RWTH AACHEN UNIVERSITY

RabbitMQ

Messaging that just works

Group DAFOT

Agenda

- Motivation
- Introduction to Advanced Message Queuing Protocol (AMQP) and RabbitMQ
- Overview of the AMQP Model
- Live coding session

Typical scenario

One bad day

User:

I don't want to wait until your app processes my images!

Product owner:

Can we also notify the user's friends when he uploads a new image?

I forgot to mention, for tomorrow...

Developer:

Damn! Our PHP module is too slow.

Your reaction

Introduction to AMQP and RabbitMQ

- Open Source message broker/ messaging middleware

- Highly scalable and stable environment for messaging with many features and patterns
- Supports AMQP, the emerging standard internet protocol for business messaging
- Supports multiple platforms and languages (e.g. Java, C#/.NET, Ruby, Python, C++, JavaScript, PHP, Go, ...)

Introduction to AMQP and RabbitMQ

What is AMQP?

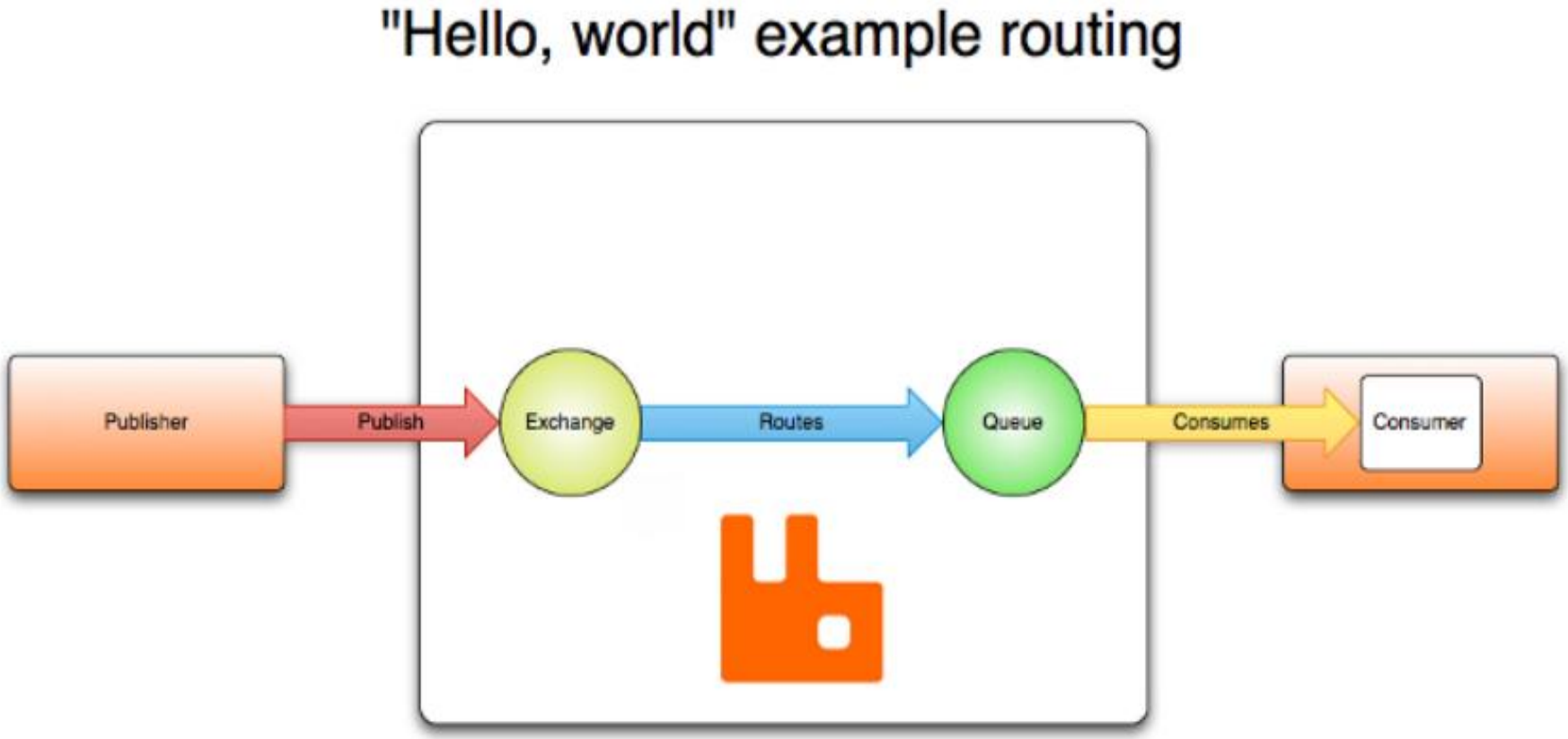
AMQP (Advanced Message Queuing Protocol) is a networking protocol that enables conforming client applications to communicate with conforming messaging middleware brokers.

