

ActiveMQ Artemis

Messaging  
for the Enterprise and IoT

Jeff Mesnil - Red Hat

# Jeff Mesnil

- Software Engineer at Red Hat
- Core developer on WildFly Application Server, lead for its messaging component

<http://jmesnil.net/>

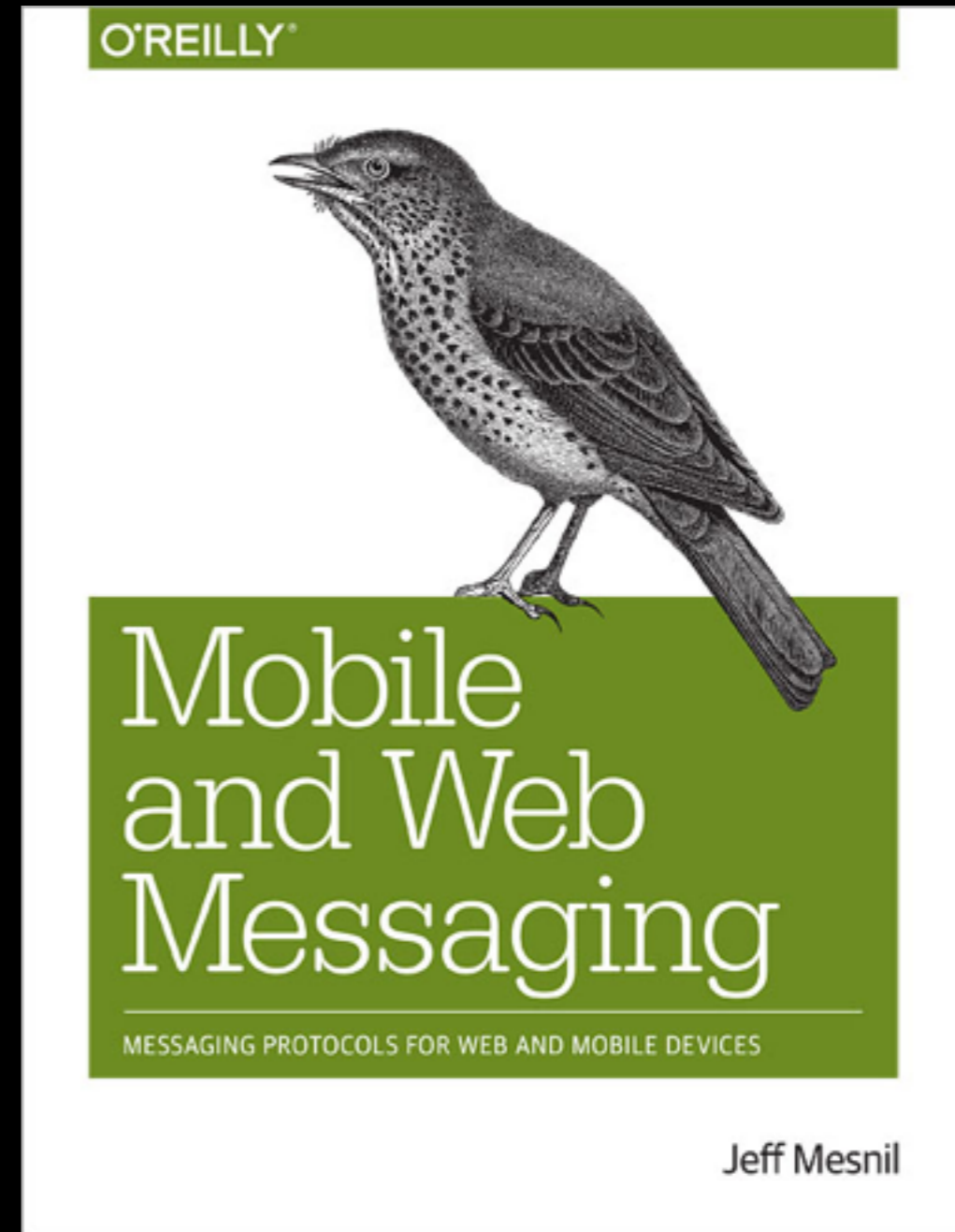
jmesnil@gmail.com

@jmesnil

# Mobile & Web Messaging

Messaging Protocols for Web and Mobile Devices

- Published by O'Reilly Media in 2014
- MQTT & STOMP
- iOS & Web Browsers



# Summary

- Apache ActiveMQ
  - Subproject
- ActiveMQ Artemis
- Messaging Models & Protocols
- Demo
- Q & A

