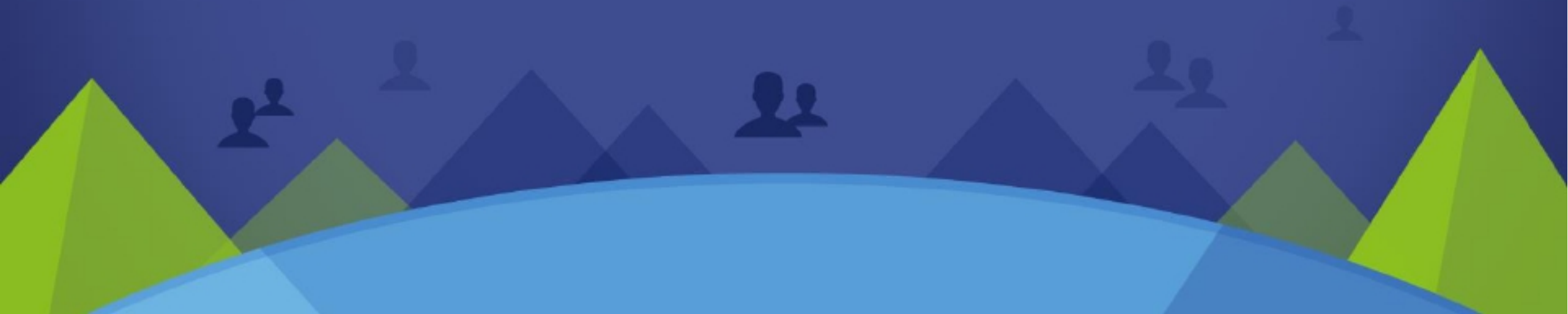

RabbitMQ

哈尔滨工业大学（威海）计算机科学与技术学院 朴学峰

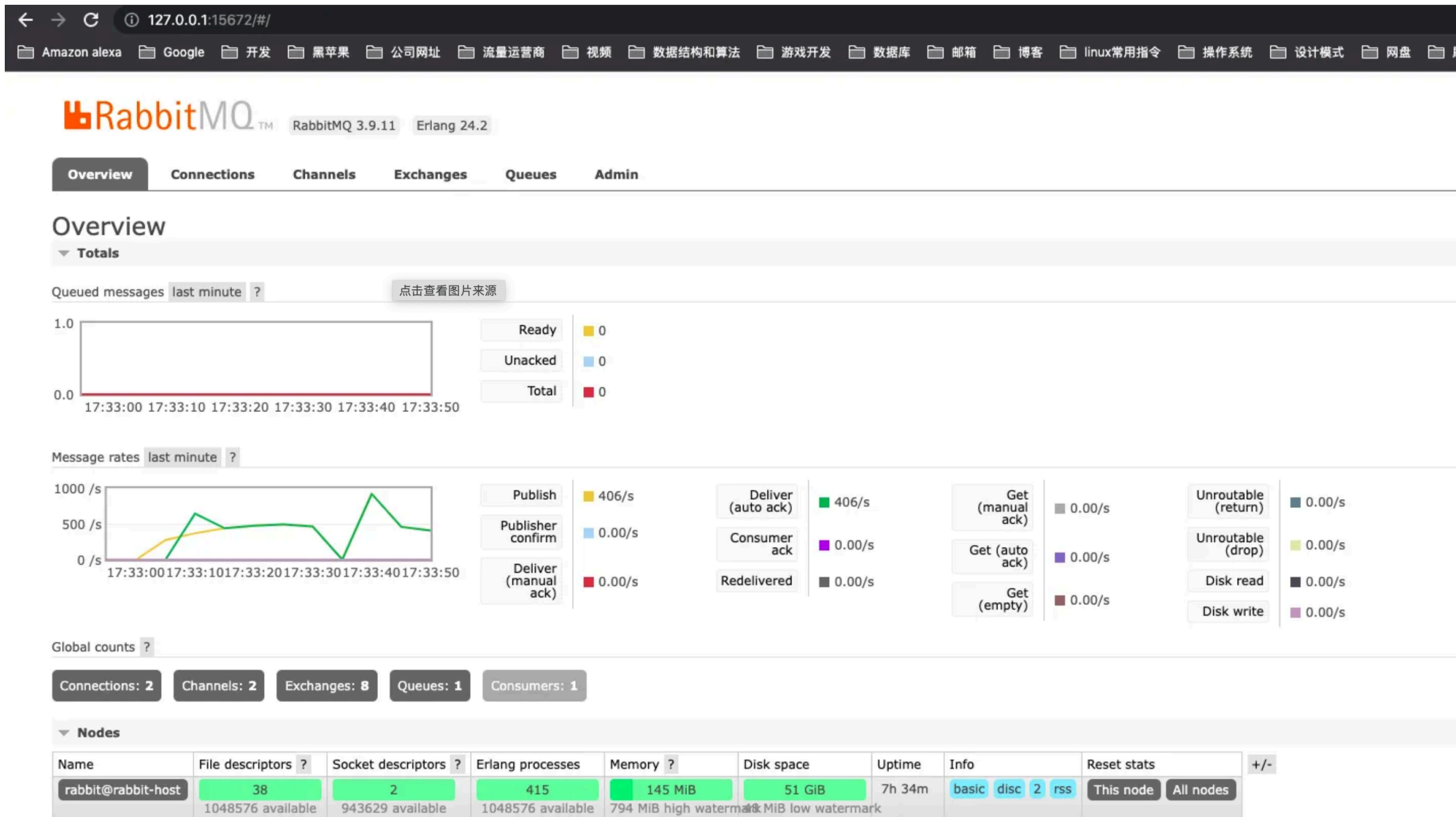


1. RabbitMQ 概念
2. RabbitMQ 工作模式及原理

RabbitMQ

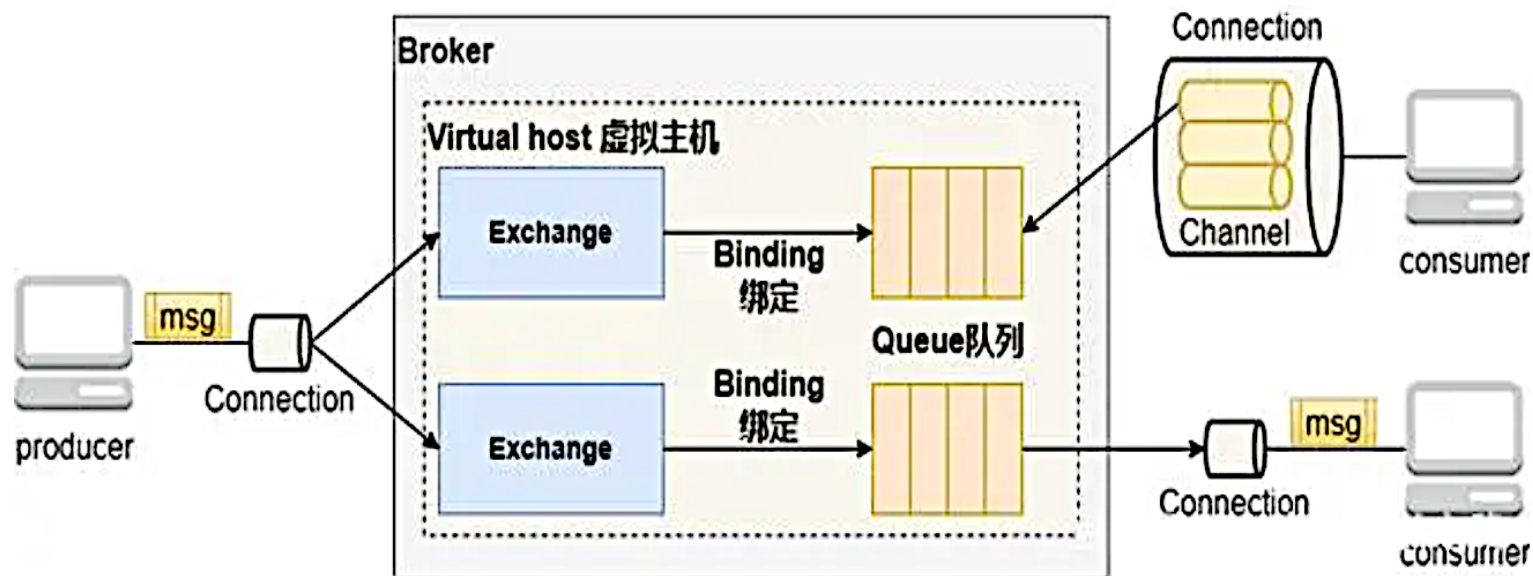
1. RabbitMQ 是一套开源的消息队列服务软件，实现了高级消息队列协议（AMQP：Advanced Message Queuing Protocol）的开源消息代理软件，又称面向消息的中间件，**is open source message brokers**。
2. 使用 ErLang 语言编写
3. 2007年，起初由 Rabbit 科技有限公司开发，于2010年被VMware 旗下的一家公司（SpringSource）收购
4. 具有可伸缩性和消息持久化的特性
5. [RabbitMQ 的安装](#)