Brokers and Their Role

Messaging brokers receive messages from publishers (applications that publish them, also known as producers) and route them to consumers (applications that process them).

Overview of the AMQP Model

AMQP entities: Queues, Exchanges, Bindings



https://www.rabbitmq.com/tutorials/amqp-concepts.html

Overview of the AMQP Model

AMQP is a Programmable Protocol

AMQP entities and routing schemes are defined by applications themselves, not a broker administrator.

Applications declare the AMQP entities that they need, define necessary routing schemes and may choose to delete AMQP entities when they are no longer used.

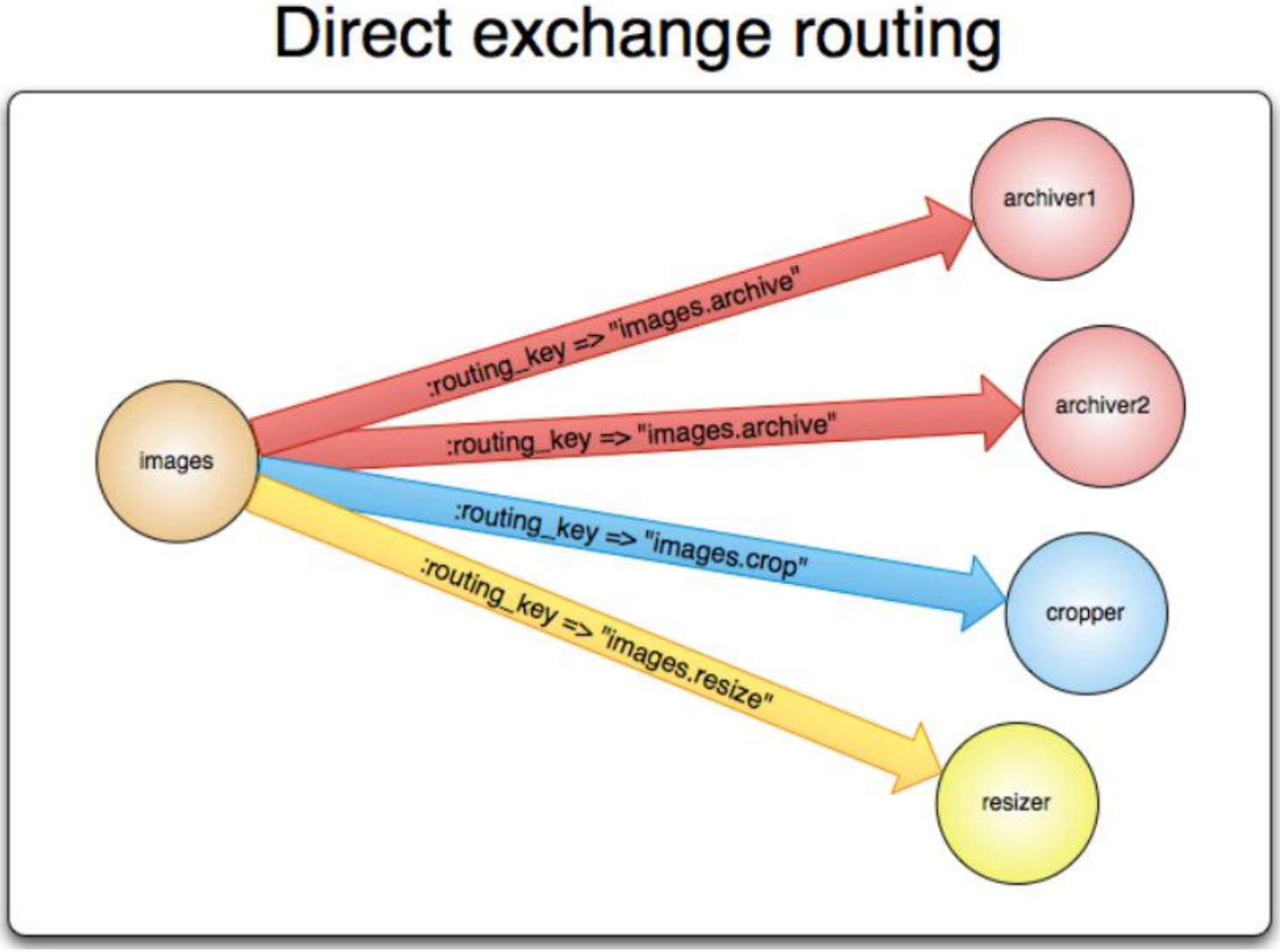
Overview of the AMQP Model

- **Exchanges and Exchange Types**

Exchanges are AMQP entities where messages are sent. Exchanges take a message and route it into zero or more queues. The routing algorithm used depends on the exchange type and rules called bindings.

