

# RabbitMQ

月薪上万如此简单~

1

MQ的介绍与RabbitMQ的安装

2

RabbitMQ 快速入门

3

RabbitMQ 的工作模式

4

SpringBoot 整合 RabbitMQ



# MQ的介绍

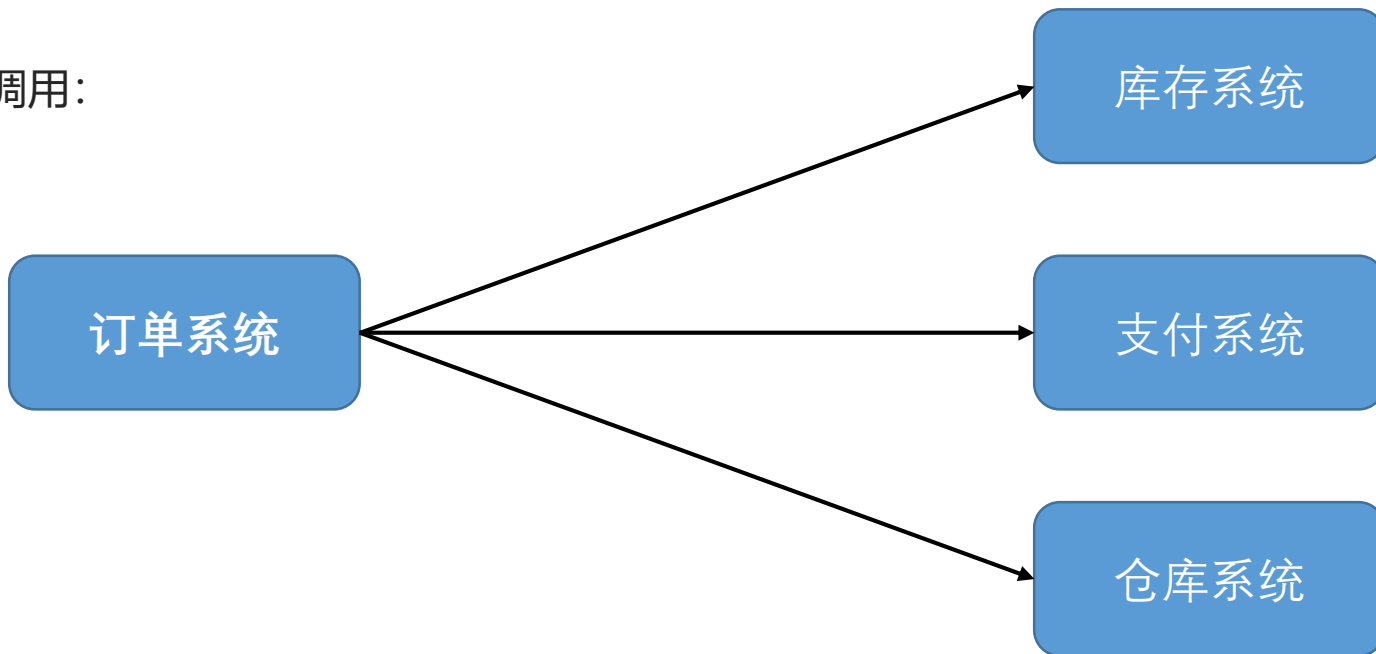
- MQ基本概念
- 常见MQ
- RabbitMQ
- RabbitMQ安装

# 一、MQ的介绍

## 1.MQ概述

MQ (Message Queue)：消息队列，是基础数据结构中FIFO的一种数据结构。一般用来解决流量削峰、应用解耦、异步处理等问题，实现高性能，高可用，可伸缩和最终一致性架构。

