

Skillbox

# Распределенная архитектура

**Event-Driven Architecture Style**

**Андрей Гордиенков**

Solution Architect  
ABAX

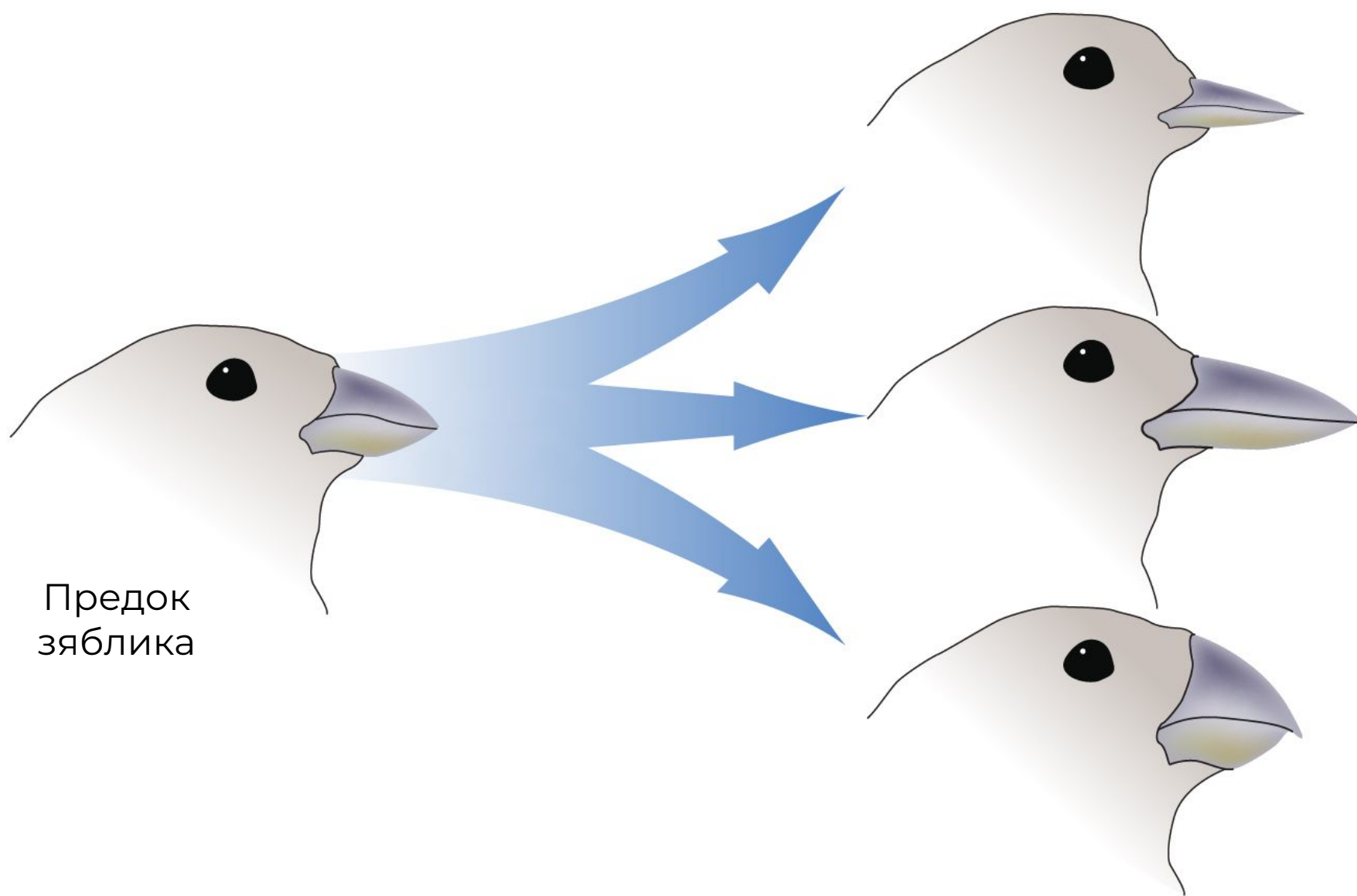
## **В прошлом видео**

- Рассмотрели архитектурный стиль Space-Based
- Затронули проблемы кэширования данных
- Кратко поговорили о плюсах асинхронных каналов связи с точки зрения обслуживания систем

## **В этом видео**

- Максимальное асинхронное взаимодействие компонентов системы в рамках подхода Event-Drive
- Понятия и принципы

# Причина - следствие



Было

Стало

Почему?

# Причина - следствие

Вчера

Сегодня

Ваш счет:

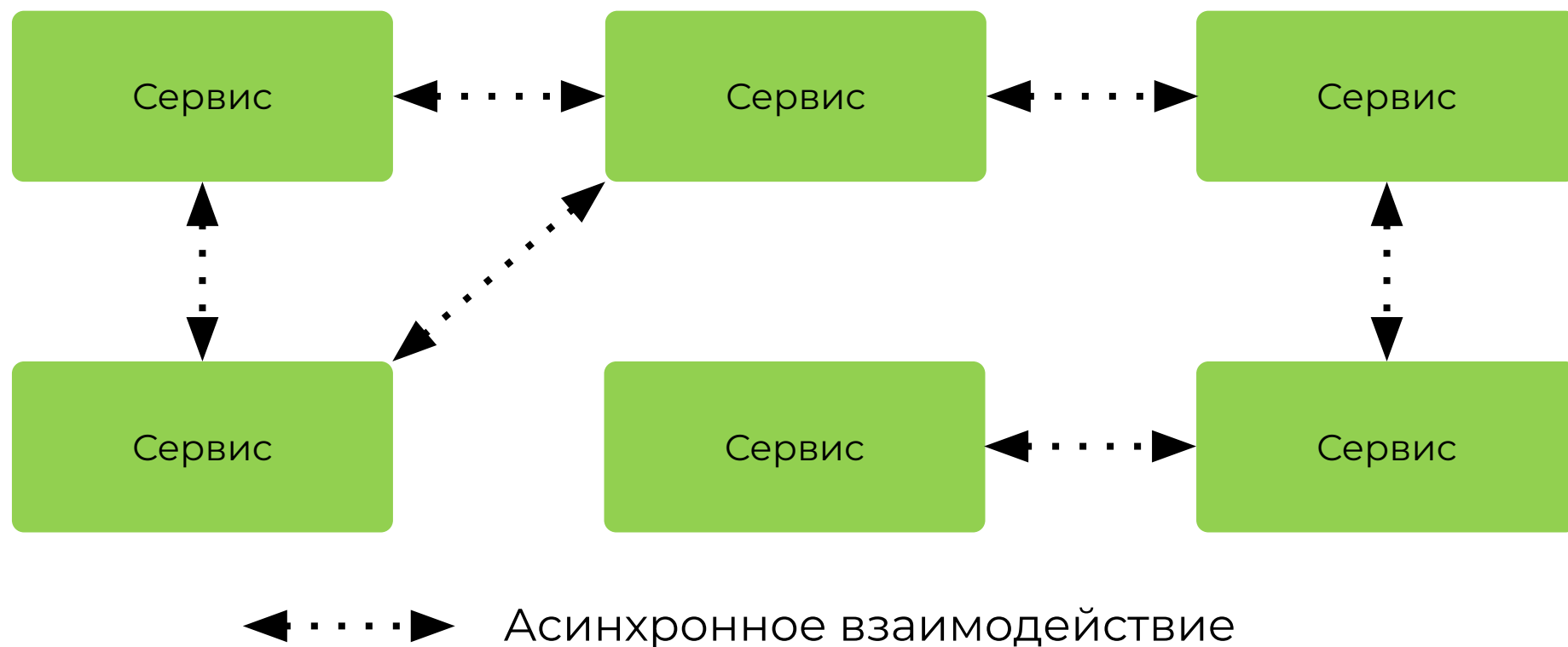
23000₹

Ваш счет:

5000₹

Почему?

