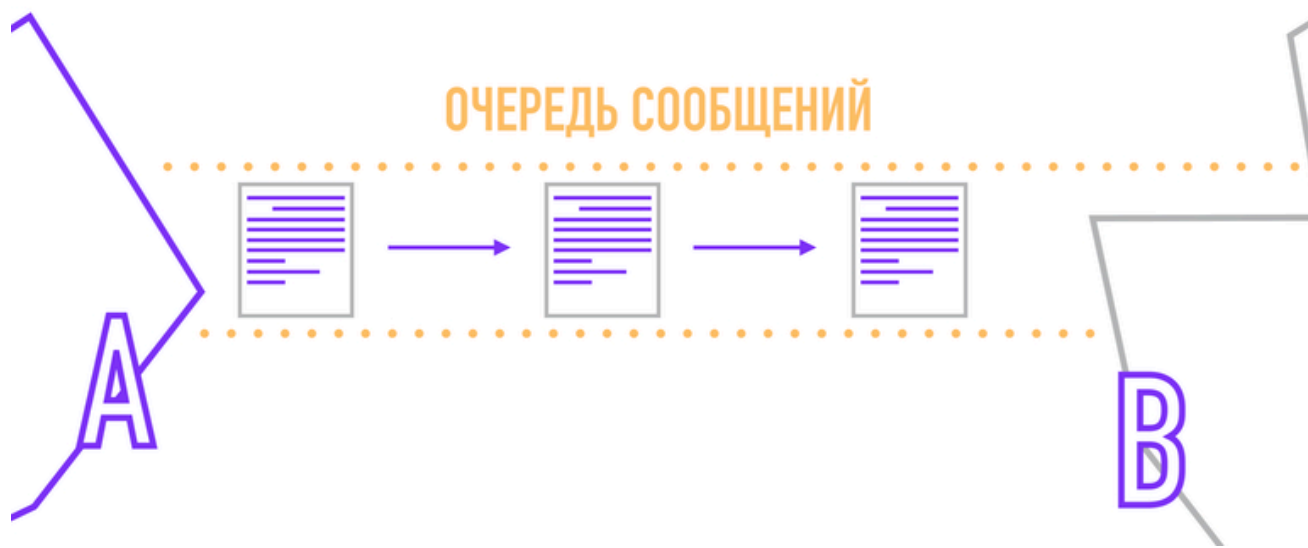


Это отдельный сервис, который отвечает за хранение и доставку данных от сервисов-отправителей к сервисам-получателям с помощью модели Pub/Sub

Модель предполагает, что асинхронное взаимодействие осуществляется согласно следующей логике двух ролей:

- Publishers - публикуют новую информацию в виде сгруппированных по некоторому атрибуту сообщений
- Subscribers - подписываются на потоки сообщений с определенными атрибутами и обрабатывают их

Группирующим сообщения атрибутом выступает "очередь", которая нужна, чтобы разделять потоки данных - таким образом, получатели могут подписываться только на те группы сообщений, которые их интересуют.



Очередь

Ее можно представить как канал связи, натянутый между писателем и читателем.

Писатели кладут сообщения в очередь, после чего они "проталкиваются" читателям, которые подписаны на эту очередь. Один читатель получает одно сообщение за раз, после чего оно становится недоступно другим читателям.

Под сообщением же подразумевается единица данных, обычно состоящая из тела сообщения и метаданных брокера.

Если говорить в общем - тело сообщения представляет из себя набор байт определенного формата. Получатель обязательно должен знать этот формат, чтобы после получения сообщения иметь возможность десериализовать его тело для дальнейшей обработки. Использовать можно любой формат, однако, важно помнить об

обратной совместимости, которую поддерживают, например, бинарный `Protobuf` и фреймворк `Apache Avro`.

AMQP - Advanced Message Queuing Protocol

По такому принципу работает большинство брокеров сообщений, построенных на **AMQP (Advanced Message Queuing Protocol)** - протокол, который описывает стандарт отказоустойчивого обмена сообщениями посредством очередей.

Протокол обеспечивает следующие преимущества:

- **Слабая связанность.** Она достигается за счет асинхронной передачи сообщений: то есть, отправитель скидывает данные и продолжает работать, не дожидаясь ответа от получателя, а получатель вычитывает и обрабатывает сообщения, когда удобно ему, а не когда они были отправлены. В данном случае очередь можно сравнить с почтовым ящиком, в который почтальон кладет ваши письма, а вы их забираете, когда удобно вам
- **Масштабируемость.** Если сообщения появляются в очереди быстрее, чем консьюмер успевает их обрабатывать (речь идет не о пиковых нагрузках, а о стабильном разрыве между скоростью записи и обработки), можно запустить несколько экземпляров приложения-консьюмера и подписать их на одну очередь. Этот подход называется "горизонтальным" масштабированием, а экземпляры одного сервиса принято называть репликами. Реплики сервиса-консьюмера будут читать сообщения из одной очереди и обрабатывать их независимо друг от друга
- **Эластичность.** Наличие между приложениями такой прослойки, как очередь, помогает справляться с пиковыми нагрузками: в этом случае очередь будет выступать буфером, в котором сообщения будут копиться и по мере возможности считываться консьюмером, вместо того, чтобы ронять приложение-получатель, отправляя данные ему напрямую
- **Гарантии доставки.** Большинство брокеров предоставляют гарантии "at least once" и "at most once"

Гарантии доставки - объяснение

- **At most once** - исключает повторную обработку сообщений, однако допускает их потерю. В этом случае брокер будет доставлять сообщения получателям по принципу "отправил и забыл". Если получатель не смог по какой-то причине обработать сообщение с первой попытки, брокер не будет осуществлять переотправку.
- **At least once** - гарантирует получение сообщения получателем, однако при этом есть вероятность повторной обработки одних и тех же сообщений. Зачастую данная гарантия достигается за счет механизма "acknowledgement/negative acknowledgement",

который предписывает совершать переотправку сообщения, если получатель по какой-то причине не смог его обработать.

Таким образом, для каждого отправленного сообщения брокером существует три состояния:

- Получатель вернул Ack (успешная обработка)
- Вернул Nack (неуспешная обработка)
- Разорвал соединение

Последние два сценария приводят к переотправке сообщения и повторной обработке.

Однако брокер может произвести повторную отправку и при успешной обработке сообщения получателем. Например, если получатель обработал сообщение, но завершил его работу, не отправив сигнал Ack брокеру.

В этом случае брокер снова положит сообщение в очередь, после чего оно будет обработано повторно, что может привести к ошибкам и порче данных, если разработчик не предусмотрел механизм устранения дублей на стороне получателя.

Стоит отметить, что существует еще одна гарантия доставки, которая называется "exactly one". Ее трудно достичь в распределенных системах, но при этом она же является наиболее желаемой.