回顾系统调用：

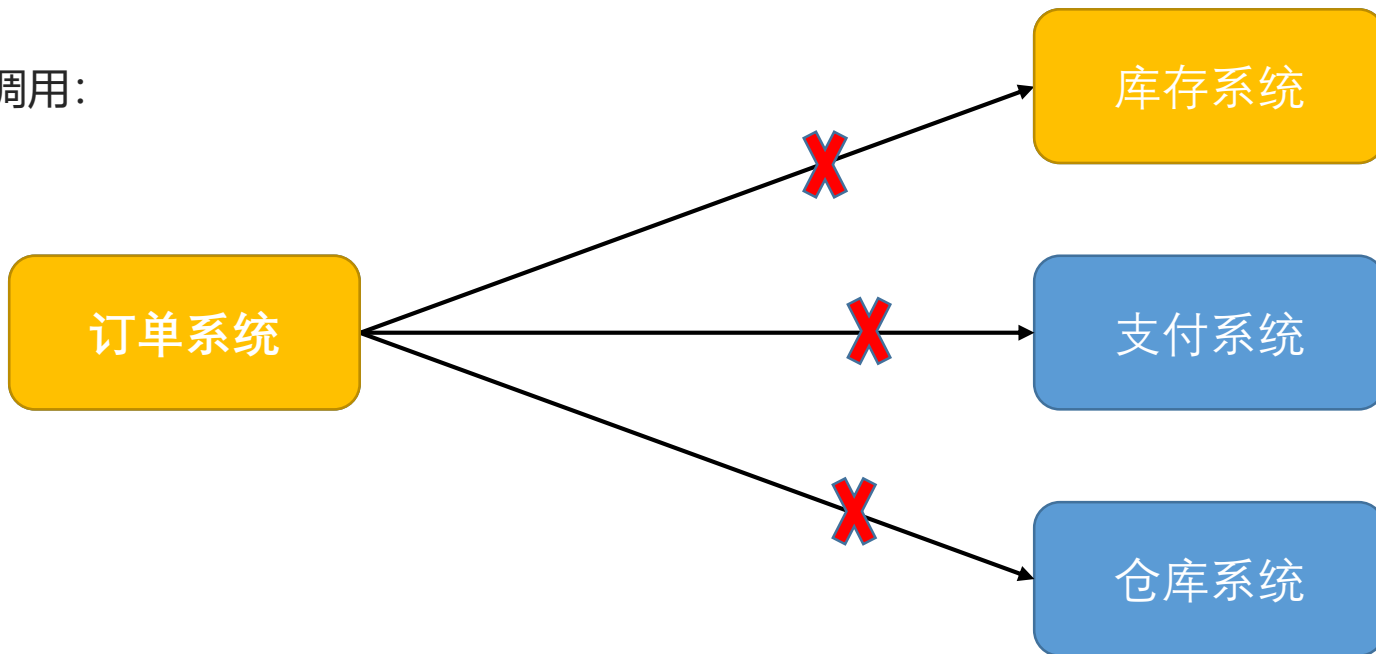


# 一、MQ的介绍

## 1.MQ概述

MQ (Message Queue)：消息队列，是基础数据结构中FIFO的一种数据结构。一般用来解决流量削峰、应用解耦、异步处理等问题，实现高性能，高可用，可伸缩和最终一致性架构。

回顾系统调用：

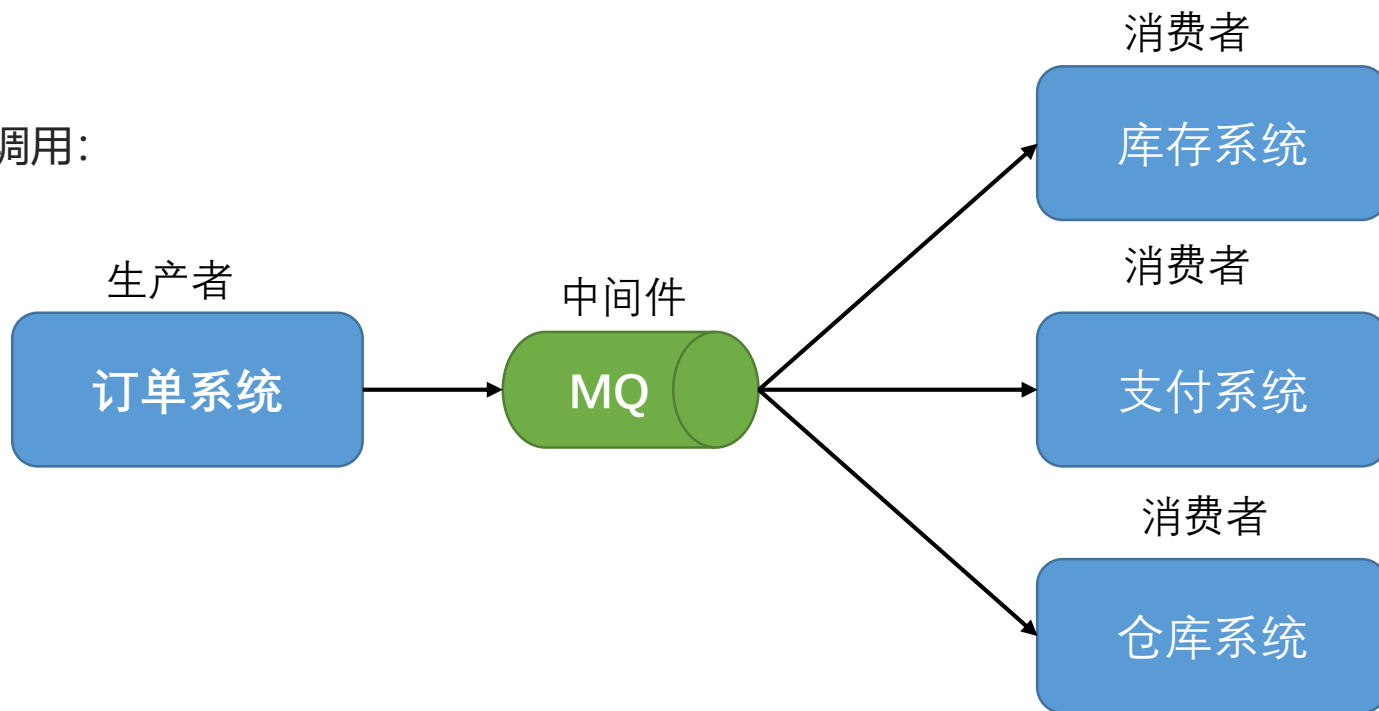


# 一、MQ的介绍

## 1.MQ概述

MQ (Message Queue)：消息队列，是基础数据结构中FIFO的一种数据结构。一般用来解决流量削峰、应用解耦、异步处理等问题，实现高性能，高可用，可伸缩和最终一致性架构。

回顾系统调用：



# 一、MQ的介绍

## 1.MQ概述

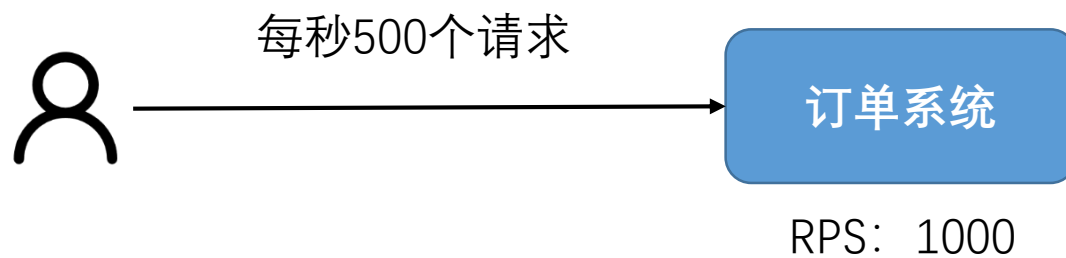
小结:

- ◆ MQ:消息队列, 用于消息传递的中间件
- ◆ 发送消息方称为生产者
- ◆ 接收消息方称为消费者

# 一、MQ的介绍

## 2.MQ优点

### ➤ 流量消峰

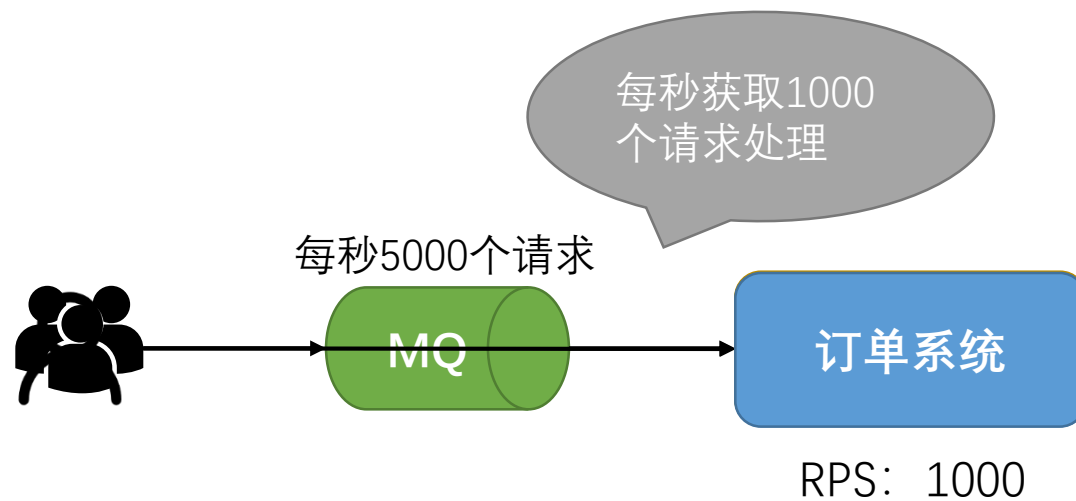




# 一、MQ的介绍

## 2.MQ优点

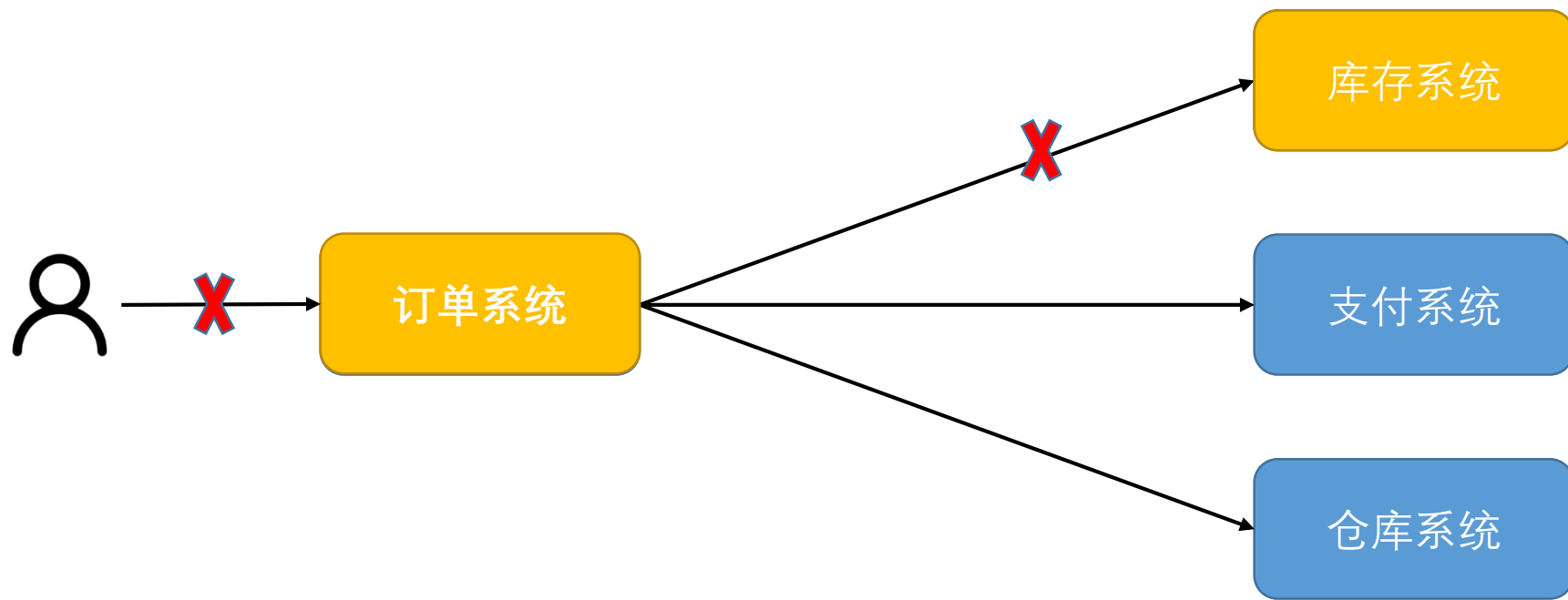
### ➤ 流量消峰



# 一、MQ的介绍

## 2.MQ优点

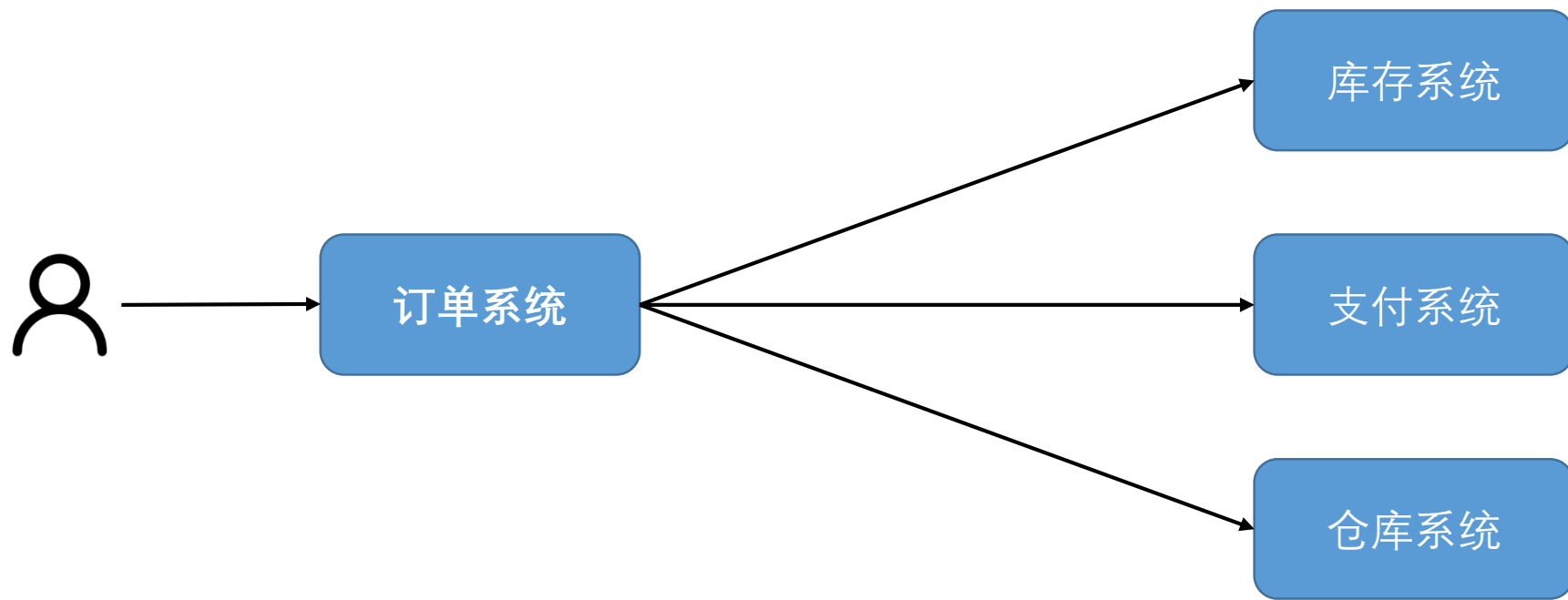
### ➤ 应用解耦



# 一、MQ的介绍

## 2.MQ优点

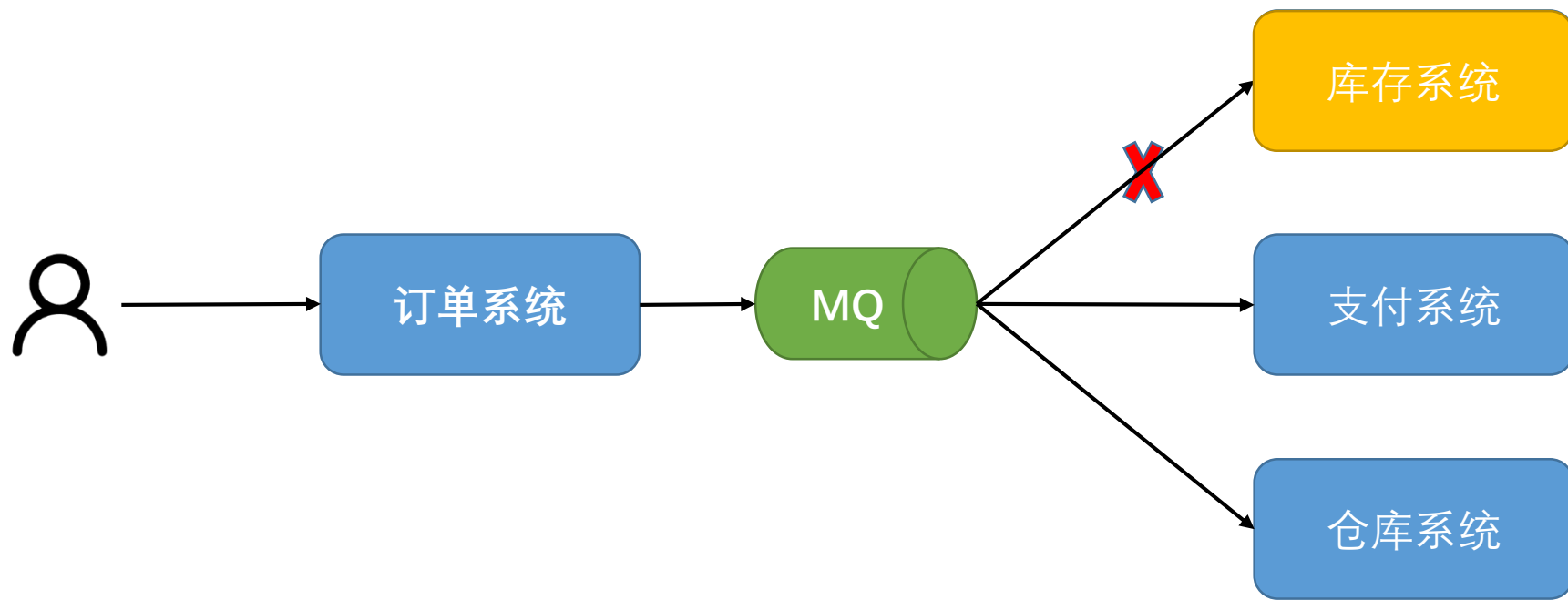
