

极客大学 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

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

第 26 课 1. RabbitMQ

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
```

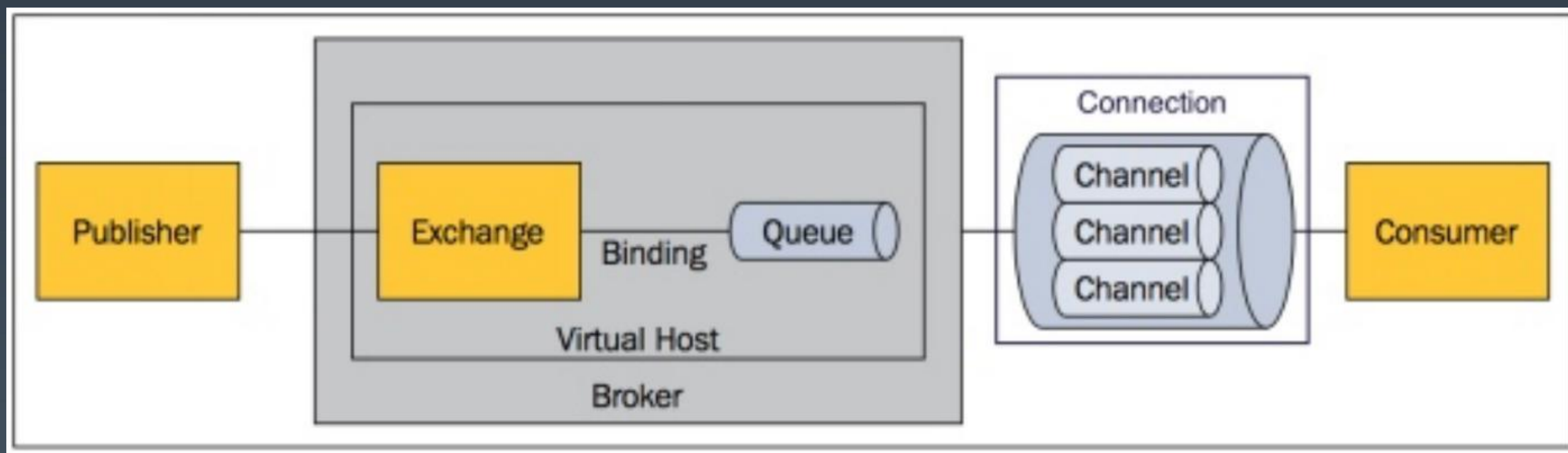
```
> rabbitmqctl list_queues、rabbitmqctl status
```

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

```
> rabbitmqadmin get queue=kk1 -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。