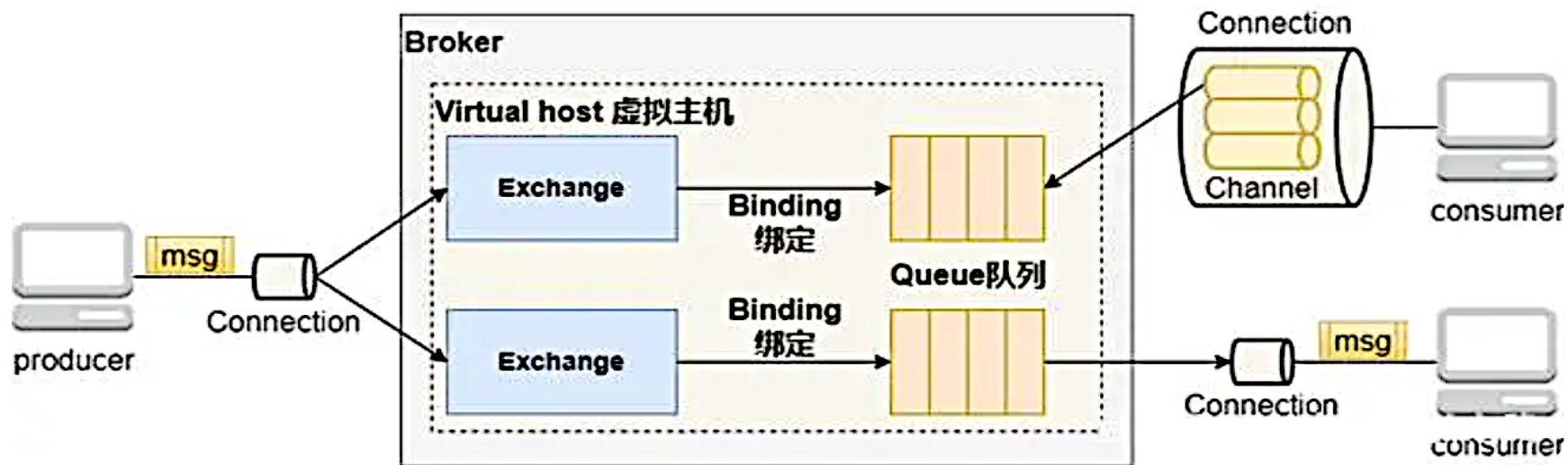


RabbitMQ 五大模块

1. Connection (链接)
2. Channel (通道、信道)
3. Exchange (交换)
4. Queue (队列)
5. Virtual Host (虚拟主机)



RabbitMQ 工作模式



- 中间的 Broker 表示 RabbitMQ 服务
- 每个 Broker 里至少有一个 Virtual Host,
- 每个 Virtual Host 里有自己的 Exchange 和 Queue, 且 Exchange 和 Queue 进行了绑定
- 生产者和消费者通过与 Broker 建立 Connection 保持连接, 并在 Connection 的基础上建立若干个 Channel, 最终用 Channel 发送和接收消息

