APACHECON
NORTH AMERICA

MESSAGING FOR THE CLOUD
with

ActiveMQ and Karaf

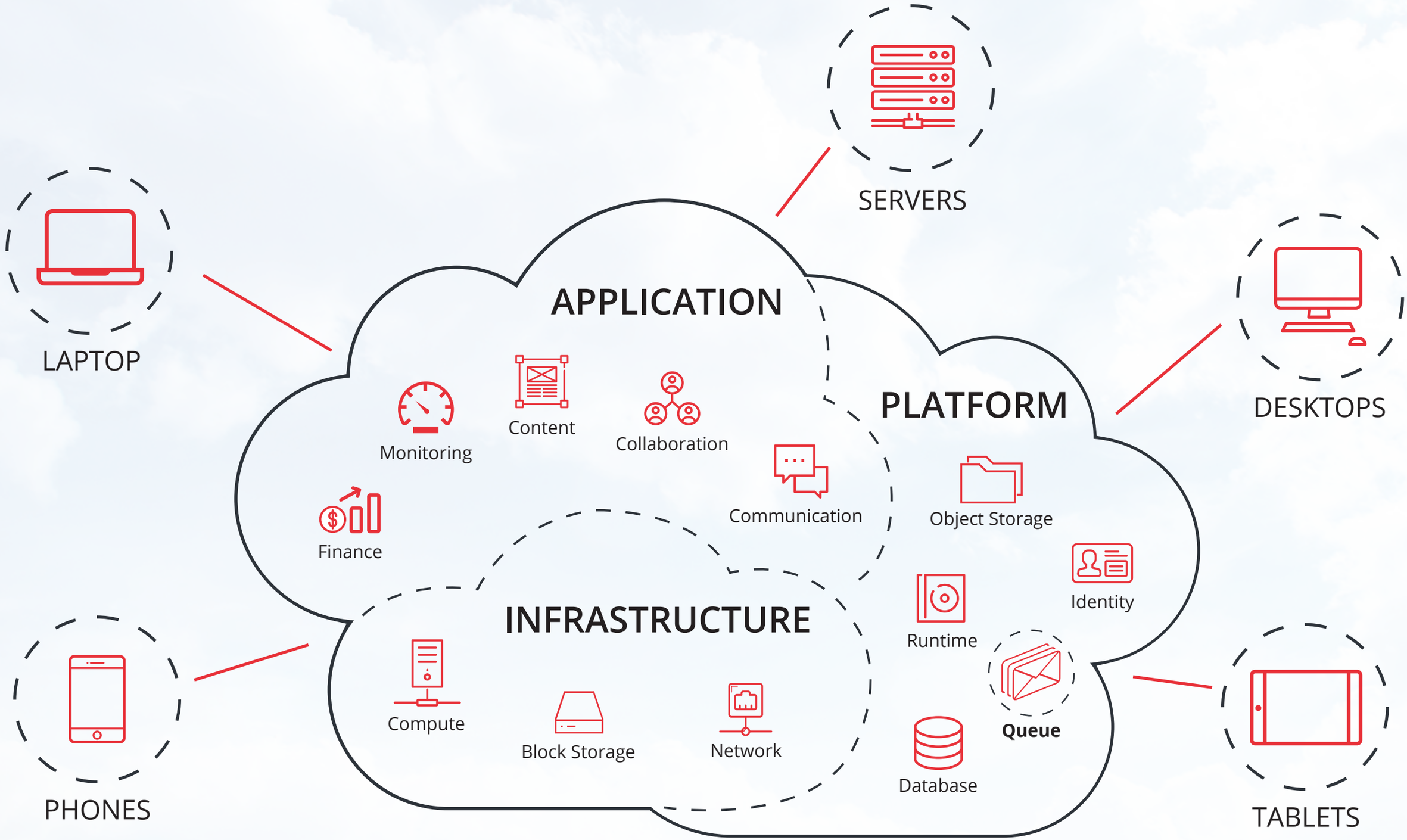
Hadrian Zbarcea & Jamie Goodyear

Cloud Computing

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. "

The NIST Definition of Cloud Computing

Cloud Computing Layers



Characteristics

http://dx.doi.org/10.6028/NIST.SP.800-145

On-demand self-service

A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity

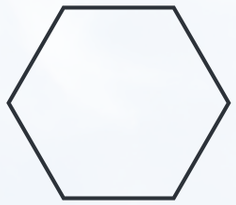
Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service