第 26 课 2. RocketMQ

安装

1、直接安装，从<http://rocketmq.apache.org/dowloading/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
```

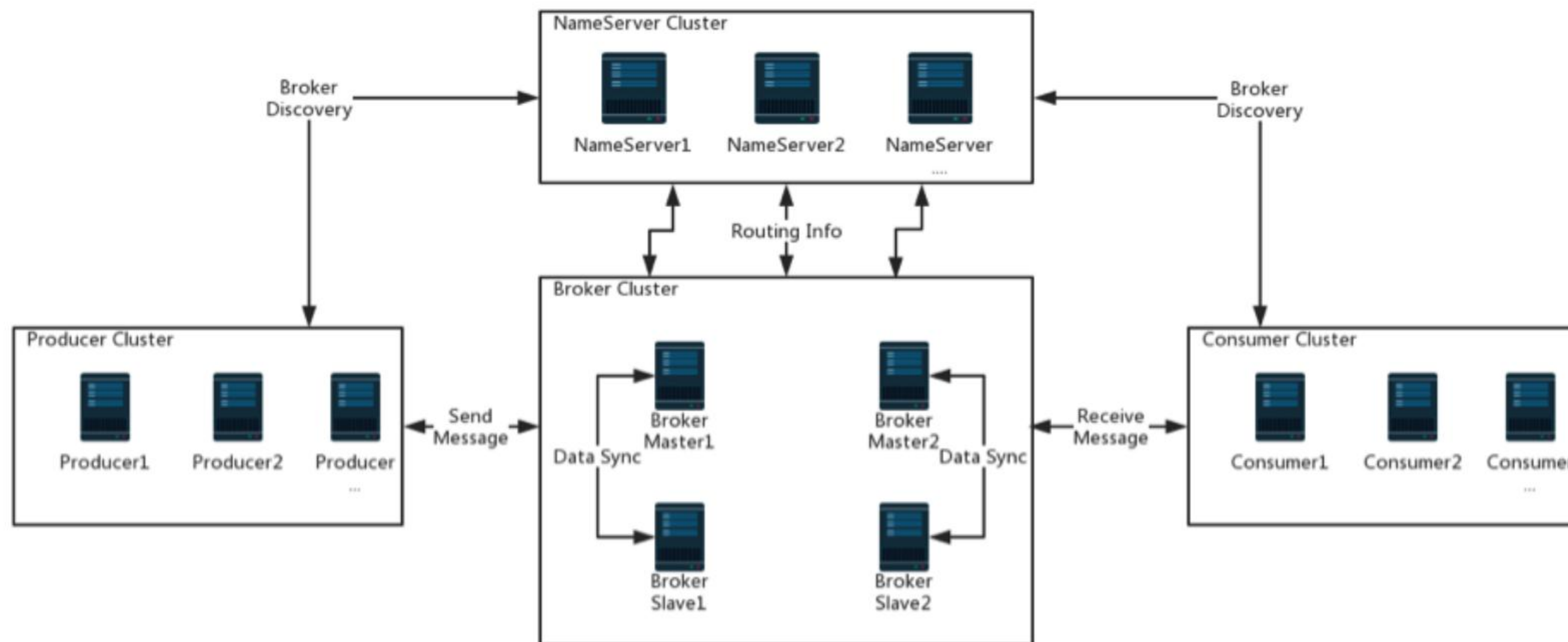
参见：<http://rocketmq.apache.org/docs/quick-start/>

2、docker安装

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

RocketMQ介绍

示例代码。



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

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。

第 26 课 3. Pulsar

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"
```