1. Connection

每个生产者和消费者要通过 RabbitMQ 发送和接收消息，首先要与 RabbitMQ（Broker）建立连接，这个连接就是 Connection，Connection 是 TCP 的长连接

2. Channel

Channel 是在 Connection 的基础上建立的虚拟（逻辑）通道，RabbitMQ 的大部分操作都是使用 Channel 完成。如发布消息、消费消息等

疑问：已经有了 Connection，为什么还需要建立 Channel 来进行发布和消费？

→ 多线程环境下，建立多个 Connection 的成本 vs 建立多个 Channel 的成本

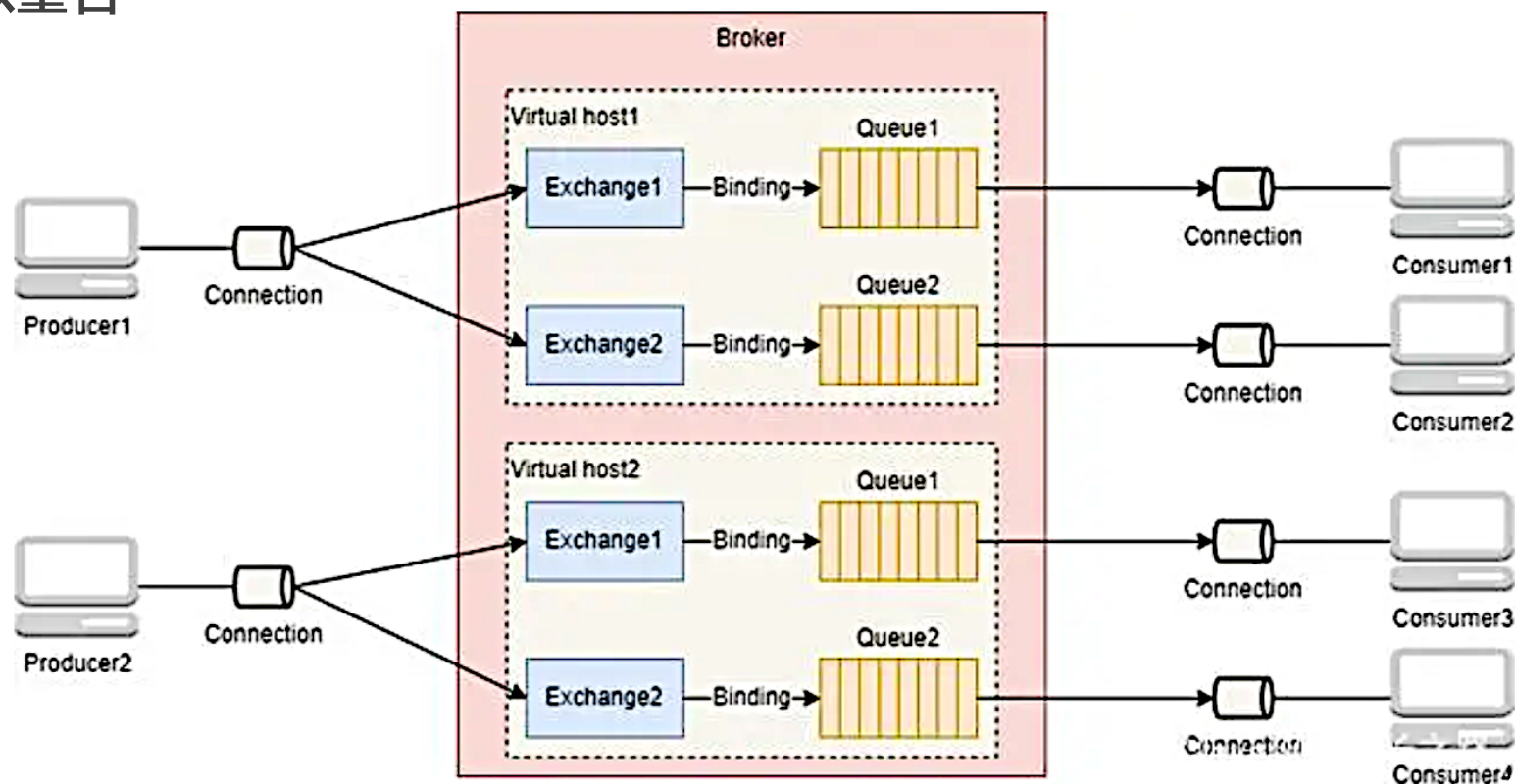
疑问：那么怎么区分 Channel 之间的消息？

→ ChannelID

3. Virtual Host

一个 Broker 可以有多个 Virtual Host，每个 Virtual Host 都有一套自己的 Exchange and Queue，但同一个 Virtual Host 中的 Exchange 之间、Queue 之间不能重名，而不同的 Virtual Host 中是可以重名

