

极客时间 Java 进阶训练营

第 26 课

分布式消息--其他 MQ 介绍与动手写 MQ



KimmKing

Apache Dubbo/ShardingSphere PMC

# 个人介绍

Apache Dubbo/ShardingSphere PMC

前某集团高级技术总监/阿里架构师/某银行北京研发中心负责人

阿里云 MVP、腾讯 TVP、TGO 会员

10 多年研发管理和架构经验

熟悉海量并发低延迟交易系统的设计实现

# 目录

1. RabbitMQ

2. RocketMQ

3. Pulsar

4. EIP/Camel/Spring Integration

5. 动手写 MQ \*

6. 总结回顾与作业实践



# 安装

## 1、直接安装

brew install rabbitmq //macos

apt/yum install rabbitmq-server //linux

choco install rabbitmq //windows

> rabbitmq-plugins enable rabbitmq\_management

## 2、docker 安装

docker pull rabbitmq:management # 注意不带后缀就不会有 web 控制台

docker run -itd --name rabbitmq-test -e RABBITMQ\_DEFAULT\_USER=admin -e RABBITMQ\_DEFAULT\_PASS=admin -p 15672:15672 -p 5672:5672 rabbitmq:management

docker exec -it rabbitmq-test /bin/bash

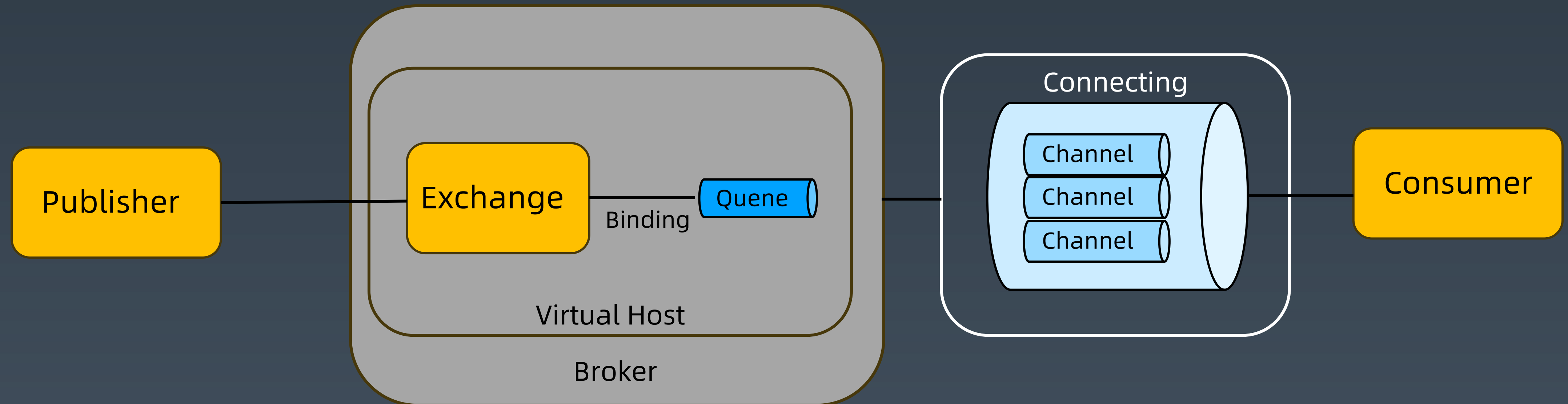
> rabbitmqctl list\_queues、rabbitmqctl status

> rabbitmqadmin declare queue name=kk01 -u admin -p admin

> rabbitmqadmin get queue=kk01 -u admin -p admin

# RabbitMQ 的核心概念

queue/exchange/routekey/binding



# spring-amqp 操作 RabbitMQ 演示

示例代码。

- 1、spring-amqp 封装好了 Template
- 2、rabbitmq-client 直接操作





# 安装

## 1、直接安装

从<http://rocketmq.apache.org/downloading/releases/>下载4.8.0，解压，即可。

```
nohup sh bin/mqnamesrv &
```

```
nohup sh bin/mqbroker -n localhost:9876 &
```

```
> export NAMESRV_ADDR=localhost:9876
```

```
> sh bin/tools.sh org.apache.rocketmq.example.quickstart.Producer
```

```
> sh bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer
```