### ➤ 应用解耦



# 一、MQ的介绍

## 2.MQ优点

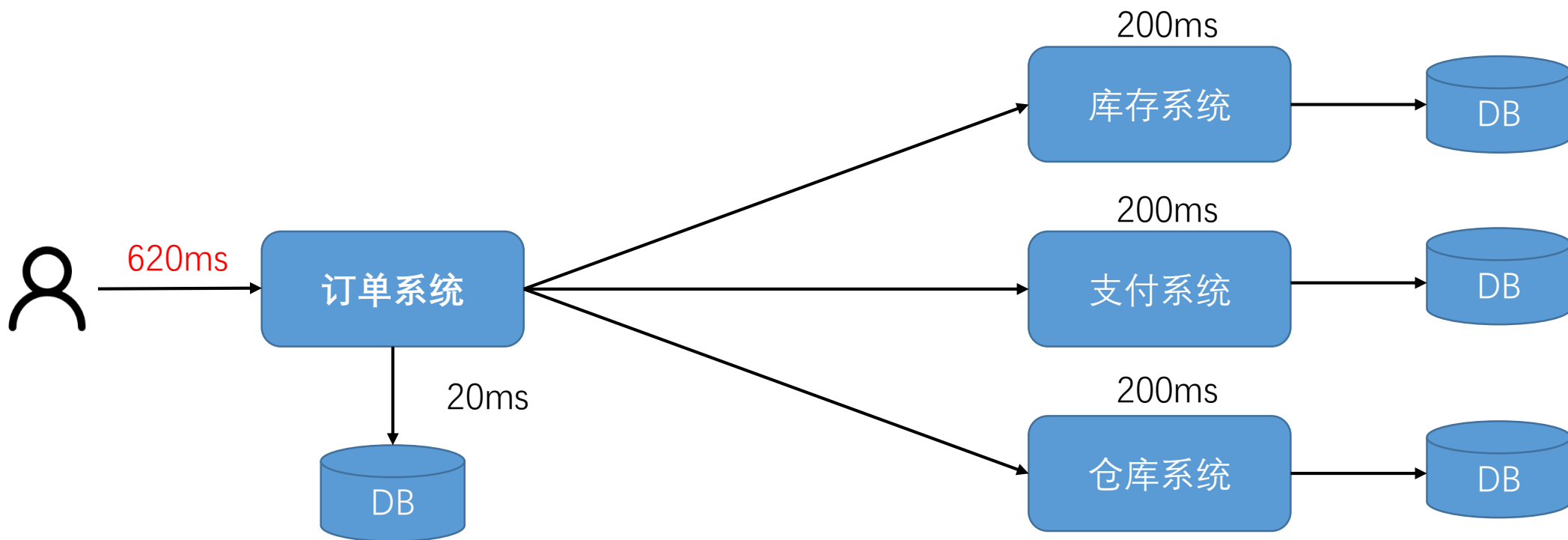
### ➤ 应用解耦



# 一、MQ的介绍

## 2.MQ优点

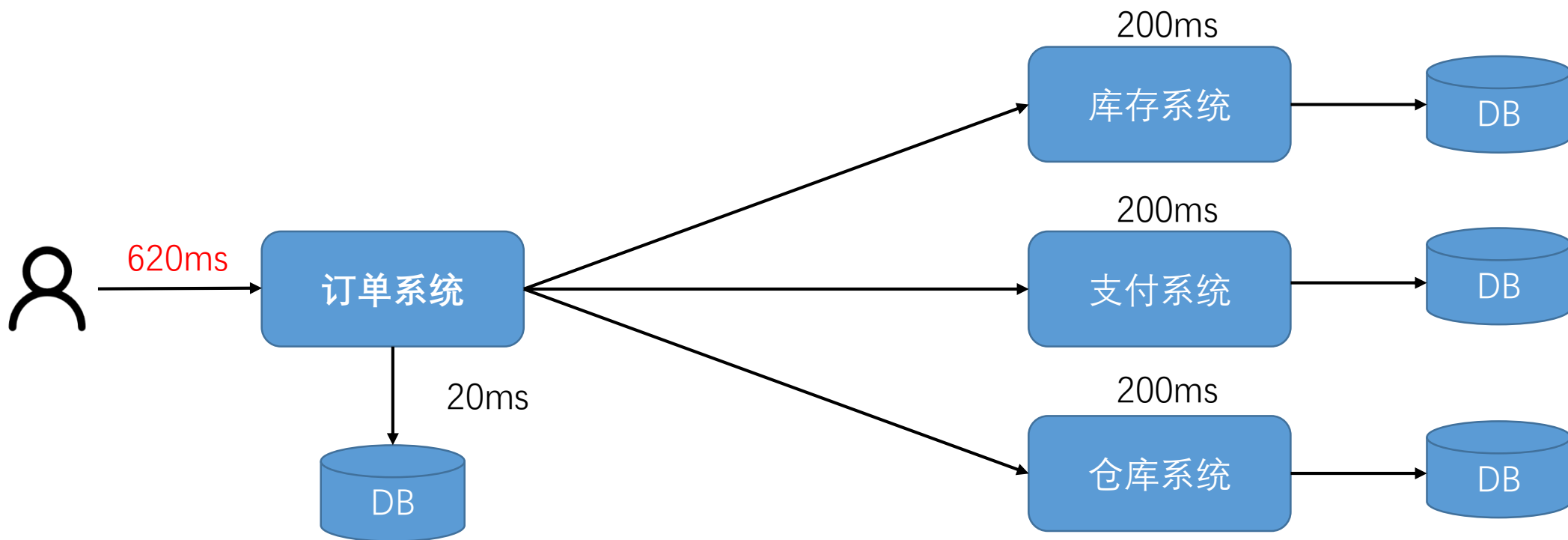
### ➤ 异步处理



# 一、MQ的介绍

## 2.MQ优点

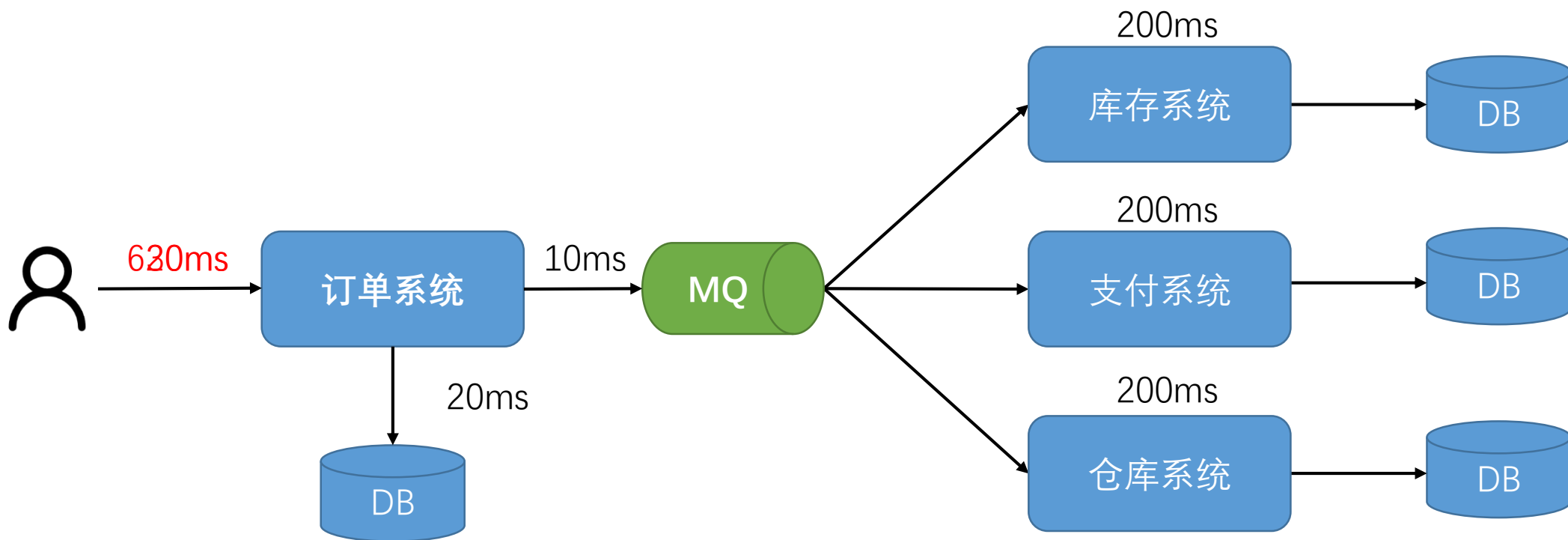
### ➤ 异步处理



# 一、MQ的介绍

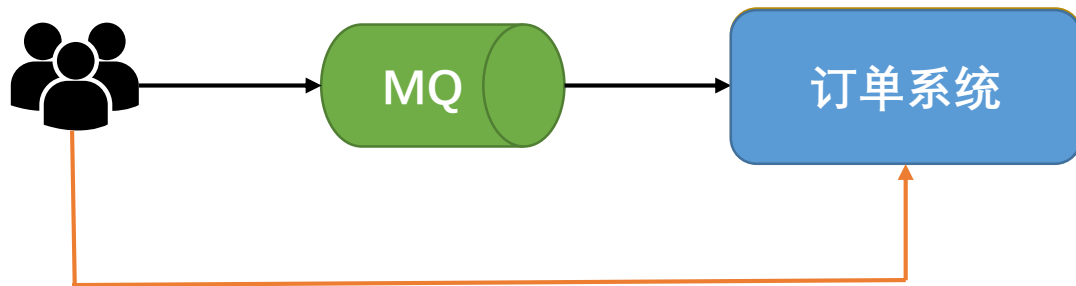