# Apache ActiveMQ

- <http://activemq.apache.org>
- Top level project from Apache Software Foundation
- ActiveMQ
- CMS
- NMS <http://activemq.apache.org/>
- Apollo
- Artemis

# Apache ActiveMQ

- "Most popular and powerful open source messaging and Integration Patterns server"
- Based on OpenWire messaging protocol
- Supports many messaging protocols (STOMP, AMQP, MQTT)
- Provides JMS 1.1 API
- Pluggable transports (in-vm, TCP, SSL, NIO, UDP, multicast, JGroups, JXTA)
- Advanced features (message groups, virtual destinations, etc.)
- Persistence using JDBC or high-performance journal
- Clustering, client-server, peer based communication

# ActiveMQ CMS & MMS

- CMS - C++ Messaging Service
  - Client library
  - STOMP & OpenWire protocols
- NMS - .Net Messaging API
  - ActiveMQ, STOMP, MSMQ, EMS, WCF, AMQP

# Apollo

- "Next generation" of messaging
- Radically different threading and message dispatching architecture
  - Reactor-based thread model (HawtDispatch)
  - Protocol agnostic
  - Written in Scala
- Supports many messaging protocols (OpenWire, STOMP, AMQP, MQTT)



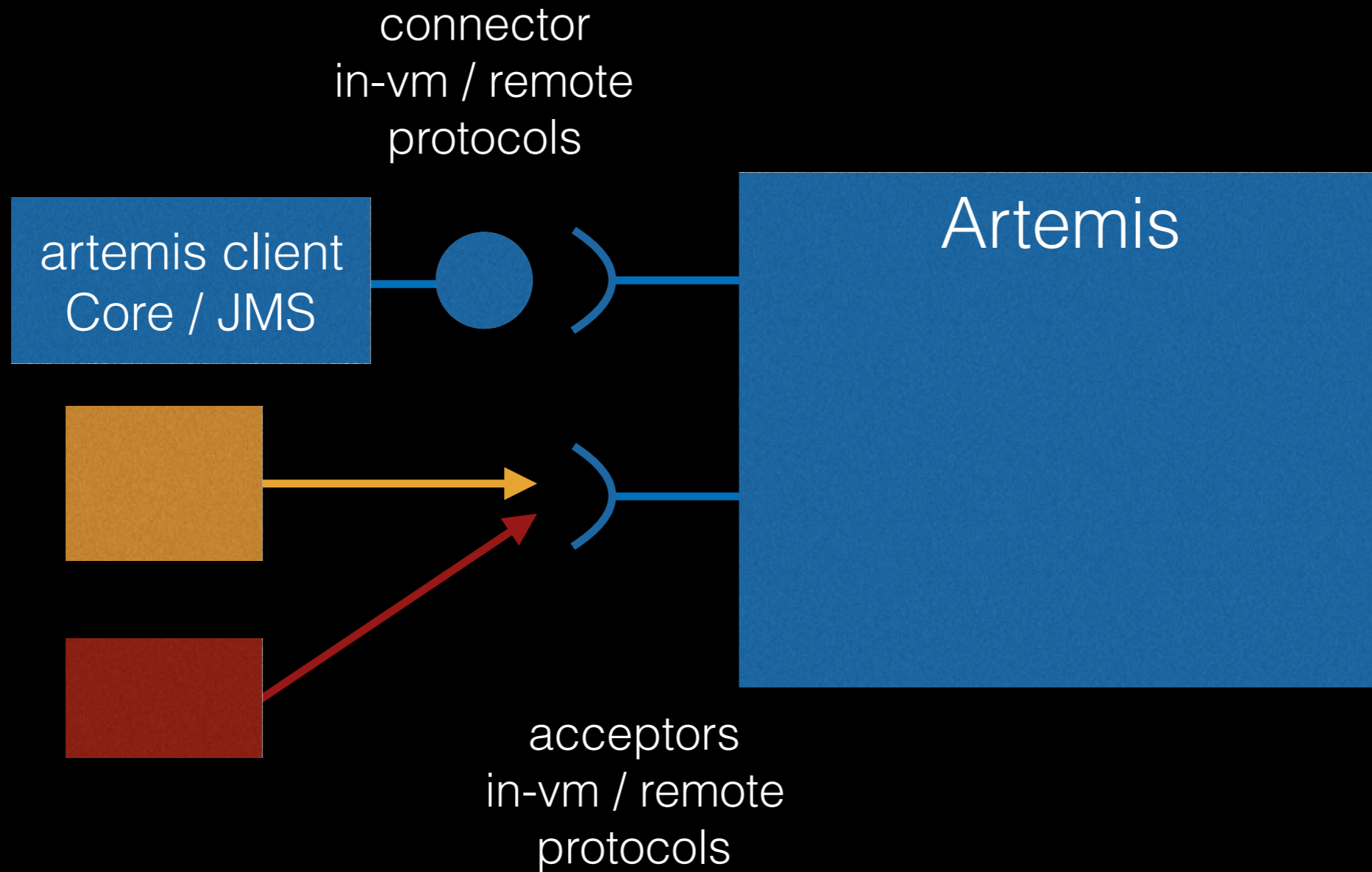
# Artemis

- New Subproject (2015)
- Code donation from Red Hat based on HornetQ
- Non-blocking architecture (Netty / NIO)
- High-performance NIO Journal
- Supports many messaging protocols (AMQP, HornetQ Core, STOMP)
- Provides JMS 2.0 API

# Artemis

- 1.1.0 released in September
  - OpenWire and MQTT protocols
  - CLI
- Integrated in WildFly 10

# Artemis Architecture



# Journal

- Persistence store
  - Persistent messages, Bindings
  - Large messages
- Fast (NIO)
- Faster (libAIO on Linux)
- No JDBC (yet)

# High Availability

- Active / Passive
- Shared store
- Replication (UDP, JGroups)
- Colocated Backup
- Scale Down