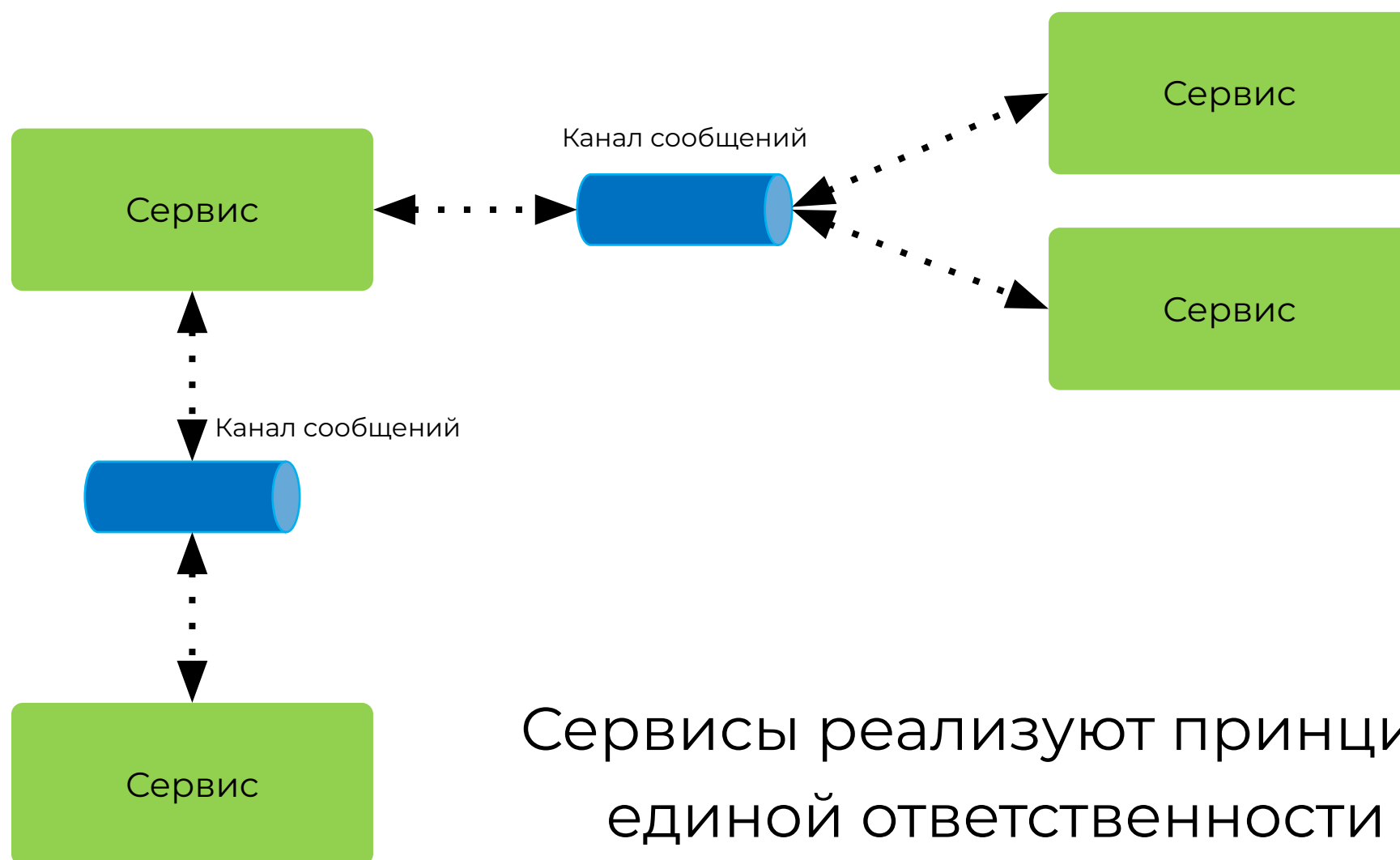
# Причина - следствие



Плюсы:

- Независимость разработки сервисов
- Внутренняя простота
- Простота разворачивания новых версий

