FreshBooks

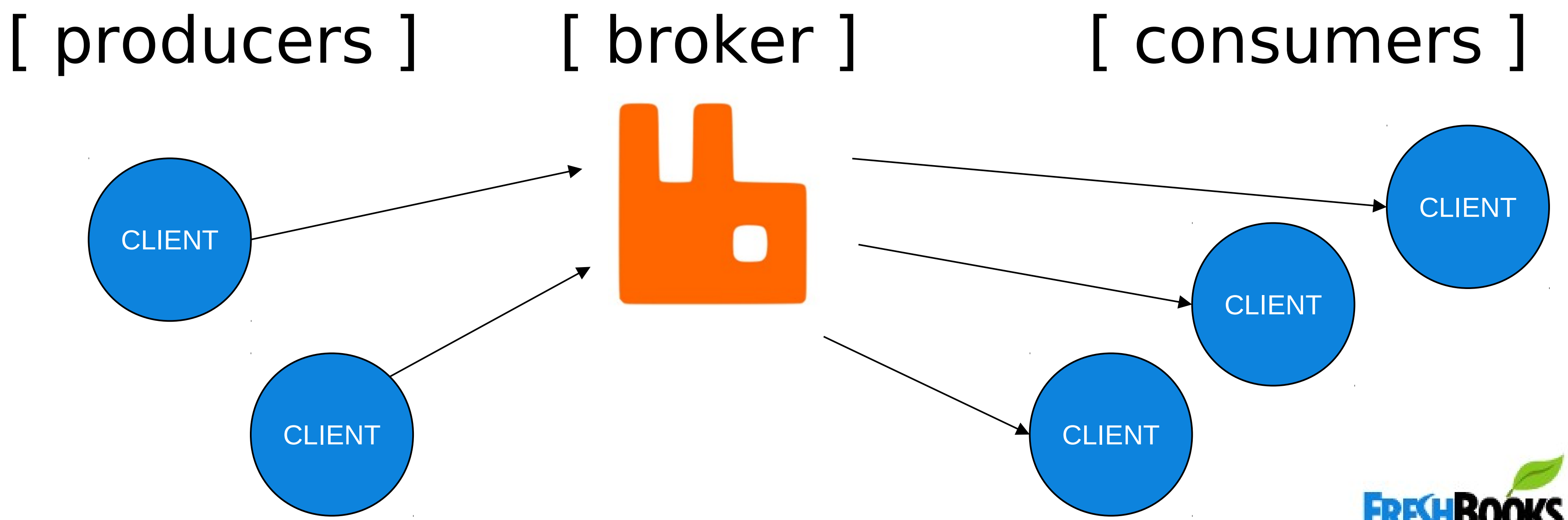
painless billing

RabbitMQ

Introduction

What is RabbitMQ?

- It's a messaging server
- It talks AMQP (Advanced Message Queueing Protocol)
- Open source, written in Erlang

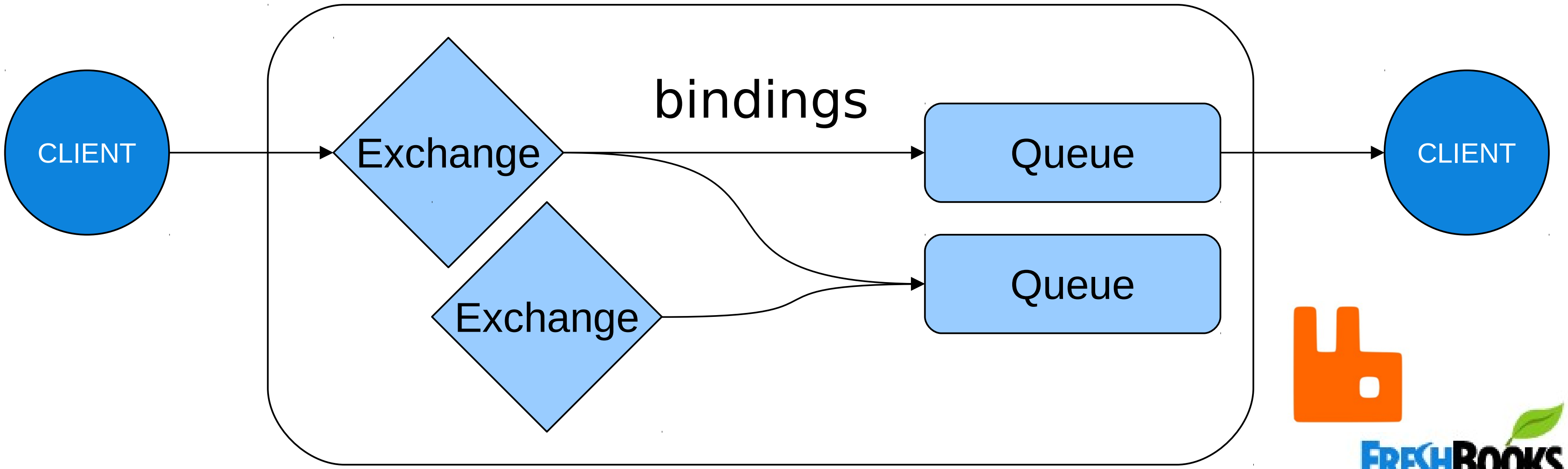


Benefits

- Push notifications
- Distribute load on multiple workers
- Asynchronicity
  - Performance (asynchronous I/O)
  - Consumer fault tolerance (buffering)
- Decouple publishers and consumers
  - Time, space, language, concerns
- Built-in clustering
  - **Webscale**!

AMQP entities

- Producer publish a message to an **exchange**
- Exchange decide which **queue**(s) to copy the message using the **bindings** rules and the message headers
- Broker push or consumer pull the message
- Consumer acknowledge reception



Pika – python client library

- Example1 connecting to the server and opening a channel
  - Continuation passing style
  - The channel is the handle we're going to use to create exchanges, bindings, queues and publish/consume messages

Pika – python client library

- Example2: create the Exchange
  - Name
  - Type: direct vs fanout vs topic
  - Durable (survive broker restart?)
  - Auto-delete (when queue are done?)

Pika – python client library

- Example3: publish a message
  - exchange
  - routing_key
  - Body
  - Properties (delivery_mode, ...)

Pika – python client library

- Example 4 consumer: create a Queue
  - Exchange declaration should match producer
  - Name
  - Durable (survive broker restart?)
  - Exclusive
  - Auto-delete (when last consumer unsubscribe)
  - queue_bind()
  - basic_ack()

Pika – python client library

- 2 more things:
  - Messages need delivery_mode=2 **and** queue/exchange need durable=True if you don't want to loose data
  - Virtualhosts: for isolating multiple environments (users, queues, exchanges, ...)

Questions?

FRESHBOOKS painless billing