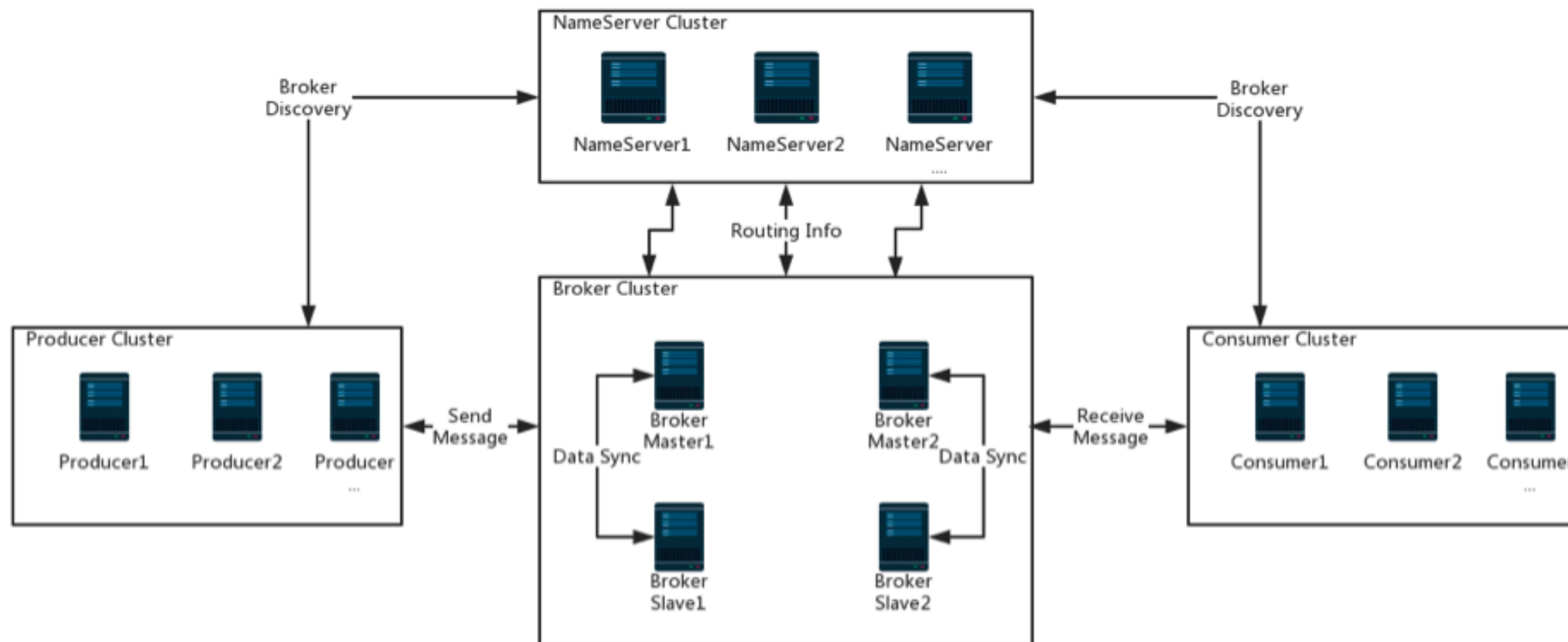
参见：<http://rocketmq.apache.org/docs/quick-start/>

## 2、docker安装

挺麻烦的，参见<https://github.com/apache/rocketmq-docker>

# RocketMQ 介绍

总体结构。



# RocketMQ 介绍

## 1、与 Kafka 的关系

2、区别：作为 Kafka 的重新实现版，没太大本质区别（百事~可口）

1) 纯 Java 开发，用不用zk

2) 支持延迟投递，消息追溯 ==> 说实话，意义不太大

3) 多个队列使用一个日志文件，所以不存在 kafka 过多 topic 问题 ==> 这个仁者见仁

参见 <http://rocketmq.apache.org/docs/motivation/>

# RocketMQ 演示

示例代码。



# 安装

## 1、下载安装

通过 <http://pulsar.apache.org/zh-CN/download/> 下载2.7.0版本

解压压缩包，即可。详细文档可以参见：<http://pulsar.apache.org/docs/zh-CN/>

```
> bin/pulsar standalone
```

```
> bin/pulsar-client consume topic1 -s "first-subscription"
```