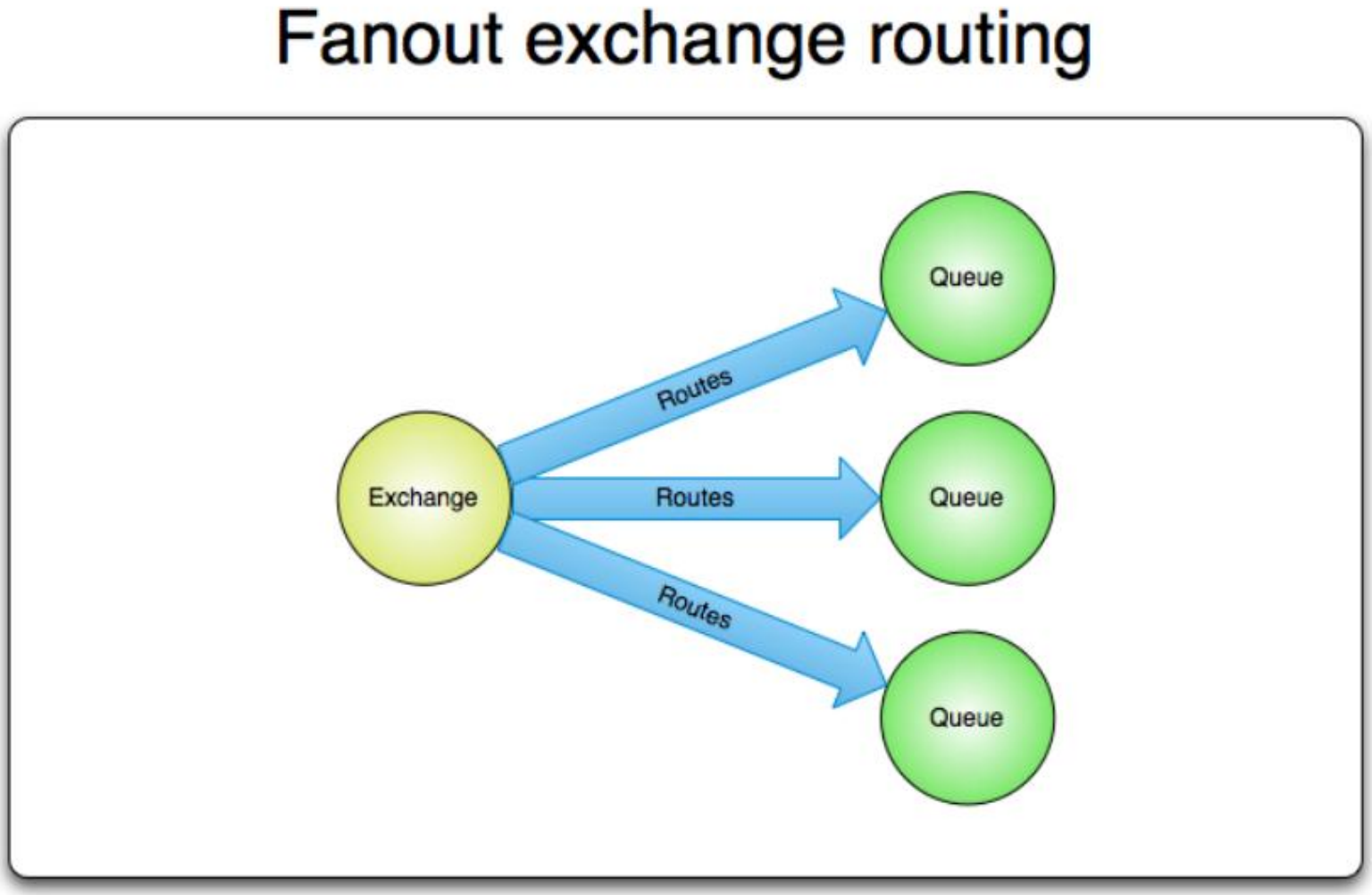
- **Types:**
  - Direct exchange
  - Fanout exchange
  - Topic exchange
  - Headers exchange