## 2.MQ优点

### ➤ 异步处理



# 一、MQ的介绍

## 2.MQ缺点



### ➤ 提高了系统复杂度，降低了系统稳定性和可用性

- 引入了第三方技术，增加了宕机的风险
- 如何保证系统高可用

### ➤ 带来了数据一致性问题

- 各系统间的调用增加了不确定性，如何保证最终一致性
- 怎么处理消息丢失的情况



# 一、MQ的介绍

## 2.MQ优缺点

小结:

### ■ 优点

- 流量消峰：避免流量高峰时期导致系统挂掉，提高系统稳定性
- 应用解耦：降低系统间依赖，提高系统容错性
- 异步处理：提速增效，提高用户体验感

### ■ 缺点

- 提高了系统复杂度，降低了系统稳定性和可用性
- 带来了数据一致性问题

# 一、MQ的介绍

## 3. 常见的MQ介绍

MQ名称	开发语言	单机吞吐量	消息延迟	功能特性	所属公司	协议
<b>Rabbit</b>	Erlang	万级	微秒	①：性能极其好并发能力强②：延时低，社区活跃，管理界面丰富	Rabbit	<b>AMQP</b> ，XMPP，SMTP，STOMP
<b>RocketMQ</b>	Java	十万级	毫秒级	①：阿里系产品，性能极佳②：金融互联网经过双11实战检验	Alibaba	自定义
<b>Kafka</b>	Scala&Java	十万级	毫秒级	①：大数据领域应用最多	Apache	自定义协议，社区封装了http协议支持

AMQP，即 Advanced Message Queuing Protocol（高级消息队列协议），是一个网络协议，是应用层协议的一个开放标准，为面向消息的中间件设计。2006年，AMQP 规范发布。