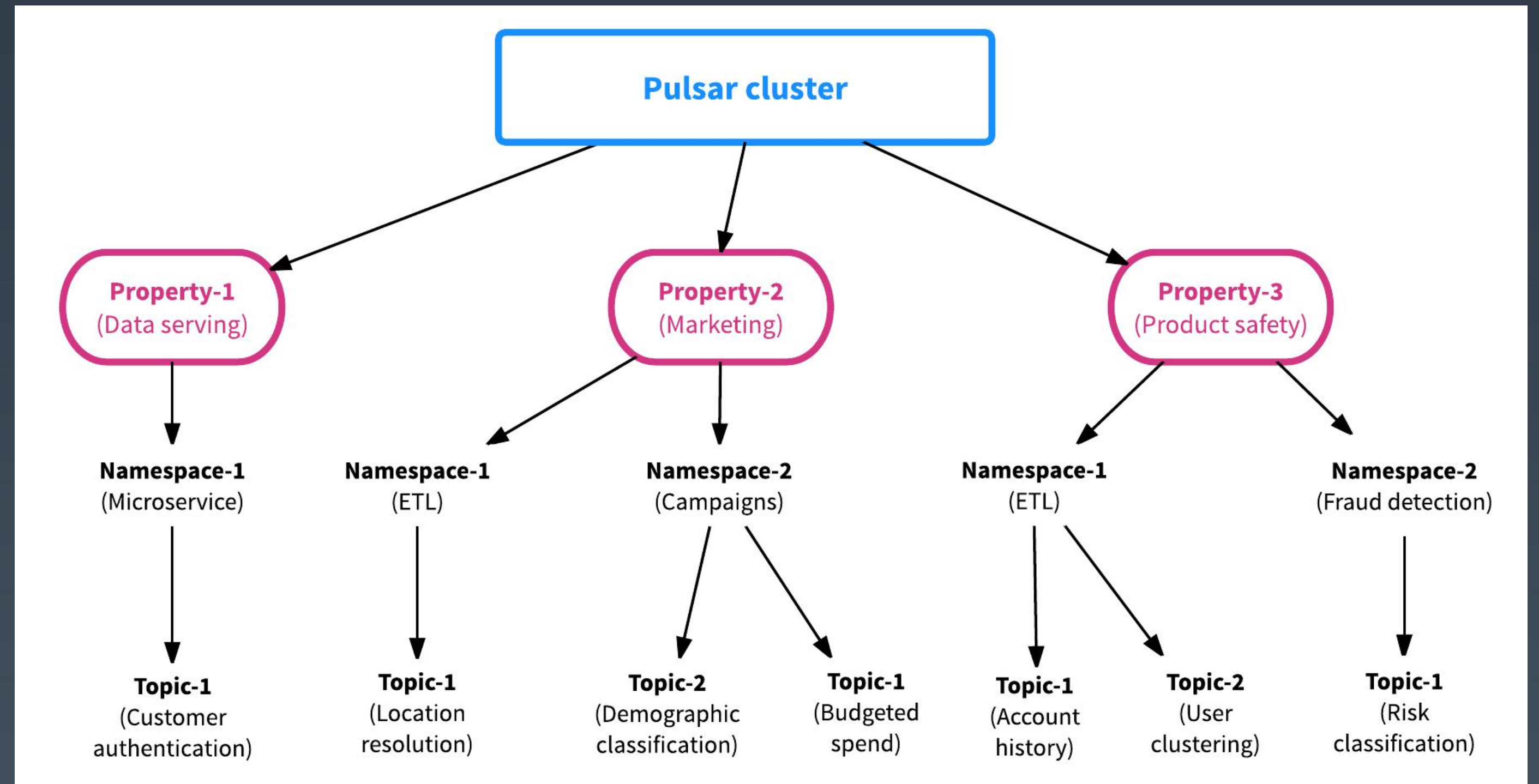
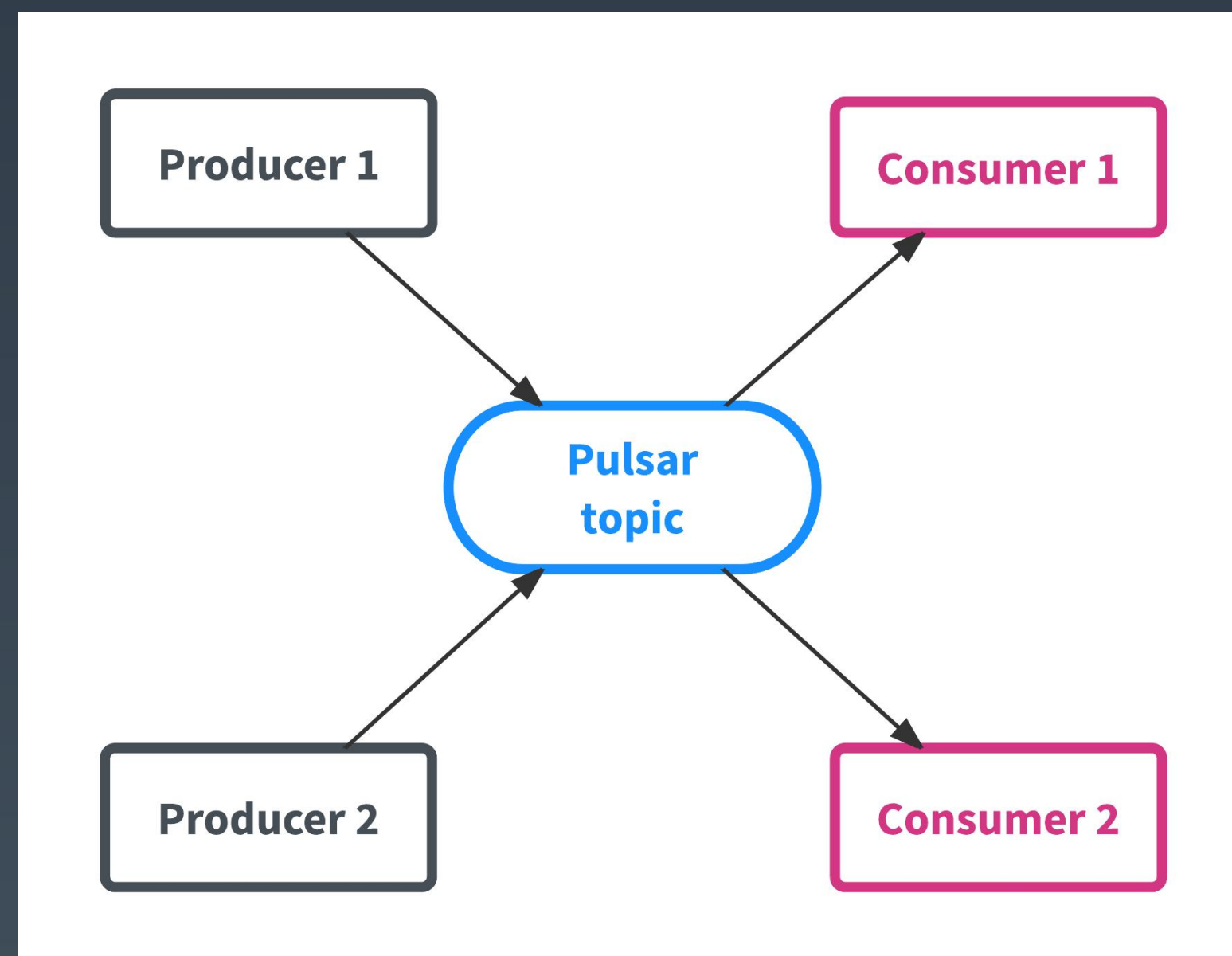
```
> bin/pulsar-client produce topic1 --messages "hello-pulsar"
```