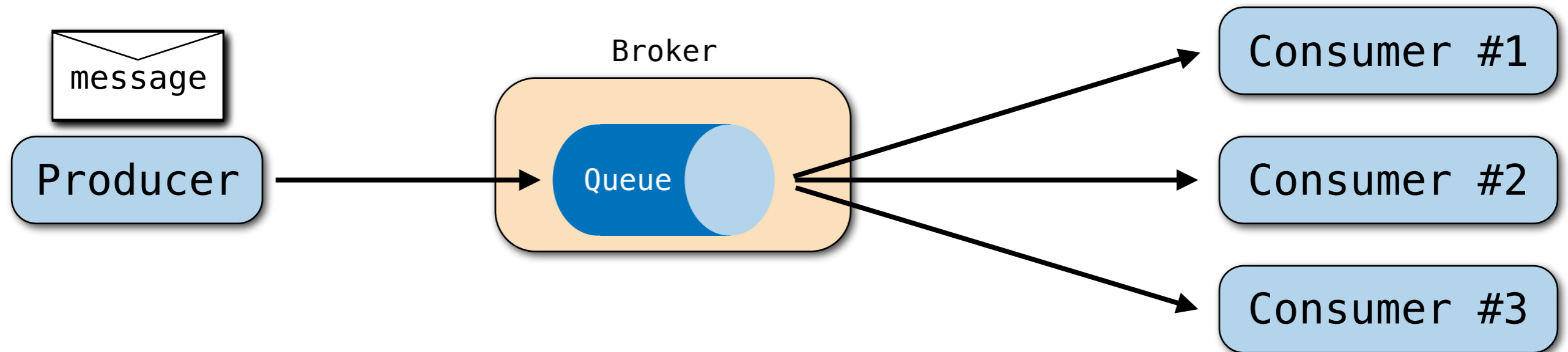
# Demo

Create and Start Artemis Broker

# Messaging Models

- Point-to-Point
- Publish/Subscribe

# Point-to-Point Model

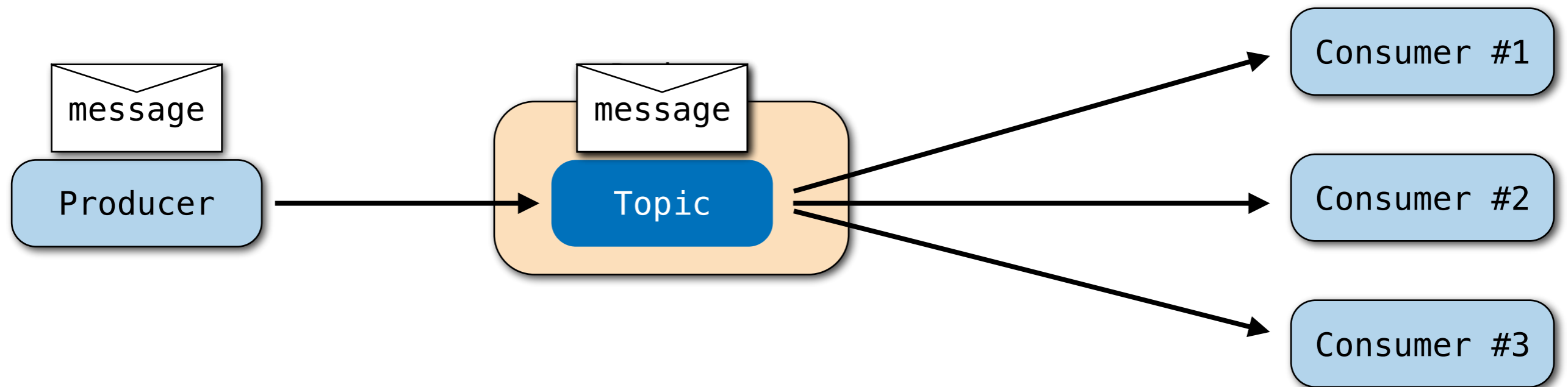




# Point-to-Point Model

- Queues
- One-to-one model
  - One message produced
  - One message consumed

# Publish/Subscribe Model



# Publish/Subscribe Model

- Topics
- One-to-many model
  - One message produced
  - Many messages consumed
- Durable subscription

# Messaging Protocols

- Enterprise
  - OpenWire
  - AMQP
  - Core / HornetQ
- First-mile
  - STOMP
  - MQTT

# OpenWire

- Used natively by ActiveMQ
- Binary Protocol
- Cross language (Java, C#, C++)
- Provides JMS 1.1 API
- <http://activemq.apache.org/openwire.html>

# AMQP

- AMQP 1.0
- Binary protocol
- “Enterprise” messaging
- Many Messaging Models
- Many client and broker implementations
- Provides JMS 1.1 API
- <http://www.amqp.org>

# Core / HornetQ

- Used by Artemis
- Binary protocol
- can provide JMS 2.0 API
- support legacy HornetQ protocol

# STOMP

- “Integration” Messaging
- Text-based protocol
- No messaging models
- Many client and broker implementations
- <http://stomp.github.io>