Overview of the AMQP Model



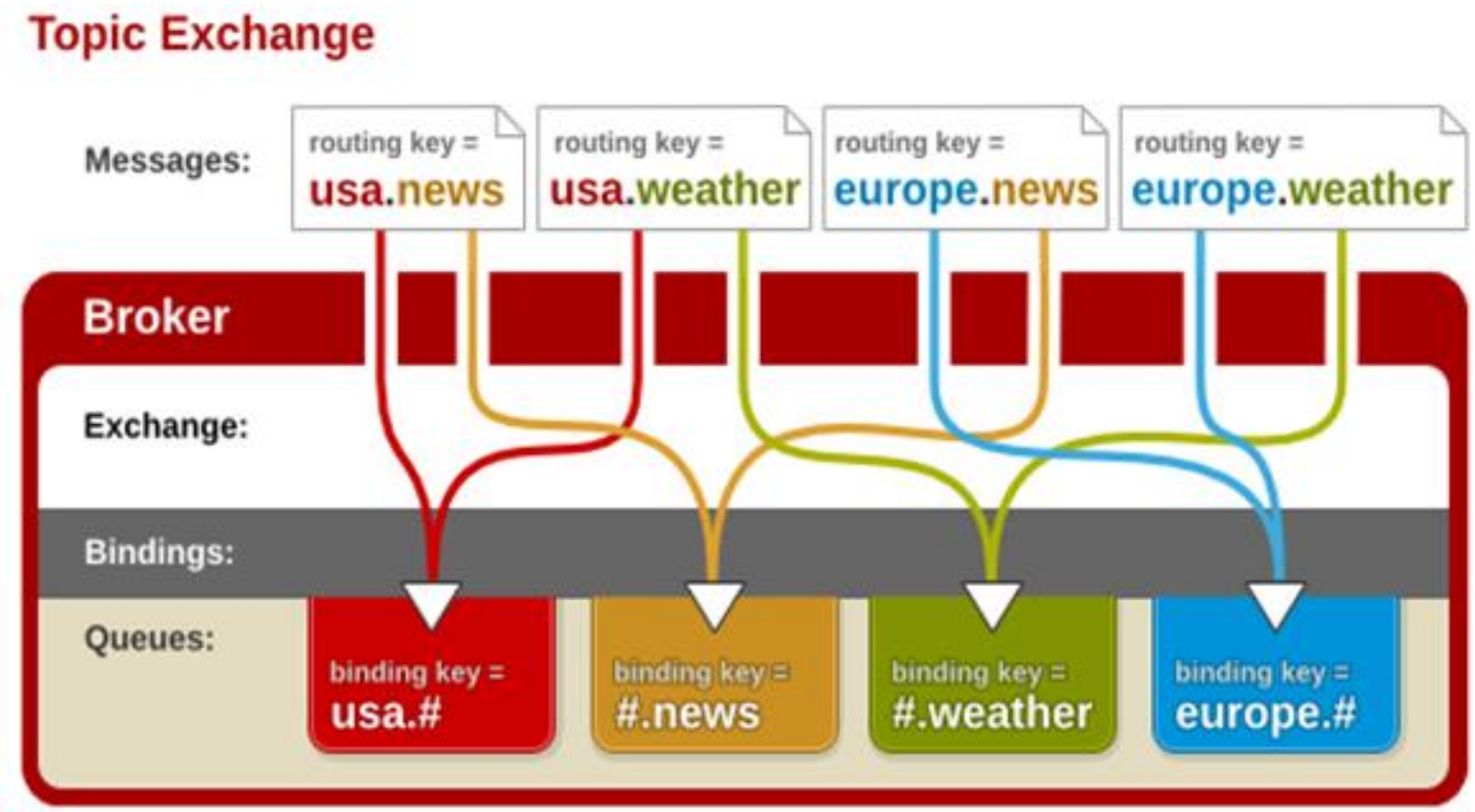
https://www.rabbitmq.com/tutorials/amqp-concepts.html

Overview of the AMQP Model



https://www.rabbitmq.com/tutorials/amqp-concepts.html

Overview of the AMQP Model



http://www.slideshare.net/rahula24/amqp-basic

Overview of the AMQP Model

- **Queues**

  Store messages that are consumed by applications.

- **Bindings**

  Bindings are rules that exchanges use (among other things) to route messages to queues.

Overview of the AMQP Model

Consumers

Storing messages in queues is useless unless applications can consume them. In the AMQP Model, there are two ways for applications to do this:

- Have messages delivered to them ("push API")
- Fetch messages as needed ("pull API")

Overview of the AMQP Model

Message Acknowledgements

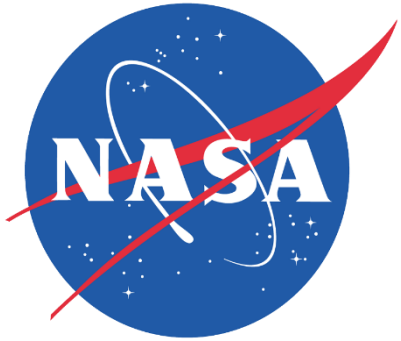
When should the AMQP broker remove messages from queues?