**优点是不同用户之间
可以相互隔离**

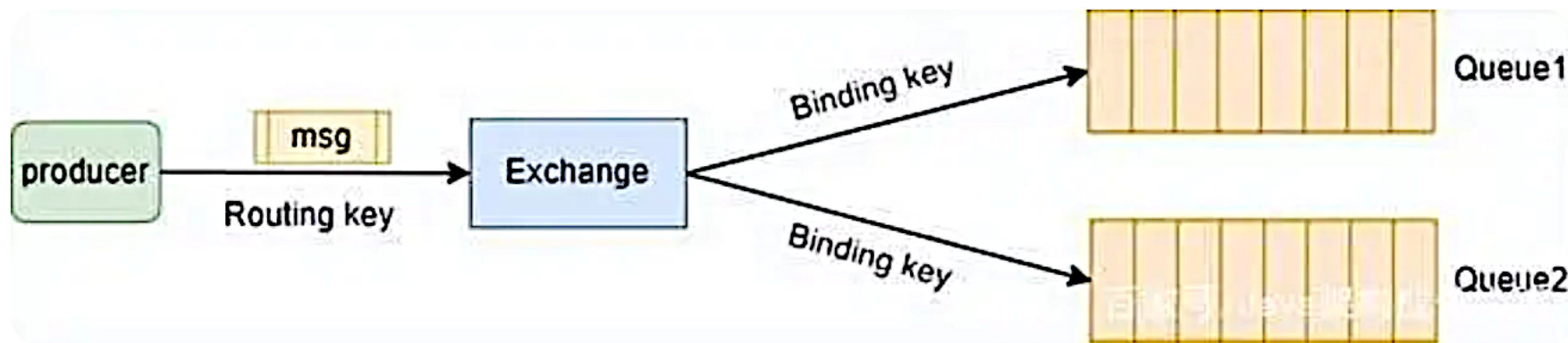


4. Queue

Queue 是用来存放消息的队列，生产者发送的消息会被存放到 Queue 中，消费者消费消息时也是从 Queue 中取走消息

5. Exchange

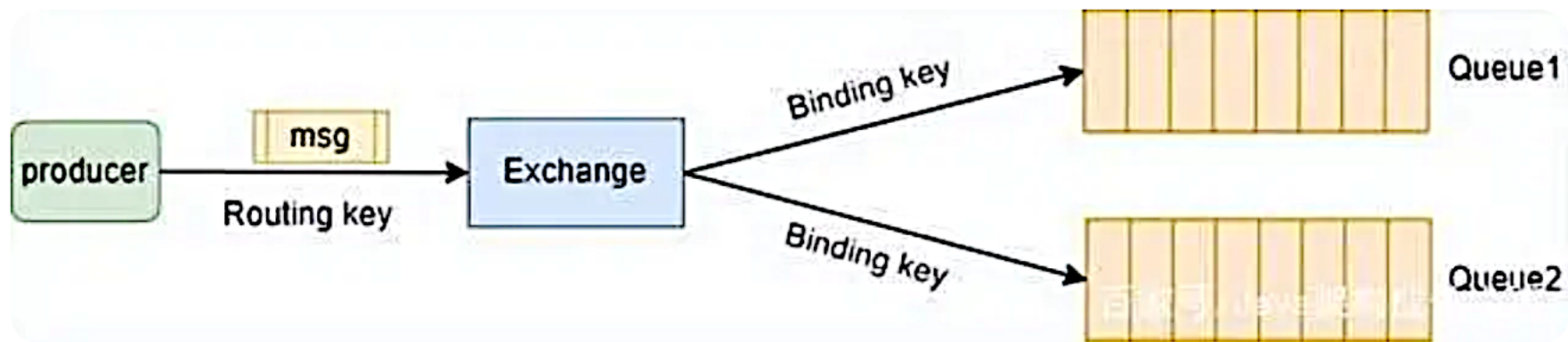
Exchange 是比较重要的概念，它是消息到达 RabbitMQ 的第一站，主要负责根据不同的分发规则将消息分发到不同的 Queue 中，供消费者订阅（消费）相关 Queue 的指定消息



