Microservice Plumbing With RabbitMQ

ASYNC MESSAGING FOR MICROSERVICES

@jsonrow
jasonrowe@gmail.com
https://github.com/JasonRowe

About Me

Past Async Messaging Experience

- MSMQ
- NServiceBus
- MassTransit

Current Tech Stack

- .NET Core
- RabbitMQ
- Docker
- Vue.JS

jasonrowe.com

PROTOLABS

RabbitMQ Overview

What is RabbitMQ?

What is AMQP?

Advanced Message Queue Protocol

Open internet protocol for business messaging

Binary wire-level protocol

AMQP was designed with the following main characteristics as goals:

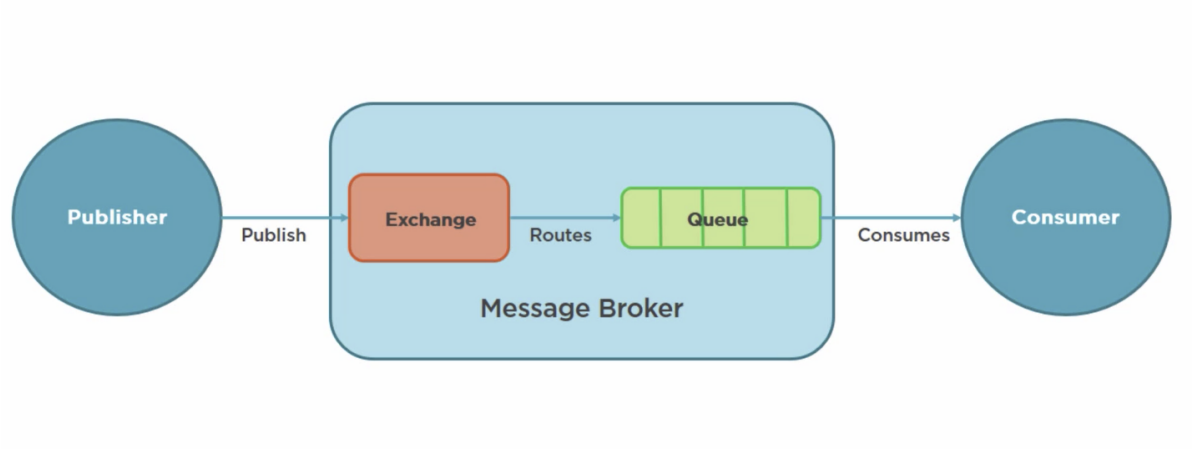
Security

Reliability

Interoperability

Standard

Open



AMQP Model

Exchange -> Binding ->[Queue]