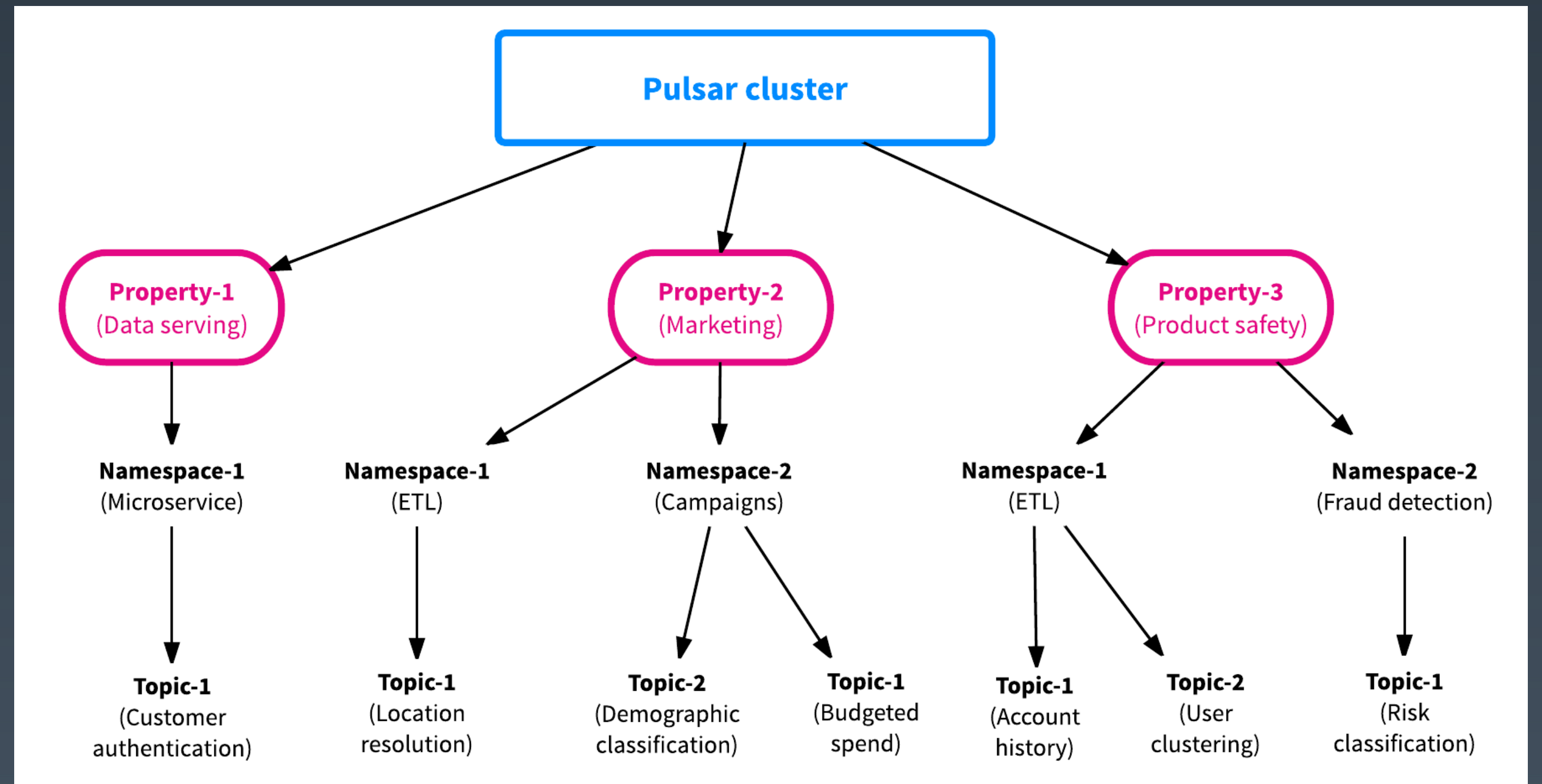
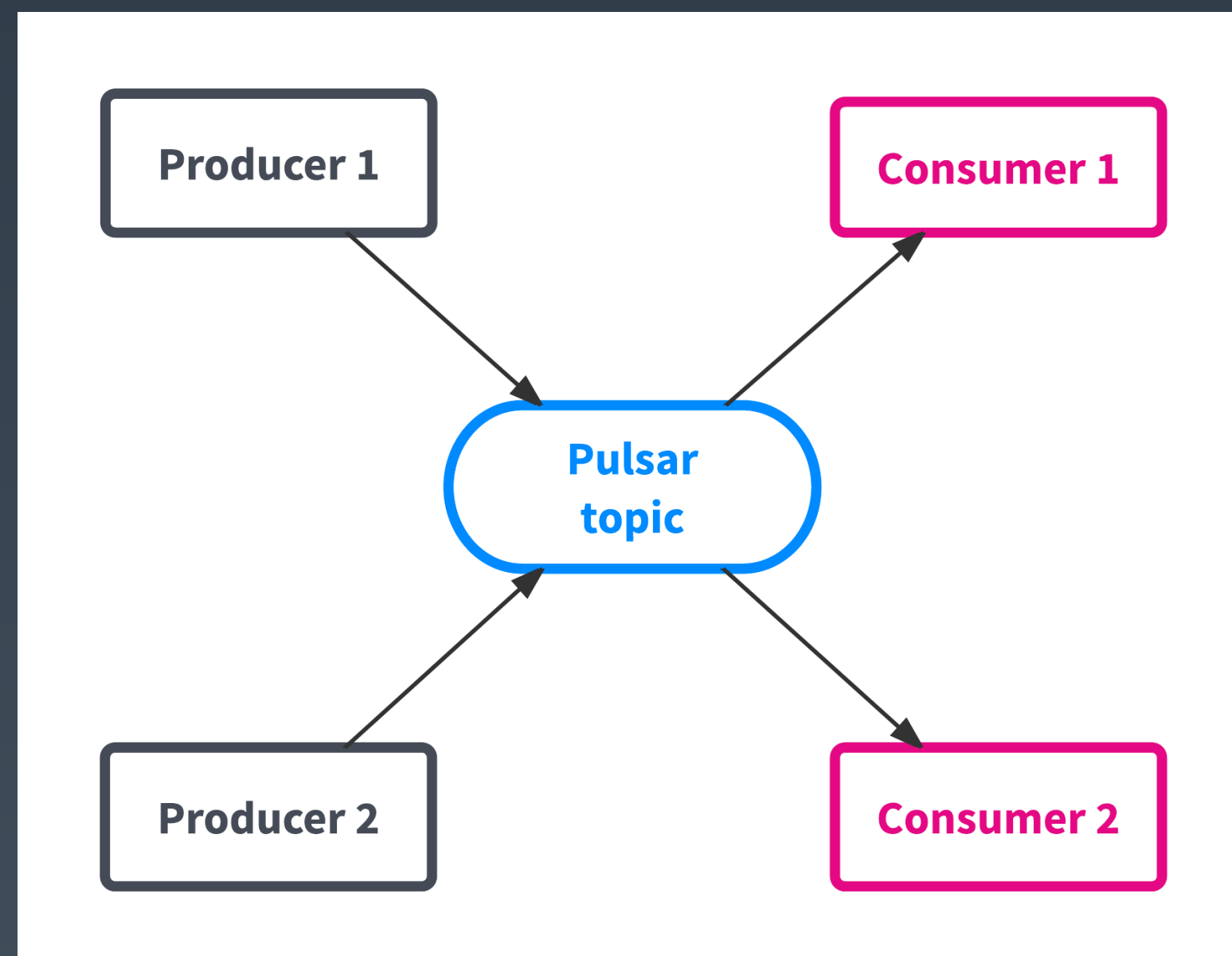
```
> bin/pulsar-client produce topic1 --messages "hello-pulsar"
```

