

Рассмотрим модели взаимодействия брокера и консьюмера. Push и Pull.

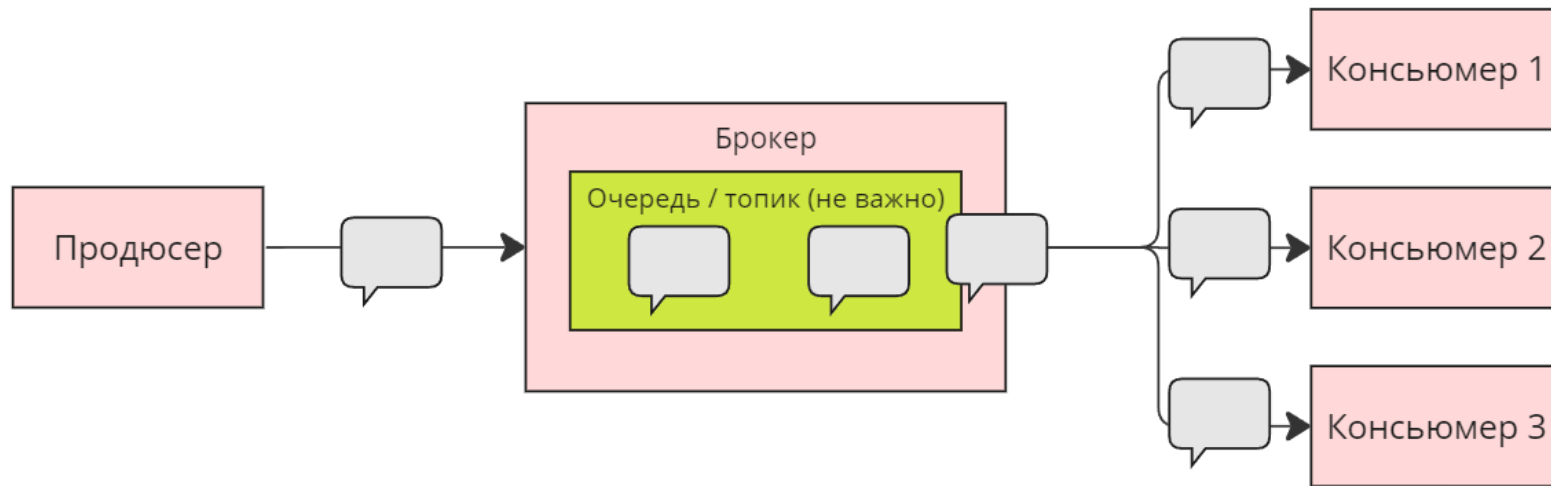
"Push" и "pull" - это термины, которые представляют модель взаимодействия между консьюмерами и брокерами. Что важно - эти термины применяются как к модели взаимодействия очереди, так и к модели издатель-подписчик. Если сказать иначе, это следующий уровень детализации принципов работы брокеров. Рассмотрим каждую модель отдельно, сразу на примере моделей очереди и подписок.

Очереди сообщений	Издатель-подписчик
Push: брокер отправляет сообщение клиенту (консьюмеру), как только оно становится доступным в очереди. Это означает, что брокер "толкает" сообщение для обработки. Клиент (консьюмер) не думает о том, когда нужно забирать новое сообщение.	Push: когда новое сообщение появляется в топике, брокер автоматически рассылает его всем активным подписчикам. Клиент (консьюмер) не думает о том, когда нужно забирать новое сообщение.
Pull: клиент (в данном случае консьюмер) активно запрашивает (тянет) сообщение из очереди, когда он готов его обработать. Клиент регулярно или по графику проверяет очередь на наличие новых сообщений и забирает их по мере необходимости.	Pull: в данной модели мы позволяем подписчикам "тянуть" сообщения из темы. Клиент регулярно или по графику проверяет очередь на наличие новых сообщений и забирает их по мере необходимости.