Exchanges

Receives and routes messages

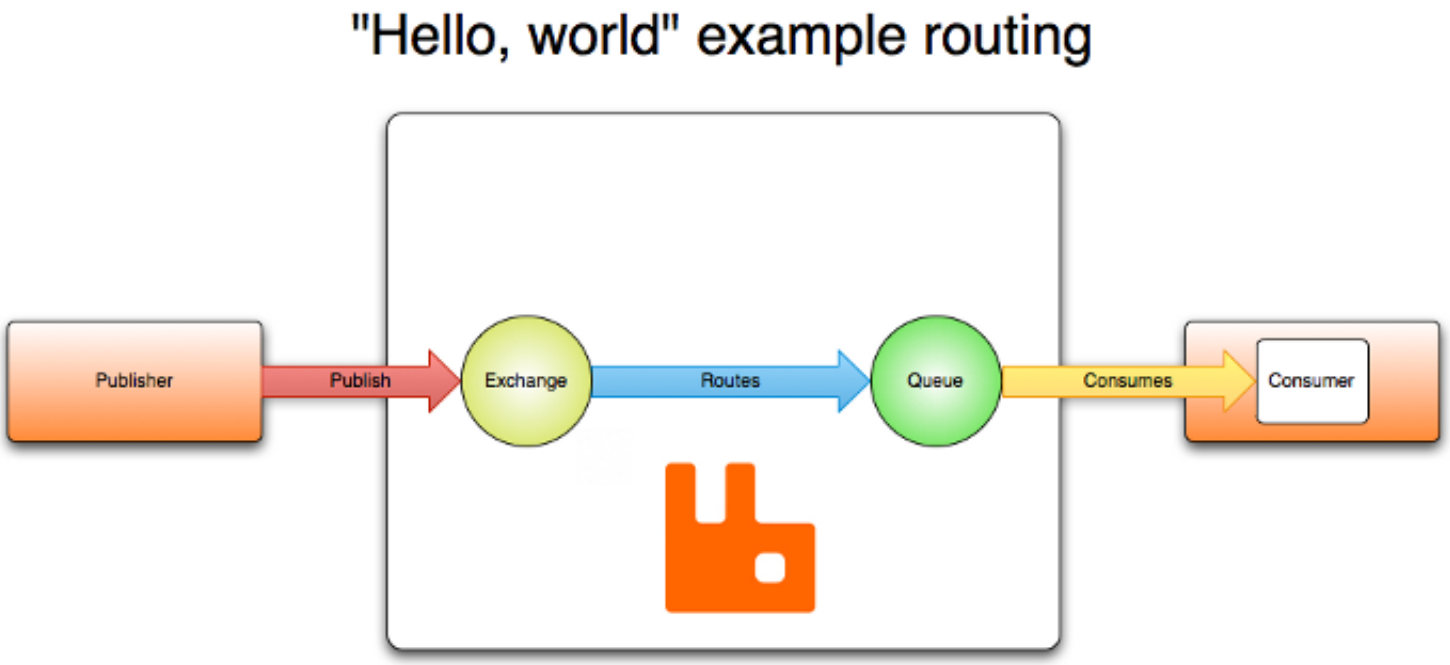
Binding

Defines the relationship between exchange and queue

Queues

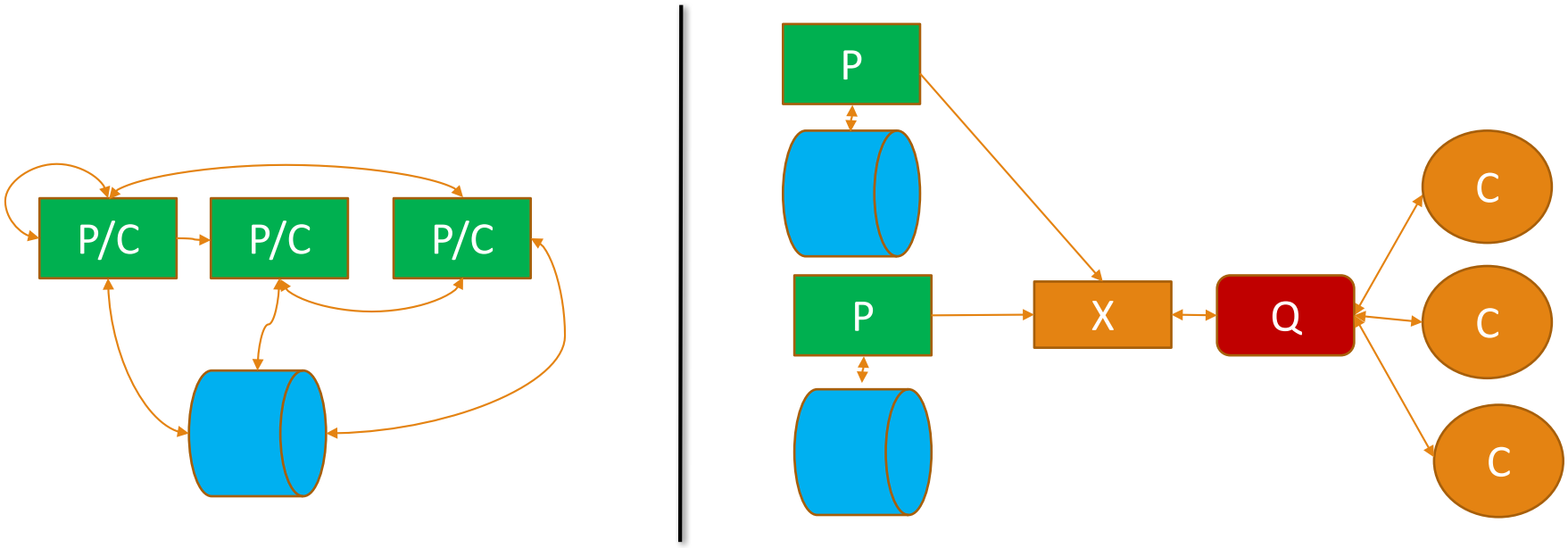
Stores messages until they are processed

RabbitMQ Hello, world.