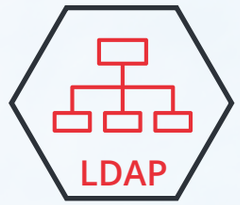
Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

LEGEND

PRODUCER



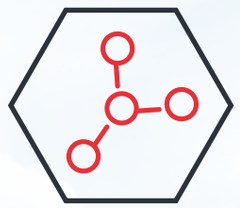
SERVICE



IDENTITY SERVICE

CONSUMER

CONTAINERIZED SERVICE



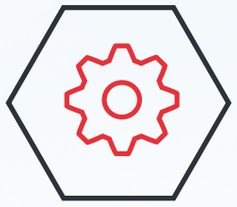
GRAPH DATABASE

MESSAGE

BROKER SERVICE

DATABASE

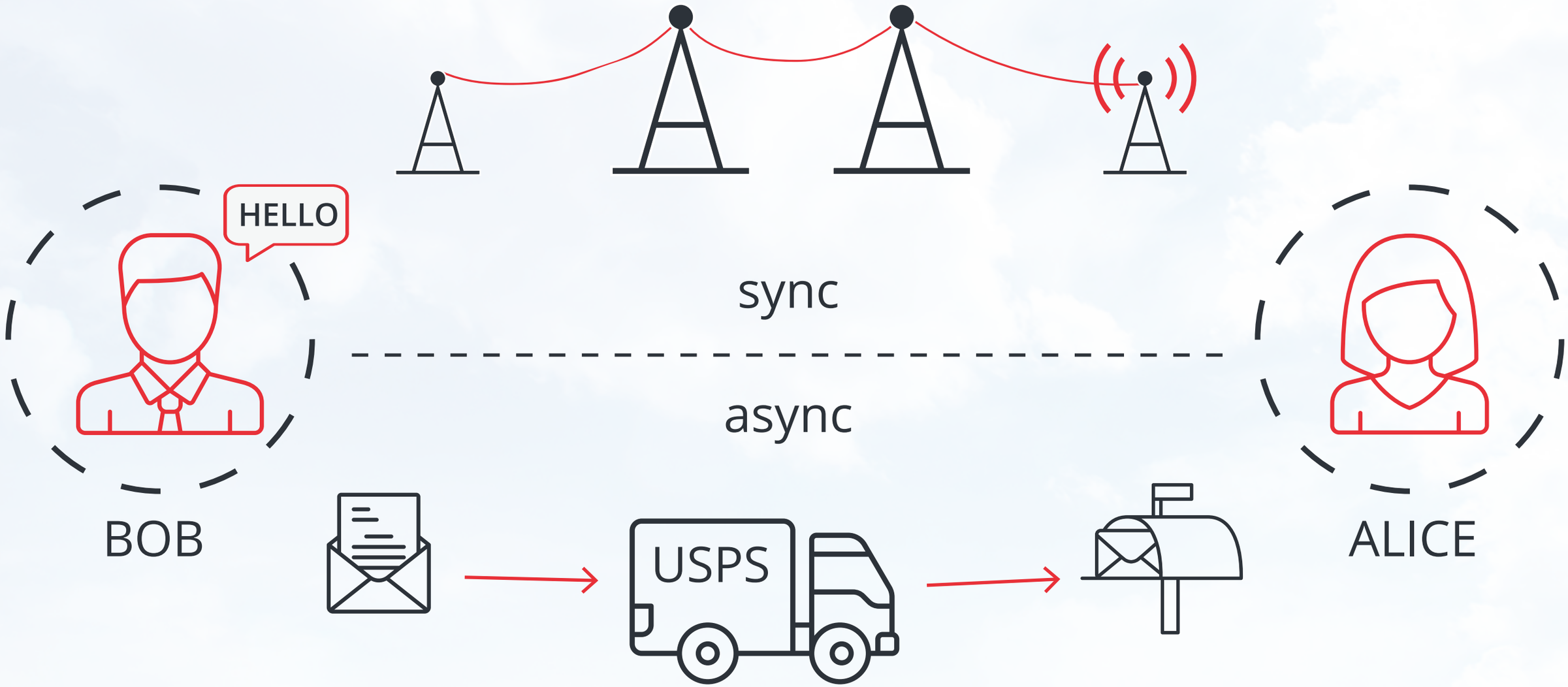
QUEUE



WORKER SERVICE

USER INTERFACE

Messaging 101

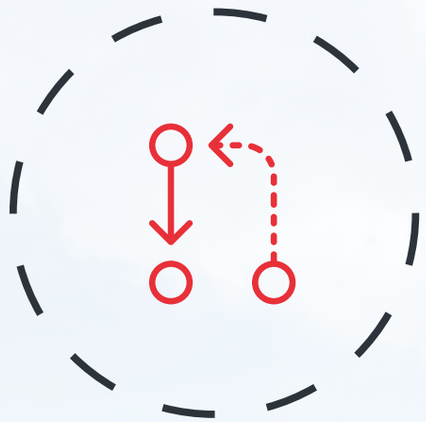


Infrastructure - QoS

- **Addressability** (address management)
- **Confidentiality** (authentication/authorization)
- **Guaranteed delivery**
- **Reliability**
- **Availability** (scalability)
- **Metrics** (bills and stamps)

