

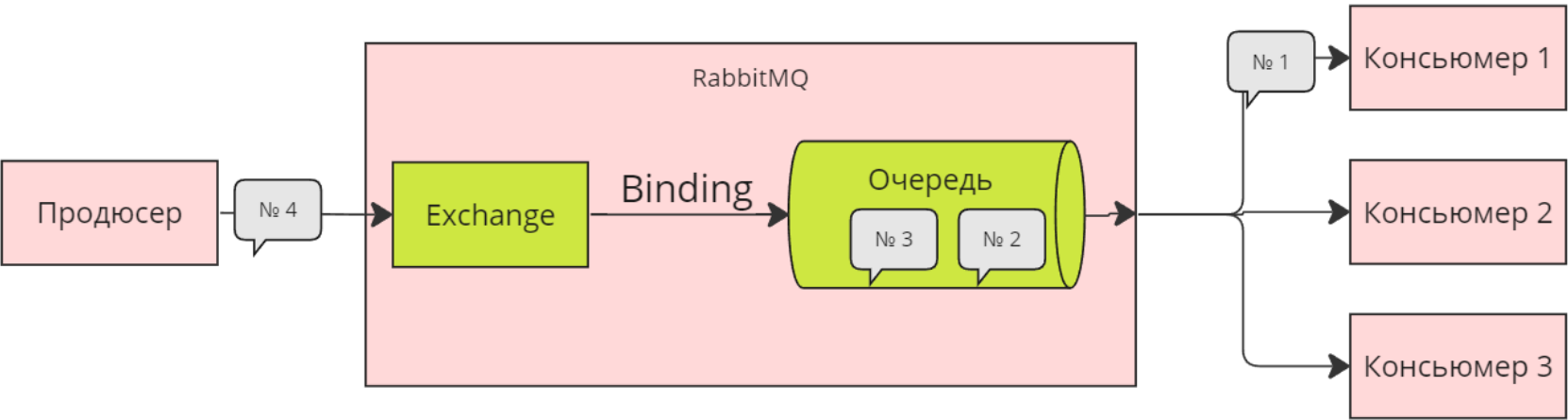
**RabbitMQ** - один из брокеров сообщений, построенный на модели очередей.

Приведем отличительные особенности RabbitMQ:

- Быстрая реализация обычной очереди для последовательной обработки сообщений от сервиса к сервису
- Реализация паттерна запрос-ответ из коробки (синхронный бизнес-процесс)
- Нет хранилища данных, сообщение удаляется после обработки. Но есть функционал записи сообщений на диск.
- Реализация паттерна "умный брокер-тупой консьюмер", логика доставки на стороне брокера
- Не подходит для огромных потоков данных

Ранее мы разбирали детально технические отличия с Kafka, еще раз проговаривать их не будем.

Давайте приступим к разбору, из чего состоит RabbitMQ. Общая схема работы:



Вам известны все понятия на данной схеме, кроме Exchange и Binding. Разберем их:

- Exchange - это по своей сути обменник или маршрутизатор, который на входе получает все сообщения от продюсеров. Важно - продюсер никогда не пишет в очередь напрямую.

- Binding - это связь между обменником и очередями (очереди может быть больше одной, конечно же). Позволяет настроить маршрутизацию сообщений в нужные очереди по ключам, Routing Key и Headers (в зависимости от настроек).

Как происходит взаимодействие:

#### 1. Продюсер отправляет сообщение в брокер

- В сообщении передает: Routing Key или Headers, чтобы сообщение было передано из обменника в нужную очередь; delivery\_mode, обозначая этим ключом, надо ли записать сообщение на диск (=1 не записываем, =2 записываем на диск); а также указывает ключи, связанные с гарантией доставки - confirm, который позволяет запрашивать подтверждения об успешной приемке сообщений и mandatory, который позволяет отправлять назад или в определенную очередь сообщения, которые не имеют настроек маршрутизации.

#### 2. Сообщение попадает в Exchange

3. Exchange ищет подходящий binding для сообщения. Сообщение не хранится в Exchange, если маршрут не найден и в сообщении не задан параметр mandatory, то сообщение будет удалено без возможности восстановления.

4. Binding можете рассматривать как настройку маршрута от Exchange к очереди. Если параметры binding заданы корректно, то Exchange отправляет сообщение в нужную очередь.

5. Сообщение из очереди отправляется одному из консьюмеров (используется модель push). Одно сообщение может получить только один консьюмер.