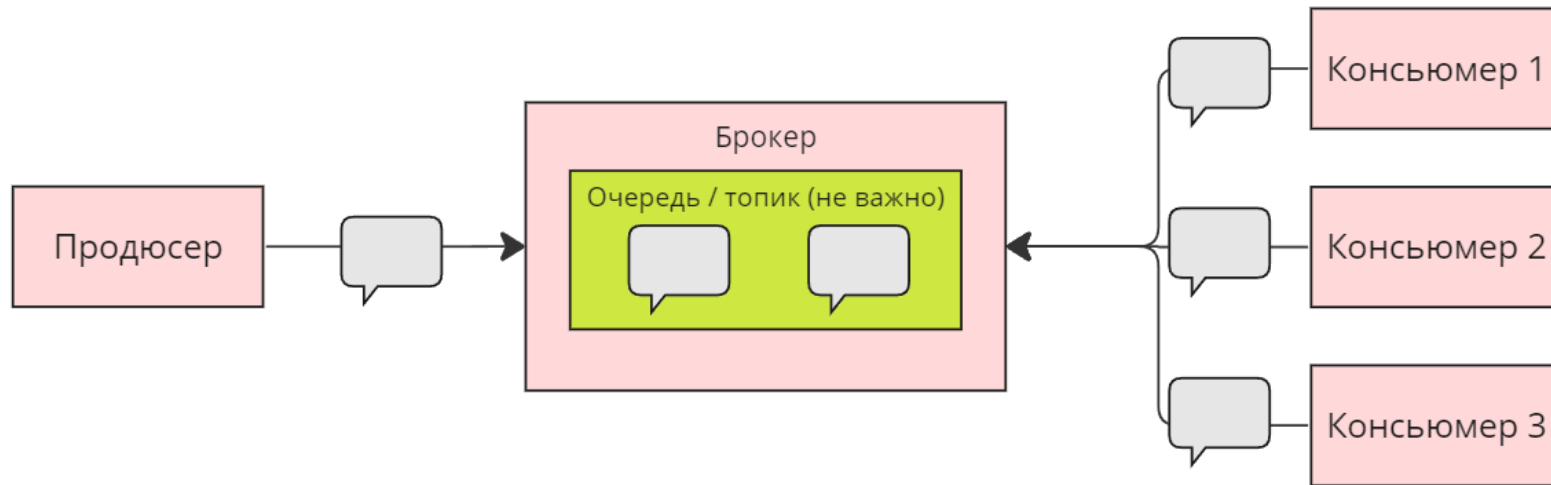
Таким образом, как push, так и pull модели используются и для очередей сообщений, и для подписок. Все зависит от конкретной реализации и брокера сообщений. Например, RabbitMQ изначально поддерживает push модель, но можно реализовать и pull модель. А Kafka, со своей моделью "журнал", поддерживает только pull.

Наглядно различие моделей выглядит так:

PUSH



PULL



Запомним главное. Push - брокер сам доставляет сообщения, pull - консьюмер сам должен позаботиться о получении сообщения. В интернете вы можете найти еще обозначения этих моделей, как "умный брокер/тупой консьюмер" и "тупой брокер/умный консьюмер" :)

Концептуальные различия данных моделей:

Push	Pull
Быстрая скорость доставки (при нужных мощностях брокер при получении сообщения сразу отправляет его)	Более медленная скорость доставки (все будет зависеть от графика опрашивания брокера и от мощности консьюмера)
Брокеру надо знать всех консьюмеров	Брокеру не нужно знать, сколько есть консьюмеров
Консьюмер может быть "тупым", не нужно писать полноценный код для получения сообщения	Консьюмер должен быть "умным", т.е. полноценный код получения сообщения на стороне консьюмера
На практике пропускная способность не более пары десятка тысяч сообщений в секунду, т.к. за доставку отвечает только брокер	На практике пропускная способность может достигать нескольких миллионов сообщений в секунду
Гарантия доставки поддерживаются брокером	Гарантия доставки по факту на стороне консьюмера.

Как вы заметили, мы ввели новый термин, гарантии доставки. Его мы подробно обсудим в следующем уроке. Пока просто запомните, что это означает определенные обязательства брокера относительно доставки сообщений от продюсера к консьюмеру.

Что получаем по итогу сравнения моделей? Если вам важна быстрая масштабируемость и вы готовы передавать на "плечи" консьюмера логику получения сообщений и гарантии доставки - вам необходимо выбирать модель pull. Если вам достаточно низкой пропускной способности и более быстрой доставки сообщений - вам подходит push модель.

Рассмотрим плюсы и минусы каждой модели подробно, помимо вышеприведенных характеристик.