Infrastructure - Technology

Content Type



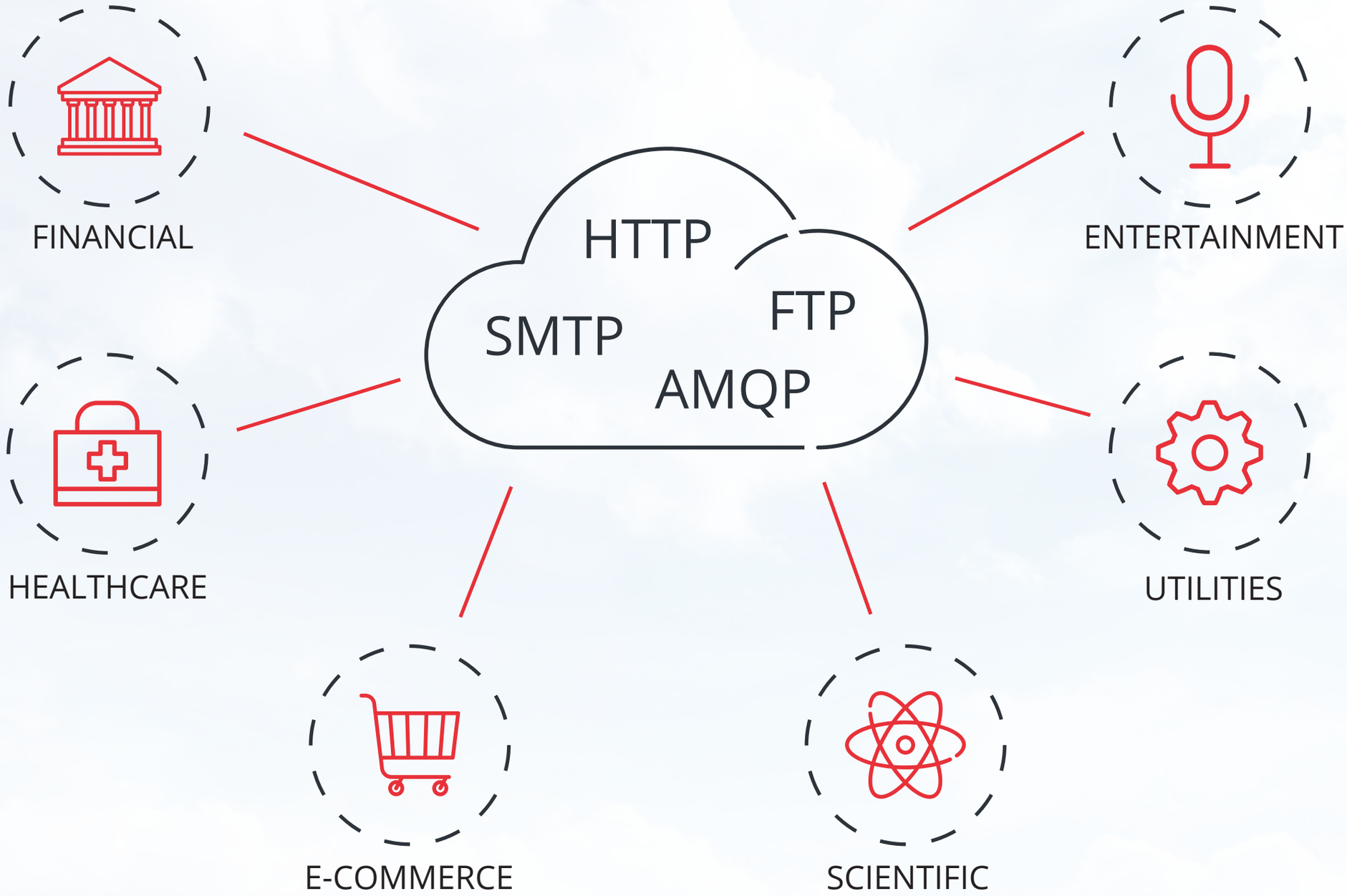
Protocols

Policies

Governance

Messaging Supply Chain

The Digital World



Apache ActiveMQ

ActiveMQ - Features

- **Based on Open Standards**
  JMS 1.1, AMQP, HTTP, XML...
- **Extensible, Pluggable Architecture**
  Configuration
  Discovery
  Transports
  Persistence
  Security
- Guaranteed delivery
- Availability (topologies)
- Resiliency (failover)
- Message Processing (Apache Camel)
- Streaming
- Governance (management, monitoring)