RabbitMQ – changing a monolith

Big Challenge - changing the communication pattern.



Message Durability

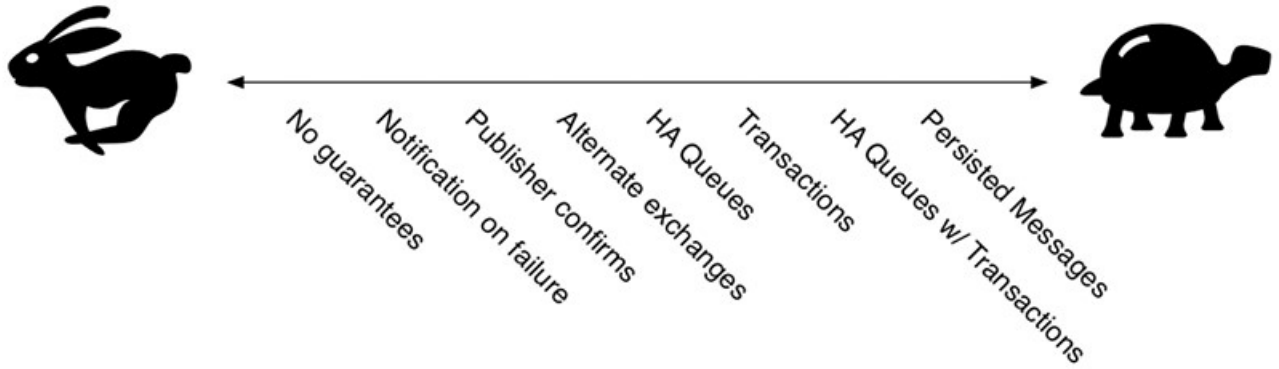
Publisher confirms

Durability exchanges and queues

message acknowledgements

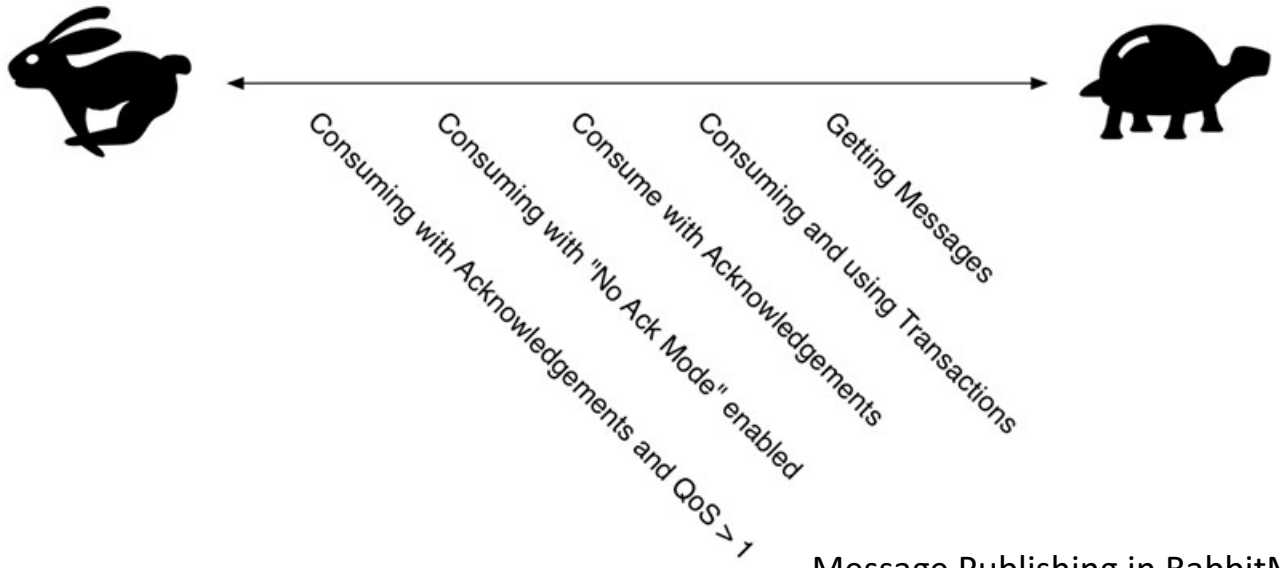
Delivery and processing confirmation

Message Publishing



Message Publishing in RabbitMQ, by Gavin M Roy

Receiving Messages



