Яндекс

HighLoad++
Siberia 2019

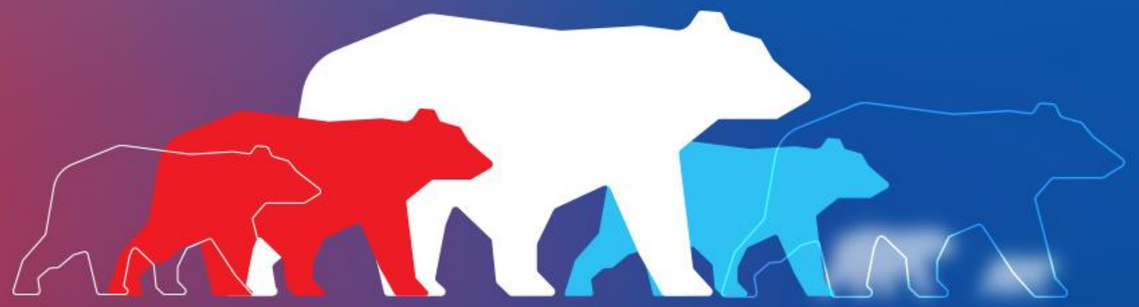
Яндекс Облако

Реализация георасперделённой персистентной очереди сообщений на примере Yandex Message Queue

Василий Богонатов, разработчик сервиса

HighLoad++ Siberia 2019

Профессиональная конференция для разработчиков высоконагруженных систем



План

01 | Зачем нужны очереди сообщений?

02 | Гарантии доставки сообщений

03 | Сравнение популярных реализаций

04 | Свойства и API YMQ

05 | Внутреннее устройство YMQ

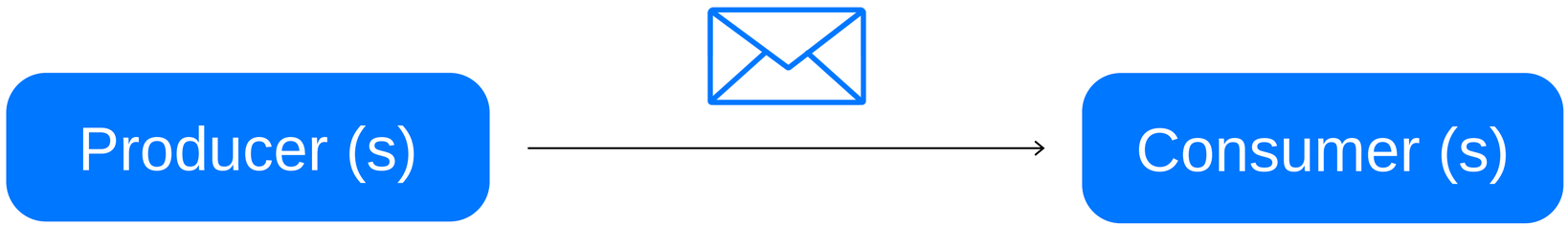
01

Зачем нужны очереди сообщений?

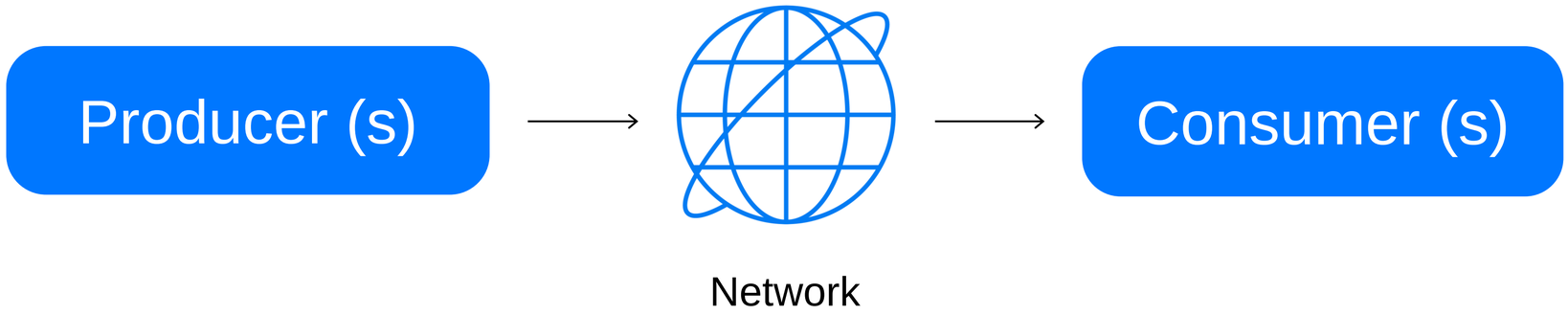
Где мы используем очереди сообщений?

- Доставка логов до хранилища
- Вычислительные задачи
- Обработка потока событий
- Взаимодействие микросервисов

