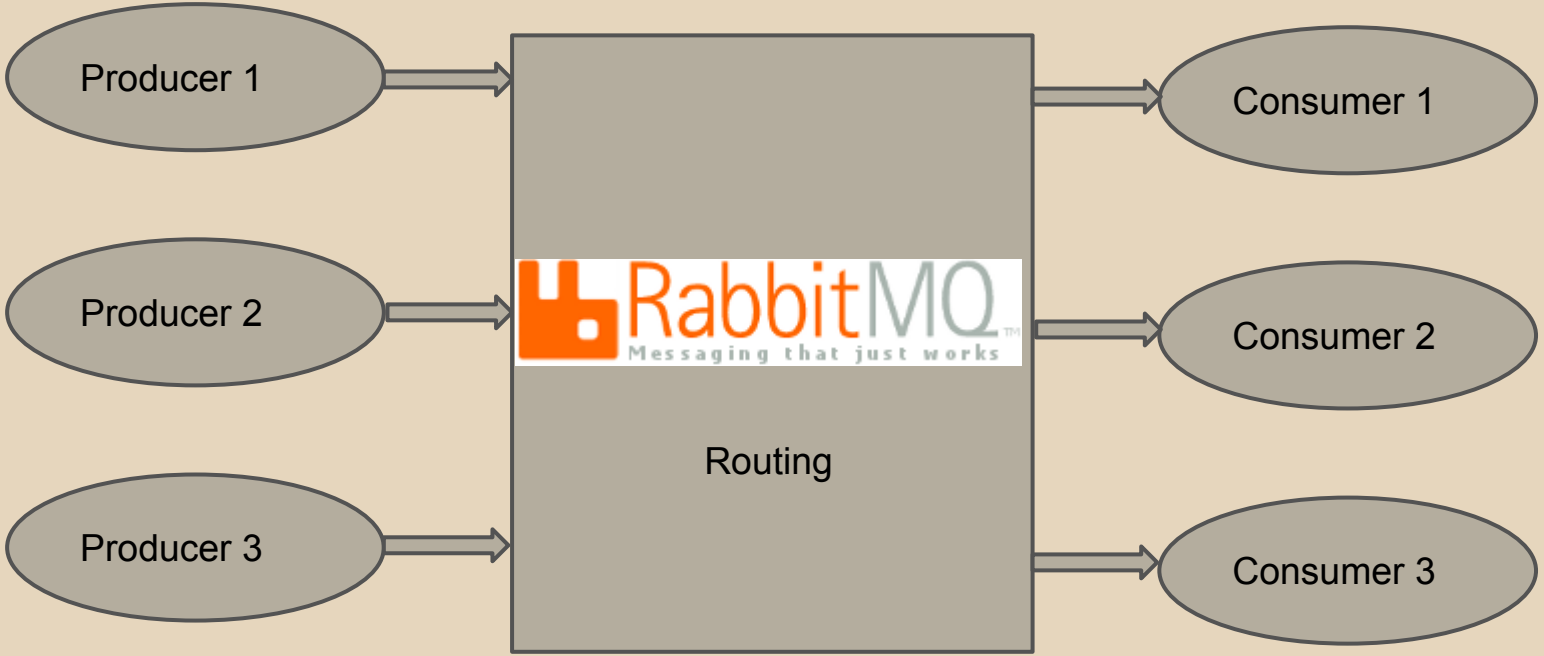
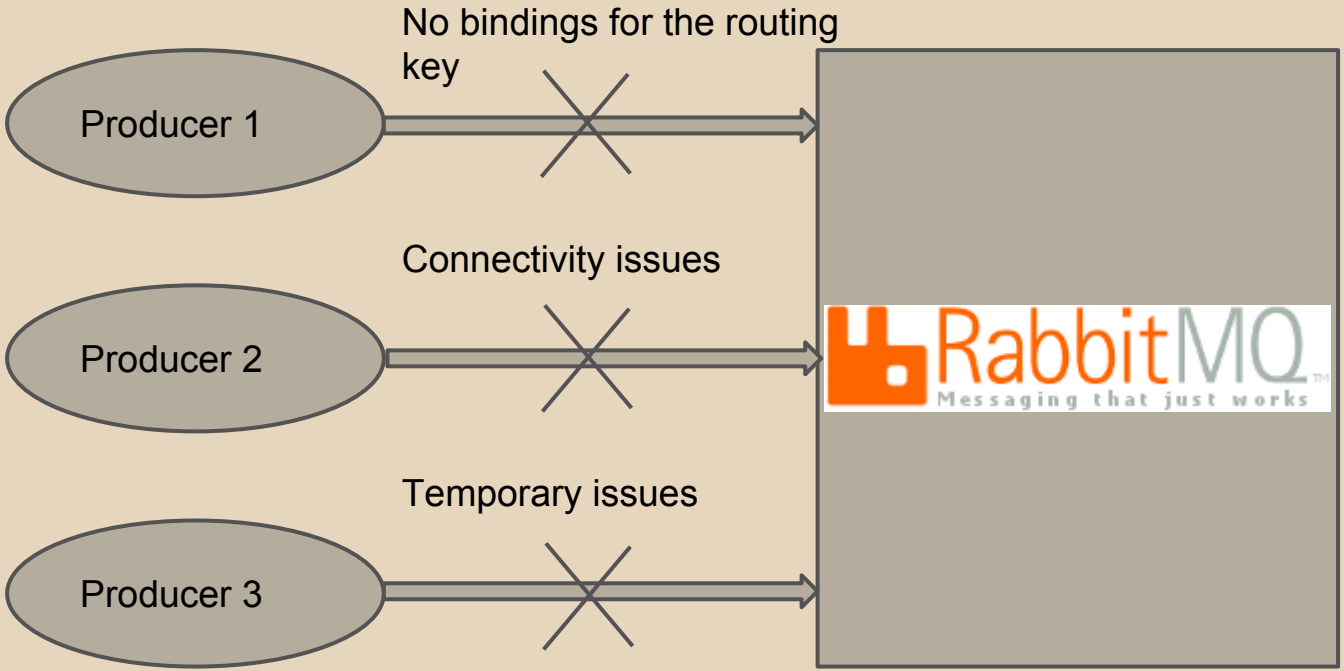
Safe Bunny