Message Publishing in RabbitMQ, by Gavin M Roy

High Availability

Enable Publisher confirms

Use Durable queues, exchanges

Extensions to consider - consistently hash and sharding

Do not enable HiPE

Prefetch configuration

Multi nodes with HA policy

High Performance

Enable HiPE

Disable LazyQueue

Short queues

Transient Messages

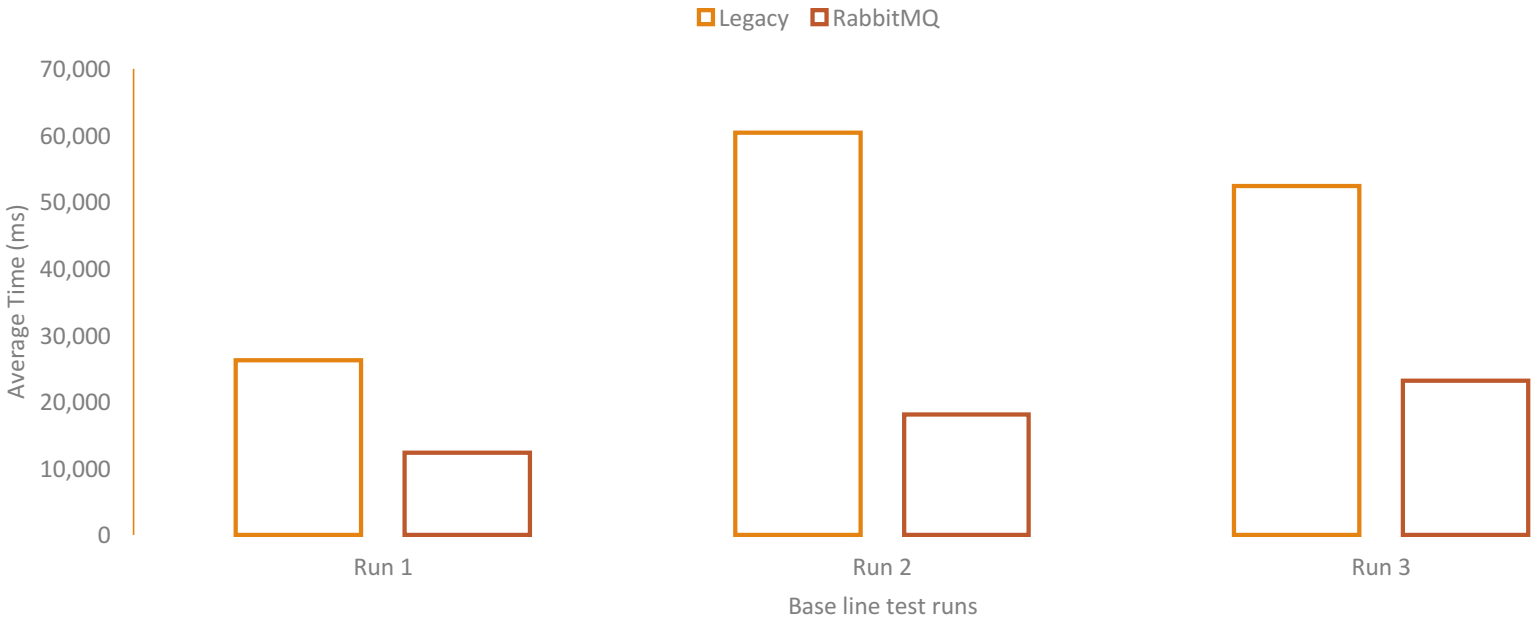
Disable HA

My Recent RabbitMQ success story.