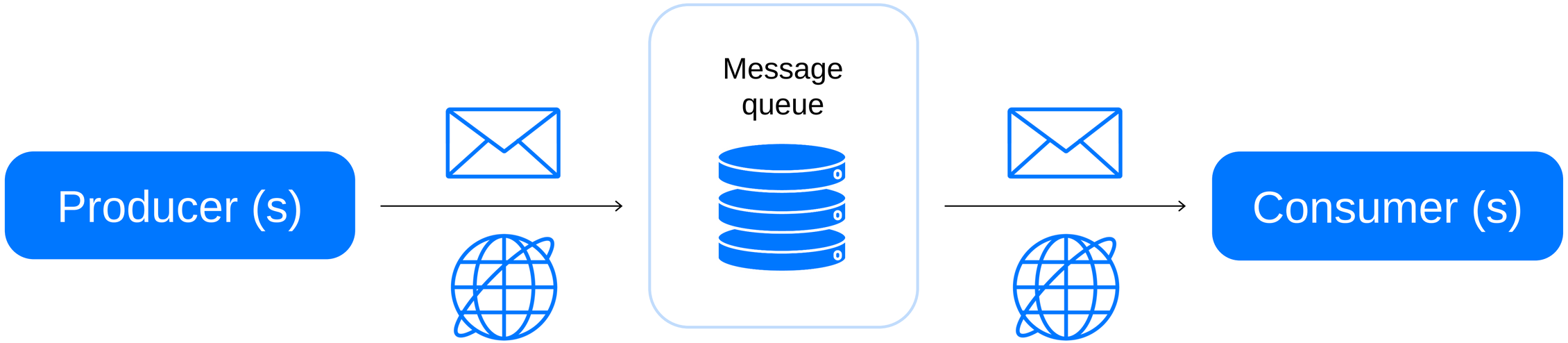
Задача надёжной передачи сообщений



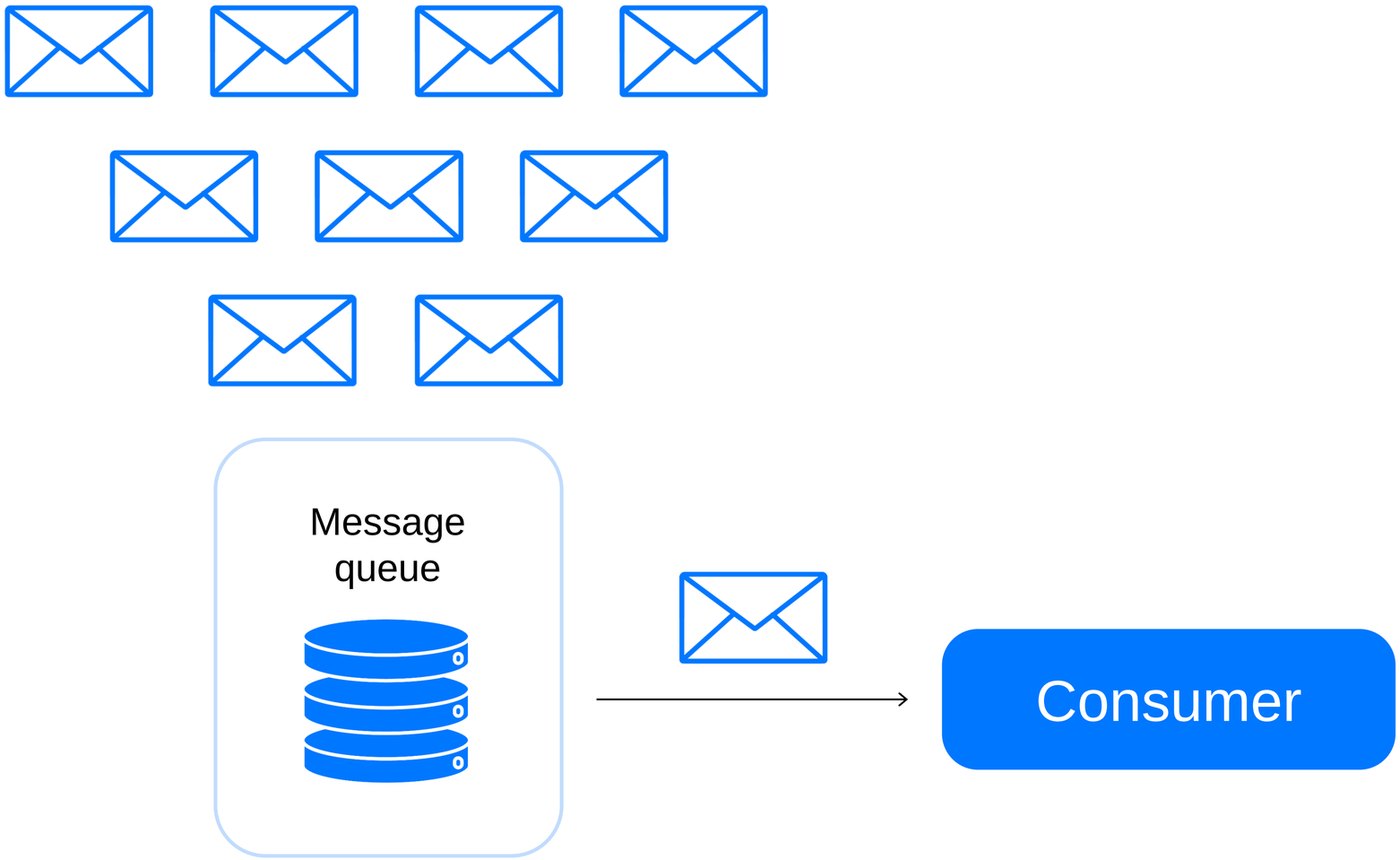
Наивное решение



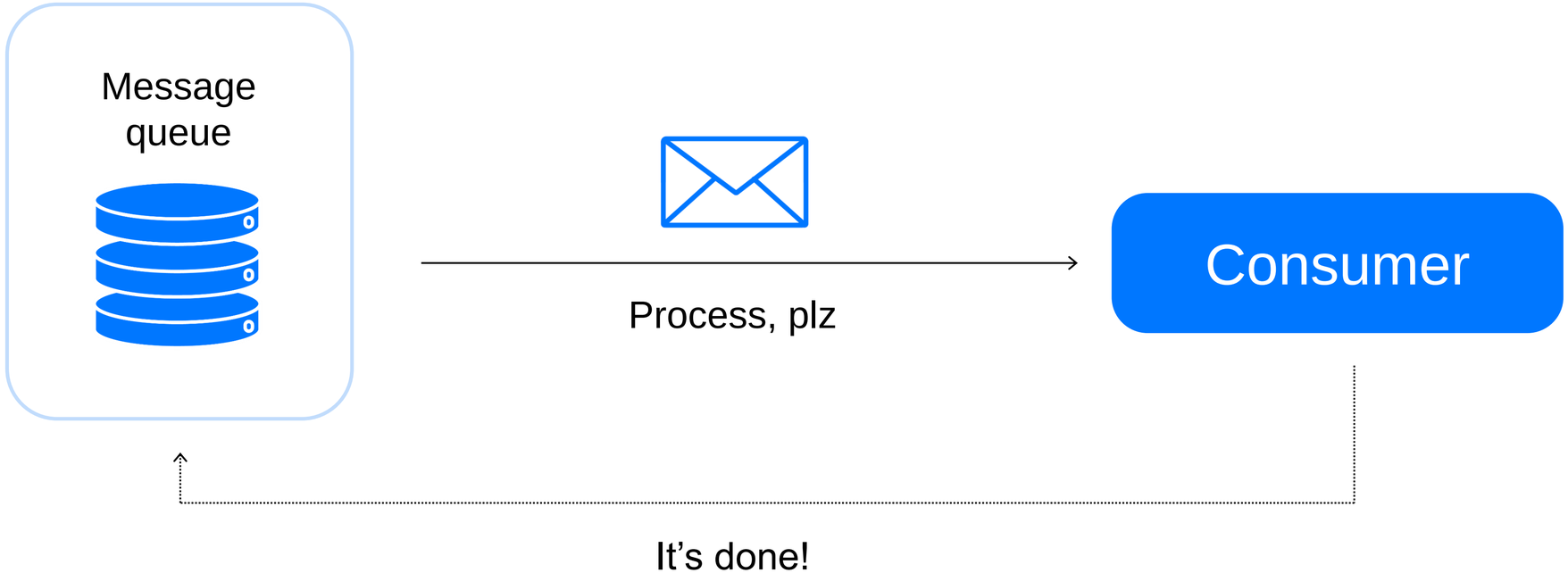
Решение с очередью сообщений



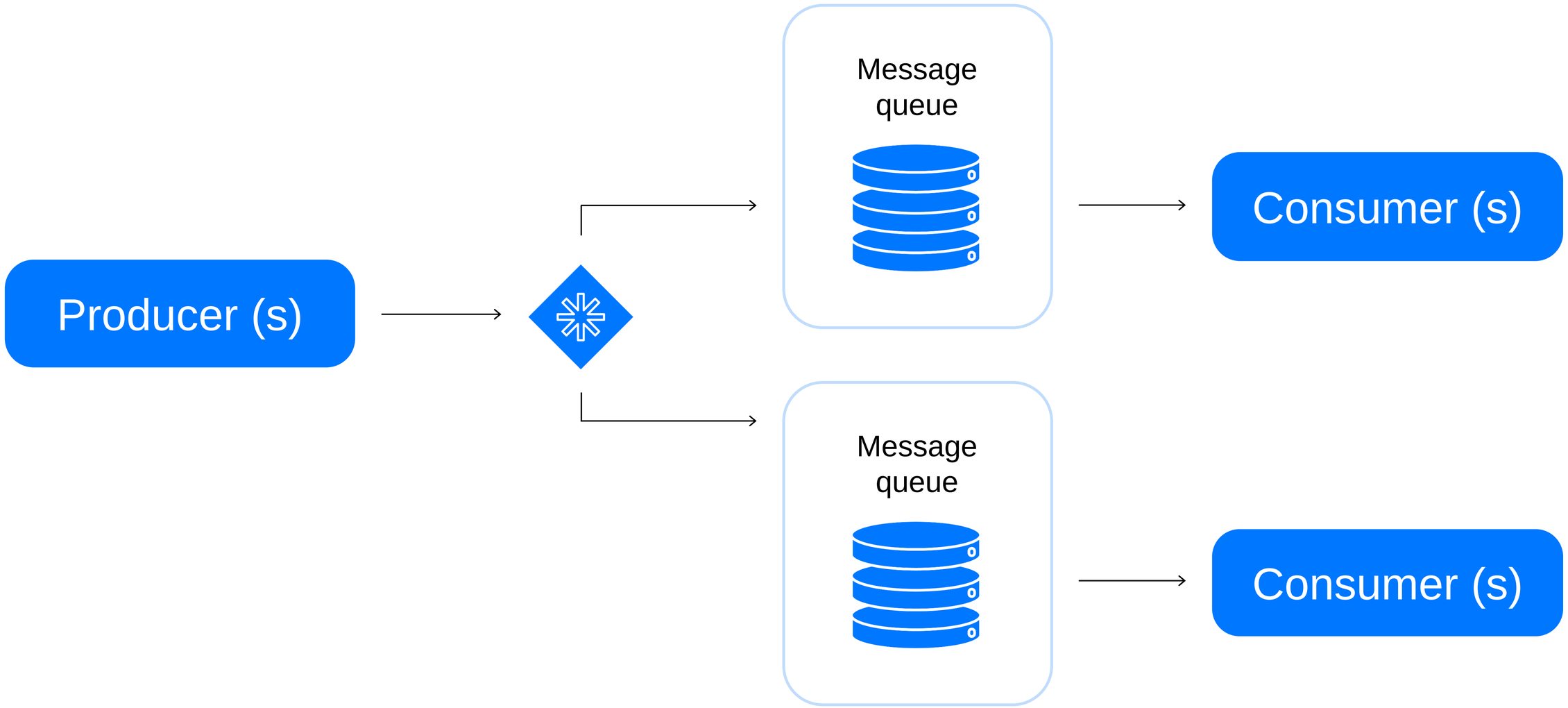
Плюшки очередей: хранение сообщений



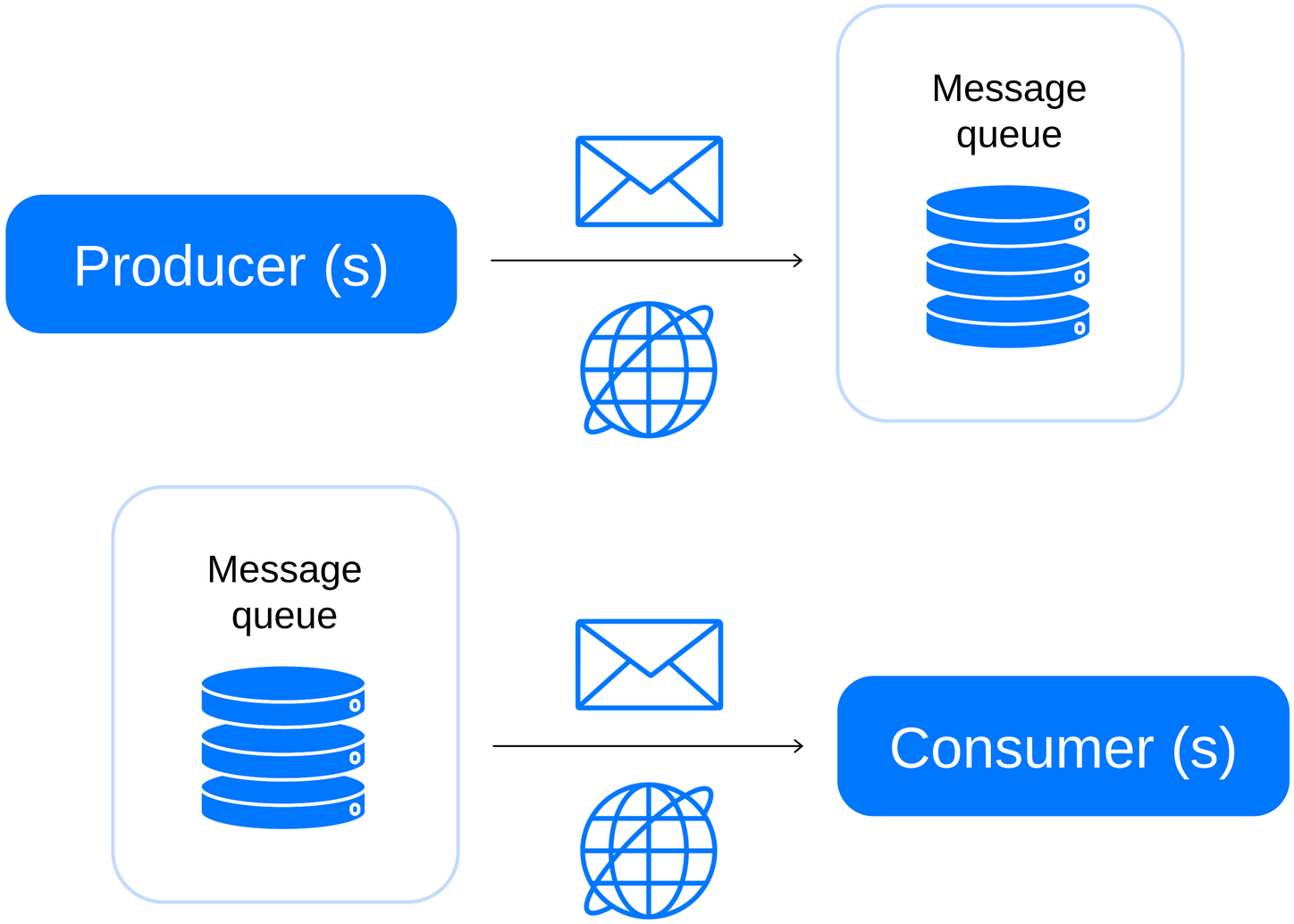
Плюшки очередей: контроль обработки



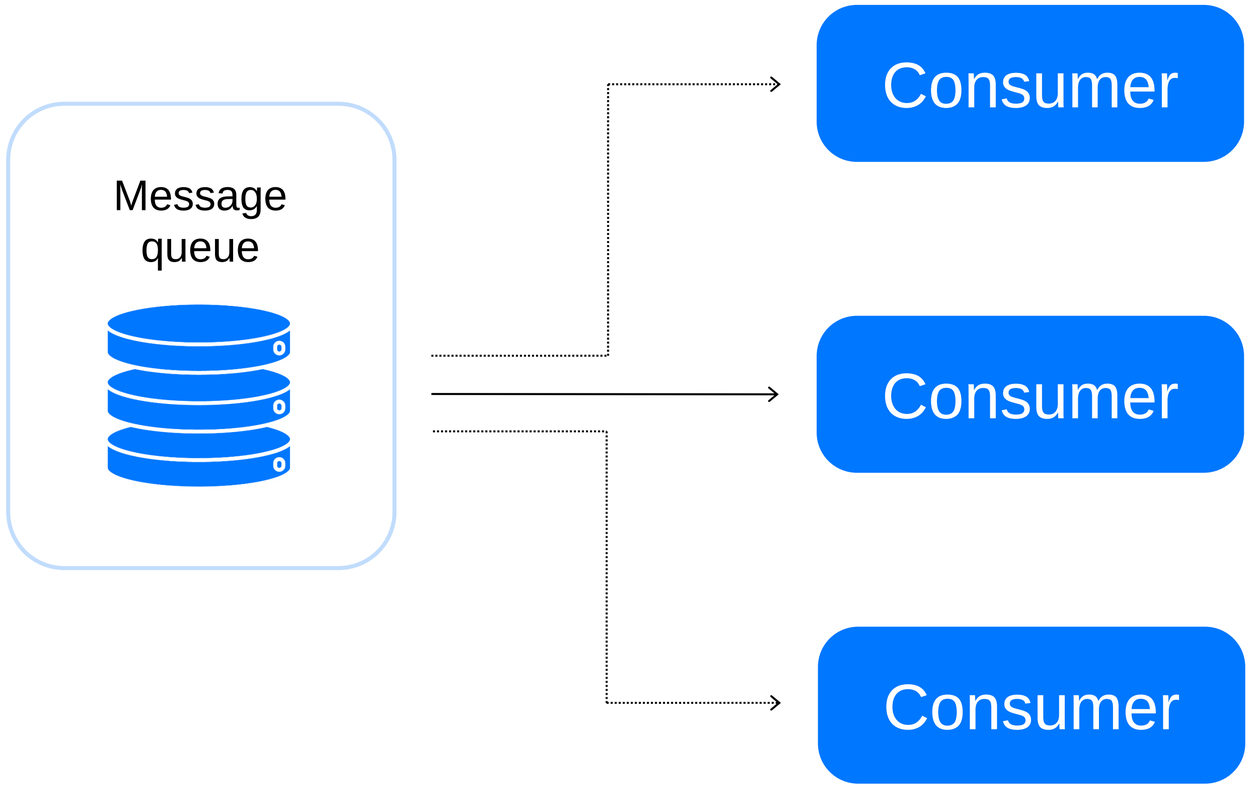
Плюшки очередей: маршрутизация



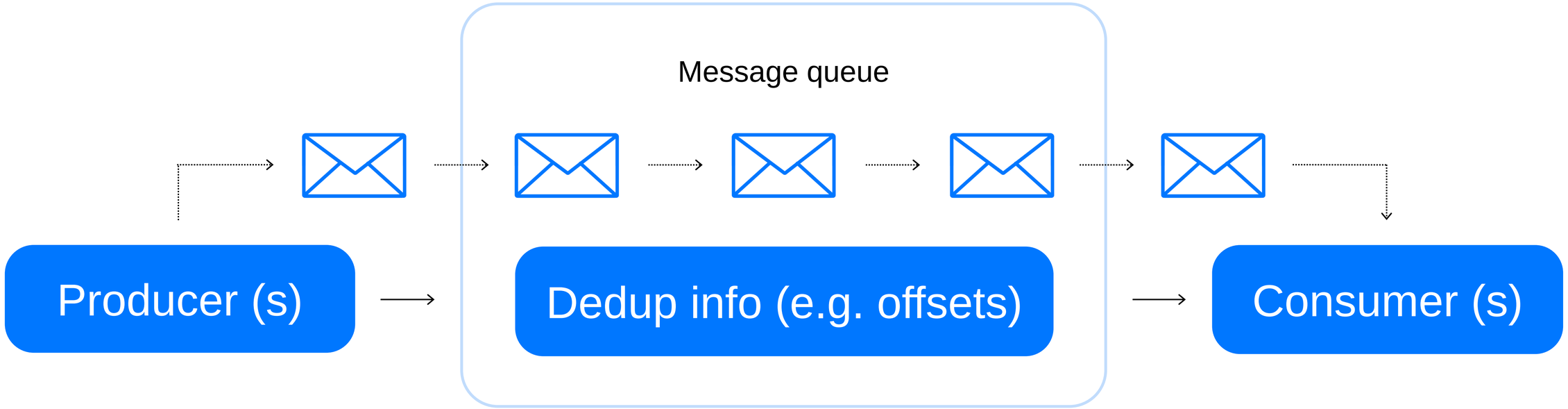
Плюшки очередей: развязка компонентов



Плюшки очередей: лёгкое масштабирование



Плюшки очередей: порядок и дедупликация



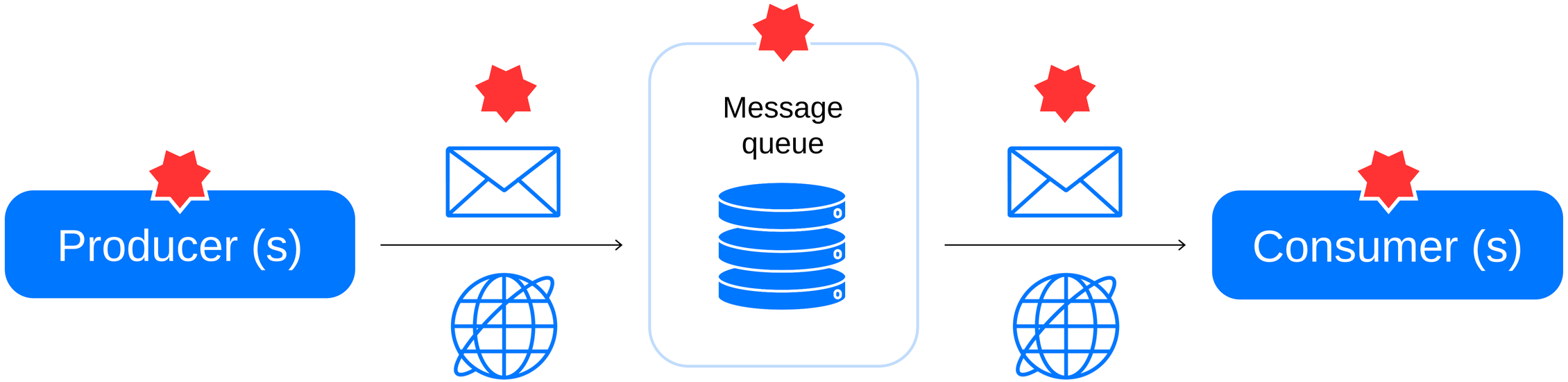
Потому что ~~можем~~ ~~NIH~~ нужно!

- RabbitMQ
- Apache Kafka