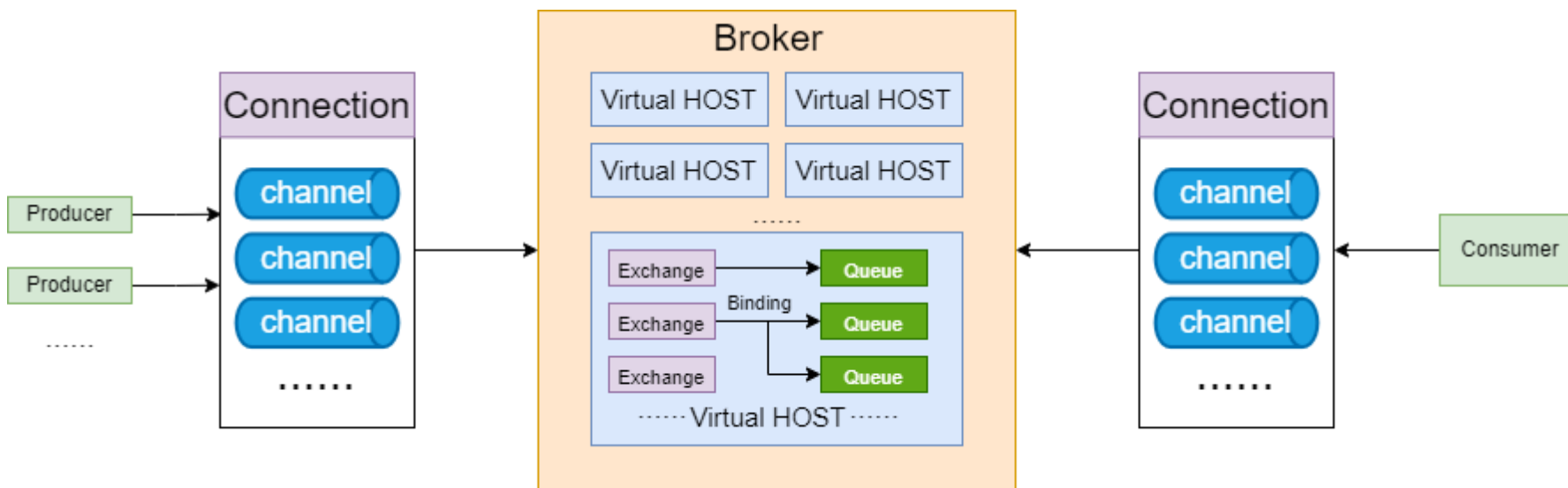
# 一、MQ的介绍

## 4.RabbitMQ概述

RabbitMQ是实现了高级消息队列协议（AMQP）的开源消息代理软件（亦称面向消息的中间件）。

官网地址：<https://www.rabbitmq.com/>

架构图：



# 一、MQ的介绍

## 5.RabbitMQ名词解释

- `Producer`:生产者, 发送消息的程序
- `Consumer`:消费者, 接收消息的程序
- `Broker`:接收和分发消息的应用, RabbitMQ Server 就是 Message Broker
- `Connection`:生产者, 消费者, Broker之间的TCP连接
- `Channel`:通信连接信道, 降低TCP连接开销
- `Virtual host`: 虚拟主机
- `Exchange`:交换机, message 到达 broker 先存放到交换机
- `Queue`:队列, 消费者从这里获取消息
- `Binding`:交换机与队列之间的虚拟连接, 用于消息分发

# 一、MQ的介绍

## 6. RabbitMQ工作模式

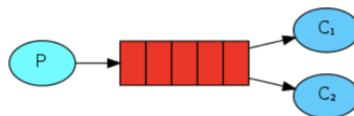
### 1 "Hello World!"

The simplest thing that does *something*



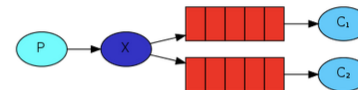
### 2 Work Queues

Distributing tasks among workers (the competing consumers pattern)



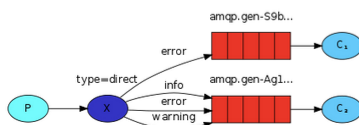
### 3 Publish/Subscribe

Sending messages to many consumers at once



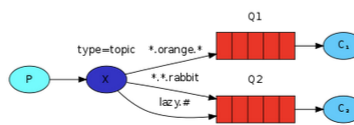
### 4 Routing

Receiving messages selectively



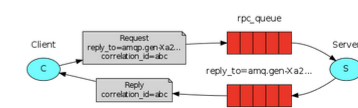
### 5 Topics

Receiving messages based on a pattern (topics)



### 6 RPC

Request/reply pattern  
example





# 一、MQ的介绍

## 7.RabbitMQ的安装

参考安装资料: [RabbitMQ单机版安装文档](#)



## RabbitMQ 快速入门

- 使用模式一完成快速入门

## 二、RabbitMQ快速入门

### 案例



#### 1. 使用简单模式完成生产者与消费者之间的通信

- 通过生产者，发送：hello rabbitMQ 给消费者，完成服务间的通信





## 二、RabbitMQ快速入门



### 实现思路

1. 通过Maven构建生产者与消费者项目
2. 添加依赖
3. 创建生产者与消费者
4. 生产者发送消息
5. 消费者接收消息



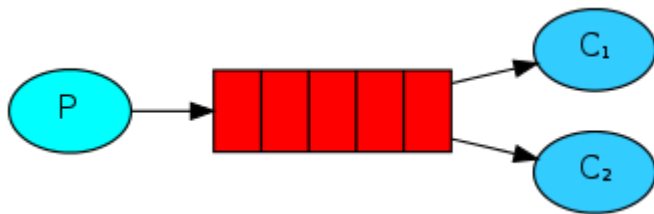
## RabbitMQ 的工作模式

- Work queues **工作队列**
- Pub/Sub **订阅模式**
- Routing **路由模式**
- Topics **通配符模式**

### 三、RabbitMQ工作模式

#### 1. Work Queues 工作队列

- ◆ 工作队列与简单模式相比，多了一个或一些消费端，多个消费端共同消费同一个队列中的消息
- ◆ 使用场景：对于消息生产速度大于消费速度场景，可以增加消费者减少单个消费者压力



## 三、RabbitMQ工作模式

### 1.Work Queues 工作队列

#### 小结

- ◆ 分发机制：轮询分发机制
- ◆ 应用场景：同一条消息，在多个消费者之间只能有一个消费，应用于只需要单节点消费的场景
  - 发送验证码
  - 发送生日提醒