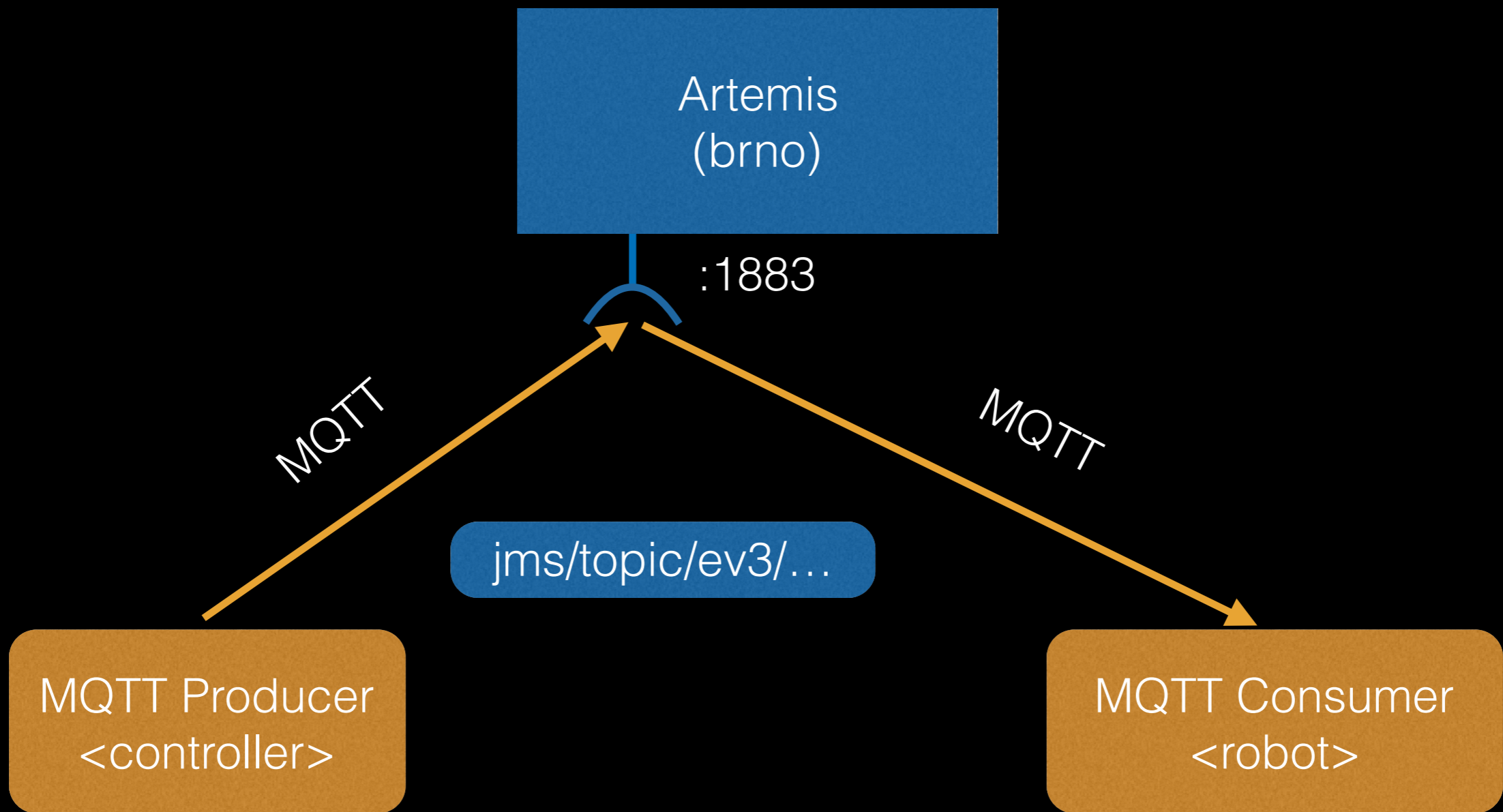


# MQTT

- Binary protocol
- “IoT” Messaging
- Publish/Subscribe
- Many client and broker implementations
- <http://mqtt.org>

# Demo

MQTT Publish and Subscribe



# Which Protocols?

- It depends
  - on your device/platform (/language)
  - on your needs (integration, performance, clustering)
  - on the messaging model it supports
- Often more than one
- Make sure that your broker supports the ones that you need