- After broker sends a message to an application
- After the application sends back an acknowledgement

Advantages of RabbitMQ

- Open-sourced project
- Reliability
- Flexible Routing
- Clustering
- Highly Available Queues
- Many clients (Java, C#/.Net, Ruby, Python, Go, …)
- Management UI
- Tracing
- Plugin System

Who uses RabbitMQ?



Google

at&t

redhat.

vmware®

Deutsche Börse Group

openstack™
CLOUD SOFTWARE

One bad day

User:

I don't want to wait until your app processes my images!

Product owner:

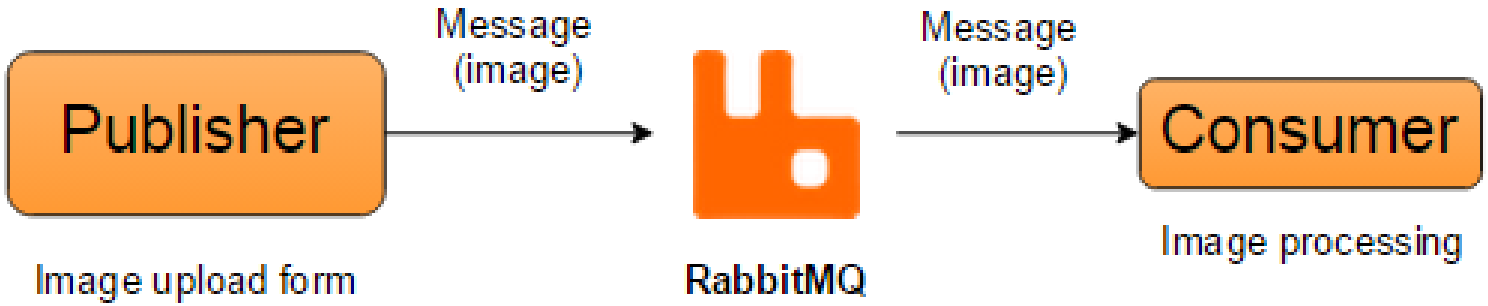
Can we also notify the user's friends when he uploads a new image?

I forgot to mention, for tomorrow...

Developer:

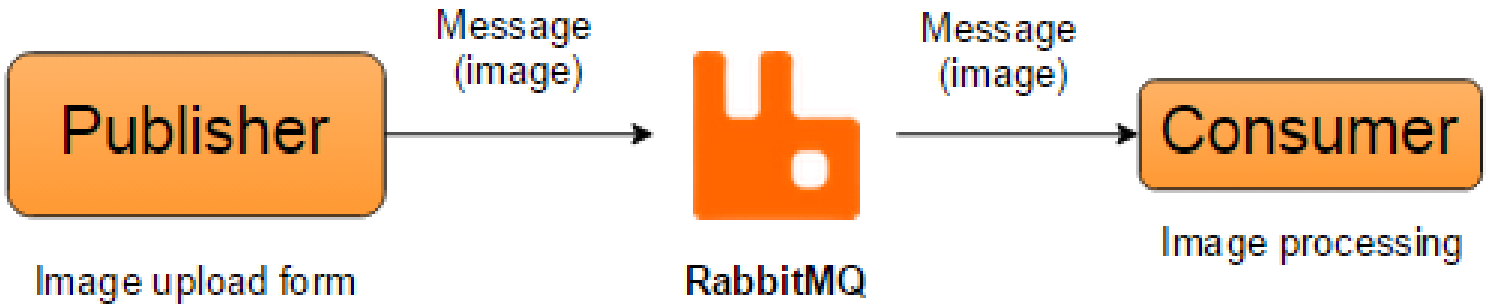
Damn! Our PHP module is too slow.