## 2、Docker安装运行

参考 <http://pulsar.apache.org/docs/zh-CN/standalone-docker/>

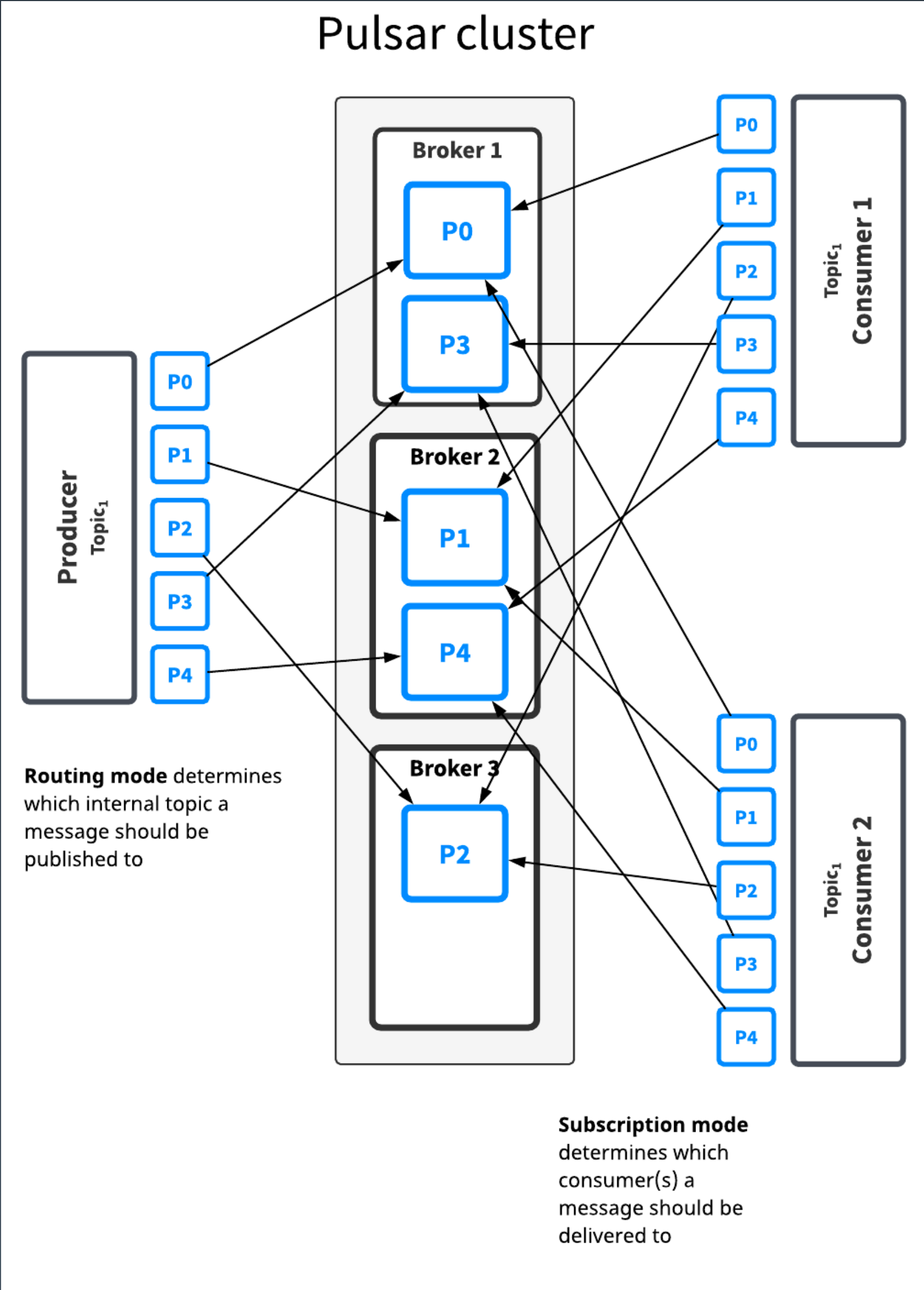
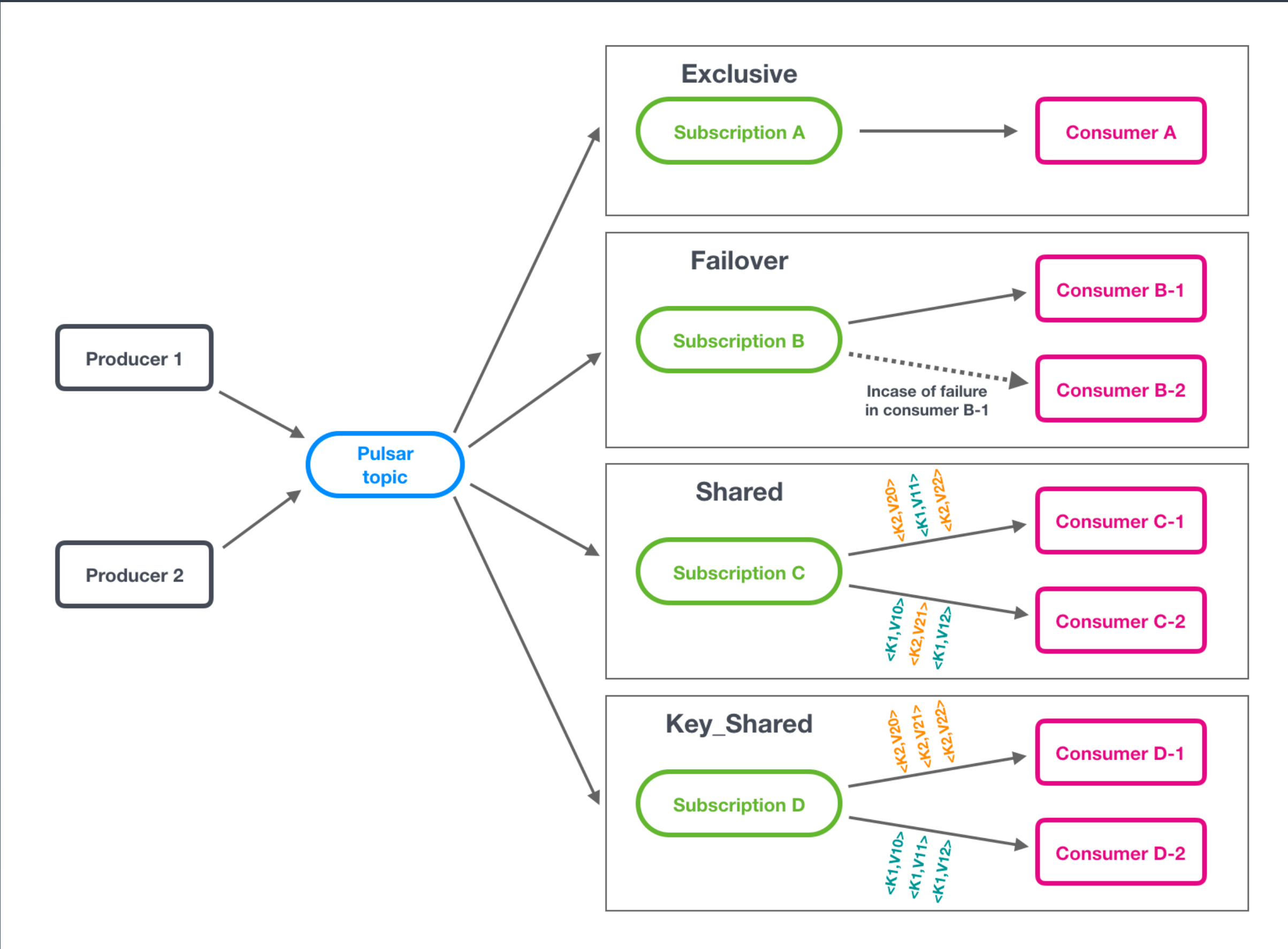
# Pulsar 介绍