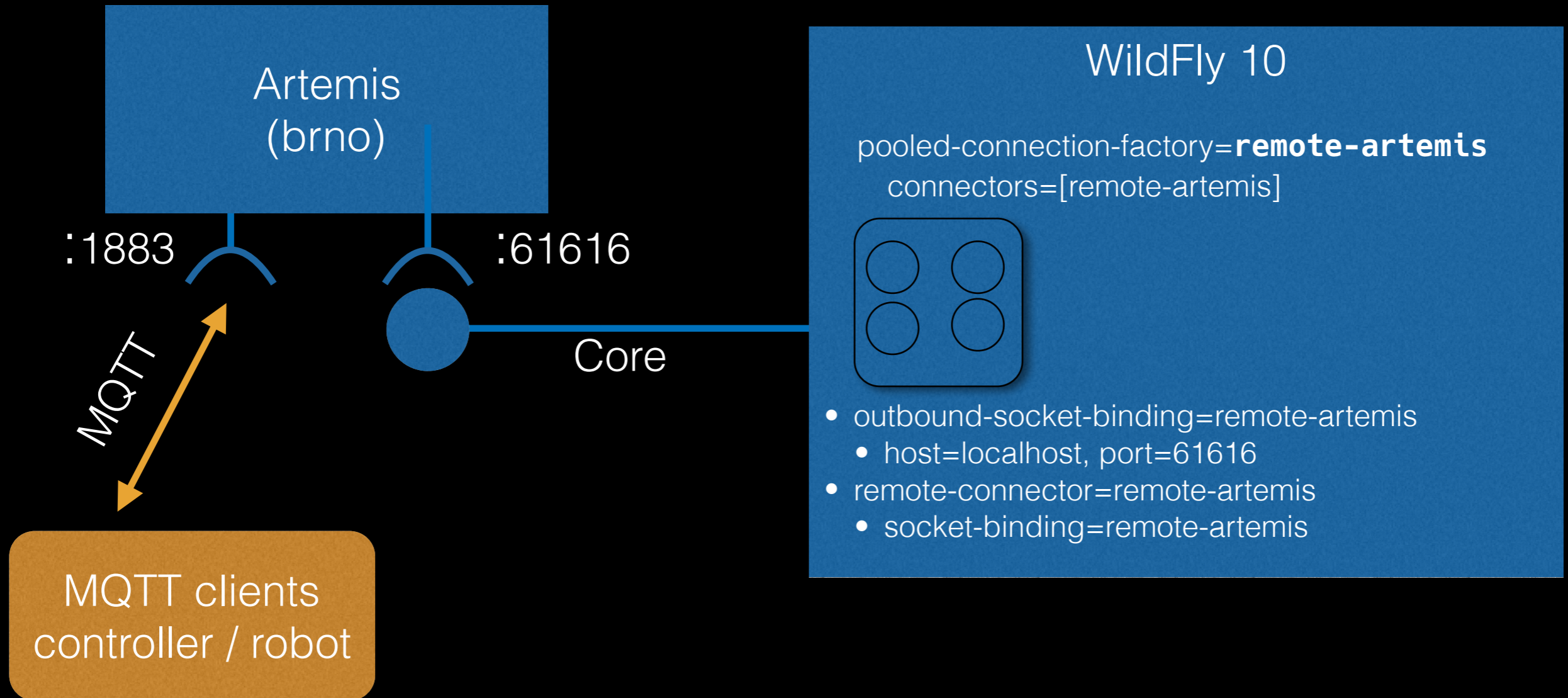
# WildFly

- Java Application Server
- Java EE7 compliant
- Upstream project for JBoss Enterprise Application Platform from Red Hat
- <http://wildfly.org>

# Artemis & WildFly

- Artemis is the broker used by WildFly 10 messaging subsystem
- JMS 2.0 / Java EE 7 compliant
- Migration from WFLY 9 (HornetQ) messaging subsystem
- Support for legacy resources & protocol