ActiveMQ - Client Bindings

ActiveMQ - Competitors

ActiveMQ

OPEN SOURCE

RabbitMQ

Qpid

kafka

ØMQ

COMMERCIAL

MSMQ

WebSphere MQ

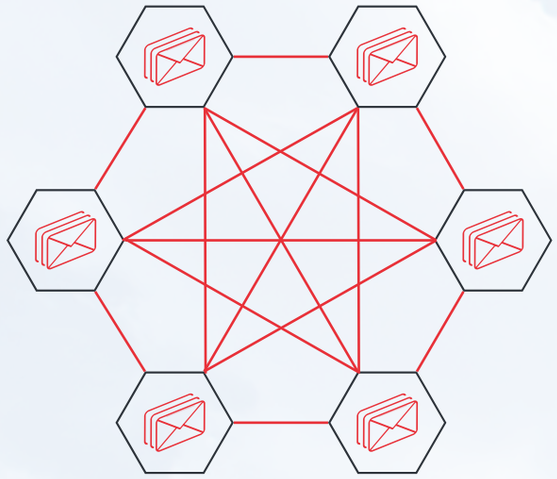
CLOUD SERVICES

SQS

rackspace

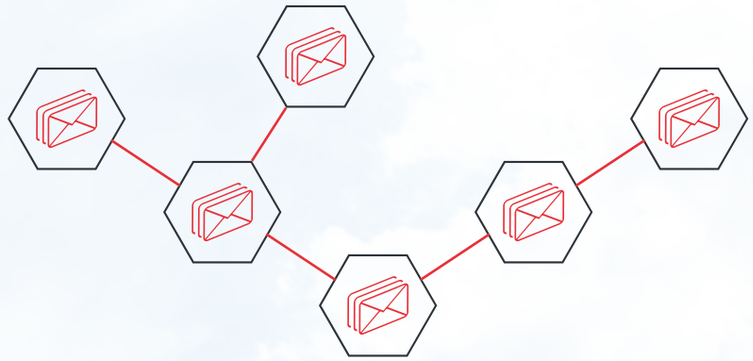
IronMQ

ActiveMQ - Topologies

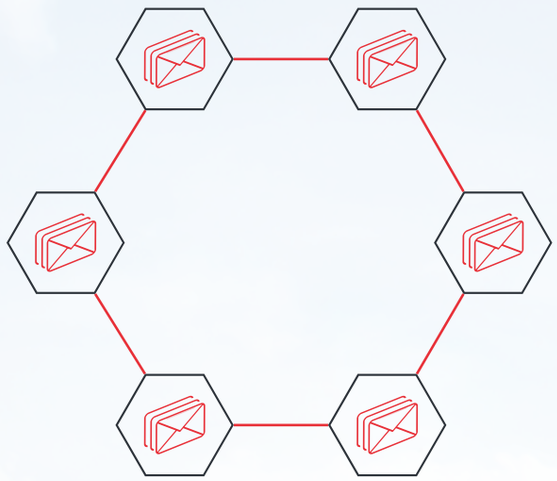


FULLY CONNECTED
HYPER CUBE

STAR
HUB & SPOKE



TREE



RING

LINE
PIPE LINE

MESH
NETWORK OF BROKERS

ActiveMQ - Reliability

ActiveMQ - Deployment

- **Standalone**
- **Embedded**
  your own app, or...