2、Docker安装运行

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

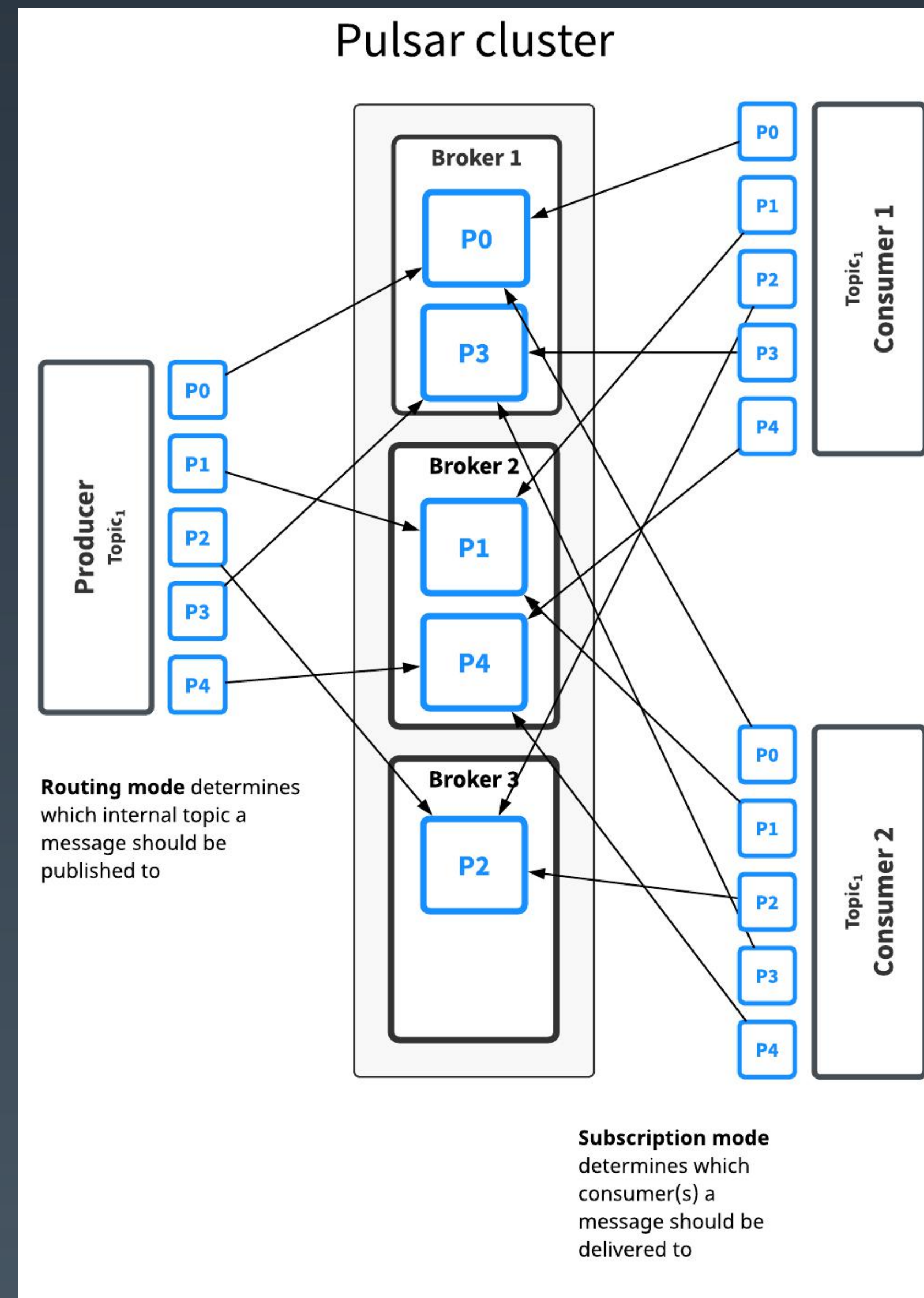
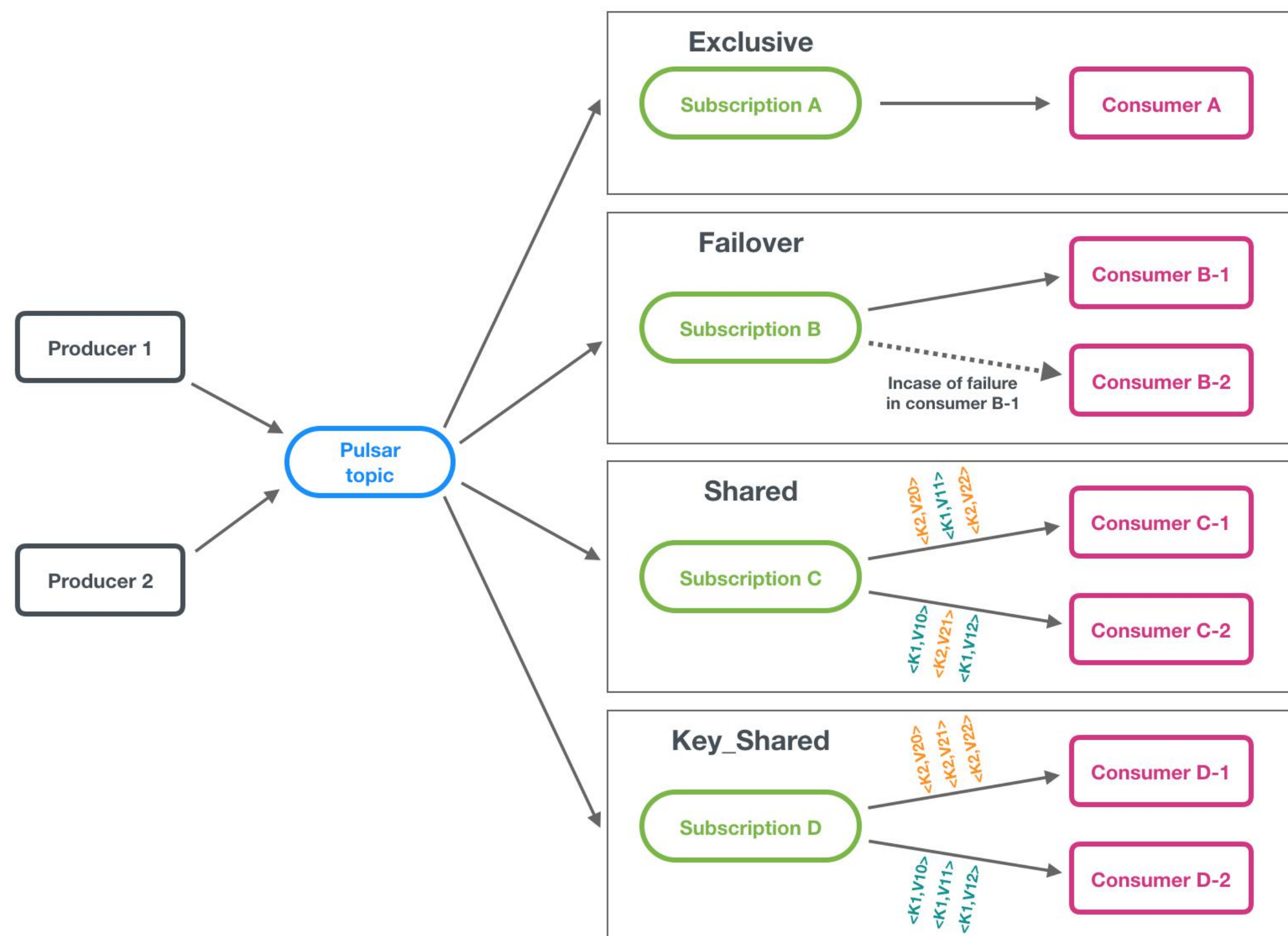
Pulsar介绍