- Apache ActiveMQ
- Amazon SQS
- Amazon Kinesis
- Azure Service Bus
- Azure Storage Queue
- Google PubSub
- IBM MQ
- Redis Pub/Sub
- Apache Apollo
- Beanstalkd
- HornetQ
- IronMQ
- NATS
- ZeroMQ
- И многие другие…

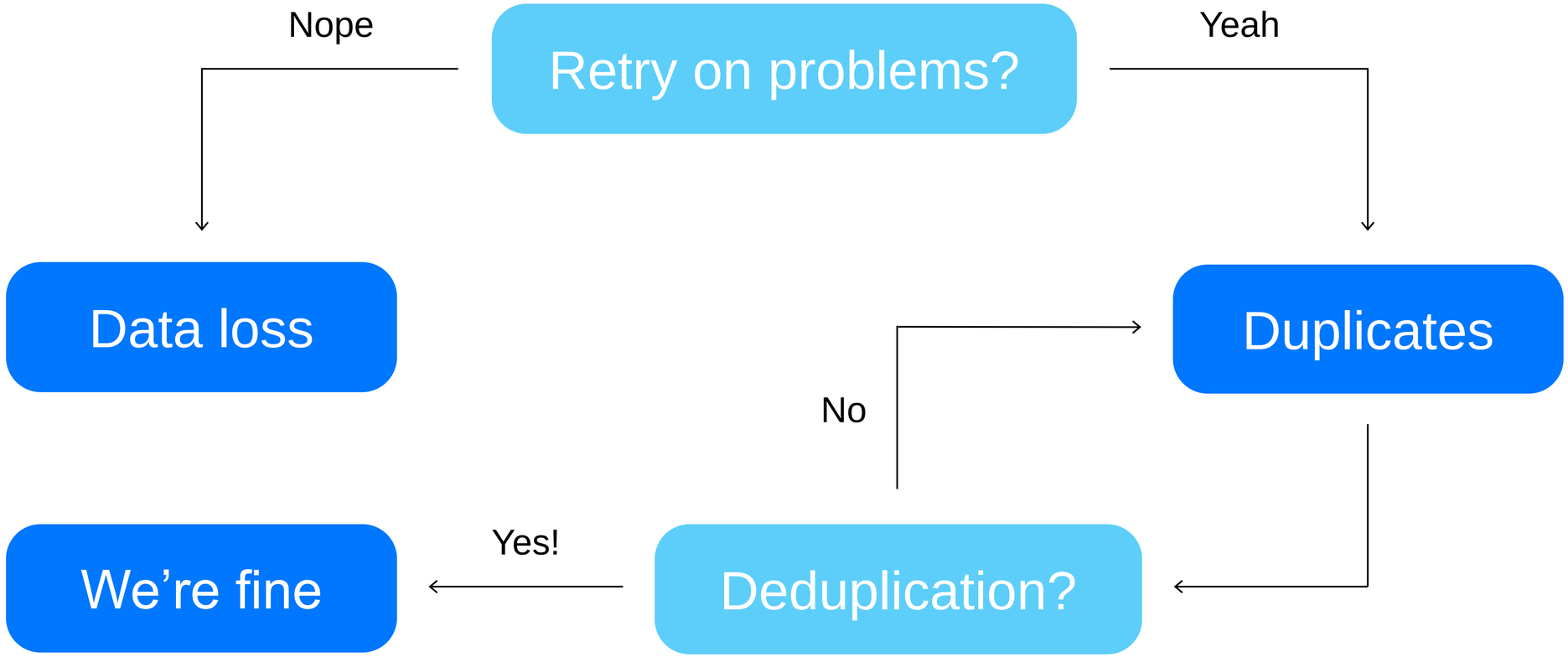
02

Гарантии доставки сообщений

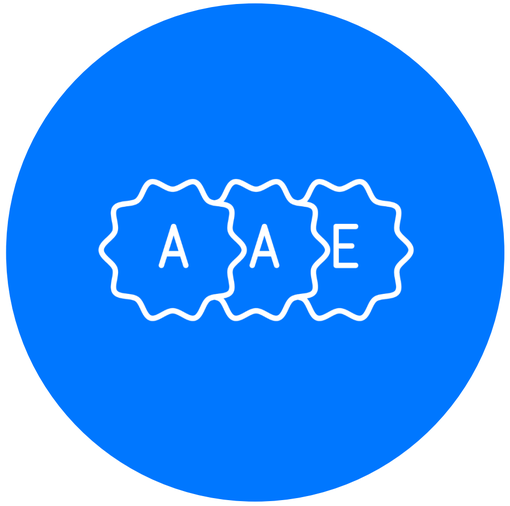
Откуда они берутся?



Что делать при отказах (помимо паники)?



Виды гарантий



- At-most-once — максимум раз, теряем данные
- At-least-once — минимум раз, получаем дубли
- Exactly-once — ровно раз, идеально!

Примеры гарантий на практике

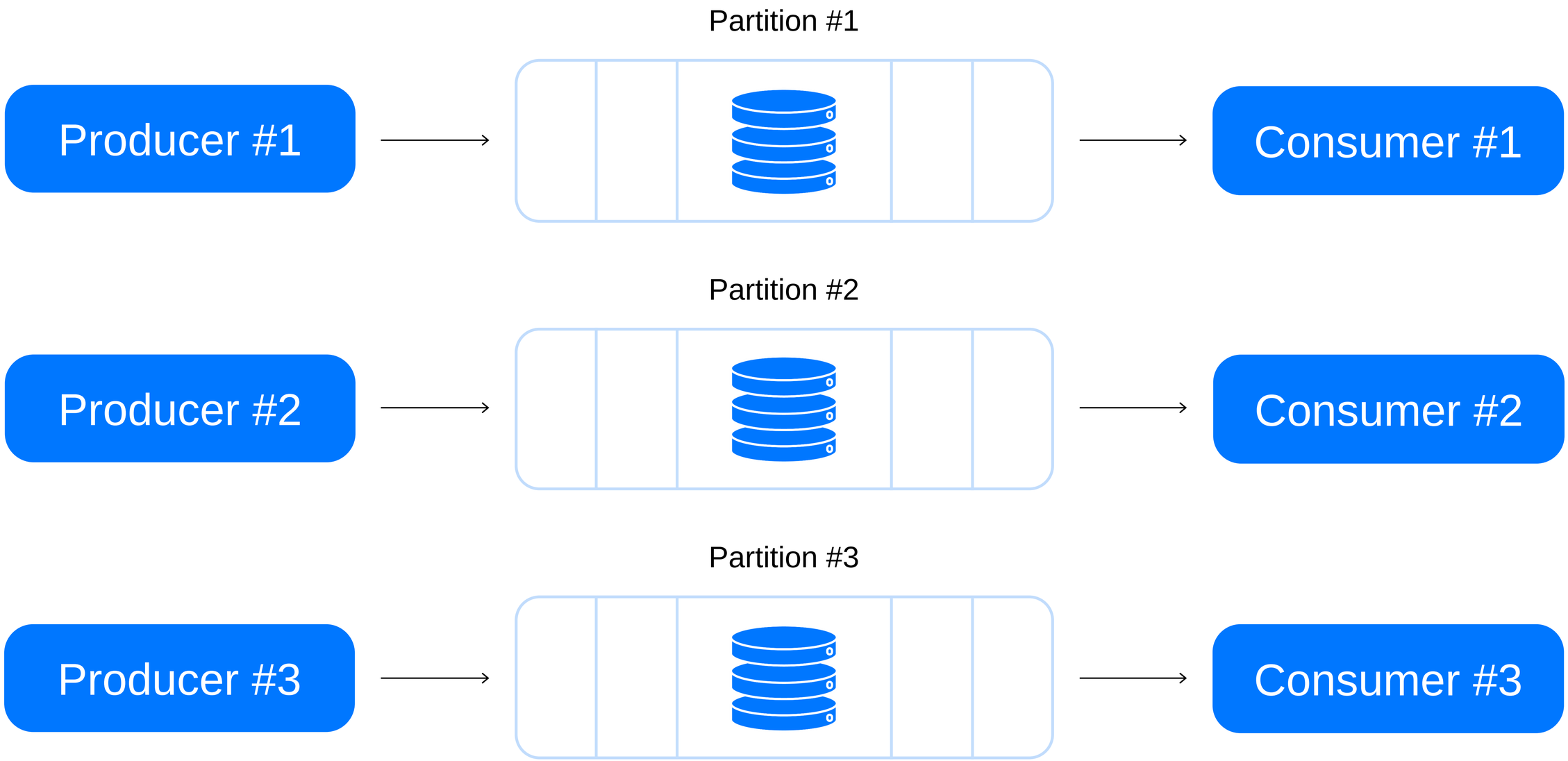
- At-most-once — реалтаймовые данные, почта
- At-least-once — почти везде
- Exactly-once — (вы верите, что у вас так)