# Event-Driven Architecture



# Event-Driven Architecture Топологии

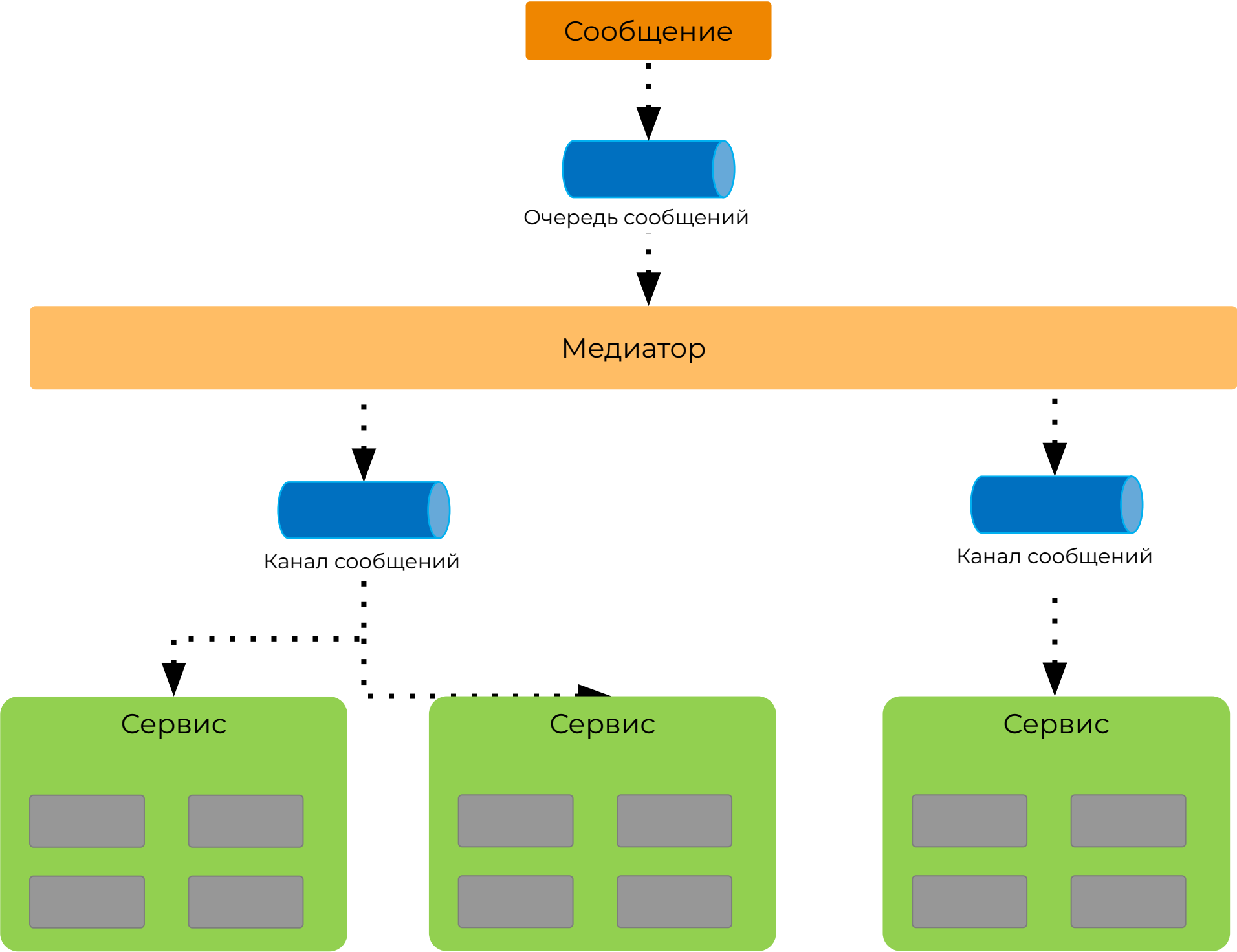
## Медиатор

- Контроль над процессом обработки сообщений

## Брокер

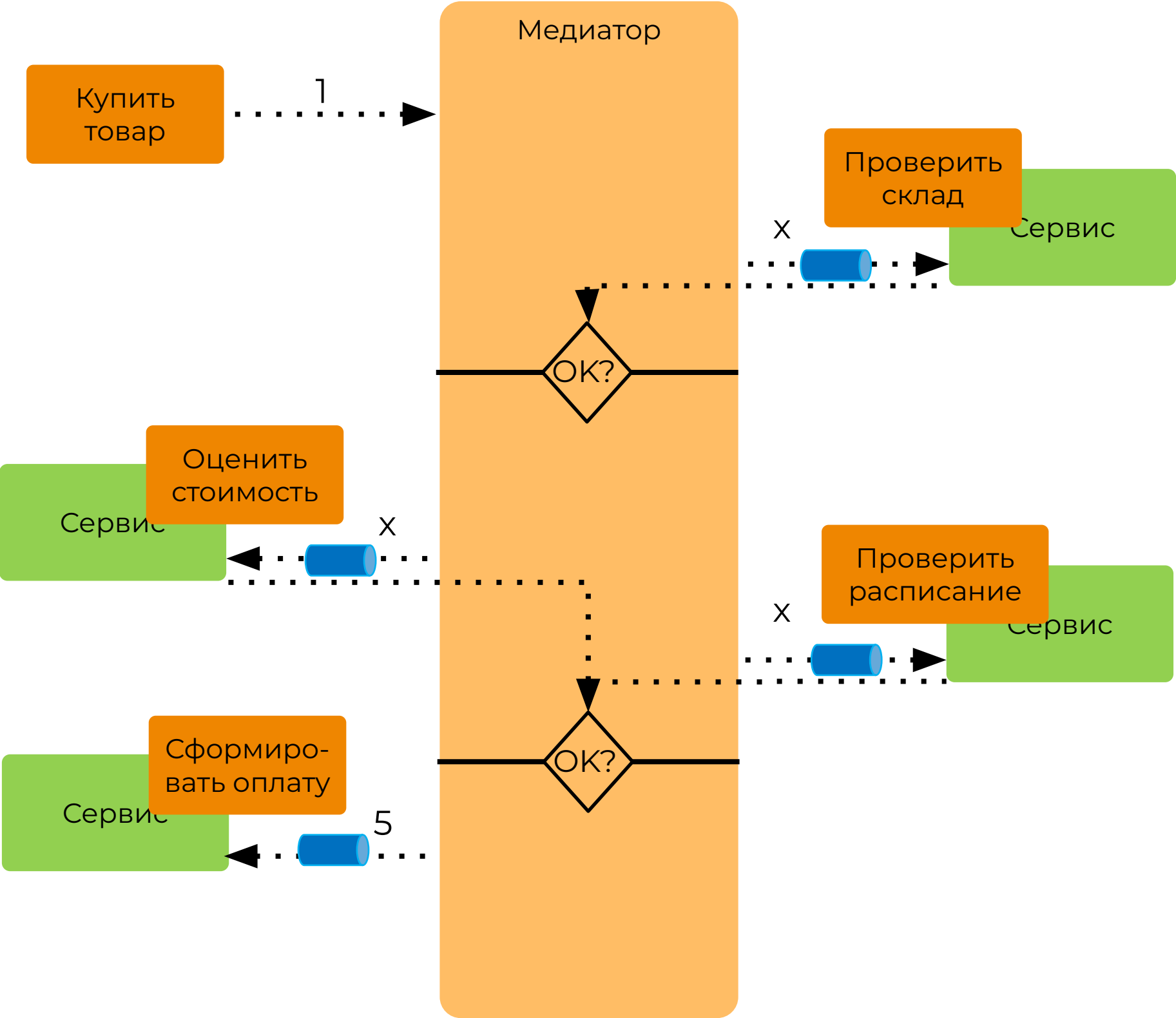
- Обратная связь
- Динамическое управление обработки сообщений

