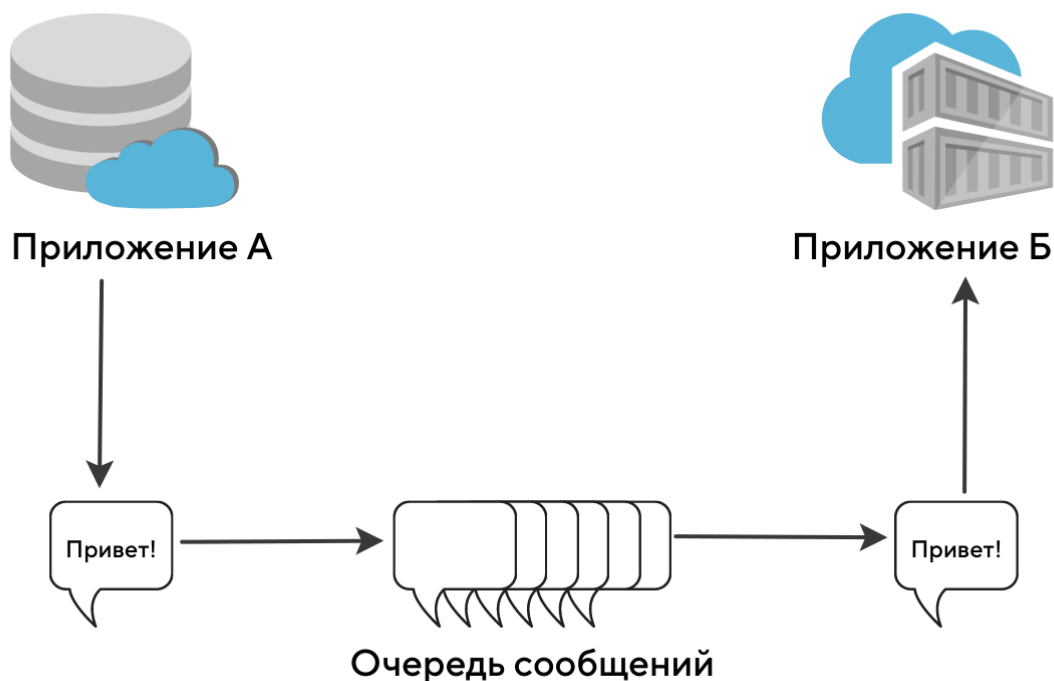


# MQ

**MQ (Message Queue или Очередь сообщений)** - это форма асинхронной коммуникации между сервисами, применяемая в бессерверных и микросервисных архитектурах. Сообщения хранятся в очереди, пока не будут обработаны и удалены. Каждое сообщение обрабатывается только один раз и только одним потребителем. Очереди сообщений могут использоваться для разъединения сложных процессов обработки, для буферизации или организации пакетной обработки, а также для сглаживания пиковых нагрузок.

Брокер сообщений представляет собой тип построения архитектуры, при котором элементы системы “общаются” друг с другом с помощью посредника. Благодаря работе брокера сообщений происходит снятие нагрузки с веб-сервисов, так как им не приходится заниматься пересылкой сообщений: всю сопутствующую этому процессу работу он берёт на себя. Очереди сообщений, по существу, являются связующим звеном между различными процессами в ваших приложениях и обеспечивают надежный и масштабируемый интерфейс взаимодействия с другими подключенными системами и устройствами.



Очереди сообщений имеют неявные или явные ограничения на размер данных, которые могут передаваться в одном сообщении, и количество сообщений, которые могут оставаться в очереди. Многие реализации очередей сообщений функционируют внутренне: внутри операционной системы или внутри приложения. Такие очереди существуют только для целей этой системы.

Другие реализации позволяют передавать сообщения между различными компьютерными системами, потенциально подключая несколько приложений и несколько операционных систем. Эти системы очередей сообщений обычно обеспечивают расширенную функциональность для обеспечения устойчивости, чтобы гарантировать, что сообщения не будут «потеряны» в случае сбоя системы.

## Стандарты и протоколы

Исторически очередь сообщений использовала собственные закрытые протоколы, которые ограничивали способность различных операционных систем или языков

программирования взаимодействовать в гетерогенном (гетерогенность - наличие неодинаковых частей в структуре, в составе чего-либо) множестве сред.