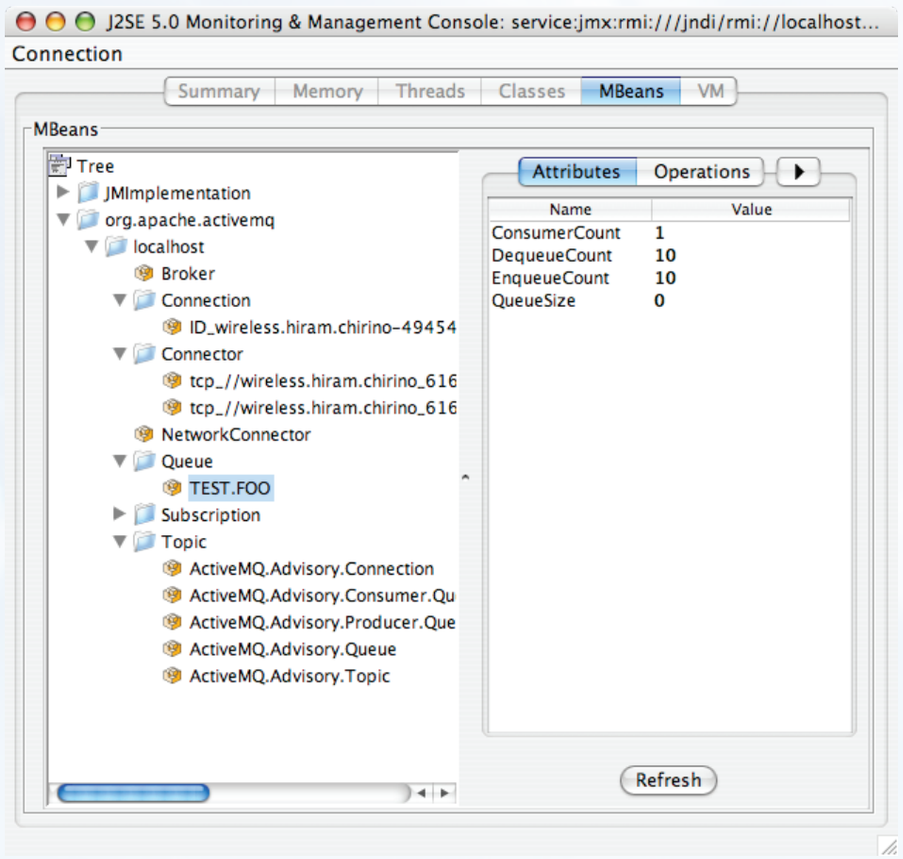
APACHE TOMCAT

karaf

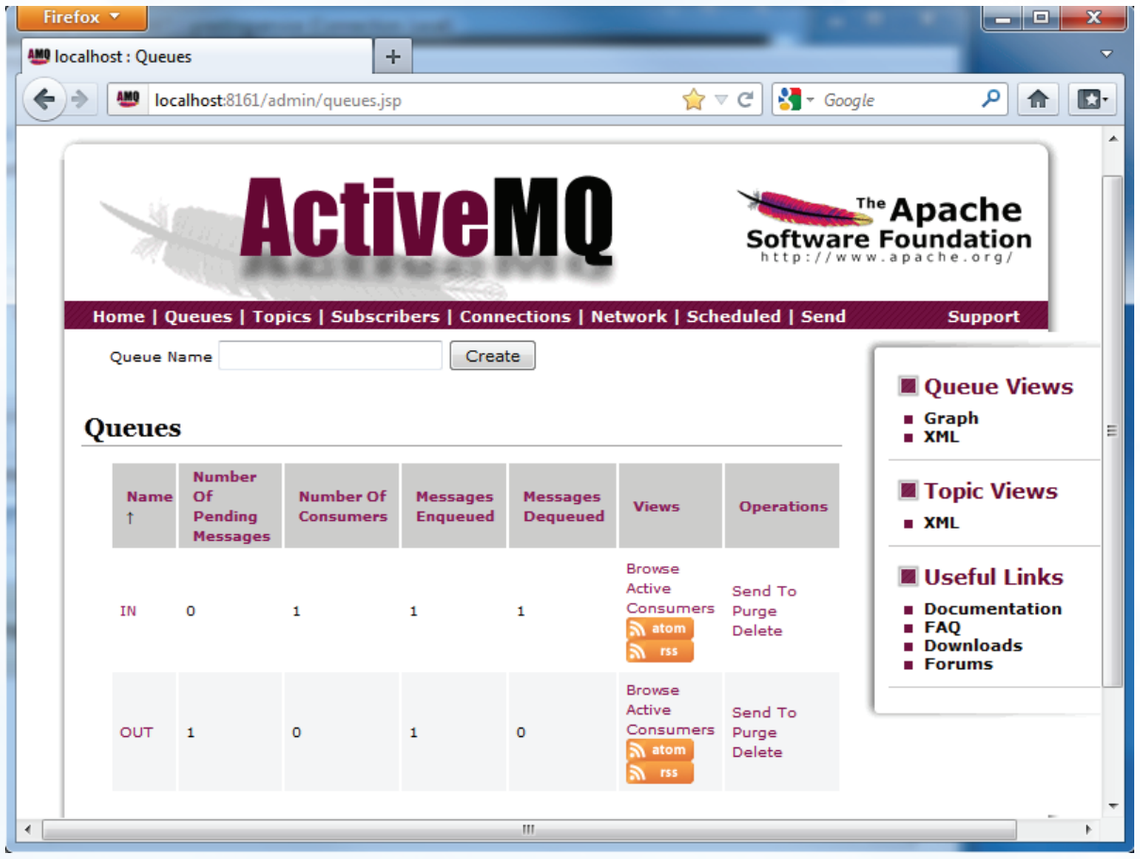
docker

ActiveMQ - Monitoring

JMX



Web Console



jolokia JMX on Capsaicin

karaf

elastic

Apache Karaf

Karaf - Features

- **Hot deployment:** simply drop a file in the deploy directory
- **Complete Console:** complete Unix-like console for managing the container.
- **Dynamic Configuration:** extensible set of commands focused on managing configuration.
- **Advanced Logging System:** supports all the popular logging frameworks (slf4j, log4j, etc)
- **Provisioning:** supports a large set of URLs where you can install your applications, plus 'features' and 'profiles"
- **Management:** enterprise-ready container, providing management metrics and operations via JMX.
- **Remote:** embeds an SSHd server for remote operations.
- **Security:** complete security framework (based on JAAS) and RBAC (Role-Based Access Control) mechanism
- **Instances:** multiple instances managed directly from a main instance (root).
  OSGi frameworks: Apache Felix Framework or Equinox

Karaf Decanter - Overview

Default implementation based on ELK

Collectors

Responsible for harvesting monitoring data.

Two kinds:

Event Drive Collectors

Polled Collectors

Appenders

Receive data from the collectors

Responsible to store data in a given backend

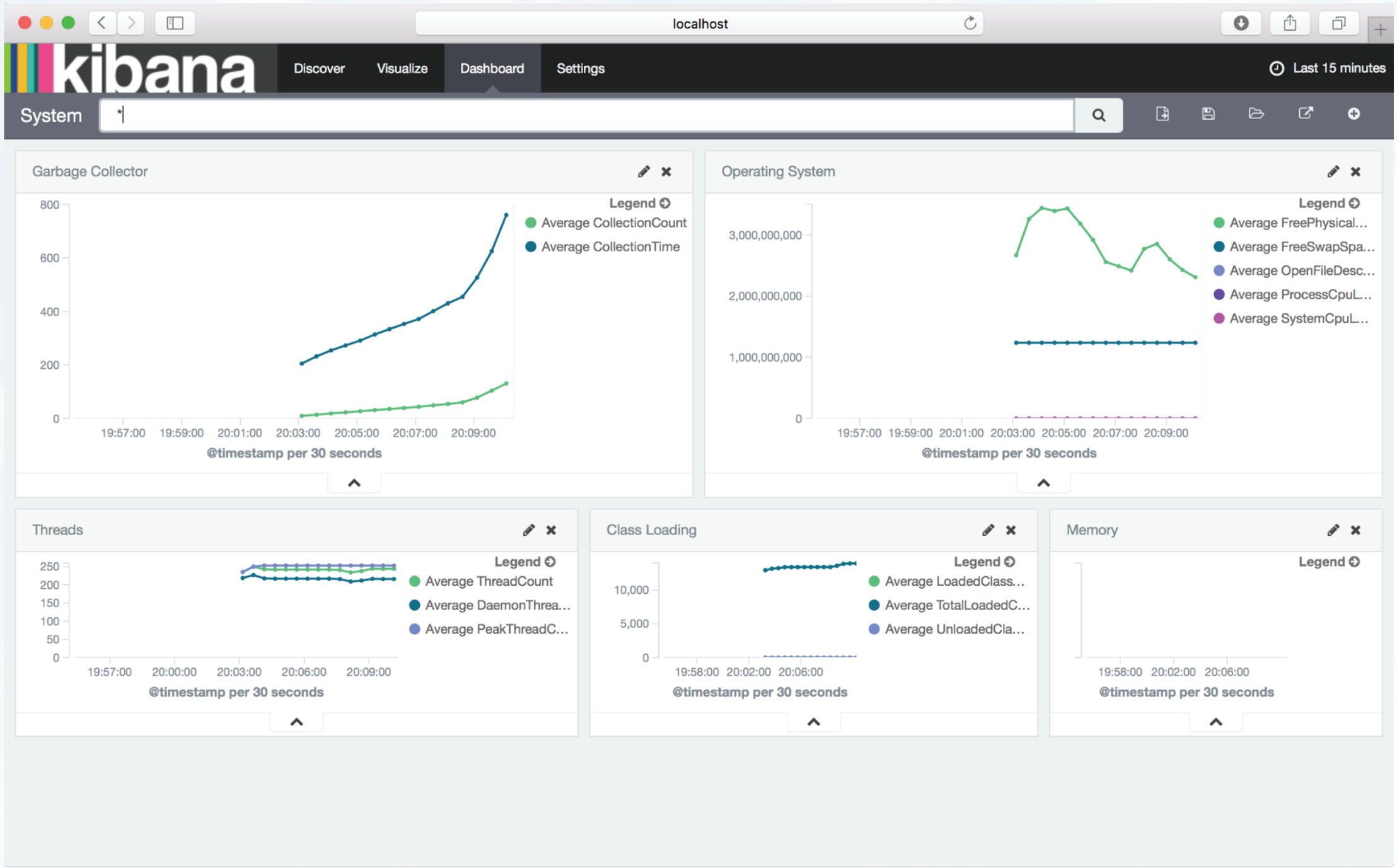
Service Level Agreements

Special kind of Appender

Receives harvested data and performs checks upon it.

if check fails, alert event is created and sent to alerters.