# Медиатор

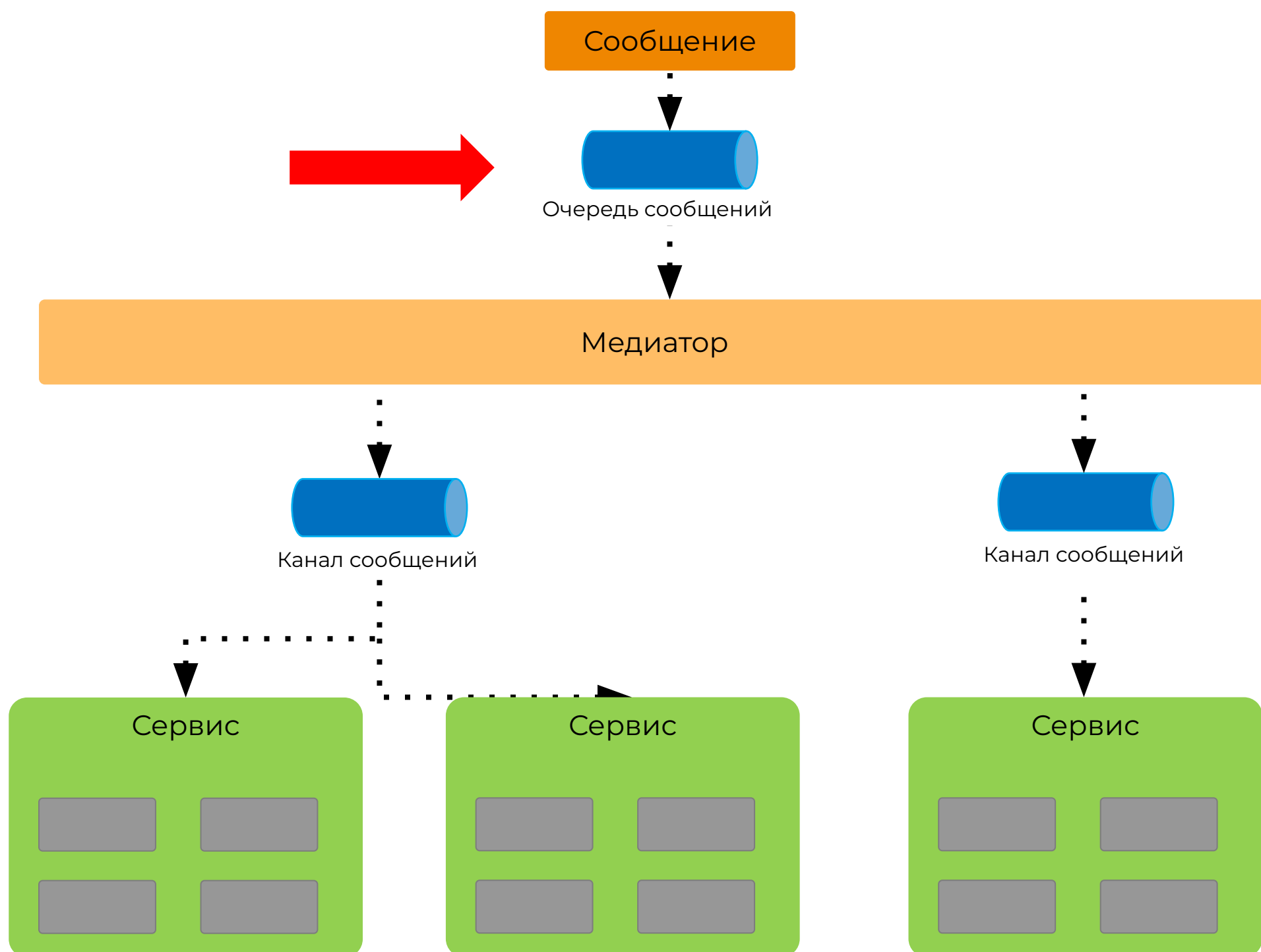




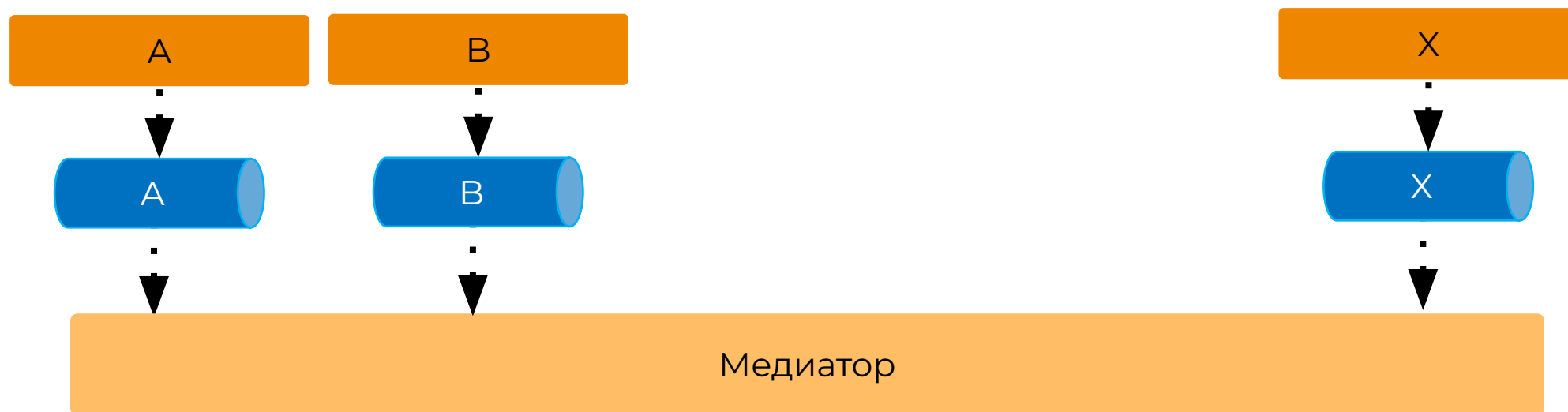
# Медиатор



# Очередь сообщений



# Очередь сообщений

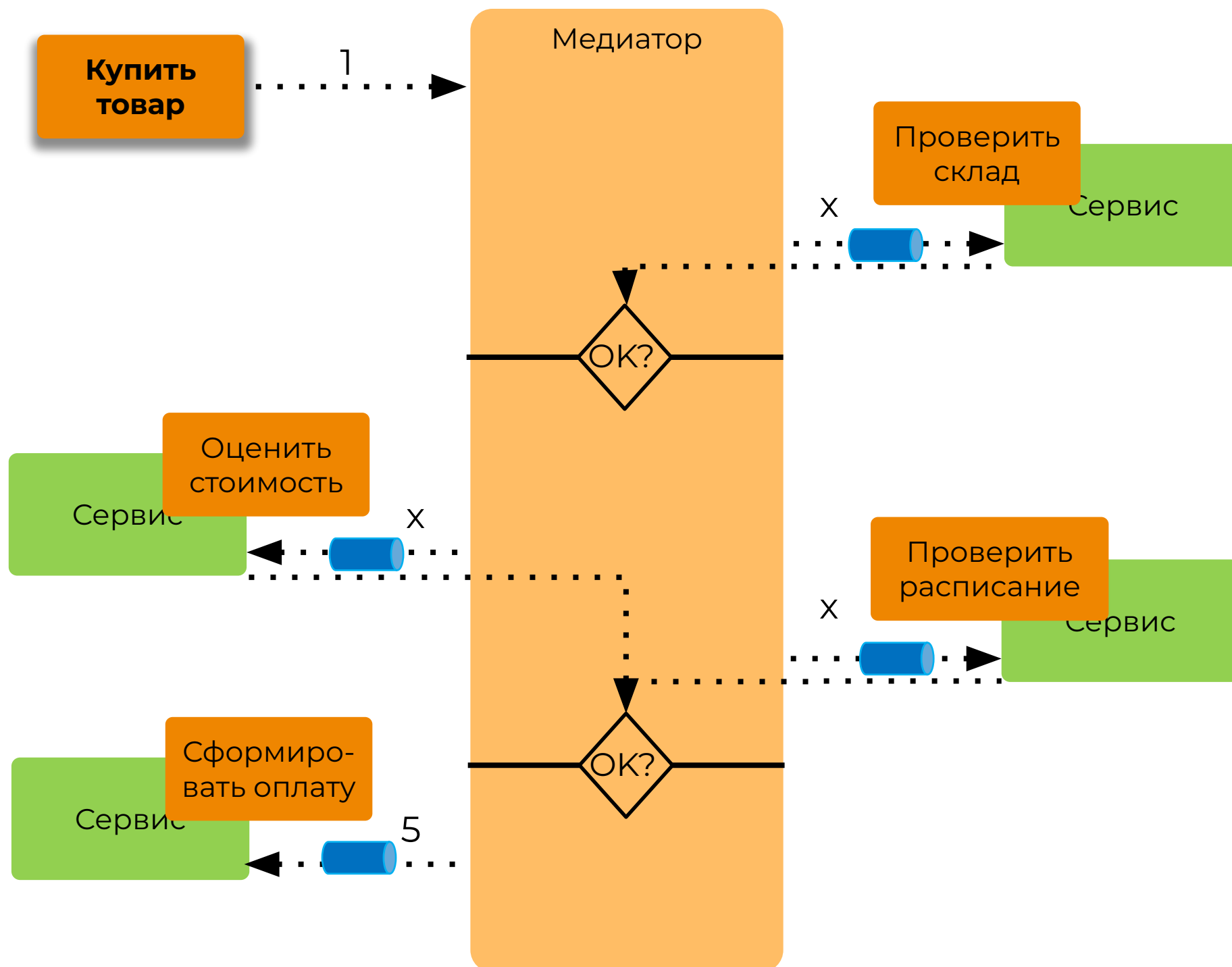


Каждому типу сообщений своя очередь.

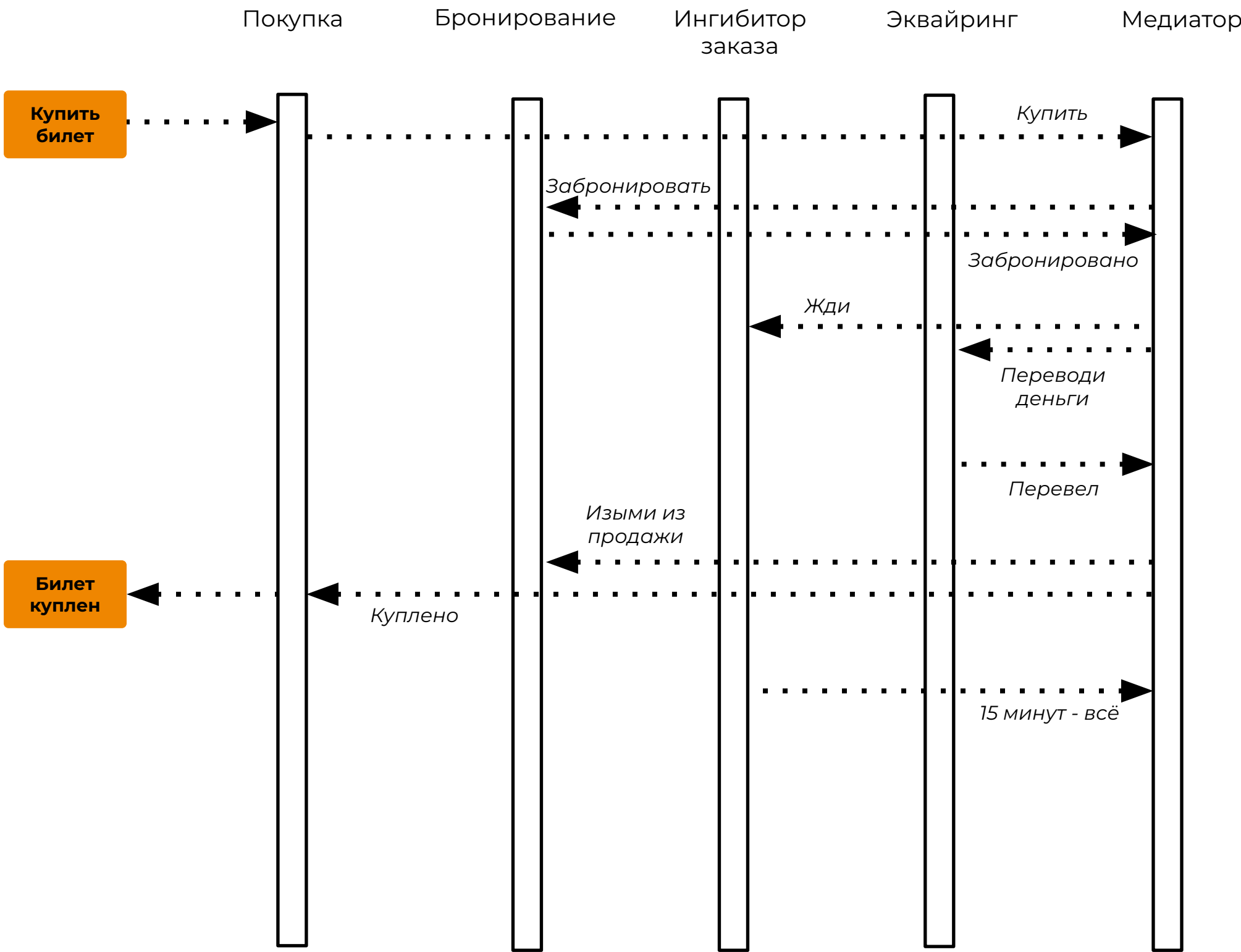
Реализация:

- Message Queue решения
- Веб-сервис
- И то и другое сразу

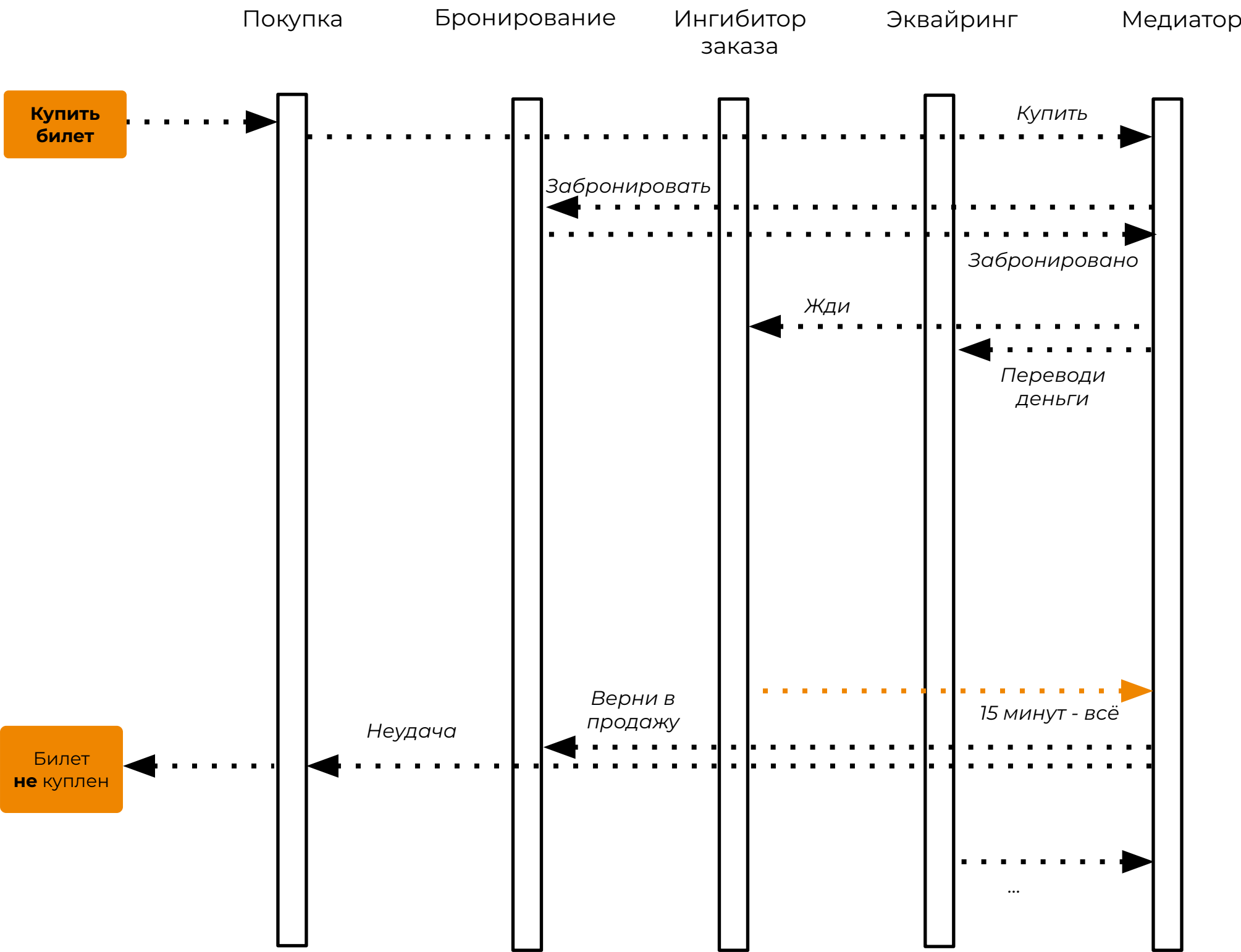
# Иницилирующее сообщение только одно



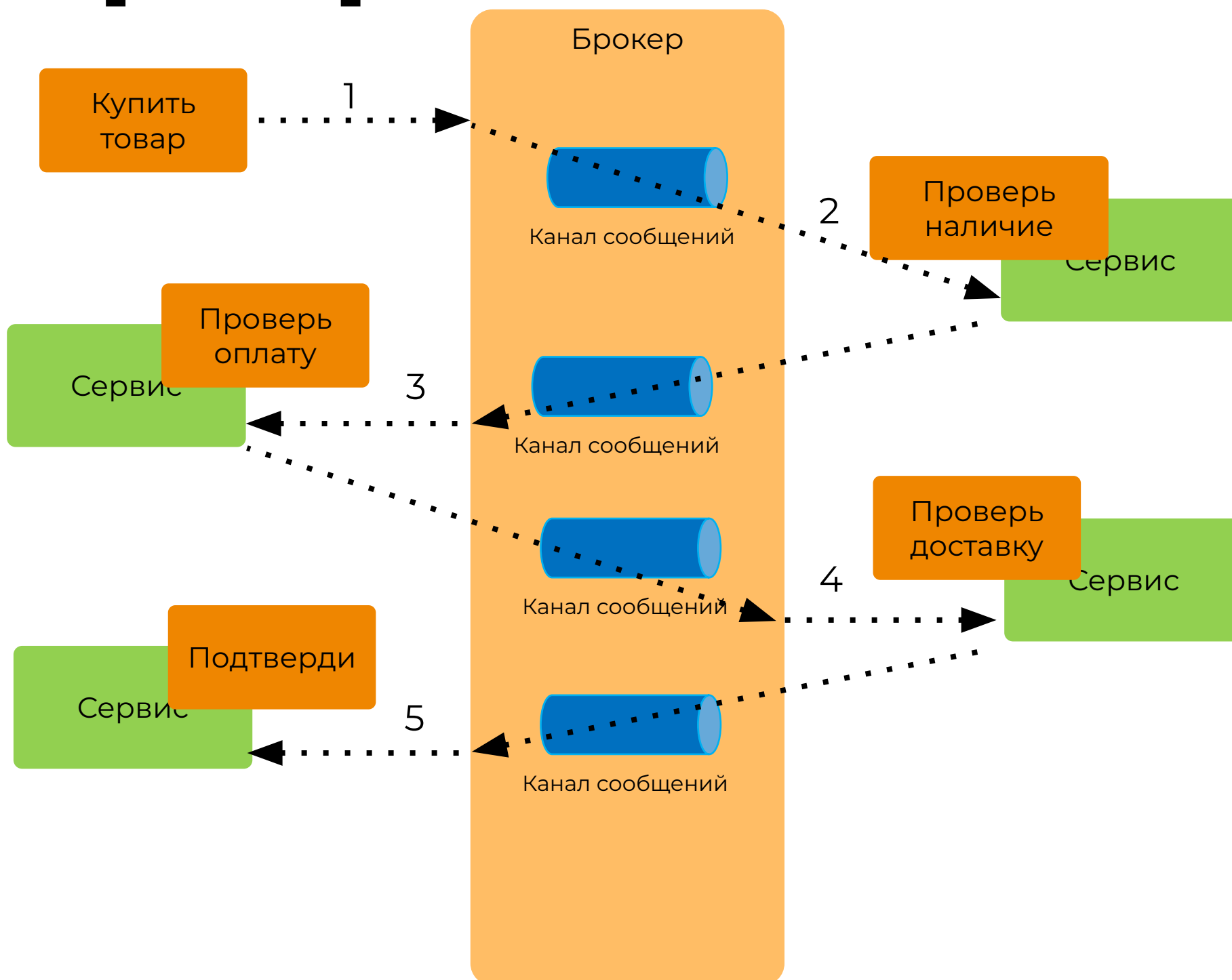
# Пример



# Пример

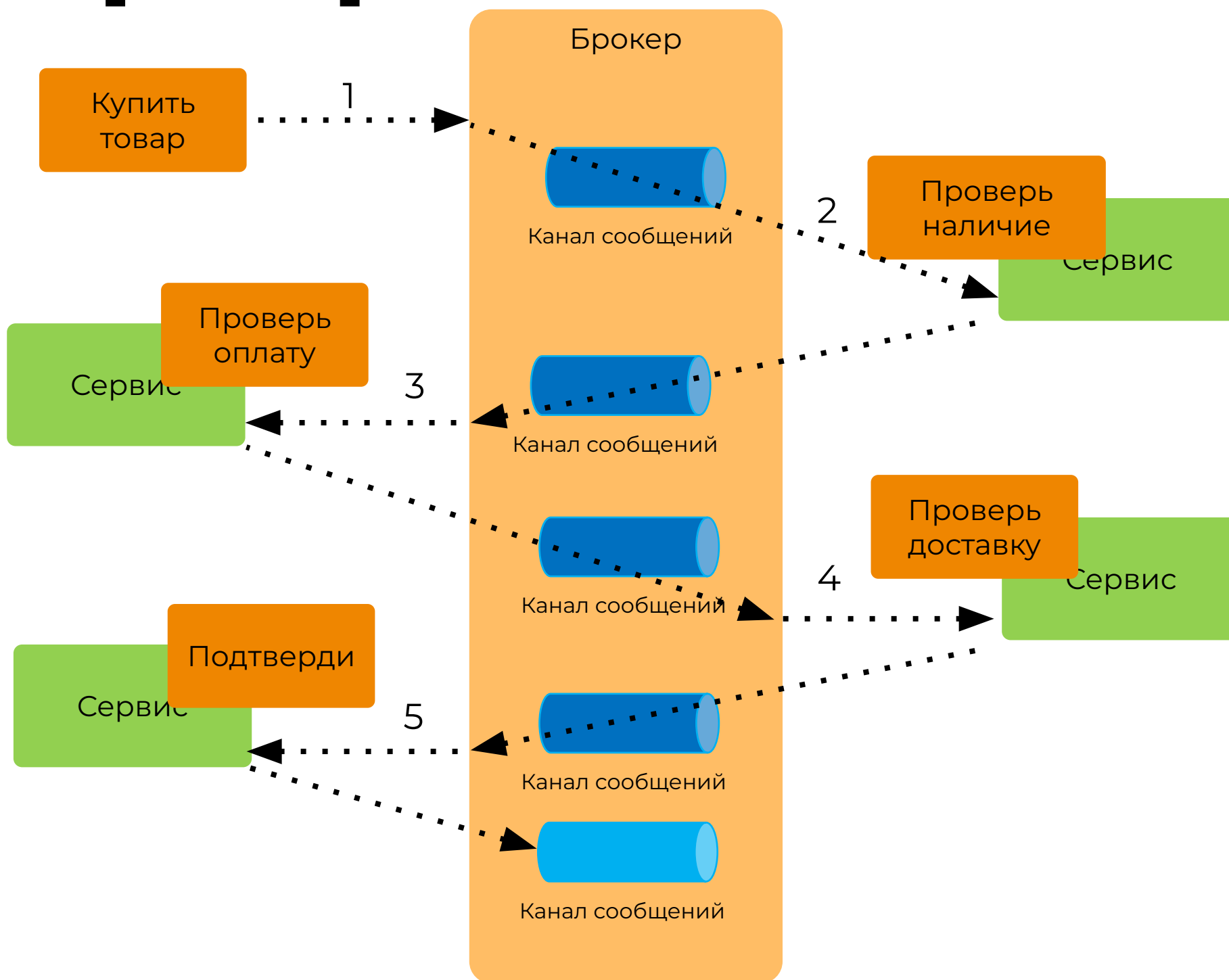


# Брокер



- Обеспечивает только доставку сообщений
- Легковесное решение

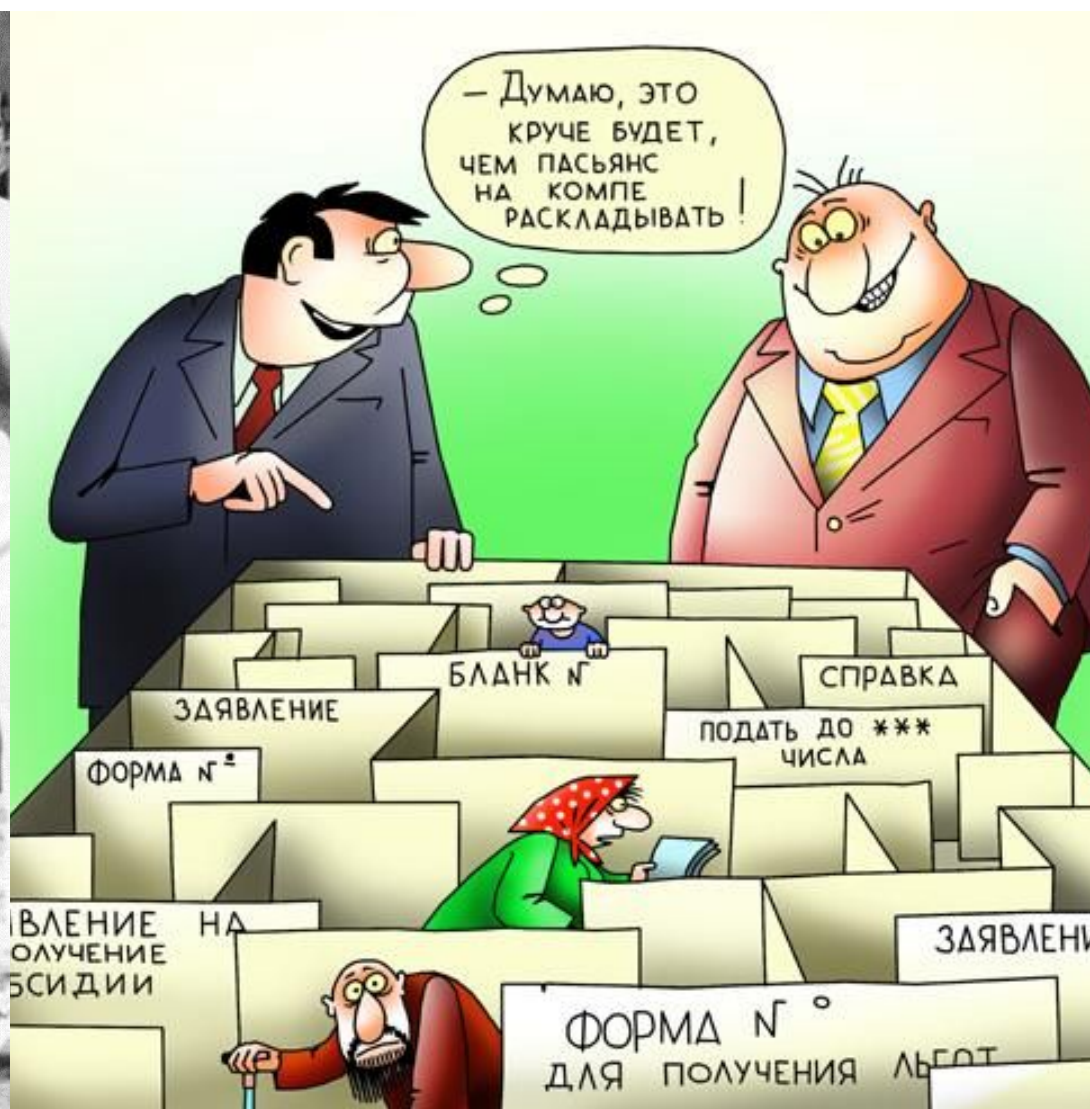
# Брокер



- Обеспечивает только доставку сообщений
- Легковесное решение

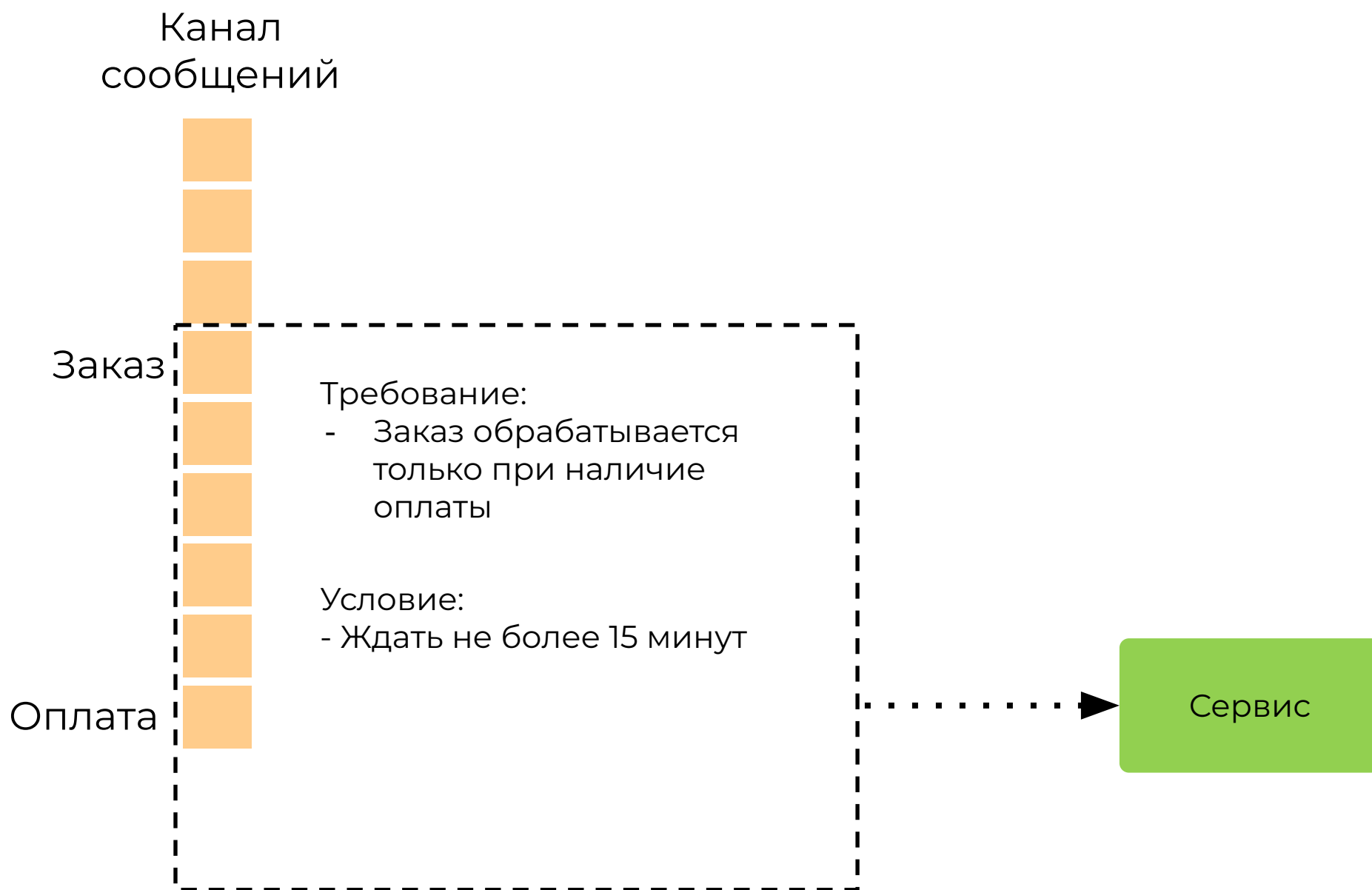


# Брокер vs Медиатор



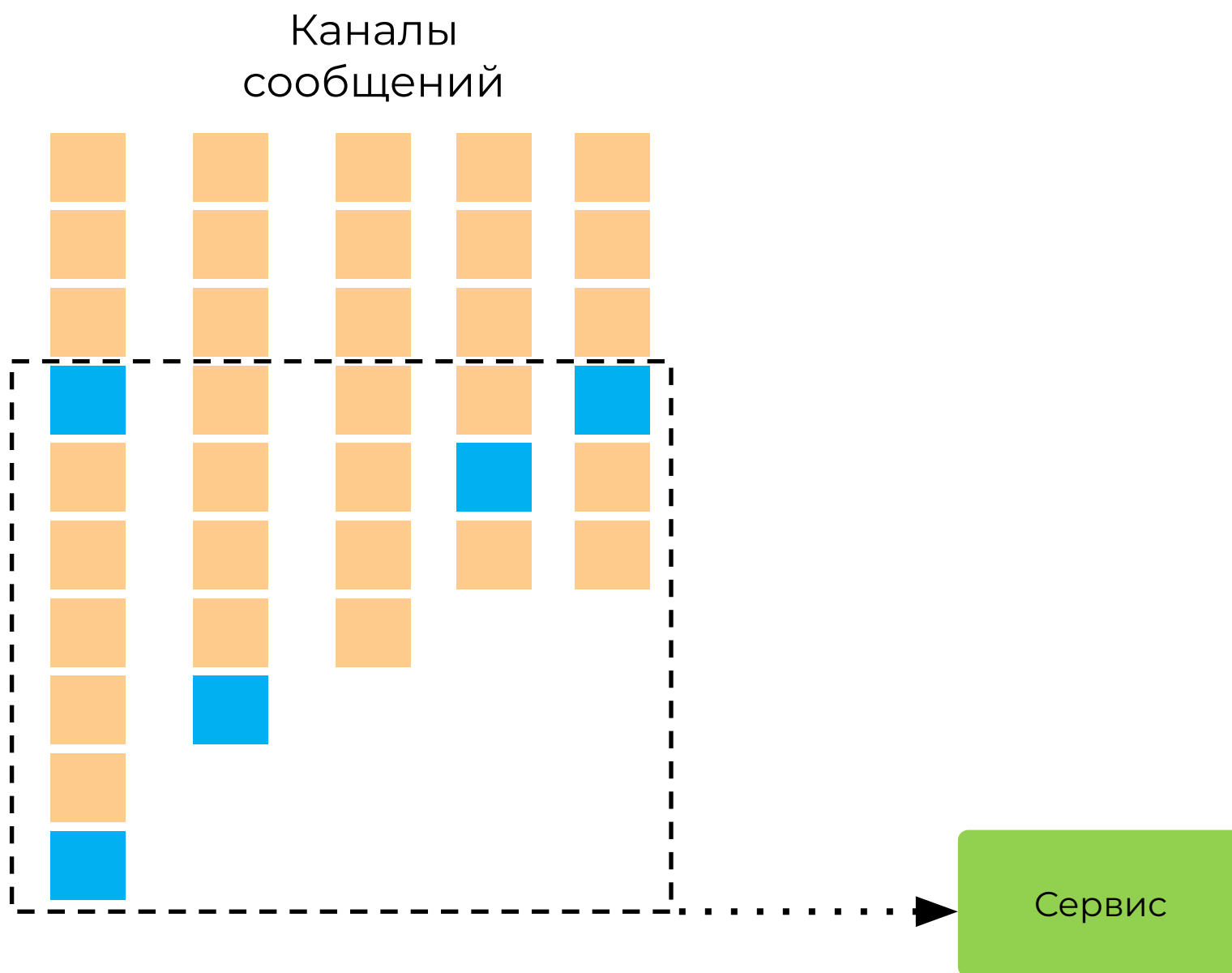