Safe RabbitMQ delivery with local queueing

RabbitMQ: Common Scenario



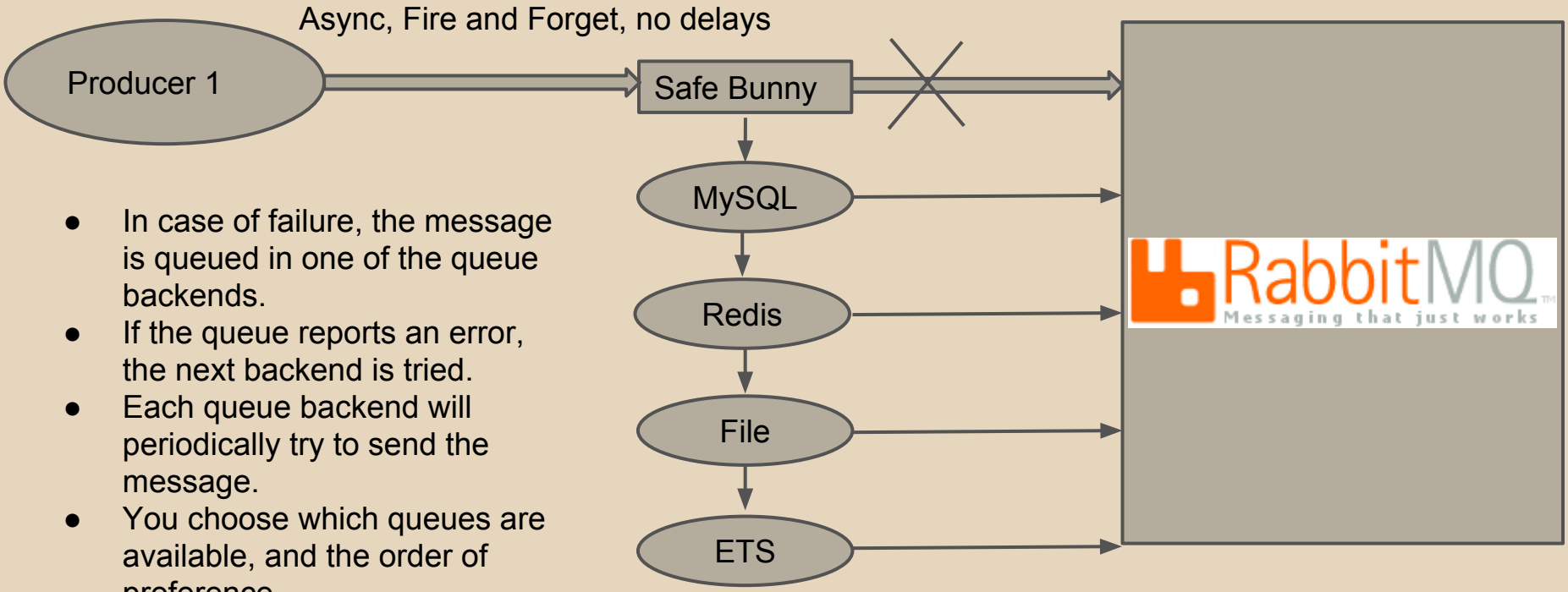
Safe Bunny: What if something fails?



Safe Bunny: Safe delivery

- Set channel in confirmation mode
- Sync operation, need to wait for server confirmation or timeout
- How/When to retry the operation?
- How to make messages survive long periods of time?

Safe Bunny: Safe delivery



- In case of failure, the message is queued in one of the queue backends.
- If the queue reports an error, the next backend is tried.
- Each queue backend will periodically try to send the message.
- You choose which queues are available, and the order of preference

Safe Bunny: delivery modes

- Direct to MQ: no local queuing: Use **safe_bunny:deliver_unsafe/3**. This will try to publish a message to the given exchange and routing key, and will not try to queue the message on failure.
- Direct to MQ, local queueing on failure: Use **safe_bunny:deliver_safe/3**. Like the above, but will queue the message in the fallback queues on failure. Fallback queues are tried in order, according to the producers option of the safe_bunny application.
- Local queuing only: Consumers will try to send the messages to the MQ. "Safest" option (because you queue first). Use **safe_bunny:queue/3**. One of the queue consumers should pick the new message and try to publish it via rabbitmq.

Safe Bunny: Examples

- safe_bunny:deliver_safe(Exchange, Key, Payload)
- safe_bunny:deliver_unsafe(Exchange, Key, Payload)
- safe_bunny:queue(Exchange, Key, Payload)

All arguments are binary(). Where:

- Exchange is the name of the exchange
- Key is the routing key
- Payload is anything you want it to be

Safe Bunny: Concurrency

- Concurrency tools: https://github.com/duomark/epocxy
- Worker Pool: https://github.com/tigertext/worker_pool

Safe Bunny: Useful links

- Jenkins, Coverage, Doc: http://ci.marcelog.name/job/safe_bunny/
- https://github.com/marcelog/safe_bunny
- Works great with MinionPools :) https://github.com/marcelog/rabbitmq_minionpool
- RabbitMQ confirmation mode: http://www.rabbitmq.com/confirms.html
- RabbitMQ erlang client: http://www.rabbitmq.com/erlang-client-user-guide.html

Thanks!