Possible flexible architecture



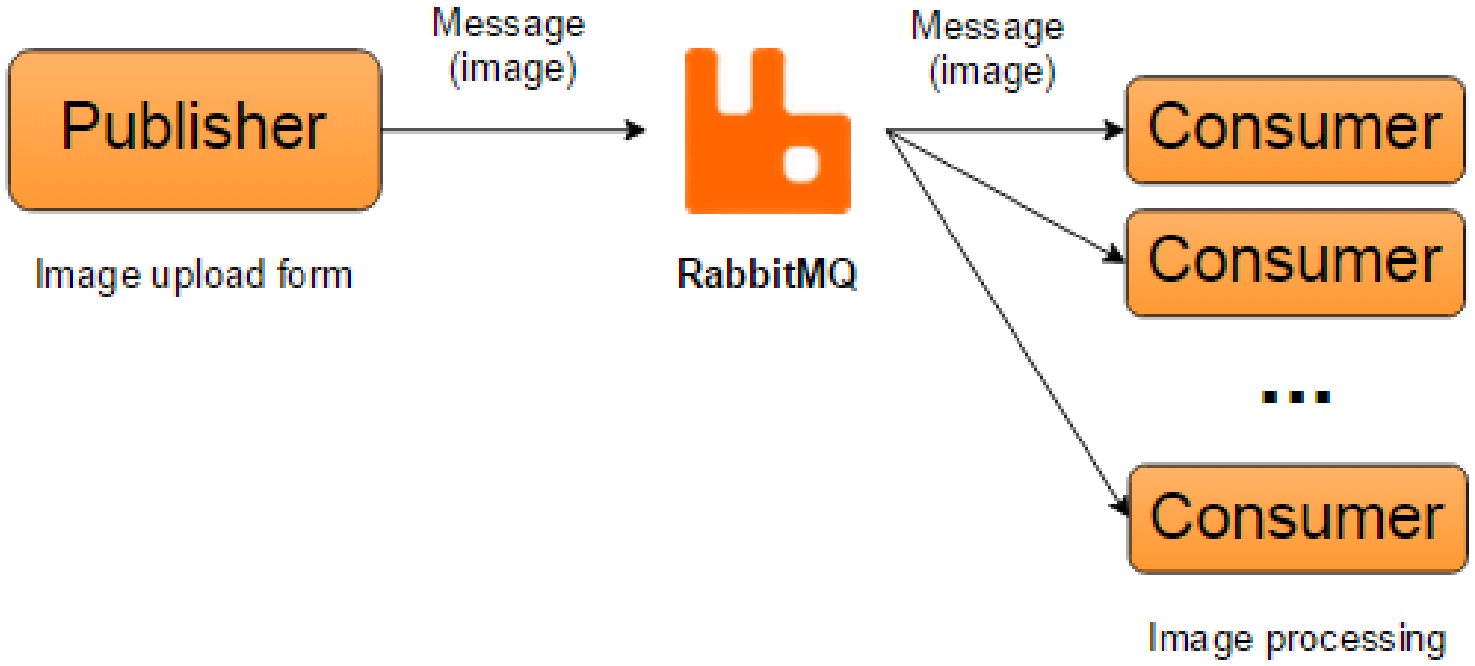
Possible flexible architecture

User is happy

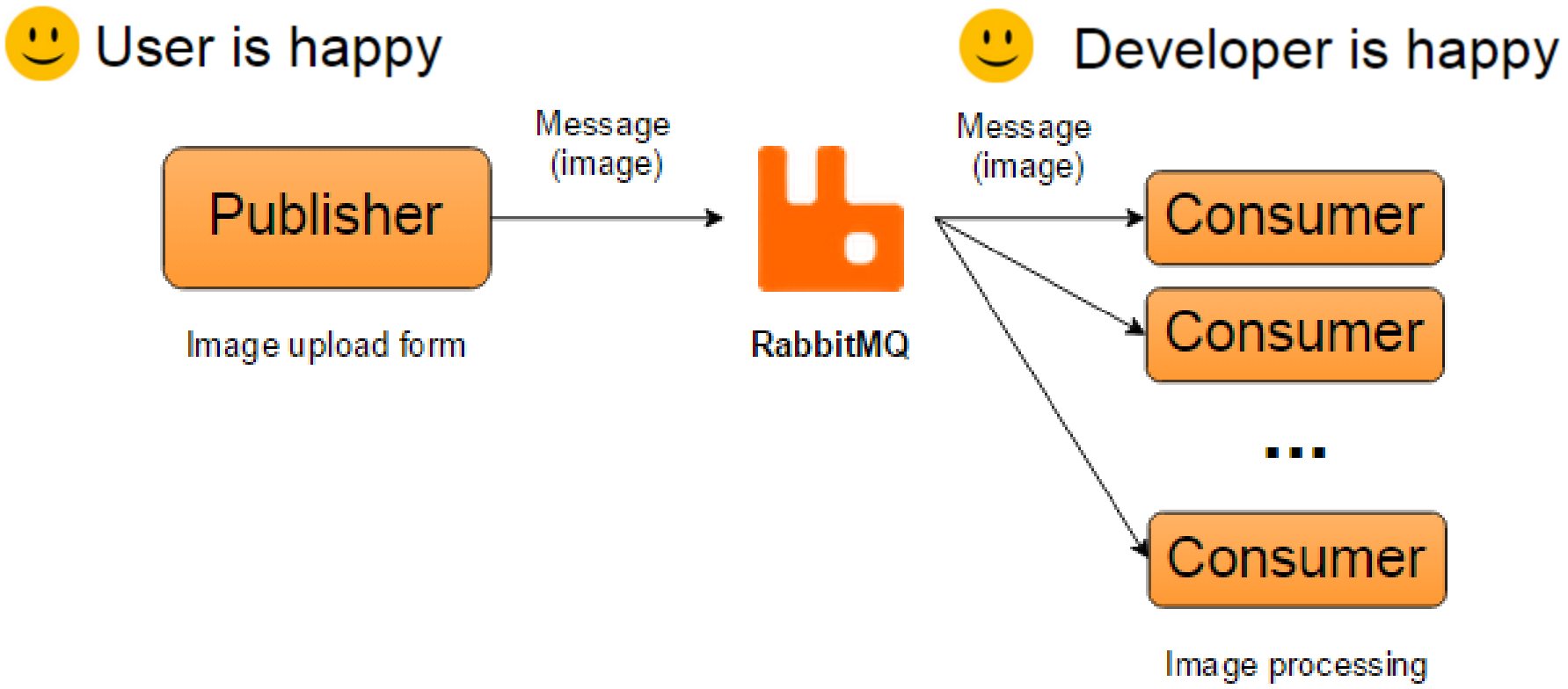