Плюсы push:

- Возможность использовать для реализации передачи данных в реальном времени. Например, в приложениях мониторинга
- Меньшая нагрузка на клиента, так как он не должен постоянно проверять наличие новых данных
- Меньше затрат на разработку клиента
- Простота реализации на стороне сервера (по опыту разработчиков)

Минусы push:

- Возможна перегрузка клиента, если он не может обрабатывать сообщения сразу
- Возможность перегрузки сервера при большом количестве активных клиентов
- Требуется настройка дополнительных механизмов обработки ошибок, если клиент не может обработать или получить сообщения

Плюсы pull:

- Клиент получает данные только тогда, когда он готов обработать новые сообщения или когда "захочет"
- Большой контроль над запросами и их частотой со стороны клиента, что позволяет оптимизировать использование ресурсов.
- Меньший объем трафика на стороне сервера, так как данные отправляются только по запросу клиента

Минусы pull:

- Возможны пропуски данных, если клиент не проверяет сервер на наличие новых сообщений слишком долго
- Потенциально увеличенная нагрузка на сервер из-за повторных запросов со стороны множества клиентов, особенно если данные обновляются часто

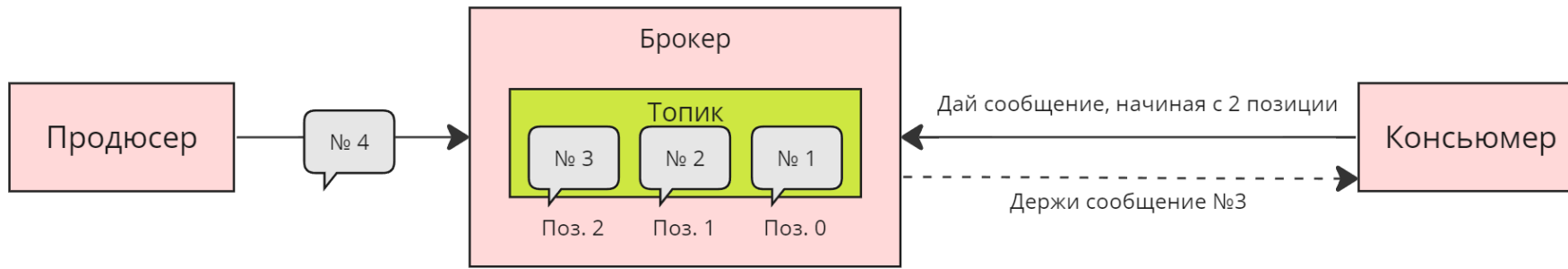
Чтобы выбрать паттерн взаимодействия брокера и консьюмеров, можете для начала ориентироваться на эти вопросы:

- 1. Какая должна быть пропускная способность у брокера?
 - 1. Приблизенно к миллиону (pull)
 - 2. Не более нескольких десятков тысяч (push)
- 2. Клиенты (консьюмеры) готовы разрабатывать логику получения и гарантий доставки сообщений на своей стороне?
 - 1. Да (pull)
 - 2. Нет (push)
- 3. Для сервиса критично получать данные мгновенно?
 - 1. Да (push)
 - 2. Нет (pull)

В данном уроке мы также затронем механизм, который присутствует у любого брокера.

Как вы заметили, за получение сообщения может отвечать или брокер или сам консьюмер (запрашивая данные самостоятельно). Чтобы брокер знал, какое сообщение читает тот или иной клиент, у брокеров есть механизм хранения этой информации. Он запоминает позицию, на которой остановился консьюмер и готов отправлять сообщение с определенного "места", чтобы не отправлять повторные сообщения (допустим, у нас топик и клиент был временно недоступен, надо продолжить работу). В большинстве случаев принцип работы один и тот же - брокер хранит оффсет, значение смещения, и использует его при начале отправке сообщения, не загружая клиента более ранними сообщениями. Расчет оффсета начинается с нуля, первое сообщение = позиция 0.

В модели pull значение оффсета также должно храниться у консьюмера, т.к. он отвечает за получение сообщений и в любой момент может потребовать выдать сообщения с определенного "места" (оффсета). Наглядно это выглядит так:



Далее мы разберем оффсеты уже в конкретных брокерах.