Комбинации гарантий

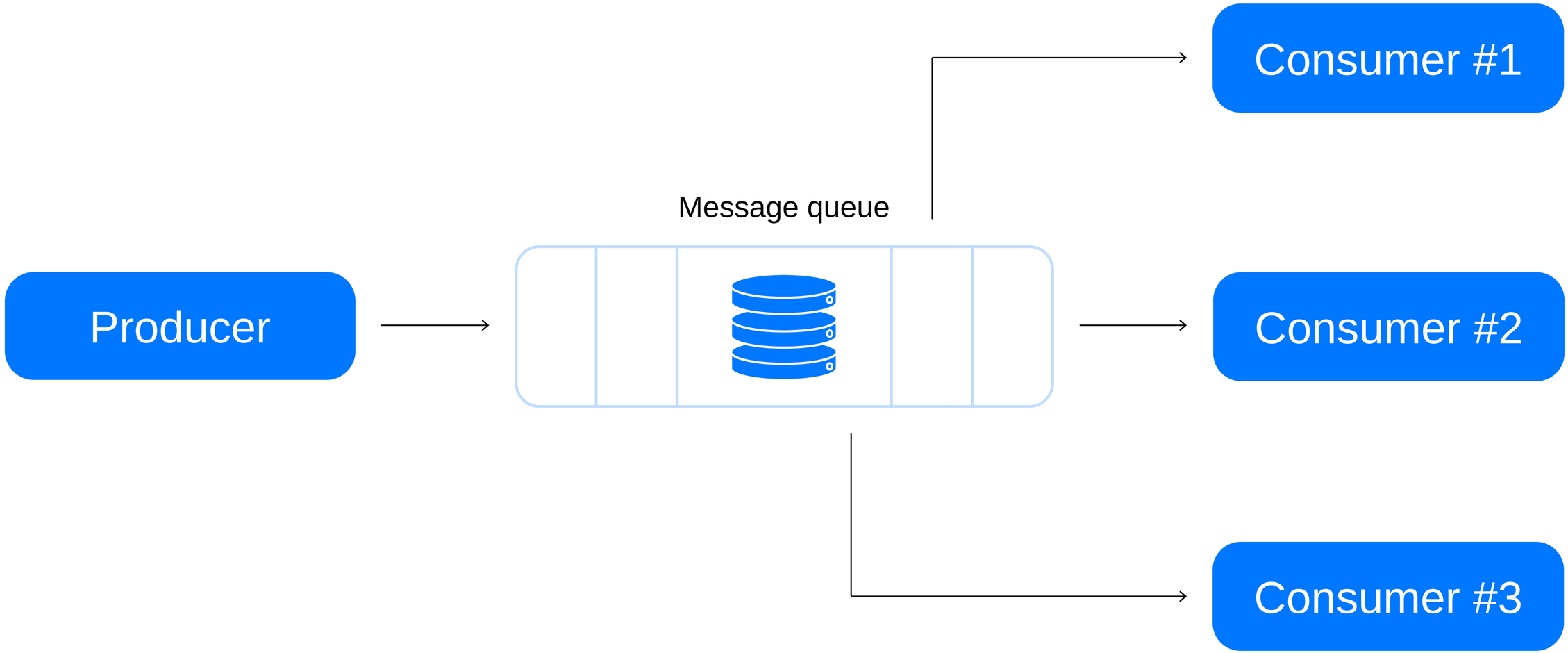
- At-least-once + at-most-once ≠ Exactly-once
- At-least-once + at-most-once = дубли + потери
- Exactly-once + ???-once = ???-once

HighLoad++
Siberia 2019

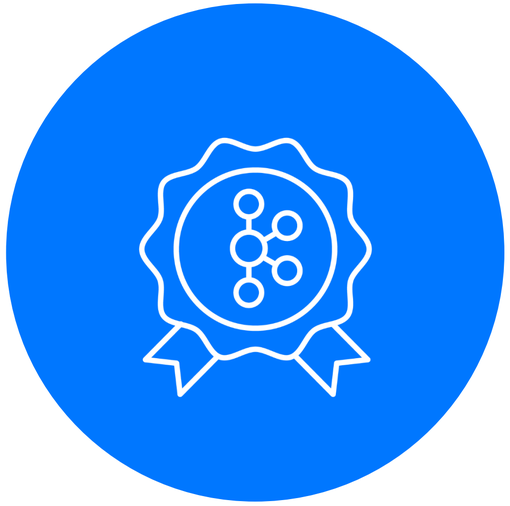
Apache Kafka



RabbitMQ

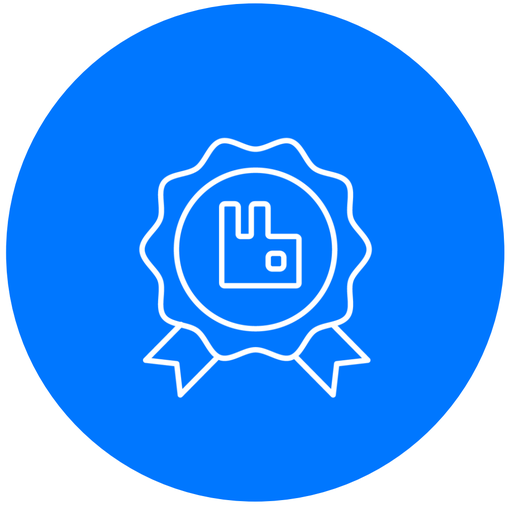


Apache Kafka: гарантии доставки



- At-most-once — коммитим offset до обработки
- At-least-once — коммитим offset после неё
- Exactly-once*** — с 0.11.0.0

RabbitMQ: гарантии доставки



- At-most-once — no ack
- At-least-once — manual ack

int fsync (int fd)

- fsync синхронизирует состояние файла в памяти с состоянием на диске

Apache Kafka: надёжность хранения

- Отвечает без явного вызова fsync

RabbitMQ: надёжность хранения

- Отвечает после fsync(), но может переключиться на несинхронизированное зеркало в качестве мастера

Bonus level:
зачем свою запилили-то?

И всё-таки?

- Целый велопарк у внутренних клиентов
- Максимально простое API и готовые библиотеки
- Стабильная работа в условиях –1 ДЦ
- Действительно надёжное хранение
- Наличие экспертизы и платформы
- Очереди для экосистемы Яндекс.Облака

04

Свойства и API YMQ

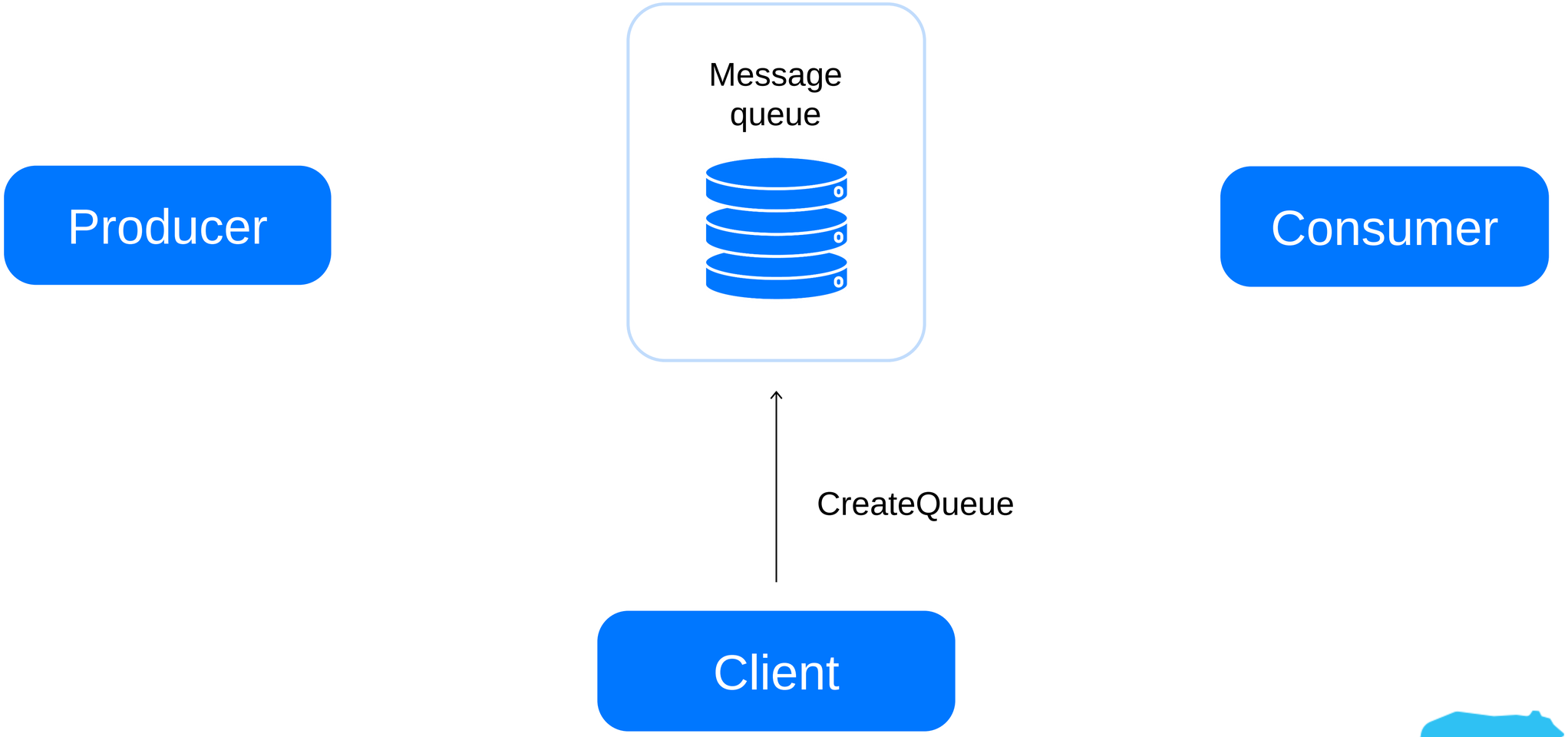
Основные свойства очередей YMQ

- Совместимы с API Amazon SQS (http-api)
- Персистентные
- Хранят данные в 3 репликах в разных ДЦ
- Реализуют паттерн “competing consumers”
- Бывают двух типов: standard и FIFO
- Обеспечивают доставку сообщений at-least-once