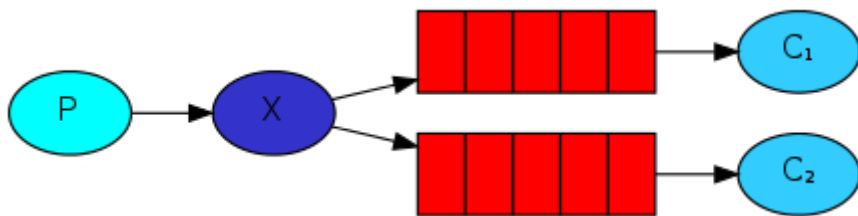
## 三、RabbitMQ工作模式

### 2. Publish/Subscribe 发布订阅模式

在订阅模型中，多了一个Exchange 角色：

Exchange：交换机（X）。接收生产者发送的消息；处理投递消息，例如递交给某个特别队列、递交给所有队列、或是将消息丢弃。具体操作根据交换机类型来定义：

- Fanout：广播，将消息交给所有绑定到交换机的队列
- Direct：定向，把消息交给符合指定routing key 的队列
- Topic：通配符，把消息交给符合routing pattern（路由模式） 的队列



■ 注意：交换机并不具备存储消息的能力，如果没有队列绑定，则消息会丢失

## 三、RabbitMQ工作模式

### 2. Publish/Subscribe 发布订阅模式

#### 小结

- ◆ 交换机需要与队列进行绑定，绑定之后；一个消息可以被多个消费者都收到。
- ◆ 发布订阅模式与工作队列模式的区别：
  - 工作队列模式不用定义交换机，而发布/订阅模式需要定义交换机
  - 发布/订阅模式的生产方是面向交换机发送消息，工作队列模式的生产方是面向队列发送消息
  - 发布/订阅模式需要手动设置队列和交换机的绑定

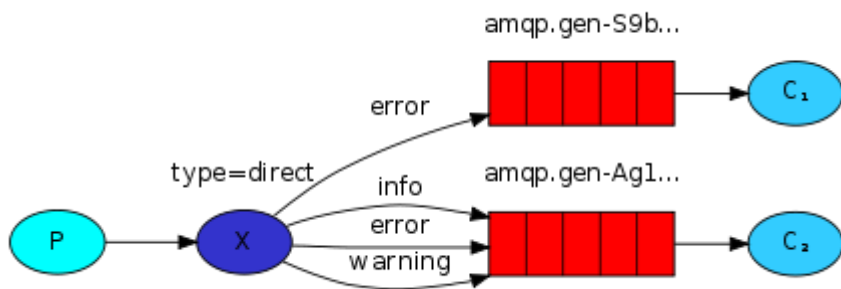
## 三、RabbitMQ工作模式

### 3. Routing 路由模式

RoutingKey 编写规则：由一个或多个单词组成，多个单词用.分割，比如order.save

在路由模型中：

- ◆ 队列与交换机的绑定需要指定路由RoutingKey
- ◆ 生产者在向 Exchange 发送消息时，也必须指定消息的 RoutingKey
- ◆ 交换机分发的规则根据 RoutingKey 来分发给相应队列



## 三、RabbitMQ工作模式

### 3. Routing 路由模式

#### 小结

- ◆ 生产者发送消息时要指定RoutingKey
- ◆ 队列与交换机绑定也需要RoutingKey



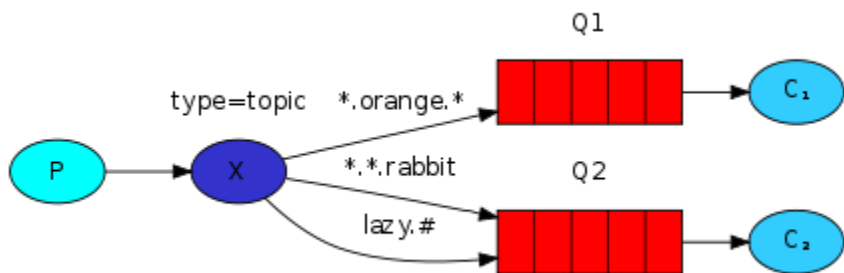
### 三、RabbitMQ工作模式

#### 4. Topics 通配符模式

Topics模式可以通过通配符配置 RoutingKey, 会更加灵活

通配符规则:

- ◆ # 匹配一个或多个词。如`order.#` 可以匹配 `order.save` `order.product.update...`
- ◆ \* 匹配一个单词。如`order.*` 可以匹配 `order.save`



## 三、RabbitMQ工作模式

### 4. Topics 通配符模式

#### 小结

- ◆ Topic 通配符模式可以实现 Pub/Sub 发布与订阅模式和 Routing 路由模式的功能
- ◆ Topic 通配符模式在配置routing key 的时候可以使用通配符，更加灵活好用。

### 三、RabbitMQ工作模式



## 总结

1. 简单模式 HelloWorld。一个生产者、一个消费者，不需要设置交换机使用默认的交换机
2. 工作队列模式 Work Queue 。一个生产者、多个消费者（排它关系），不需要设置交换机，使用默认的交换机
3. 发布订阅模式 Publish/subscribe。设置类型为 fanout 的交换机同时让交换机和队列绑定，交换机会将消息发送到绑定的队列
4. 路由模式 Routing。设置 direct 交换机，并且指定 routing key，交换机根据 routing key 将消息发送到对应的队列
5. 通配符模式 Topic。设置类型为 topic 的交换机，交换机和队列进行绑定，并且指定通配符方式的 routing key，交换机根据通配符分发消息到队列



## SpringBoot 整合 RabbitMQ

- SpringBoot整合RabbitMQ
- 案例讲解

## 四、SpringBoot 整合 RabbitMQ

### 1. 整合

#### 1.1 创建生产者服务

- 发布普通消息
- 发布路由消息

#### 1.2 创建消费者服务

## 四、SpringBoot 整合 RabbitMQ

### 2. 案例

1.1 创建订单服务项目

1.2 创建库存服务项目

1.3 创建支付服务项目

1.3 调用订单服务接口，通过消息队列通知扣减库存

1.4 调用支付服务接口，添加支付订单记录



月薪上万如此简单