# Pooled Connection Factory

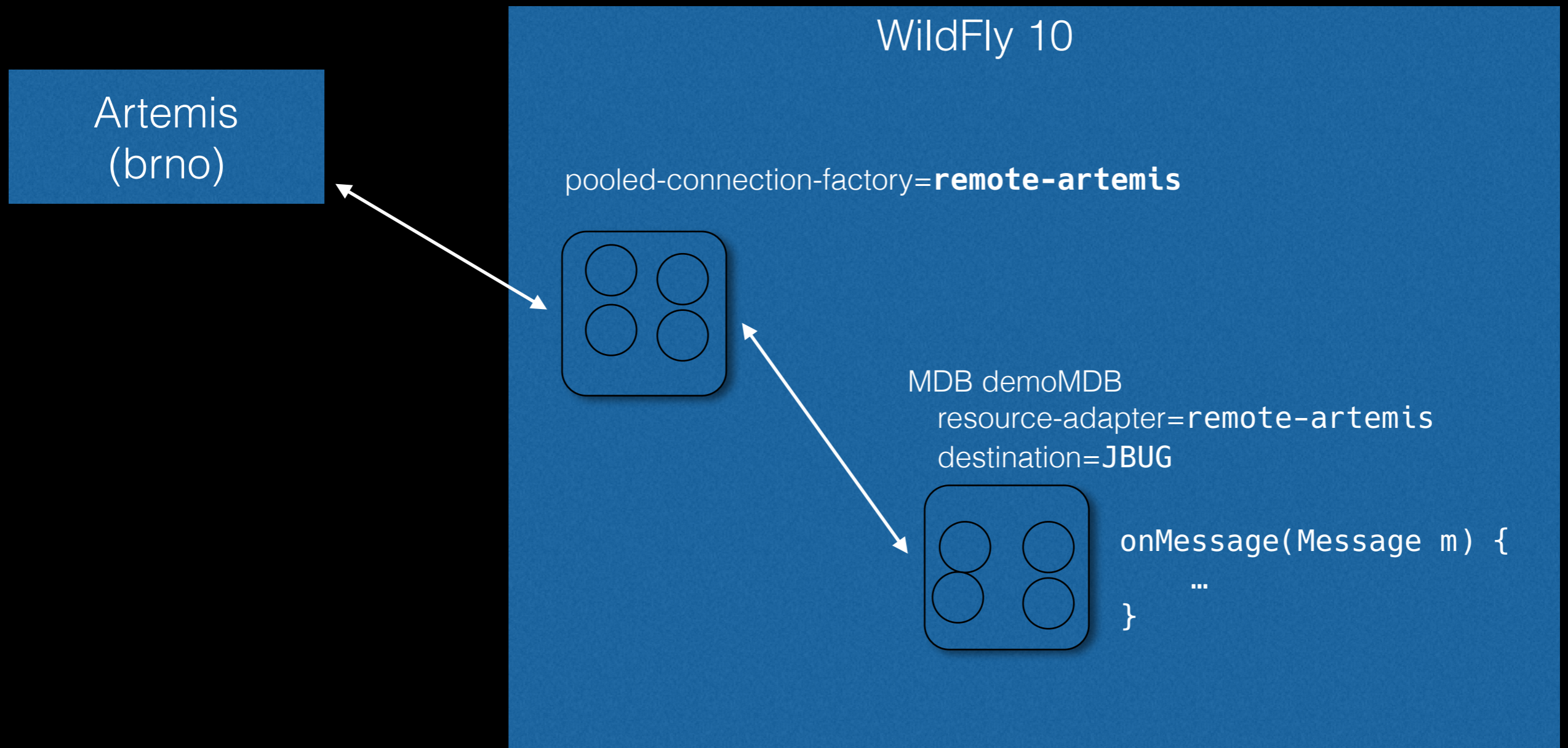


# Demo

Configure a Pooled Connection Factory



# Message-Driven Bean



# Demo

Consuming MQTT Messages from Message-Driven Bean

Q & A

# Thanks!

@jmesnil / [jmesnil@gmail.com](mailto:jmesnil@gmail.com)

#apache-activemq