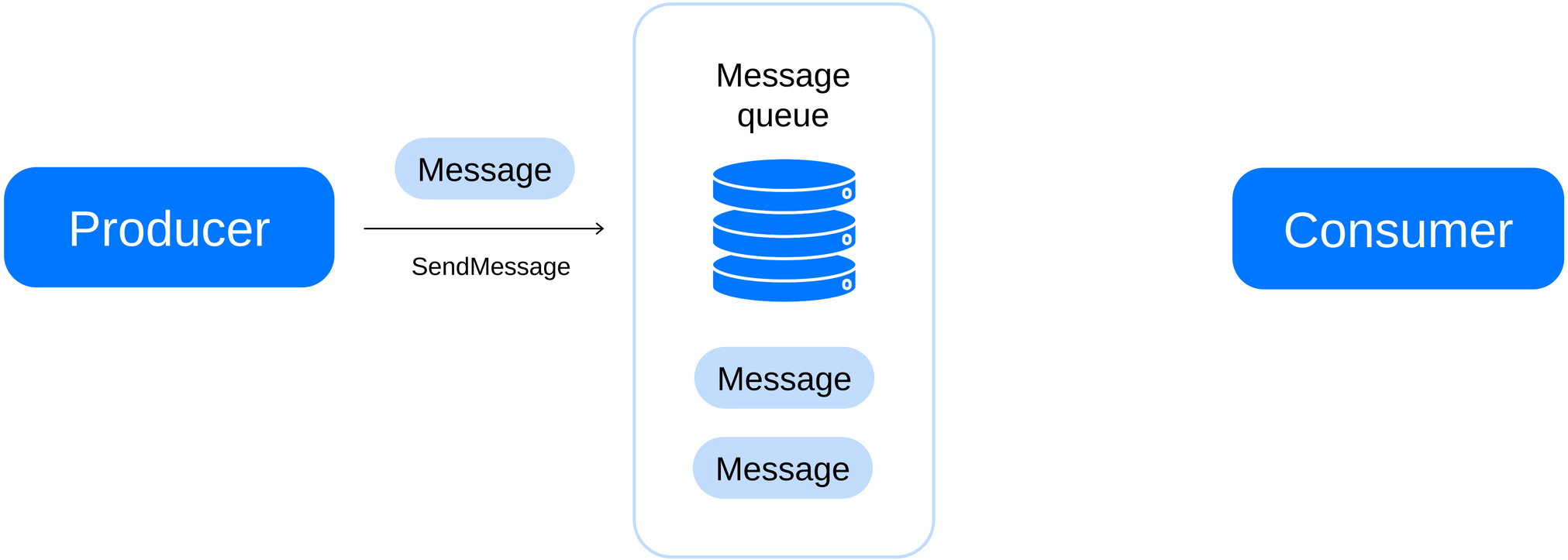
Основные методы API

- CreateQueue
- SendMessage
- ReceiveMessage
- DeleteMessage

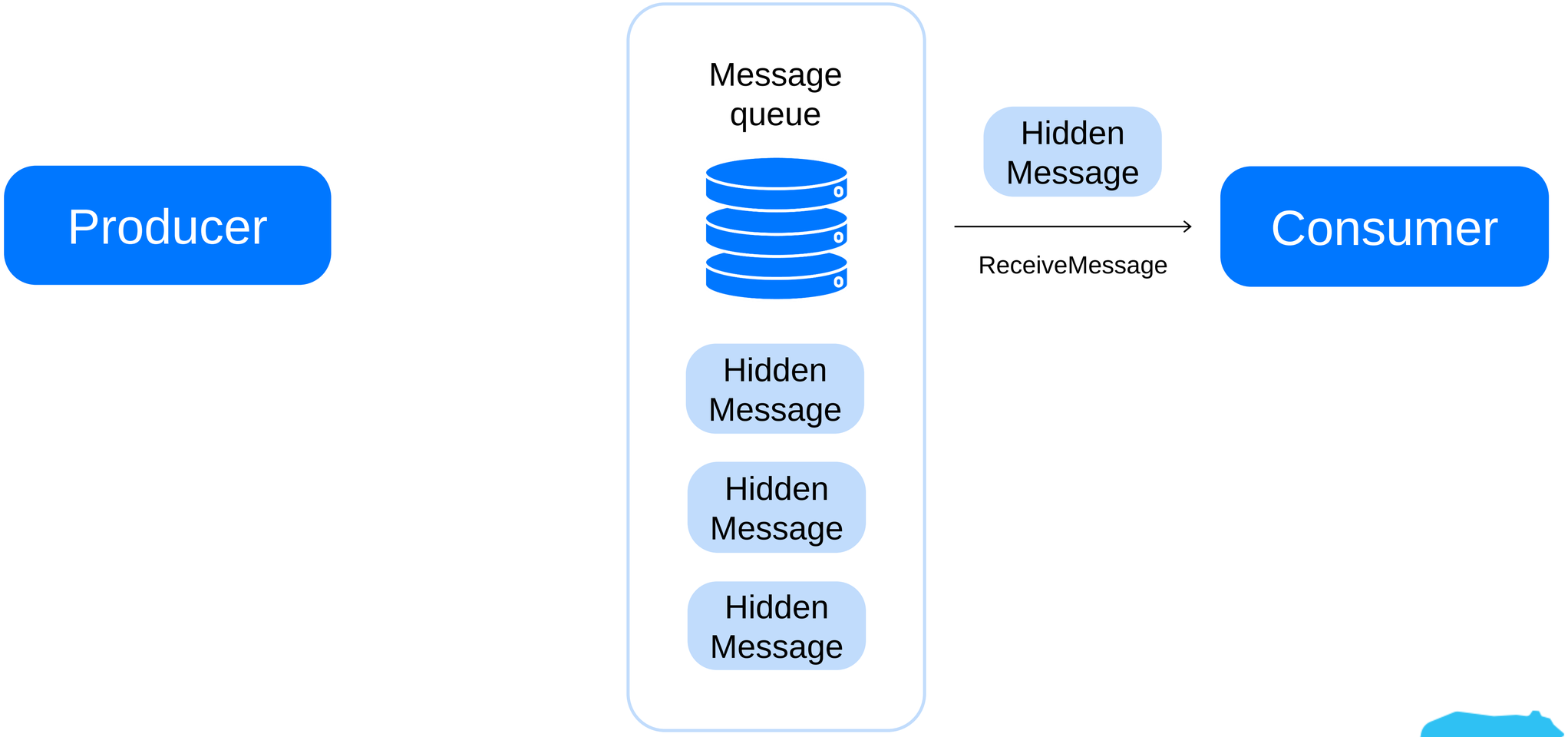
Создание очереди



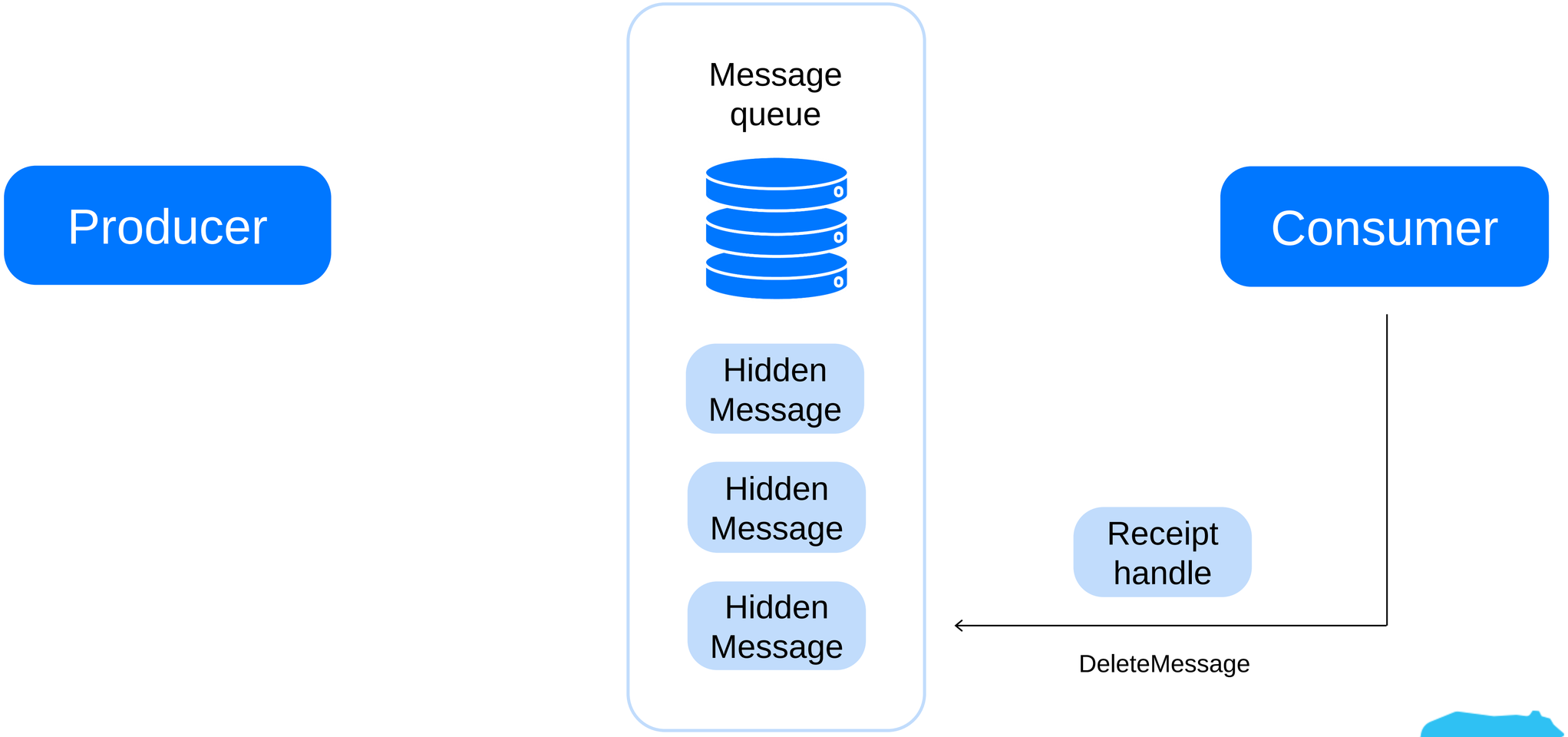
Запись сообщения



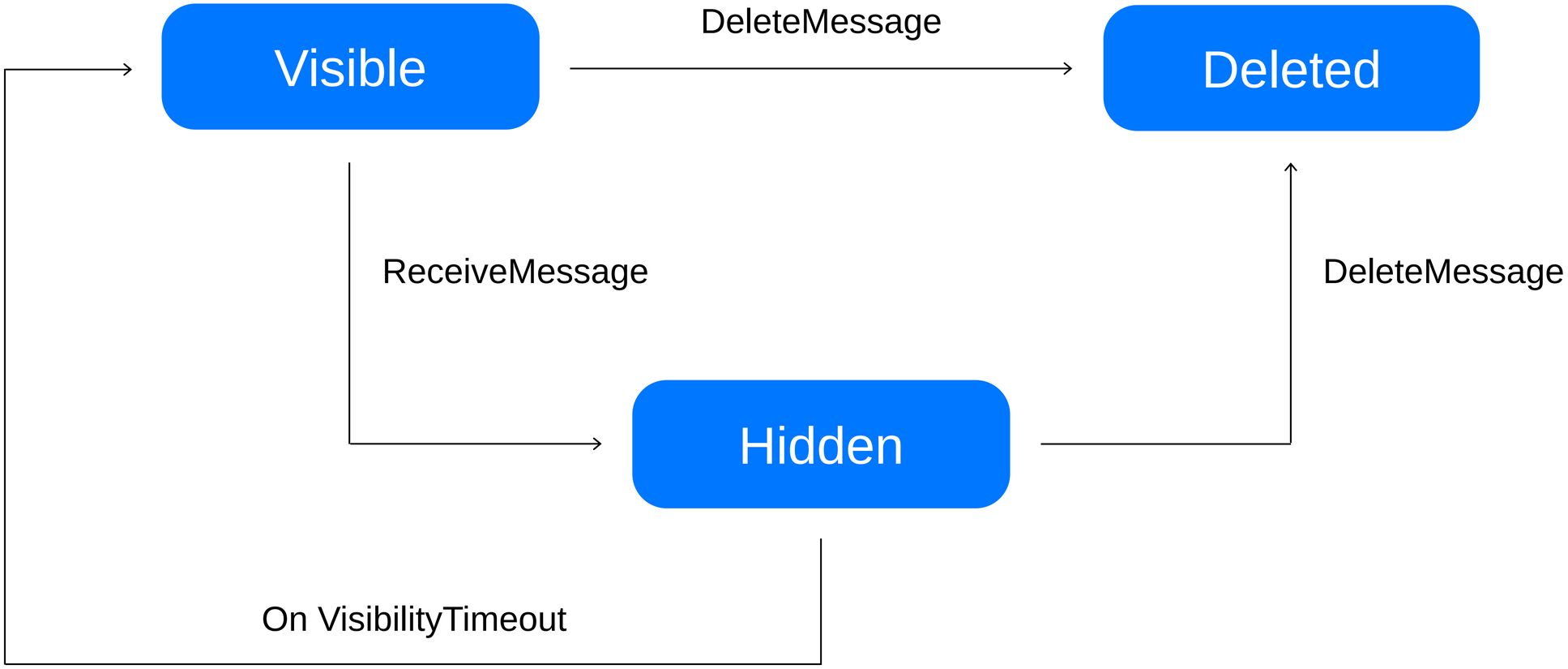
Чтение сообщения



Удаление сообщения



Visibility timeout



05

Внутреннее устройство YMQ

Платформа для Yandex Message Queue

Основой YMQ является Yandex Database (YDB)

