can scale to N+2 or N+3 architecture
- Failover logic is stored in a technology that manages other high availability resources

Overview

Figure 2



Concepts

- Nodes: Separate operating system instances in the cluster
- Resources: IP address, process, storage mount
- Failover Domains: Groups of nodes unto which Service Groups can be assigned
- Fence Devices: Integrated Lights Out (ILO), Dell Remote Access Card (DRAC), IPMI

High Availability Components

- Cluster Manager: Ties it all together, calculates quorum, communicates with other cluster components
- Resources: Resource Manager controls starting/stopping of processes, storage mounts, IP addresses, etc
- Fencing: The act of ensuring that broken nodes are removed from the cluster.
- Conga Web Interface

Title: Resilient Enterprise Messaging

Presenters: Scott Cranton & Scott McCarty

Links

- Cluster Knowledge Base Articles

  https://access.redhat.com/knowledge/articles/47987

- Best Practices

  https://access.redhat.com/knowledge/articles/40051

- Architecture Review Process

  https://access.redhat.com/knowledge/articles/53347

- Stretch Clustering