# Complex Event Processing

## Временные окна



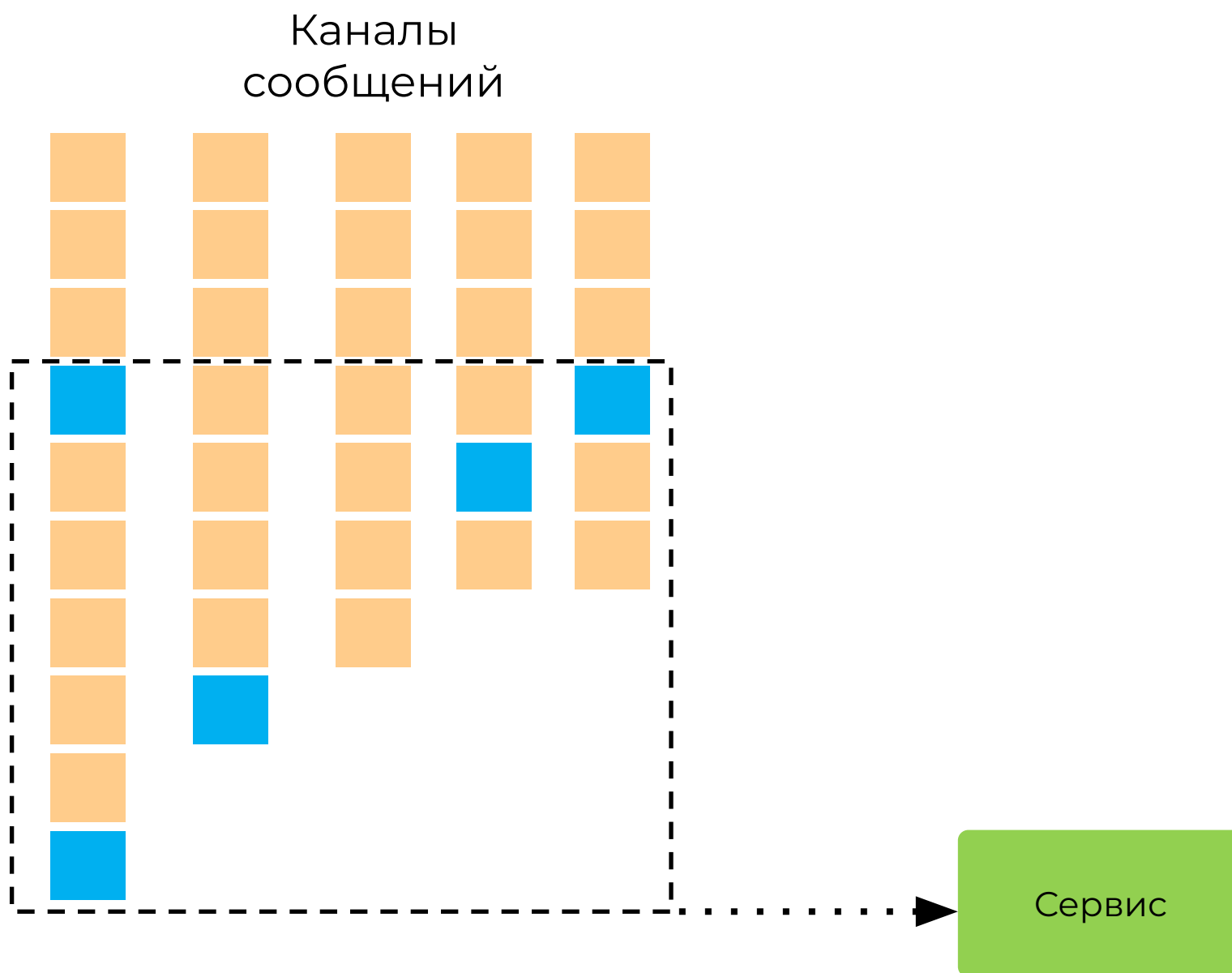
# Complex Event Processing

# Количественные окна



# Complex Event Processing

Количественные окна



# Complex Event Processing

## Применение

- Обнаружение атак на сеть
- Подозрительная активность по счету в банке

Области, где применяется сложная корреляция сообщений

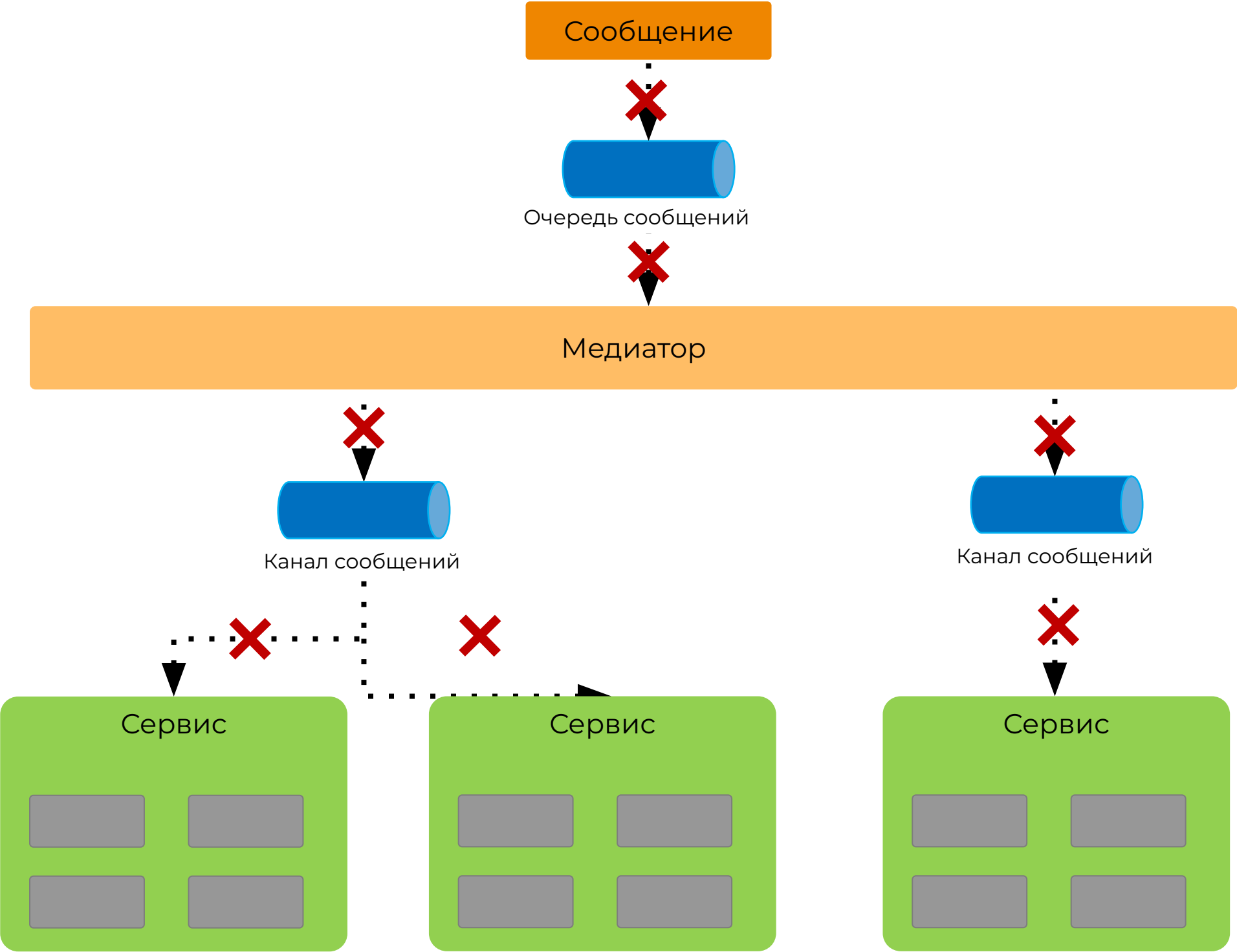
Но стоимость и сложность применения очень высока.

# Выводы

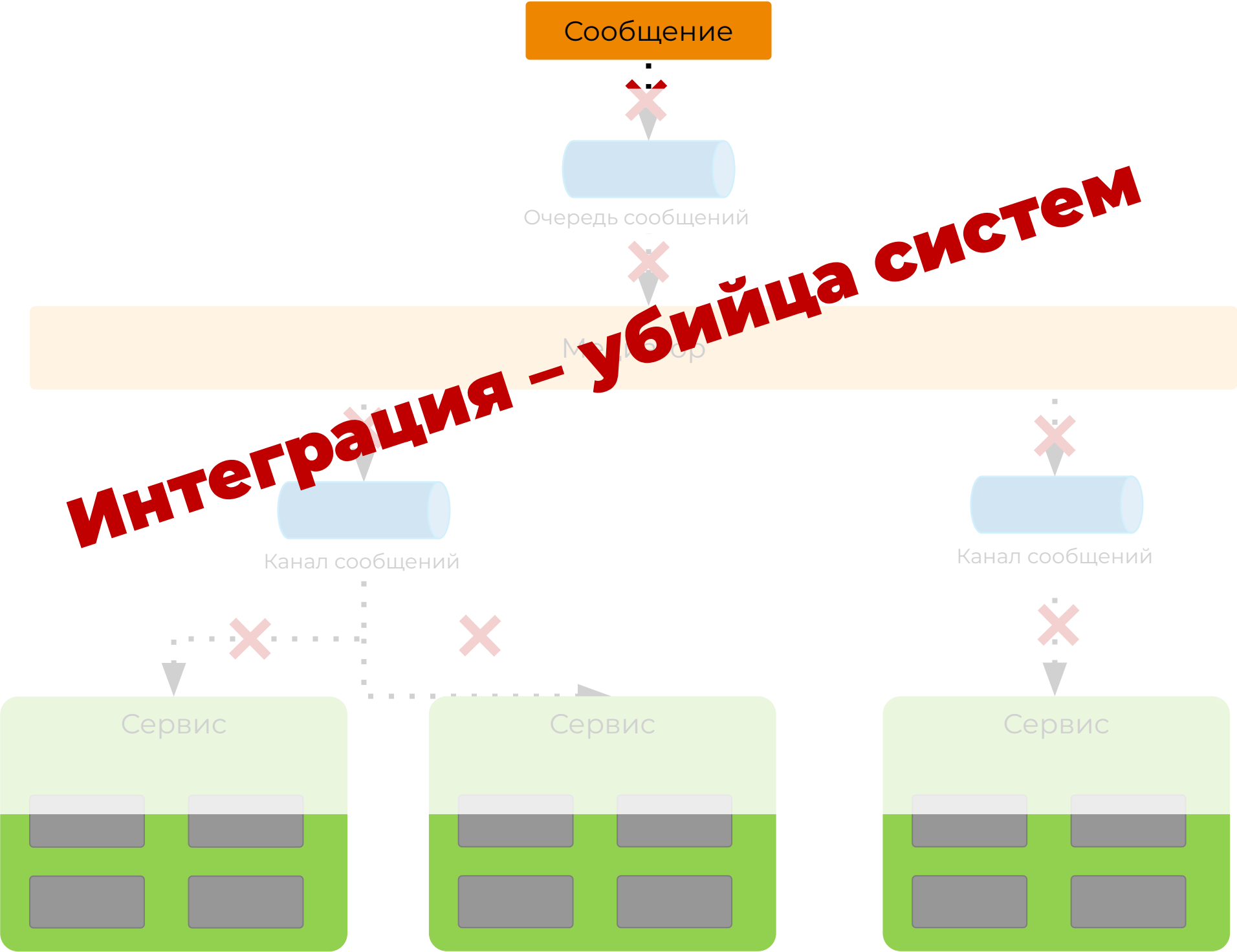
## Плюсы:

- Экстремальная независимость процессов обработки
- Точки расширения функционала создаются естественным образом
- Легкость расширения функционала
- Легкость наращивания мощностей по обработке сообщений
- Легкость разворачивания новых версий
- Хорошая тестируемость на наличие нежелательных эффектов
- Отказоустойчивость

# Отказоустойчивость



# Отказоустойчивость





# Выводы

## Минусы:

- Слабая согласованность данных на верхнем уровне системы
- Сложность организации транзакционной записи данных
- Управление (создание и поддержка) контрактов сервисов
- Высокая когнитивная сложность схем общения
- Сложность тестирования и отладки на уровне всей системы

# Самое важное

Реализация Event-Driven стиля требует много сил и времени для большой системы.

Но если всё сделано верно, то легкость бизнес-изменений вас порадует.

# CQRS + ES

- Commands and Queries Responsibility Segregation (CQRS)
- Event Sourcing (ES)

Хорошие шаблоны, но необязательные для Event-Driven стиля. Мы их разберем в другом модуле.

# Что дальше?

- Рассмотрим один из самых «свежих» стилей архитектуры - Микросервисы