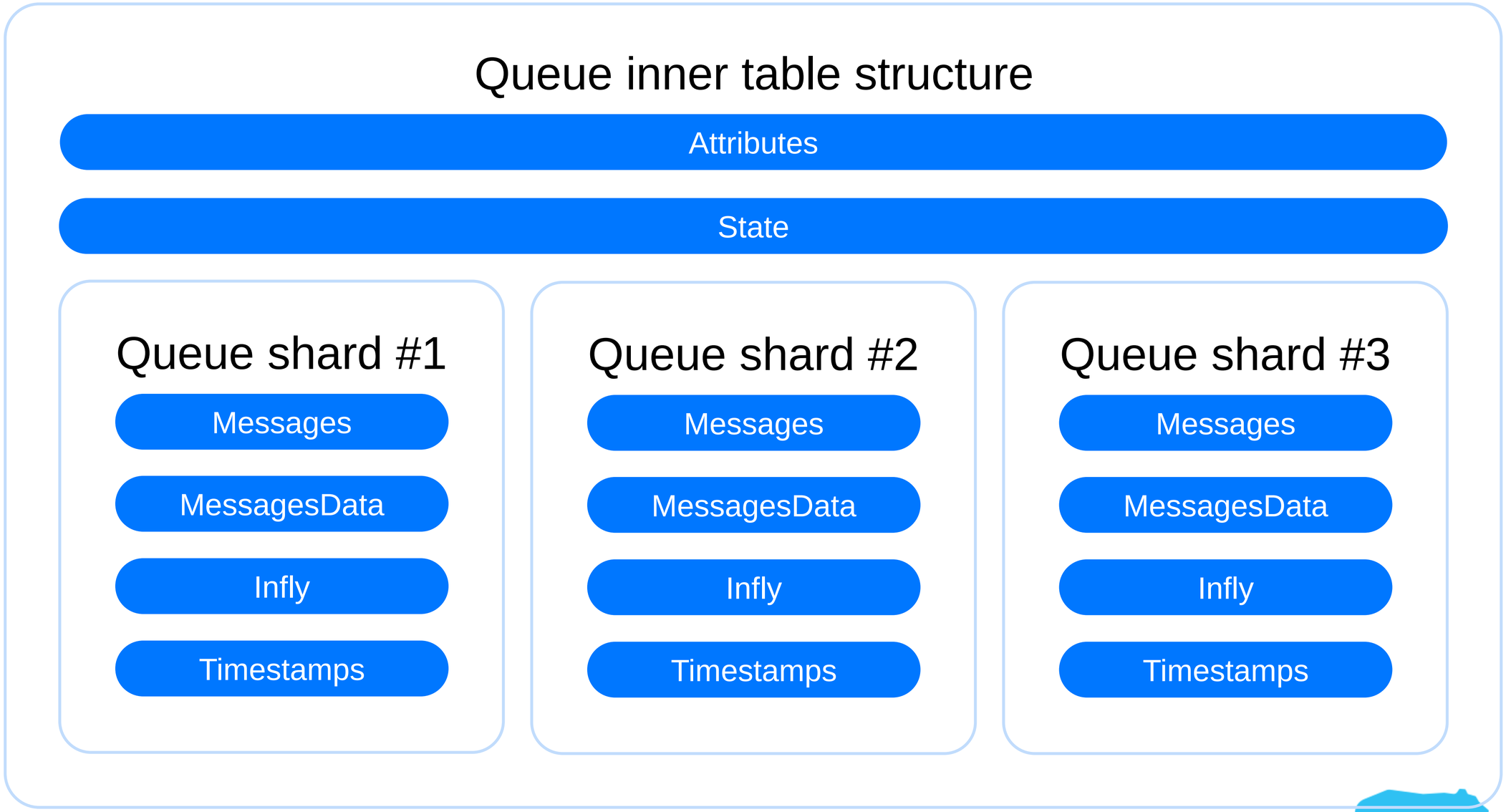
Важные свойства Yandex Database

- Распределённая
- Горизонтально масштабируемая
- Отказоустойчивая
- Поддерживает ACID-транзакции
- Поддерживает диалект SQL (YQL)
- Строго консистентная
- Очень крутая!

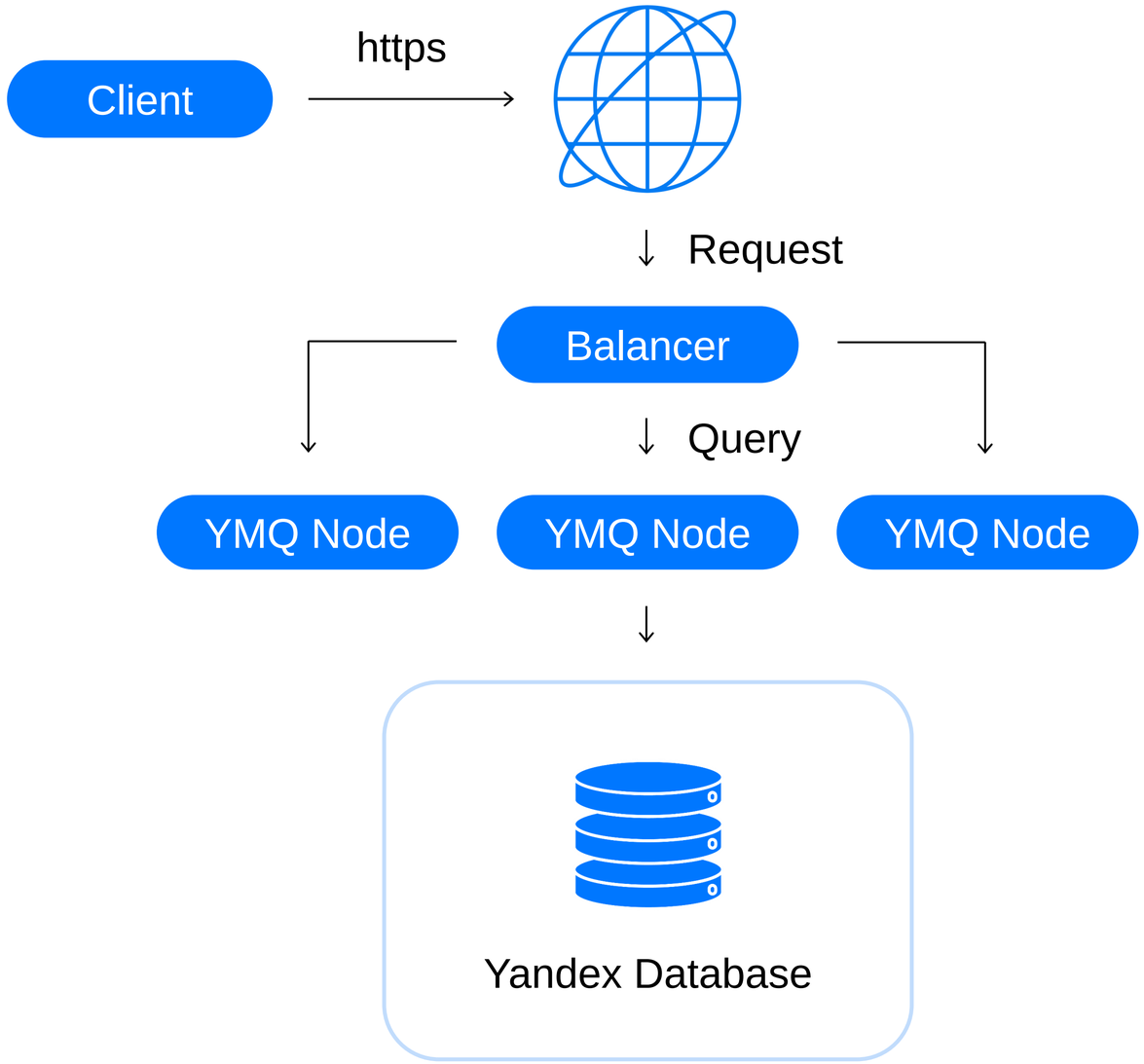
Основные идеи

- Очередь — набор шардов
- Шард — набор таблиц в базе данных
- Операции — запросы к таблицам шарда