Появились три стандарта, которые используются в реализациях очереди сообщений с открытым исходным кодом:

1. **Advanced Message Queuing Protocol (AMQP)** — многофункциональный протокол очереди сообщений. Открытый протокол прикладного уровня для передачи сообщений между компонентами системы. Основная идея состоит в том, что отдельные подсистемы (или независимые приложения) могут обмениваться произвольным образом сообщениями через AMQP-брокер, который осуществляет маршрутизацию, возможно гарантирует доставку, распределение потоков данных, подписку на нужные типы сообщений.
2. **STOMP** — простой текстовый протокол сообщений. Основан на фреймах по образцу HTTP. Фрейм состоит из команды, необязательных заголовков и необязательного тела.
3. **MQTT (ранее MQ Telemetry Transport)** — протокол облегчённой очереди сообщений, особенно для встроенных устройств. MQTT - упрощённый сетевой протокол, работающий поверх TCP/IP, ориентированный на обмен сообщениями между устройствами по принципу издатель-подписчик.

Эти протоколы находятся на разных стадиях стандартизации и реализации. Первые два работают на том же уровне, что и HTTP, MQTT на уровне TCP/IP.

## **Синхронный или асинхронный**

Многие из широко известных протоколов связи используются синхронно. Протокол HTTP, используемый во Всемирной паутине и в веб-сервисах, предлагает наглядный пример, когда пользователь отправляет запрос на веб-страницу, а затем ждёт ответа.

Однако существуют сценарии, в которых синхронное поведение не подходит. Например, AJAX (асинхронный JavaScript и XML) можно использовать для асинхронной отправки текстовых, JSON- или XML-сообщений для обновления части веб-страницы с более релевантной информацией.

### **Наиболее часто используемые сценарии использования очередей**

- **Фоновая обработка долгосрочных задач на веб-сайтах** - Сюда можно отнести задачи, которые не связаны напрямую с основным действием пользователя сайта и могут быть выполнены в фоновом режиме без необходимости ожидания с его стороны. Это обработка изображений, преобразование видео в различные форматы, создание отзывов, индексирование в поисковых системах после изменения данных, отправка электронной почты, формирование файлов и так далее.
- **Буферизация при пакетной обработке данных** - Очереди можно использовать в качестве буфера для некоторой массовой обработки, например пакетной вставки данных в БД, таким образом сокращаются накладные расходы на инициализацию и завершение каждой операции. Но для стандартной архитектуры может стать проблемой генерация данных клиентской службой быстрее, чем их может обработать получатель. Очередь же предоставляет временное хранилище для пакетов с данными, где они будут храниться до завершения обработки принимающей стороной.
- **Отложенные задачи** - Многие системы очередей позволяют производителю указать, что доставка сообщений должна быть отложена. Это может быть полезно при реализации льготных периодов. Например, вы разрешаете покупателю отказаться от размещения заказа в течение определенного времени и ставите отложенное задание в очередь. Если покупатель отменит операцию в указанный срок, сообщение можно удалить из очереди.

- Сглаживание пиковых нагрузок - Помещая данные в очередь, вы можете быть уверены, что данные будут сохранены и в конечном итоге обработаны, даже если это займет немного больше времени, чем обычно, из-за большого скачка трафика. Увеличить скорость обработки в таких случаях также возможно за счет масштабирования нужных обработчиков.
- Гарантированная доставка при нестабильной инфраструктуре - Нестабильная сеть в сочетании с очередью сообщений создает надежный системный ландшафт: каждое сообщение будет отправлено, как только это будет технически возможно.
- Упорядочение транзакций - Многие брокеры поддерживают очереди FIFO, полезные в системах, где важно сохранить порядок транзакций. Брокер назначает каждому клиенту номер в очереди. Первый в очереди будет обработан первым.
- Сбор аналитической информации - Очереди часто применяют для сбора некоторой статистики, например использования определенной системы и ее функций. Как правило, моментальная обработка такой информации не требуется. Когда сообщения поступают в веб-службу, они помещаются в очередь, а затем при помощи дополнительных серверов приложений обрабатываются и отправляются в базу данных.
- Разбиение трудоемких задач на множество малых задач - Если у вас есть некоторая задача для группы серверов, то вам необходимо выполнить ее на каждом сервере. Например, при редактировании шаблона мониторинга потребуется обновить мониторы на каждом сервере, использующем этот шаблон. Вы можете поставить сообщение в очередь для каждого сервера и выполнять их одновременно в виде небольших операций.
- Прочие сценарии, требующие гарантированной доставки информации высокого уровня отказоустойчивости - Это обработка финансовых транзакций, бронирование авиабилетов, обновление записей о пациентах в сфере здравоохранения и так далее.