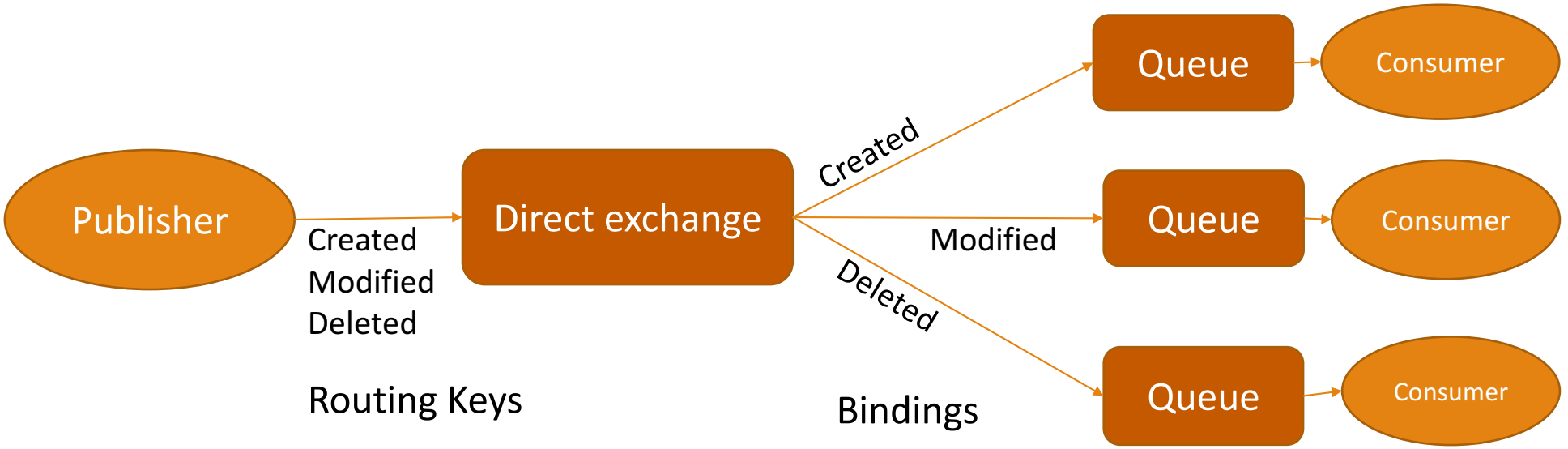
Refining a services boundaries and switching to RabbitMQ

@JSONROW

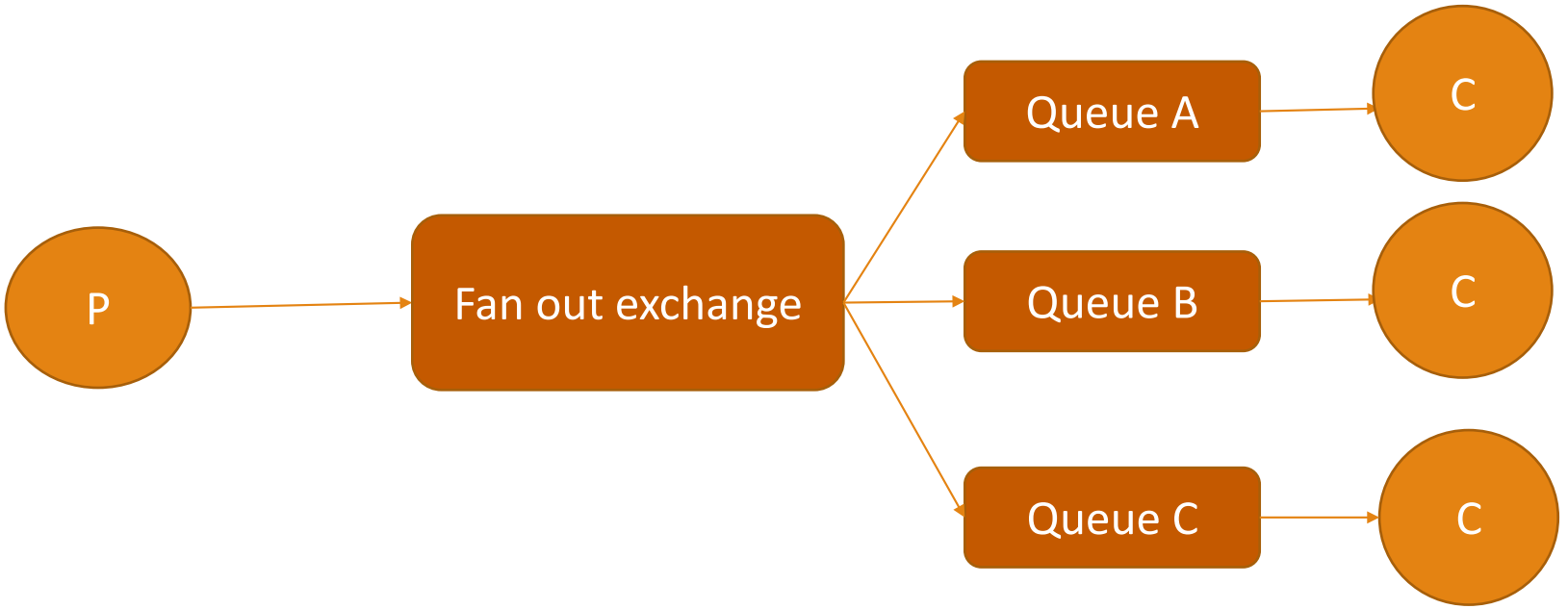
Recent RabbitMQ success story.