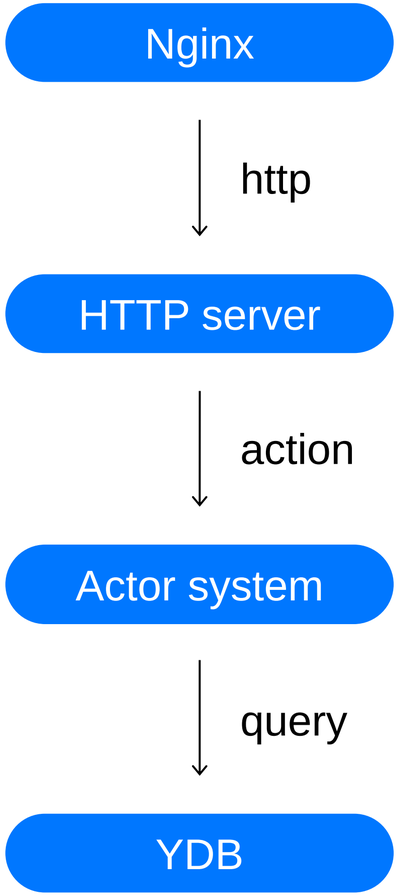
Анатомия очереди



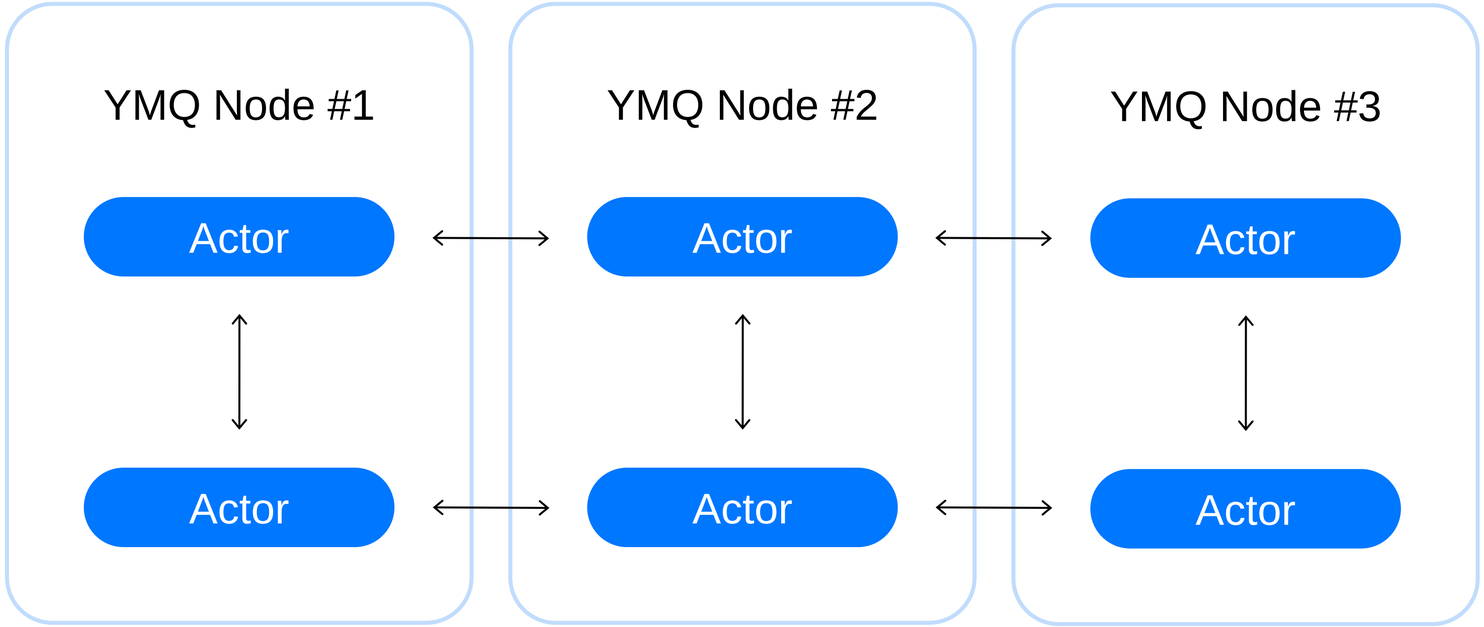
Архитектура сервиса



Внутри узла YMQ



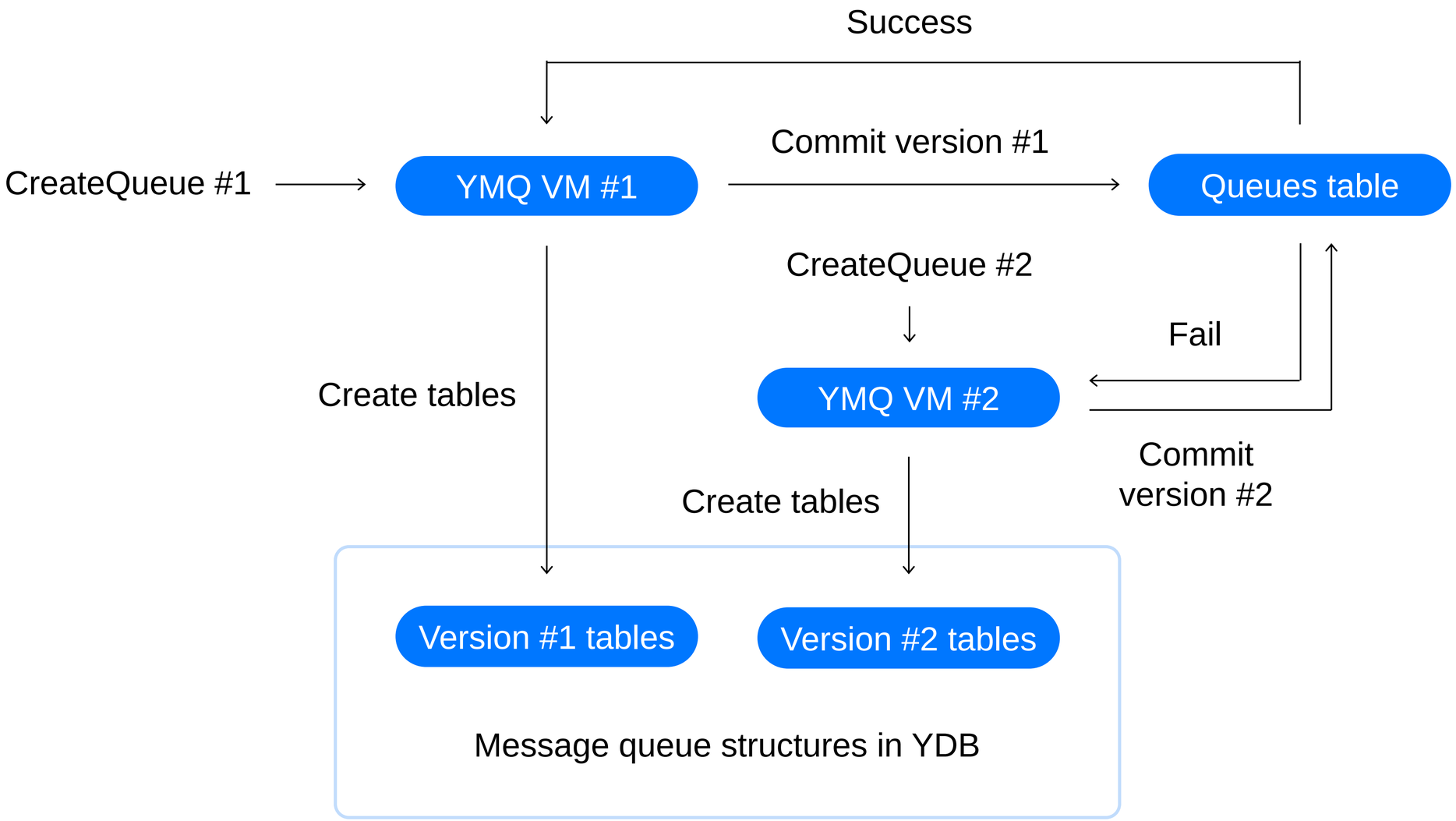
Акторная система



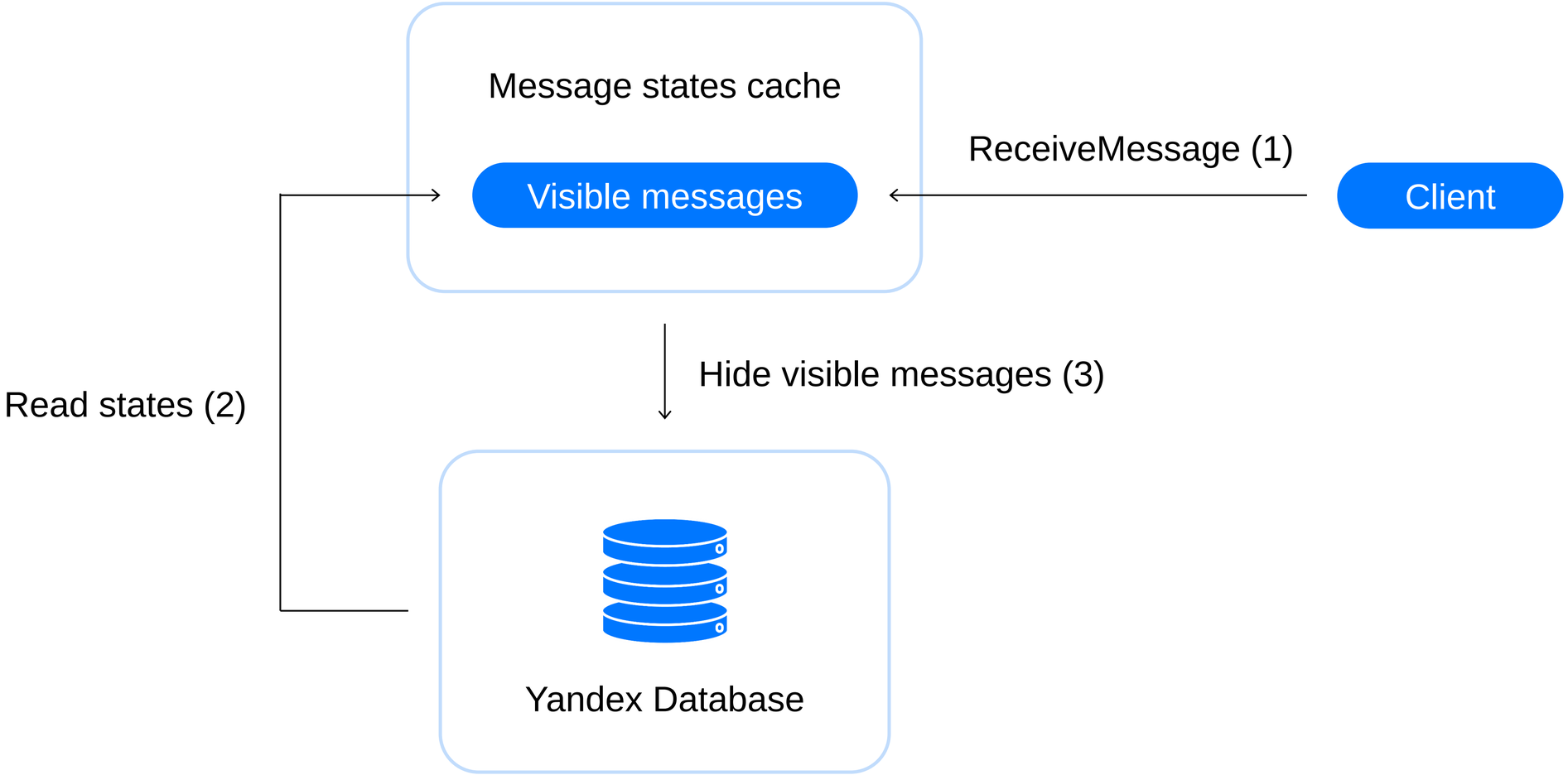
Проблема создания очередей

- В YDB (пока что) нельзя транзакционно одновременно создать несколько таблиц и записать данные
- Хост может упасть во время работы CreateQueue