基于topic, 支持namespace和多租户



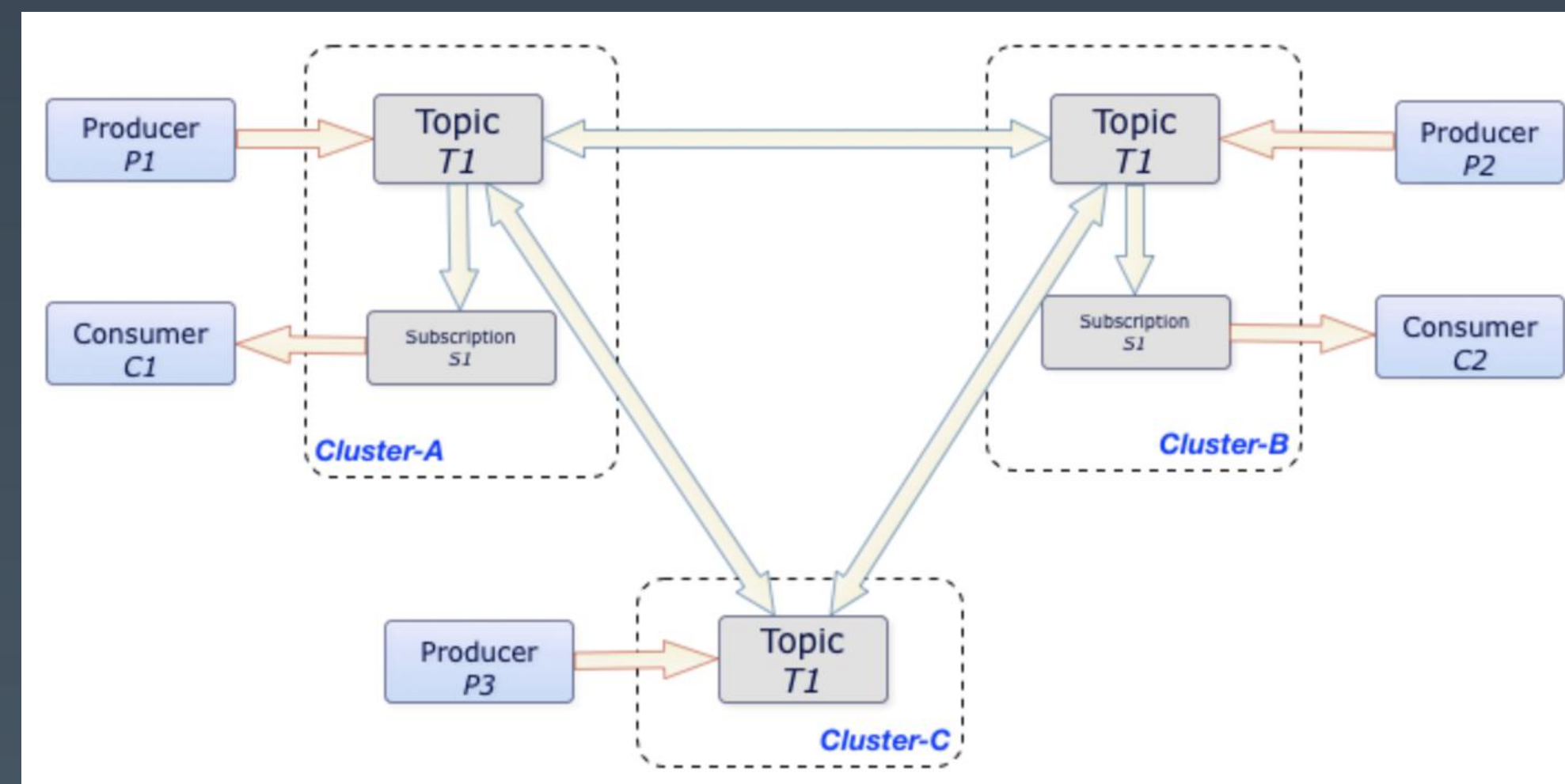
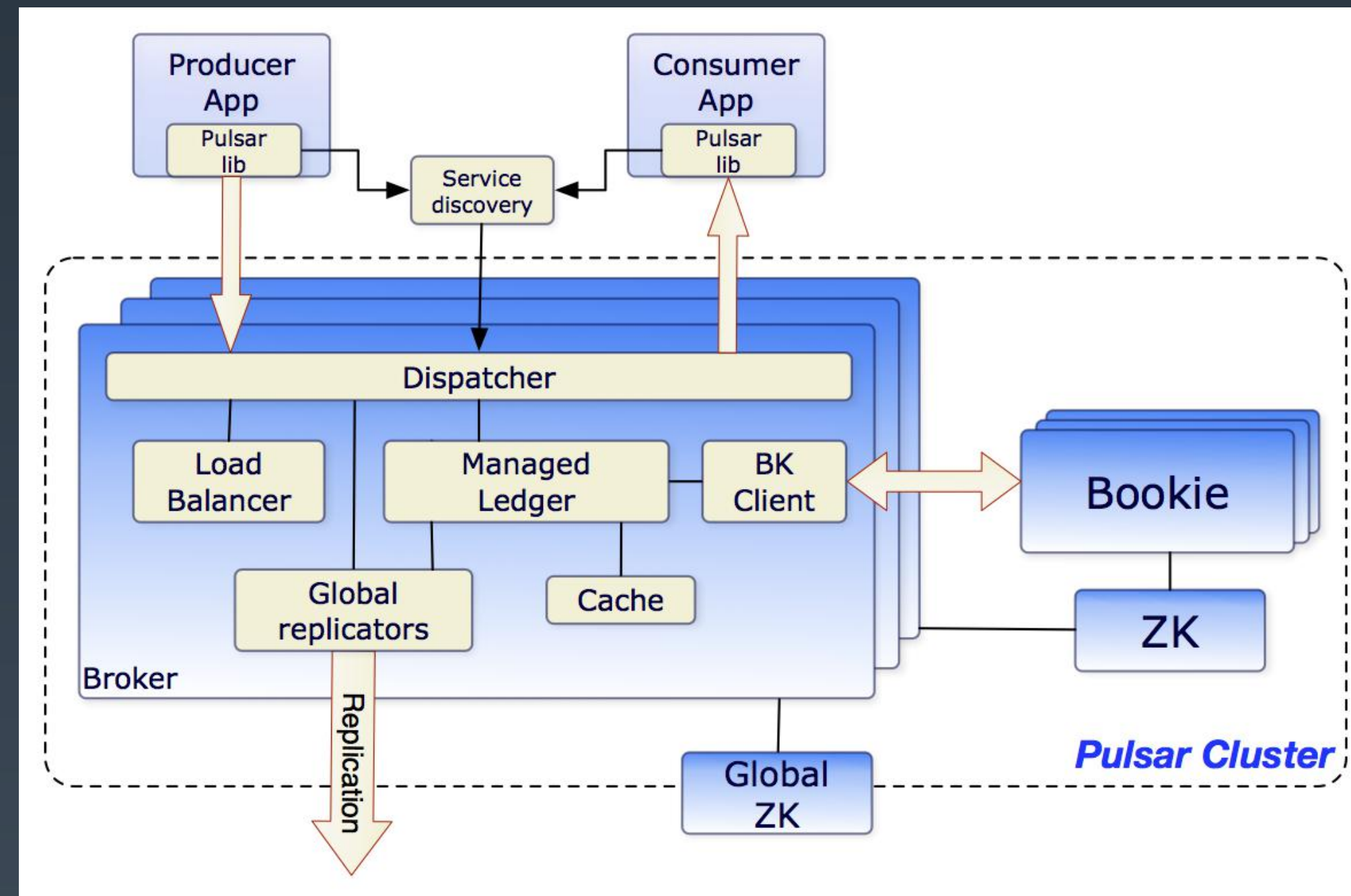
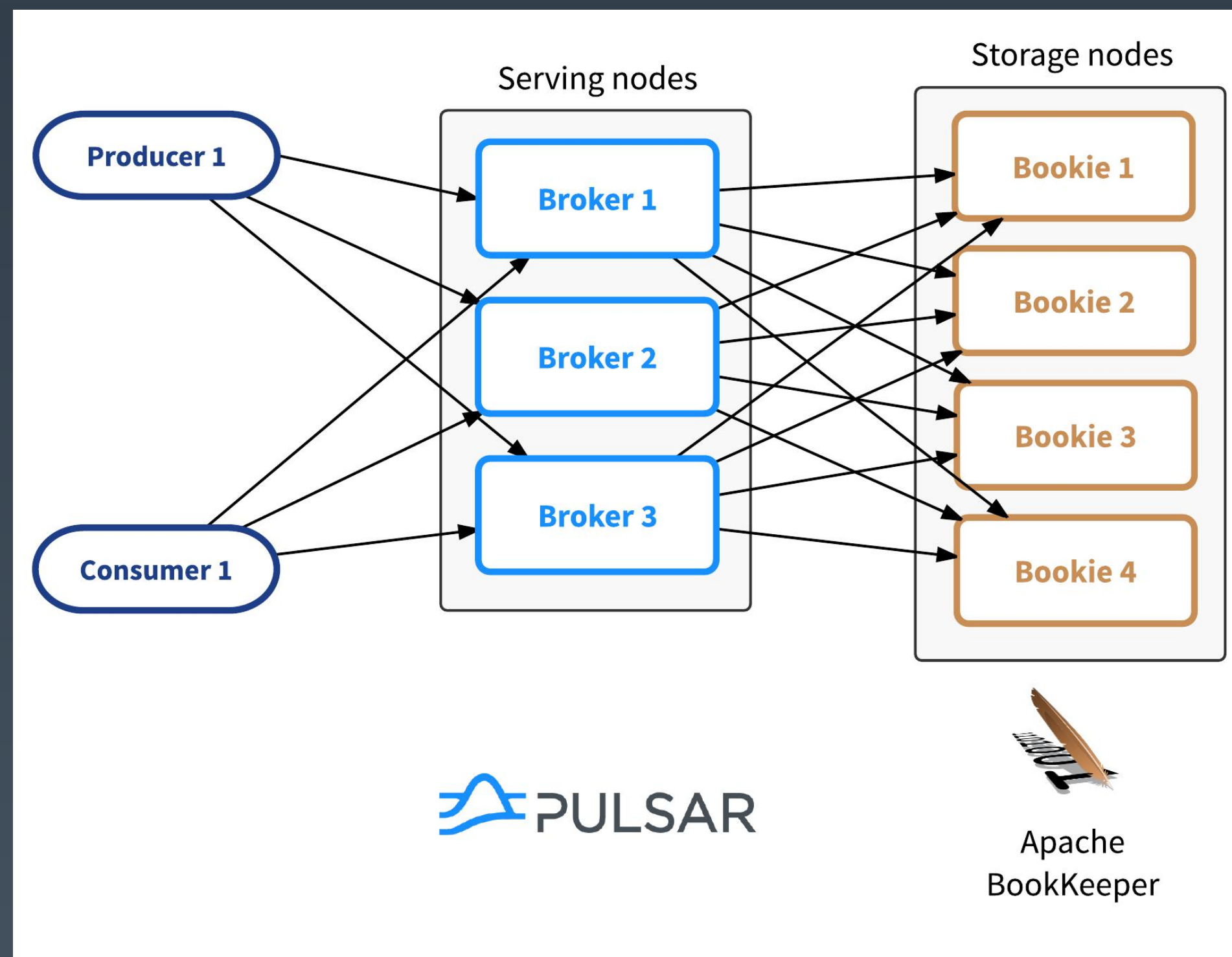
Pulsar介绍

三种消费模式，支持Partition



Pulsar介绍

计算存储分离，高可用集群



pulsar演示

示例代码。

第 26 课 4. EIP

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 26 课 5. 动手写MQ

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Queue，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

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。

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

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
 - 2) 考虑消息清理策略，包括定时清理，按容量清理等
 - 3) 考虑消息持久化，存入数据库，或WAL日志文件，或BookKeeper
 - 4) 考虑将spring mvc替换成netty下的tcp传输协议

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

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SI。
- 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☆☆☆☆☆）完成其他版本的要求。

THANKS! |  极客大学