基于 topic, 支持 namespace 和多租户



# Pulsar 介绍

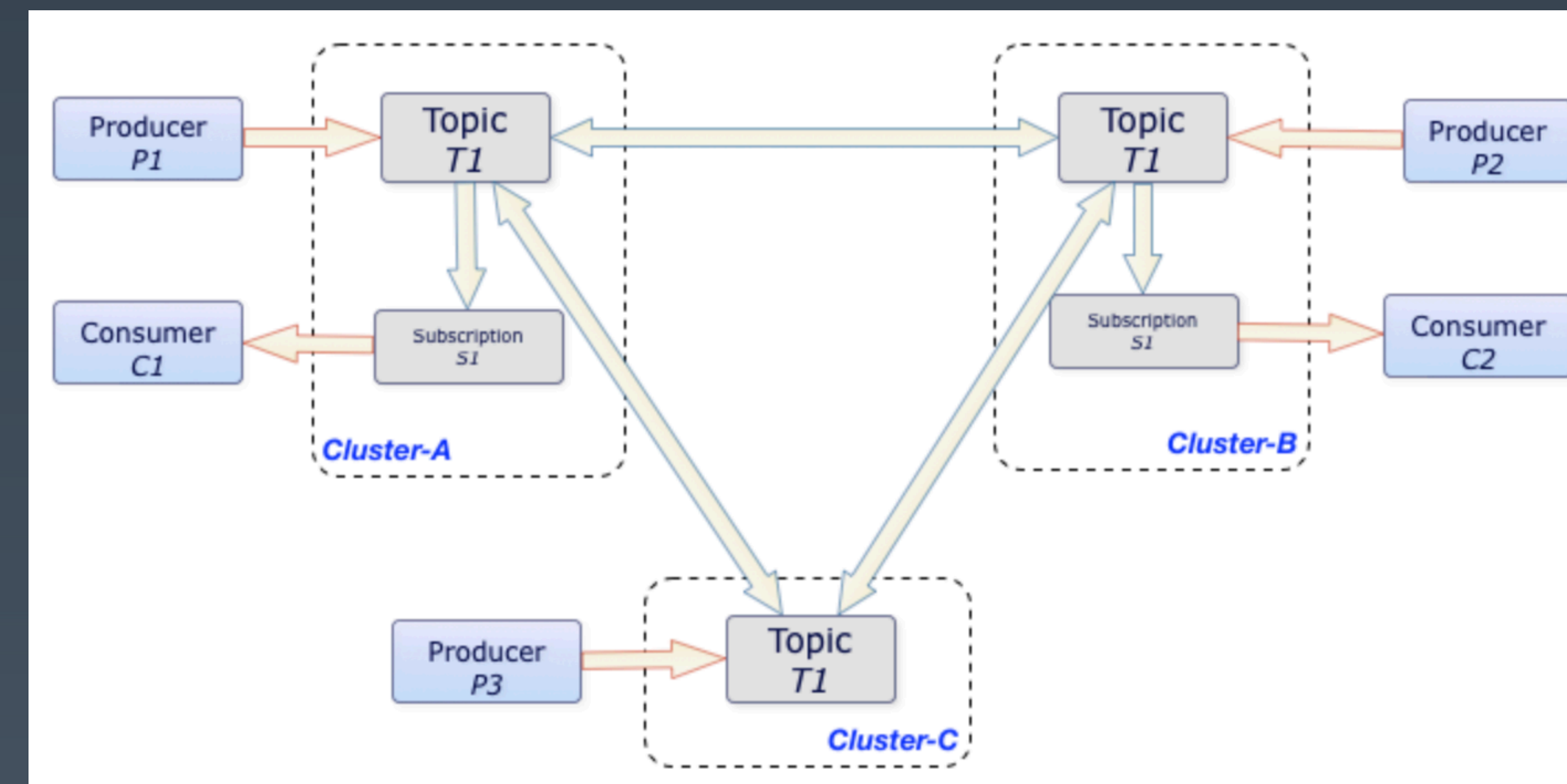