5. Exchange

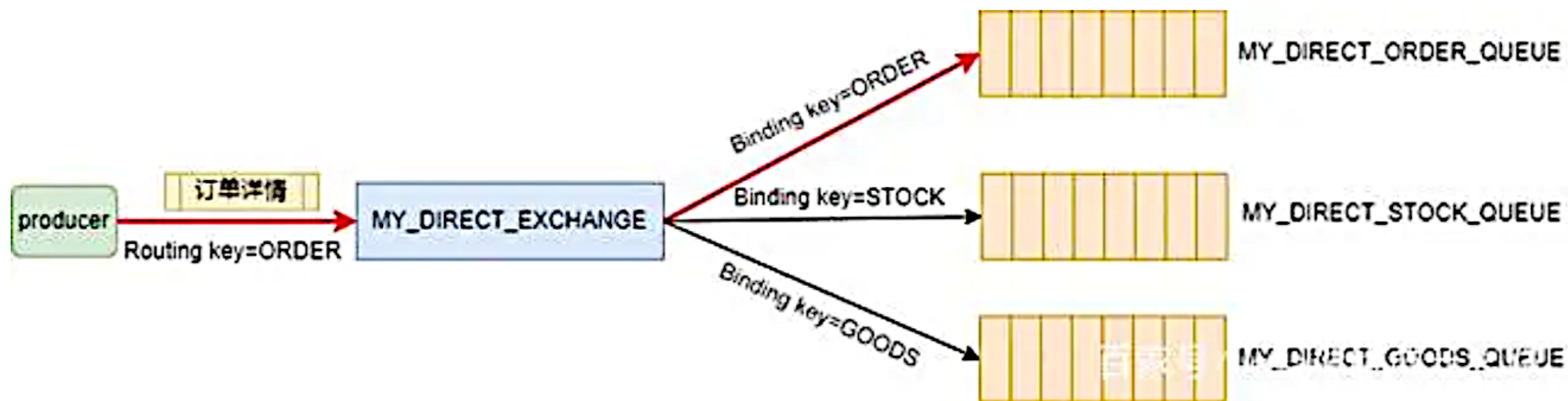
Exchange 分发机制类型: Direct、Fanout、Topic、Headers

- Routing Key : 生产者发送消息时用到 key, 生产者向 Exchange 发送消息的时候, 必须指定一个 Routing Key, Exchange 收到消息之后, 解析 Routing Key, 然后根据分发规则发送到符合规格的 Queue 中
- Binding Key: Exchange 和 Queue 之间绑定的时用到的 key