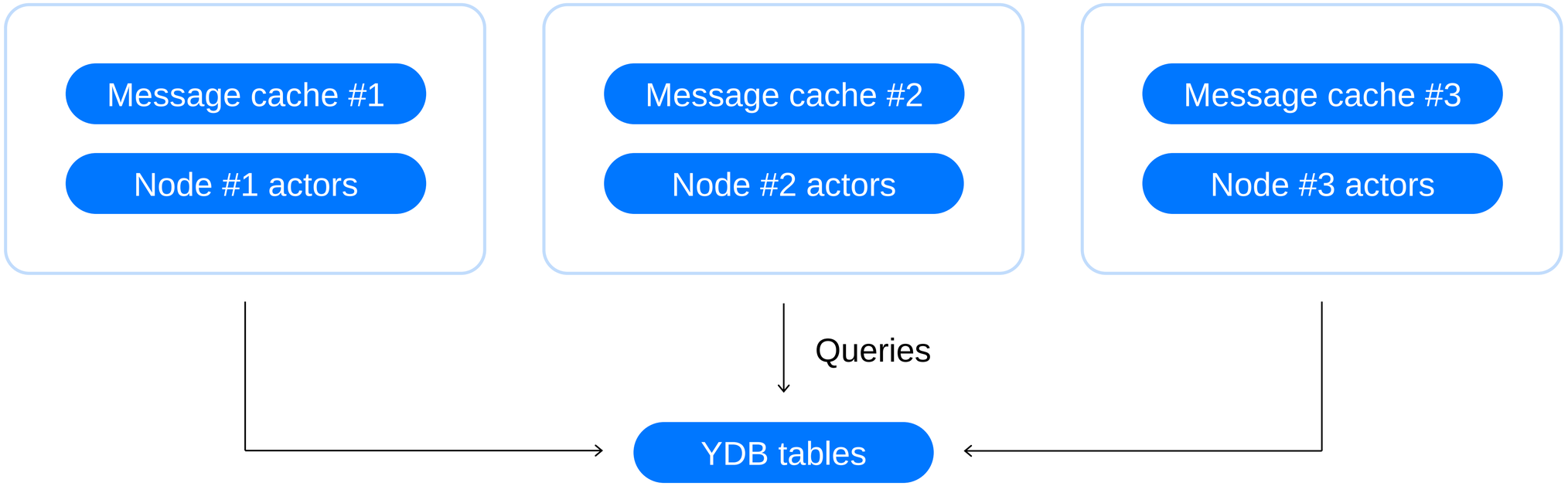
Отказоустойчивое создание очередей



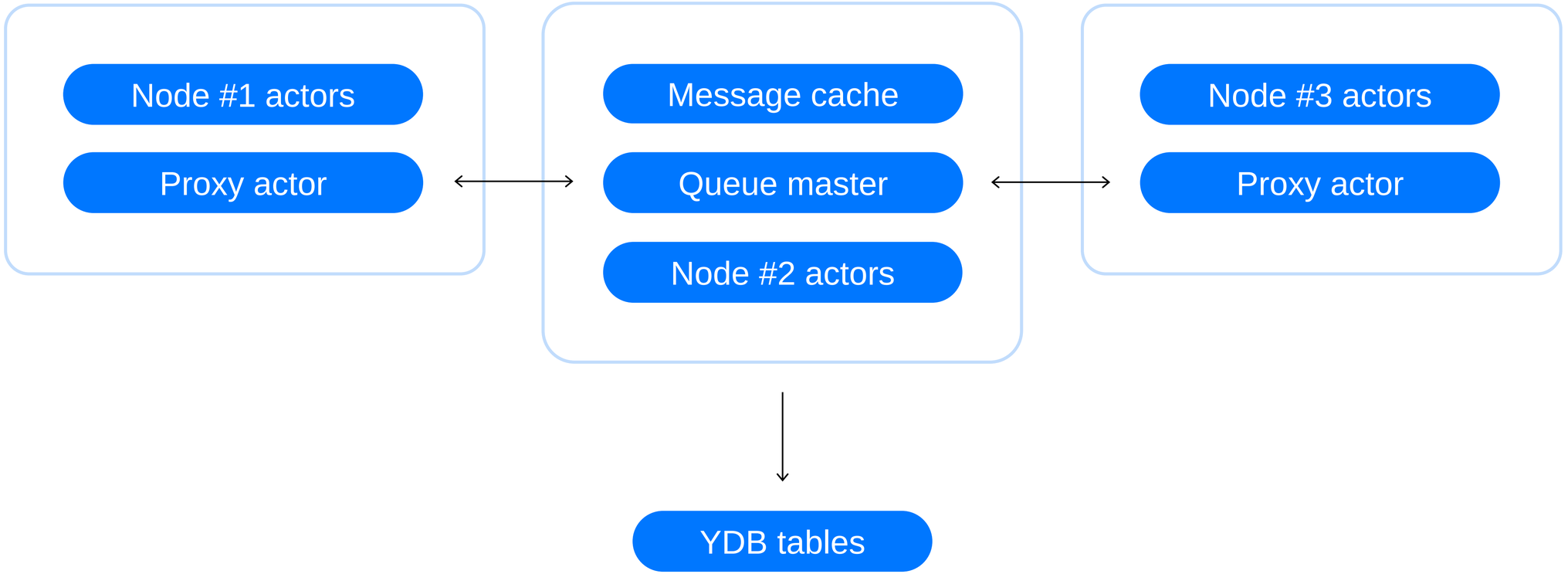
Кэш сообщений



Проблема согласованности кэшей сообщений



Решение: мастера очередей



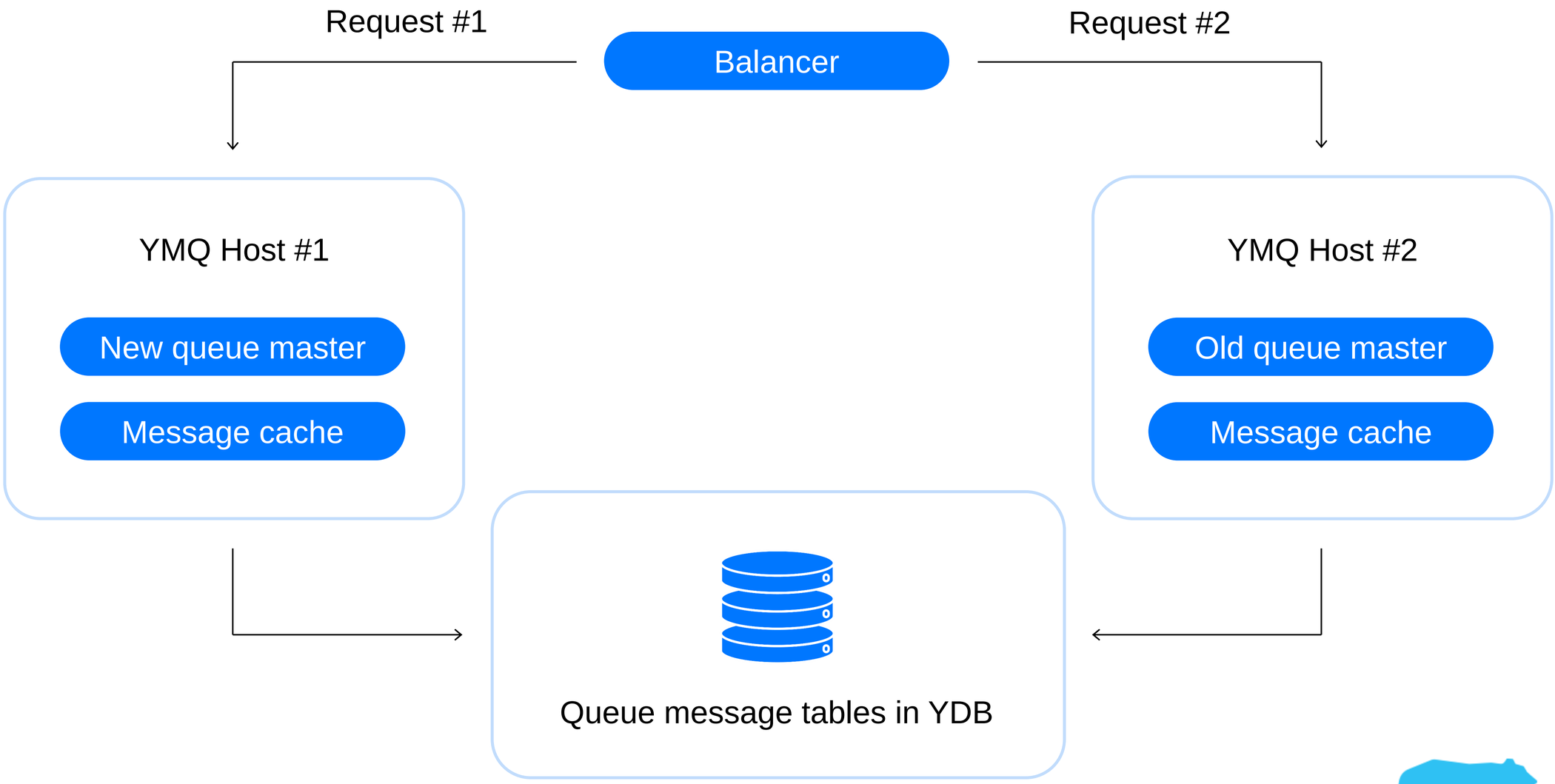
Функции мастера очереди

- Кэширование информации
- Единая точка управления запросами
- Место сбора пользовательских метрик
- Оптимизация потока запросов к базе

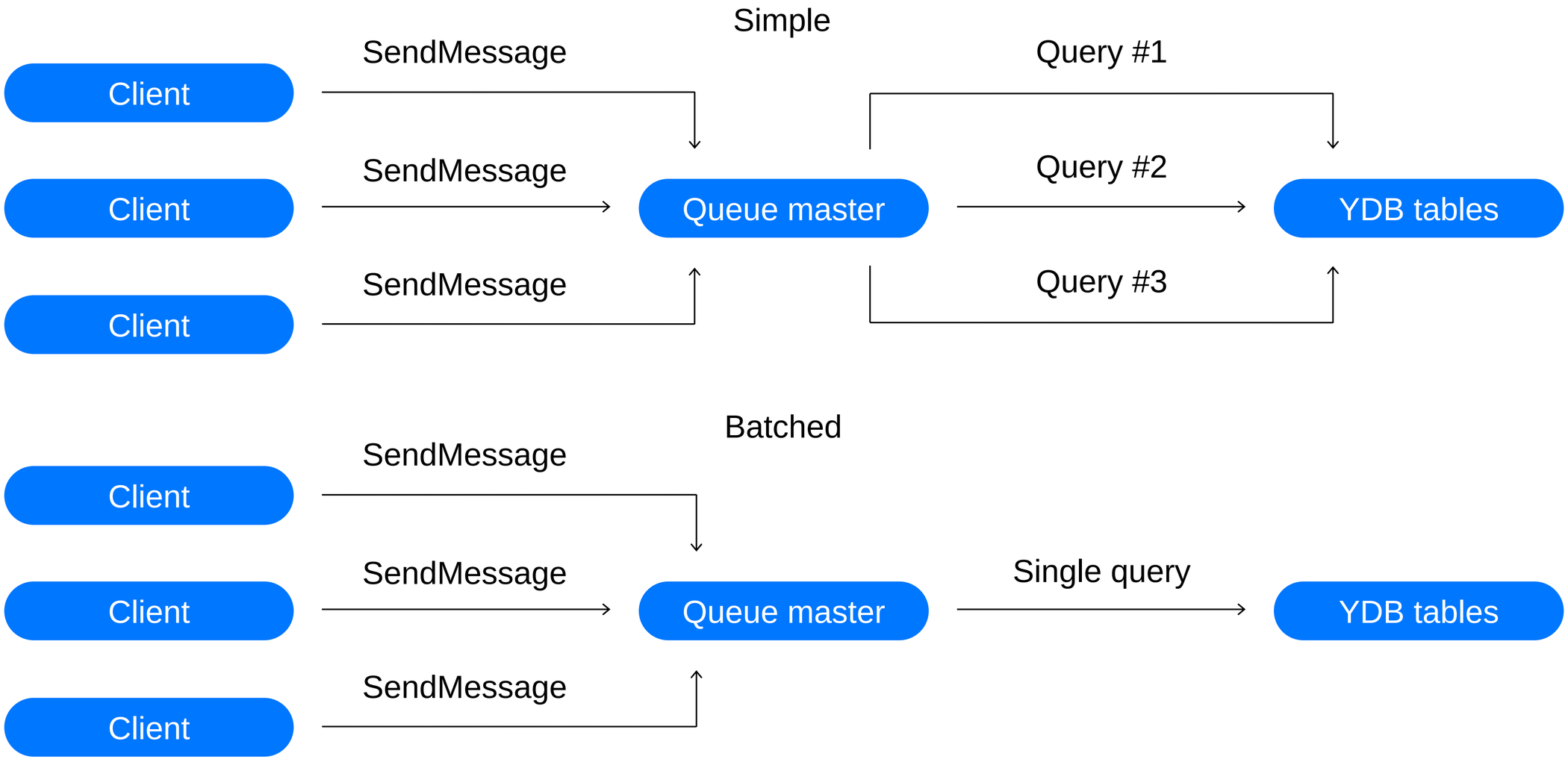
Выбор мастера

- Выбирается за счет механизмов YDB
- Мастера быстро переедут при отказе узла

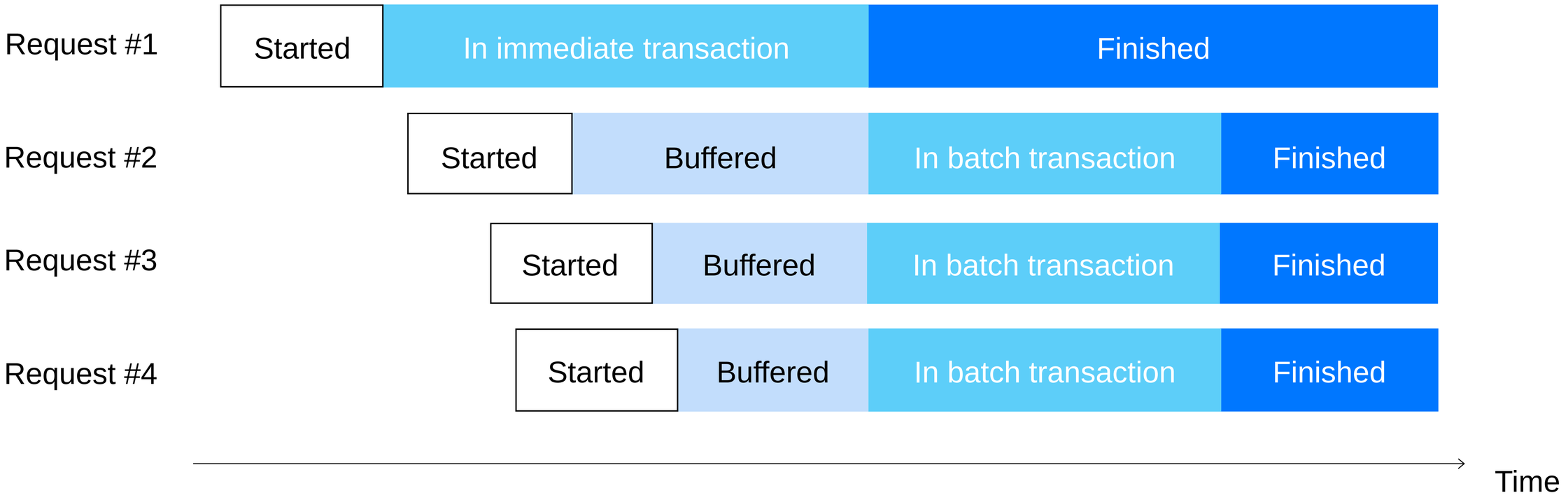
Когда мастеров в кластере несколько



Группировка однотипных запросов



Временная диаграмма батчинга



Что мы узнали сегодня

- Очередь сообщений — полезный компонент
- Exactly-once недостижима только за счет очереди
- Apache Kafka и RabbitMQ — два различных подхода
- Иногда делать своё решение выгоднее
- YMQ построен на мощной платформе YDB
- Распределённость привносит свои сложности

Яндекс Облако

Спасибо за внимание!

Василий Богонатов

Разработчик сервиса Yandex Message Queue

radix@yandex-team.ru

@drween

HighLoad++ Siberia 2019