Possible flexible architecture

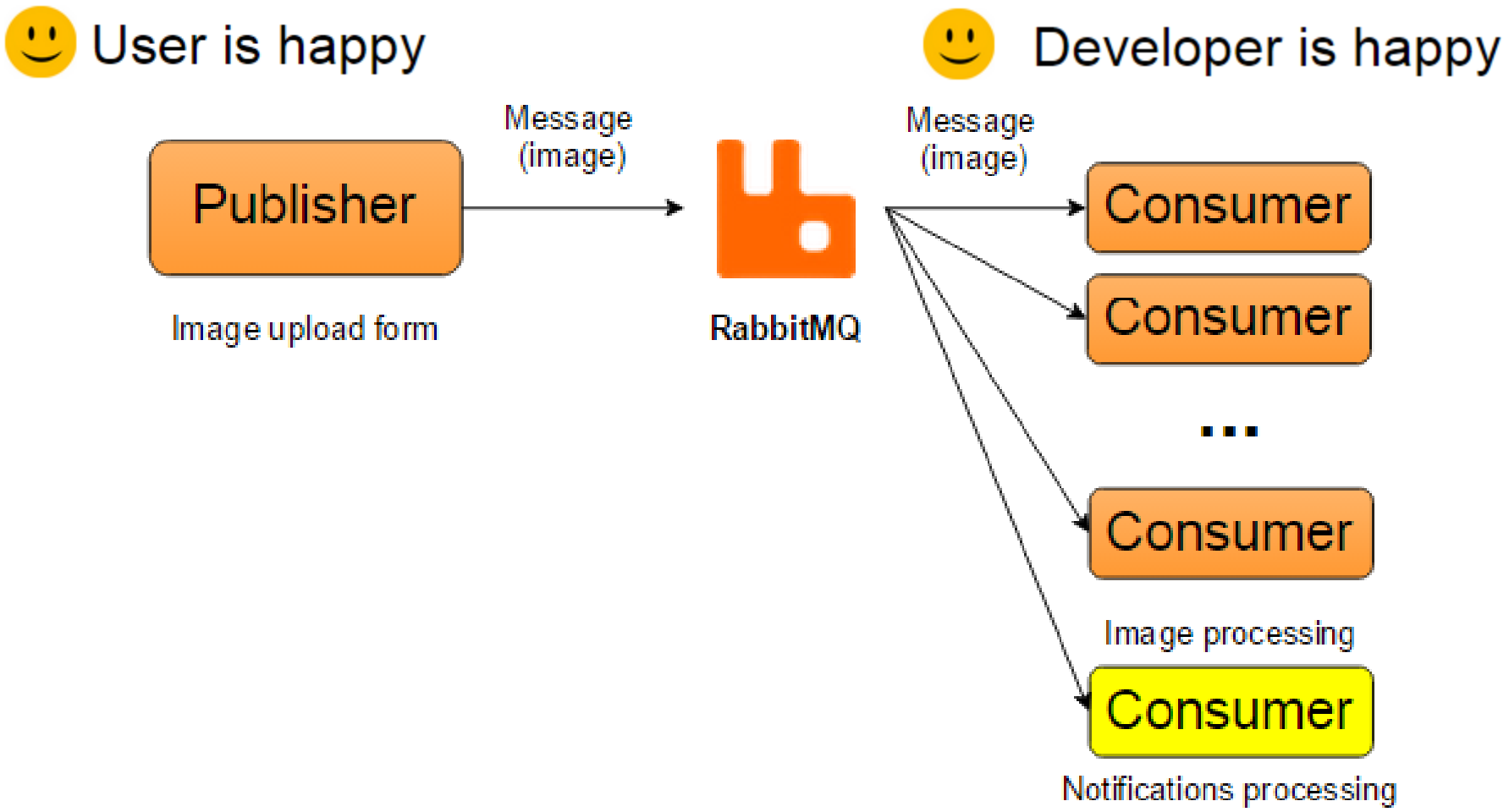
User is happy



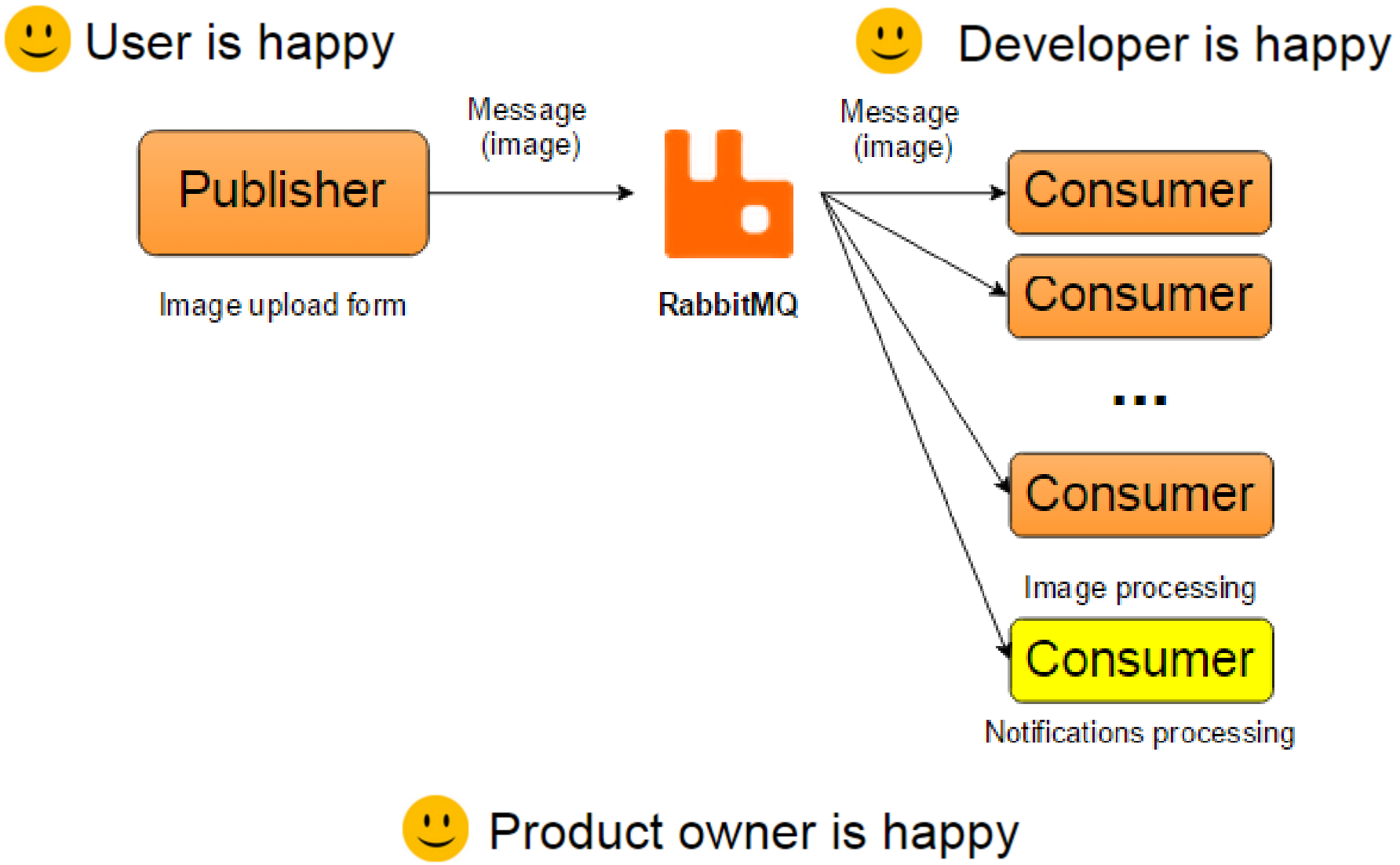
Possible flexible architecture



Possible flexible architecture



Possible flexible architecture



Wake up!

Time for demo

Questions?

Don’t implement messaging yourself. You can use RabbitMQ