Direct Exchange



Fan out example

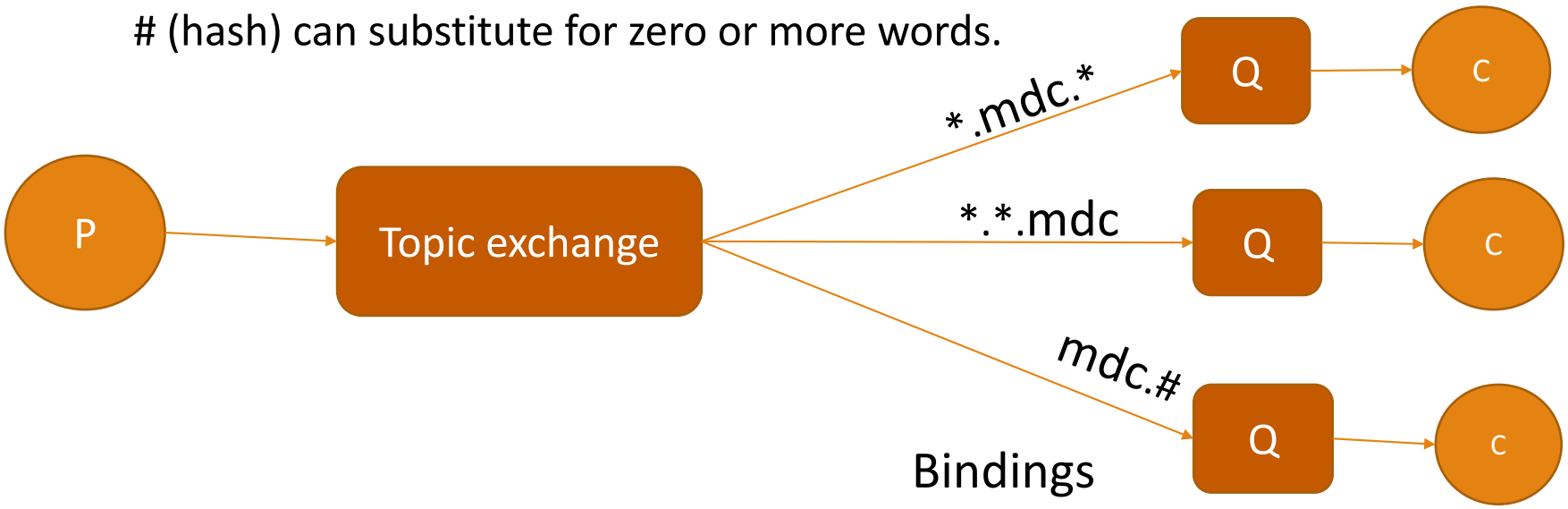


Topic exchange example

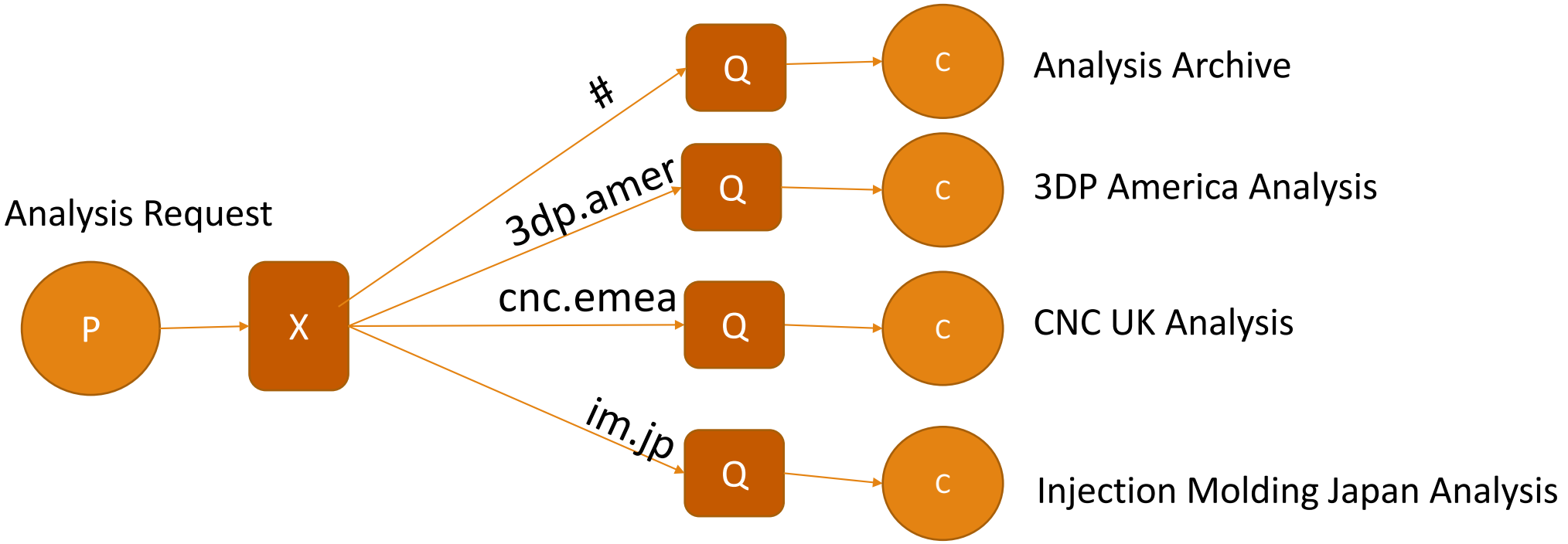
Wild card routing by **routing key**

* (star) can substitute for exactly one word.

(hash) can substitute for zero or more words.



Analysis Request Example



Competing Consumers

Rabbitmq Use Cases

Good

Event Stream

Connecting apps new and old written in different languages

When you are uncertain RabbitMQ is a certain choice

multi protocol and patterns

Federation between data centers

Bad

Large binary videos and images

Use as a data store large queues

Lessons Learned

Major RabbitMQ upgrades and Erlang upgrades comes with some downtime even on multi-node clusters. (blue green)

Connections - don't open new connections with every message.

Don't let queues grow 10,000 messages is too much limit with TTL or max size

Send persistent messages and use durable queues and exchange

Use publisher confirms when it makes sense

Take a look at prefetch and adjust if needed

Monitoring

Queue Length

Missing Consumers

How long messages have been in queue

CPU, RAM, Disk

https://www.rabbitmq.com/production-checklist.html

Summary

- RabbitMQ is a great solution for connecting async messaging for Microservices
- RabbitMQ provides options for high performance and high reliability
- Messaging doesn't need to be hard, it can be boring, and boring is good.
- It's a general purpose messaging system that may be the right choice to help breakup your monolith or connect our microservices

Questions?