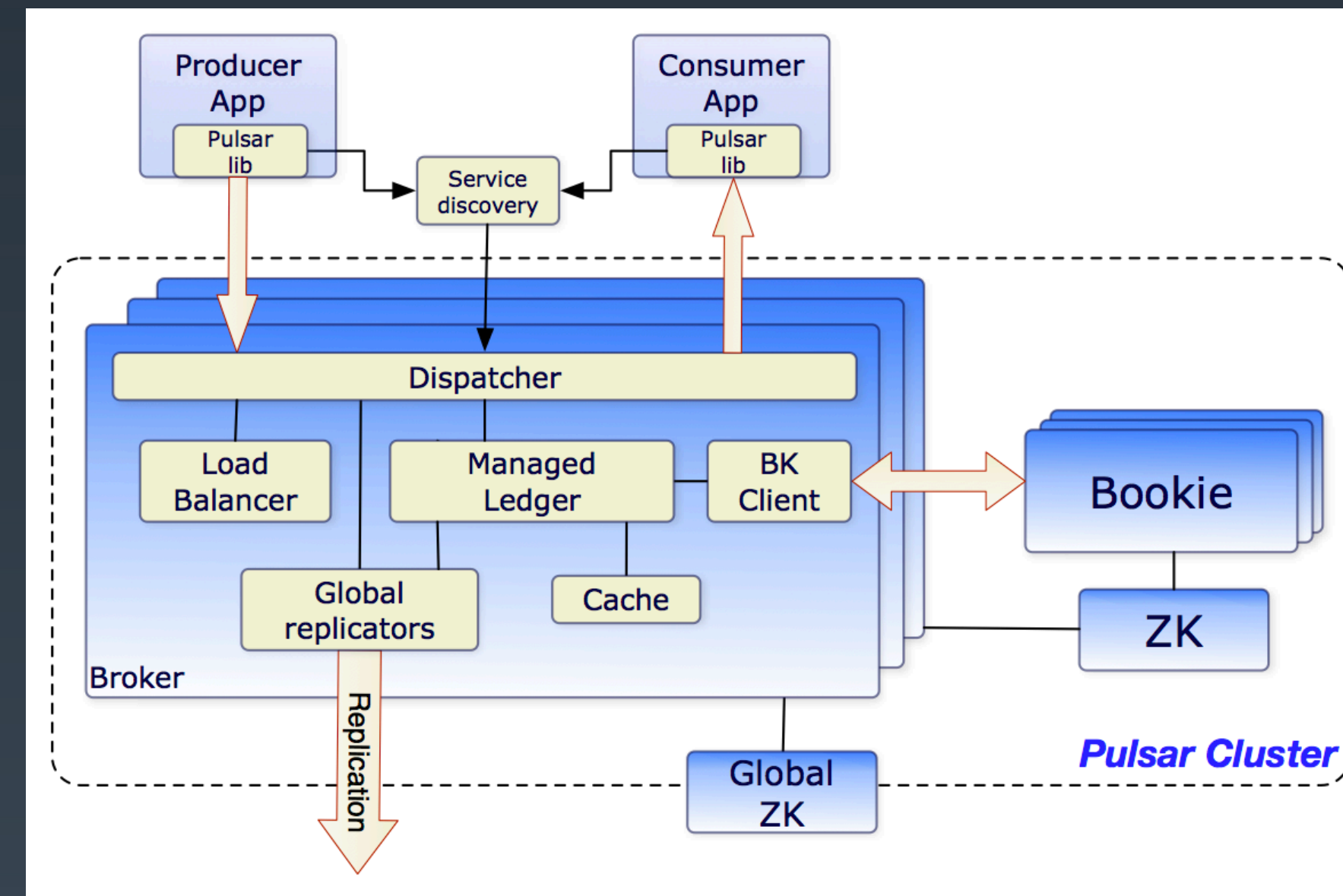
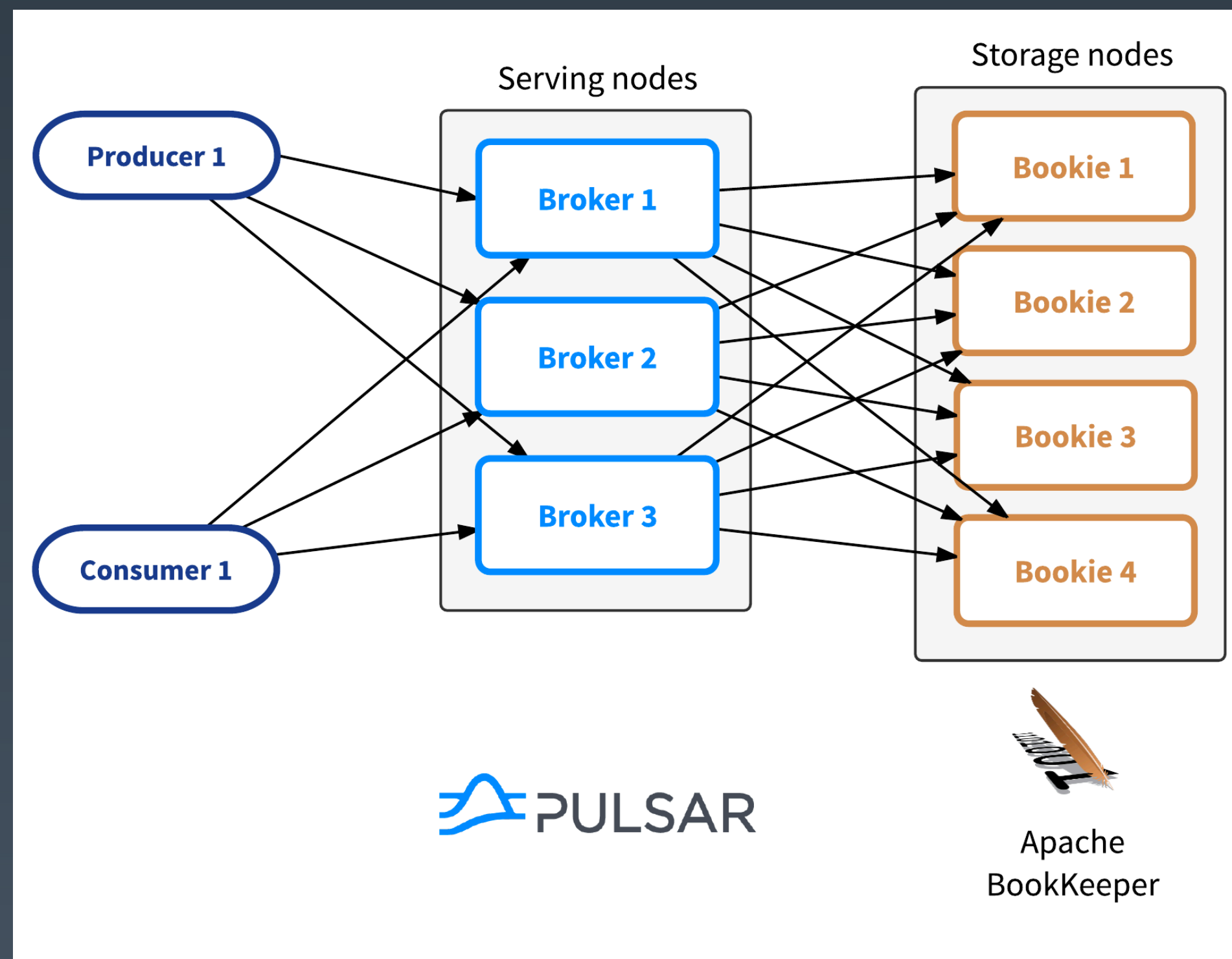
四种消费模式，支持 Partition