Karaf Decanter - Monitoring

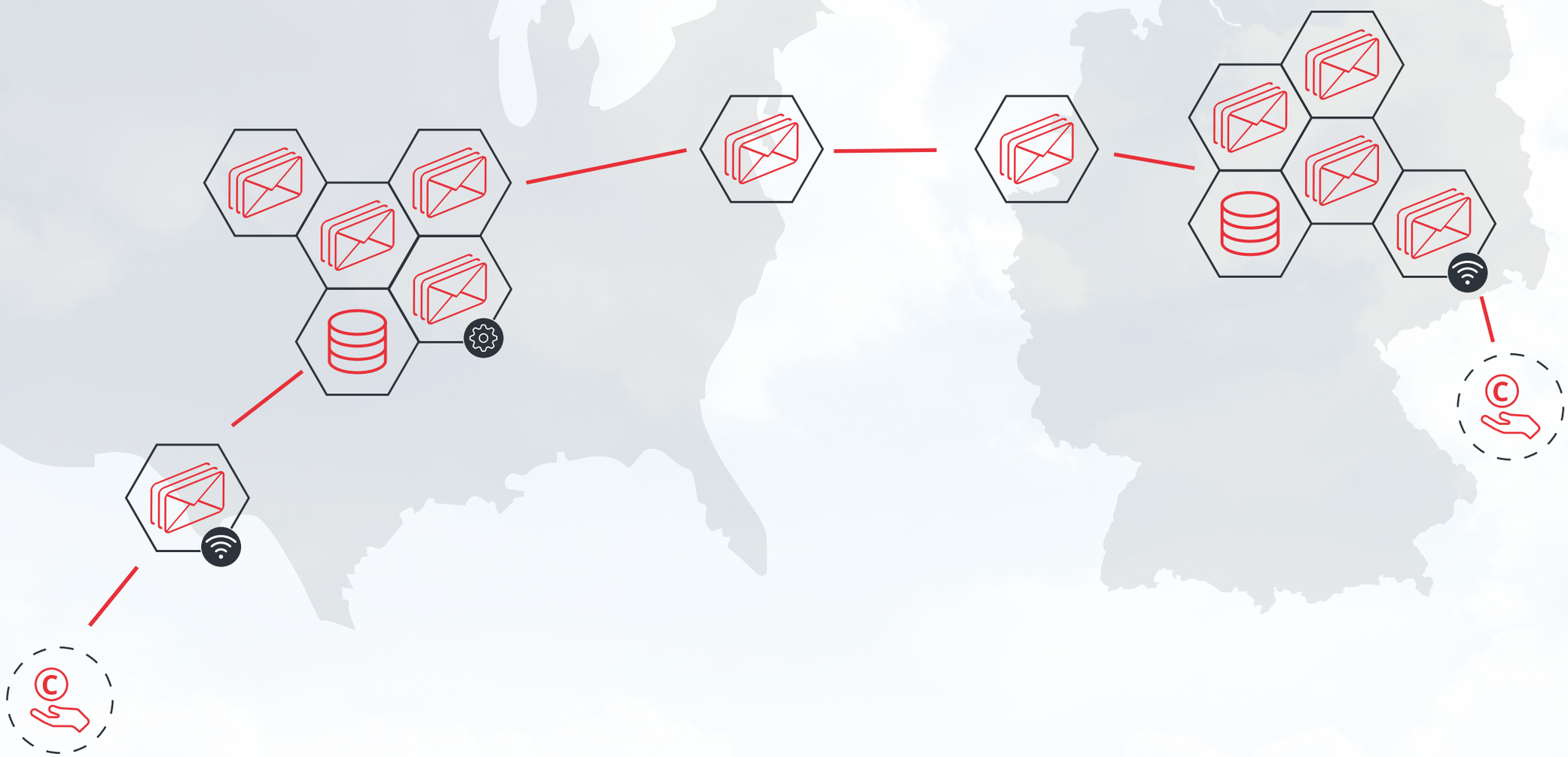


DEMO

SILKMQ

SilkMQ - What's in a name?

SilkMQ - Topology



SilkMQ - Broker Monitoring

SilkMQ is smooth, shouldn't monitoring be too?

- **We want Decanter Monitoring to be elastic like SilkMQ**
- **Currently Decanter is focussed on individual services**

  Needs more work...
- **We want to view message infrastructure at a higher level**

  How many messages enqueued and dequeued for a project?

  Status of pending messages

  SLA: What is the throughput and latency for queue XYZ (in a project)?
- **Tenants are not aware of the physical deployment infrastructure**

  ...nor should they be

SilkMQ - Broker Management

- **Use metrics and policies to scale elastically**
- **Use a service orchestrator**

apache brooklyn

RANCHER

SilkMQ - Broker Discovery

"ActiveMQ uses an abstraction called a Discovery Agent to detect remote services such as remote brokers. We can use discovery for JMS clients to auto-detect a Message Broker to connect to, or to provide Networks of Brokers"

Multi-step resolution process

- Host resolution DNS
- Pluggable resolution logic

  Multicast, zeroconf, SilkMQ

SilkMQ - Multi-tenancy

Model based on users/organizations/teams as commonly seen with Cloud Services

SilkMQ - Demo

HANDS ON

APACHECON
NORTH AMERICA

THANKS!

apifocal

savoir
TECHNOLOGIES

Hadrian Zbarcea

hadrian@apache.org

Jamie Goodyear

jgoodyear@apache.org

Team:

Alex | Cipi | Andrei | Raul | Cristi