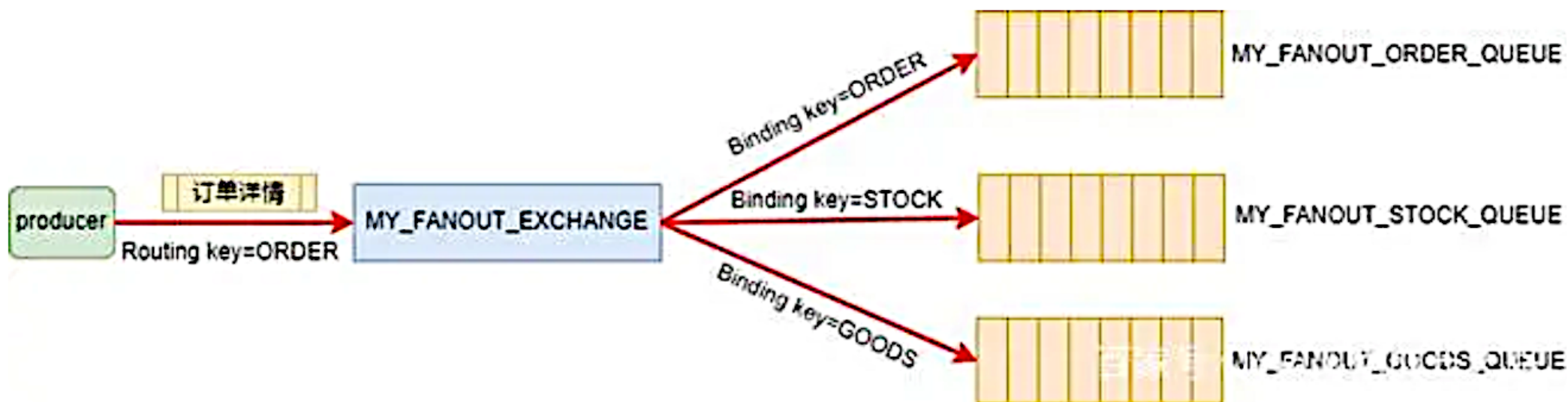
(1) Direct

Exchange 将消息转发到指定 Routing Key 的 Queue 上，也就是 Routing Key == Binding Key 时（Routing Key 与 Binding Key 的精准匹配），消息才分发到对应的 Queue 上



(2) Fanout

Fanout 是扇形的意思，该类型通常叫做广播类型。Fanout 类型的 Exchange 不处理 Routing Key，而是将消息分发到所有与它绑定的 Queue 上



(3) Topic

Topic 是主题的意思，Topic 类型的 Exchange 会根据通配符对 Routing Key 进行匹配，只要 Routing Key 满足某个通配符条件，就会分发到对应的 Queue 上

- Routing Key 必须是一串字符串，每个单词用 “.” 分隔
- 符号 “#” 表示匹配一个或多个单词
- 符号 “*” 表示匹配一个单词

例如，

“*.123” 能够匹配到 “abc.123”，但匹配不到 “abc.def.123”

“#.123” 能够匹配到 “abc.123”，也能匹配到 “abc.def.123”

(4) Headers

Headers 类型并不经常使用，即 Exchange 与 Queue 之间的绑定不再通过 Binding Key 绑定，而是通过 Arguments（条件）绑定，

例如，Exchange A 与 Queue1、Queue2、Queue3 绑定

- Queue1 的条件是 “ $x = 1$ ”，Queue2 的条件是 “ $x = 1, y = 2$ ”，Queue3 的条件是 “ $y = 2$ ”
- 生产者发送消息 Header 包含 “ $x = 1$ ”，这时 Queue1 满足条件，Queue2 不满足条件、Queue3 不满足条件，那么该消息将会分发到 Queue1 上；
- 生产者发送消息 Header 包含 “ $x = 1, y = 2$ ”，这时 Queue1、Queue2、Queue3 的条件都满足，那么该消息会分发到各个 Queue 上

THANKS