# Pulsar 介绍

计算存储分离，高可用集群



# Pulsar 演示

示例代码。



# 再谈 EIP

集成领域的两大法宝，就是 RPC 和 Messaging

也是所有 SOA/ESB 的基础。

两个开源 EIP 实现框架，Camel 和 Spring Integration

# 还是管道加过滤器模式

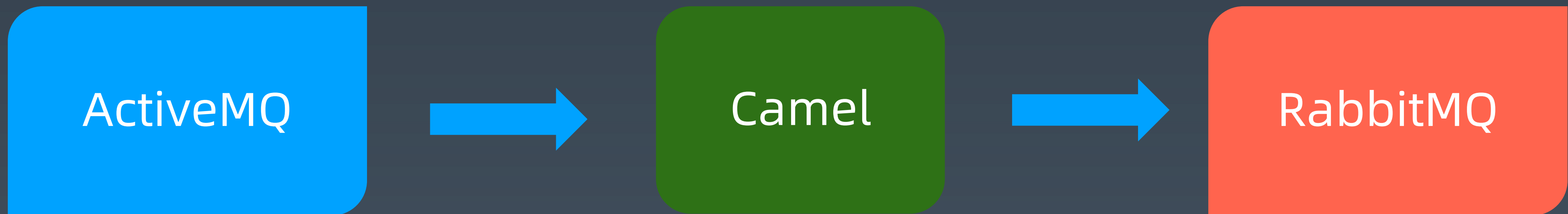
EIP 里，所有的处理，都可以看做是：

- 1、数据从一个输入源头出发；
- 2、数据在一个管道流动；
- 3、经过一些处理节点，数据被过滤器处理，增强，或者转换，或者做个业务处理等等。
- 4、最后，数据输出到一个目的地。

# 以 Camel 为例

做个好玩的 demo:

把 ActiveMQ 的消息, 自动转移到 RabbitMQ。





# 第一个版本-内存 Queue

- 1、基于内存 Queue 实现生产和消费 API（已经完成）
  - 1) 创建内存 BlockingQueue, 作为底层消息存储
  - 2) 定义 Topic, 支持多个 Topic
  - 3) 定义 Producer, 支持 Send 消息
  - 4) 定义 Consumer, 支持 Poll 消息

代码库里，本程序已经实现。



## 第二个版本：自定义 Queue

2、去掉内存 Queue，设计自定义 Queue，实现消息确认和消费 offset

1) 自定义内存 Message 数组模拟 Queue。

2) 使用指针记录当前消息写入位置。

3) 对于每个命名消费者，用指针记录消费位置。

因为数据没有真实的弹出，还在内存，容易 OOM。

不要着急，后面考虑。。。

作业相关。

# 第三个版本：基于 SpringMVC 实现 MQServer

3、拆分 broker 和 client（包括 producer 和 consumer）

1) 将 Queue 保存到 web server 端

2) 设计消息读写 API 接口，确认接口，提交 offset 接口

3) producer 和 consumer 通过 httpclient 访问 Queue

4) 实现消息确认，offset 提交

5) 实现 consumer 从 offset 增量拉取

从单机走向服务器模式。

作业相关。

## 第四个版本：功能完善 MQ

4、增加多种策略（各条之间没有关系，可以任意选择实现）

1）考虑实现消息过期，消息重试，消息定时投递等策略

2）考虑批量操作，包括读写，可以打包和压缩

3）考虑消息清理策略，包括定时清理，按容量清理、LRU 等

4）考虑消息持久化，存入数据库，或 WAL 日志文件，或 BookKeeper

5）考虑将 spring mvc 替换成 netty 下的 TCP 传输协议，rsocket/websocket

完全功能。内存不 OOM，持久化。

特别是走 TCP，可以真正做到 server->client，从而实现 PUSH 模式。

# 第五个版本：体系完善 MQ

5、对接各种技术（各条之间没有关系，可以任意选择实现）

- 1) 考虑封装 JMS 1.1 接口规范
- 2) 考虑实现 STOMP 消息规范
- 3) 考虑实现消息事务机制与事务管理器
- 4) 对接 Spring
- 5) 对接 Camel 或 Spring Integration
- 6) 优化内存和磁盘的使用

到这一步，可以业务系统里放心使用了。

# 第 26 课总结回顾

RabbitMQ

RocketMQ

Pulsar

EIP/Camel

动手做 MQ

## 第 26 课作业实践

- 1、（选做）自己安装和操作 RabbitMQ, RocketMQ, Pulsar, 以及 Camel 和 Spring Integration。
- 2、（**必做**）思考和设计自定义 MQ 第二个版本或第三个版本，写代码实现其中至少一个功能点，把设计思路和实现代码，提交到 GitHub。
- 3、（挑战☆☆☆）使用 camel 把 ActiveMQ, RabbitMQ, Kafka, RocketMQ, Pulsar, 全部串起来，实现往 ActiveMQ 的 q1 队列发送一条消息，然后走一圈，最后来到 Pulsar 的 q1 队列。
- 4、（挑战☆☆☆☆☆）完成所有其他版本的要求。期限一年。

THANKS! |  极客大学