

Predictive maintenance

Лекция 2

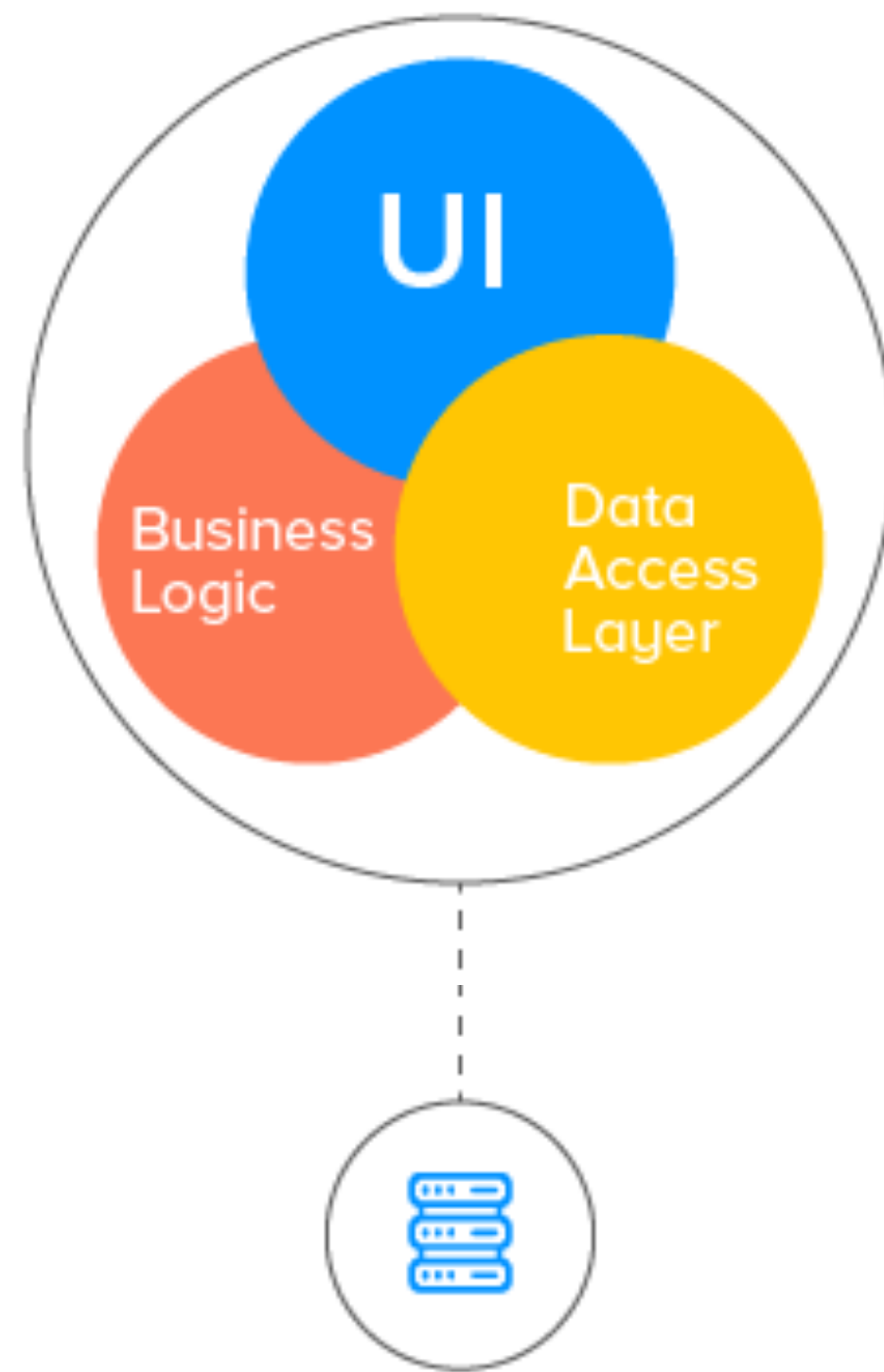
Взаимодействие сервисов

Власов Кирилл Вячеславович

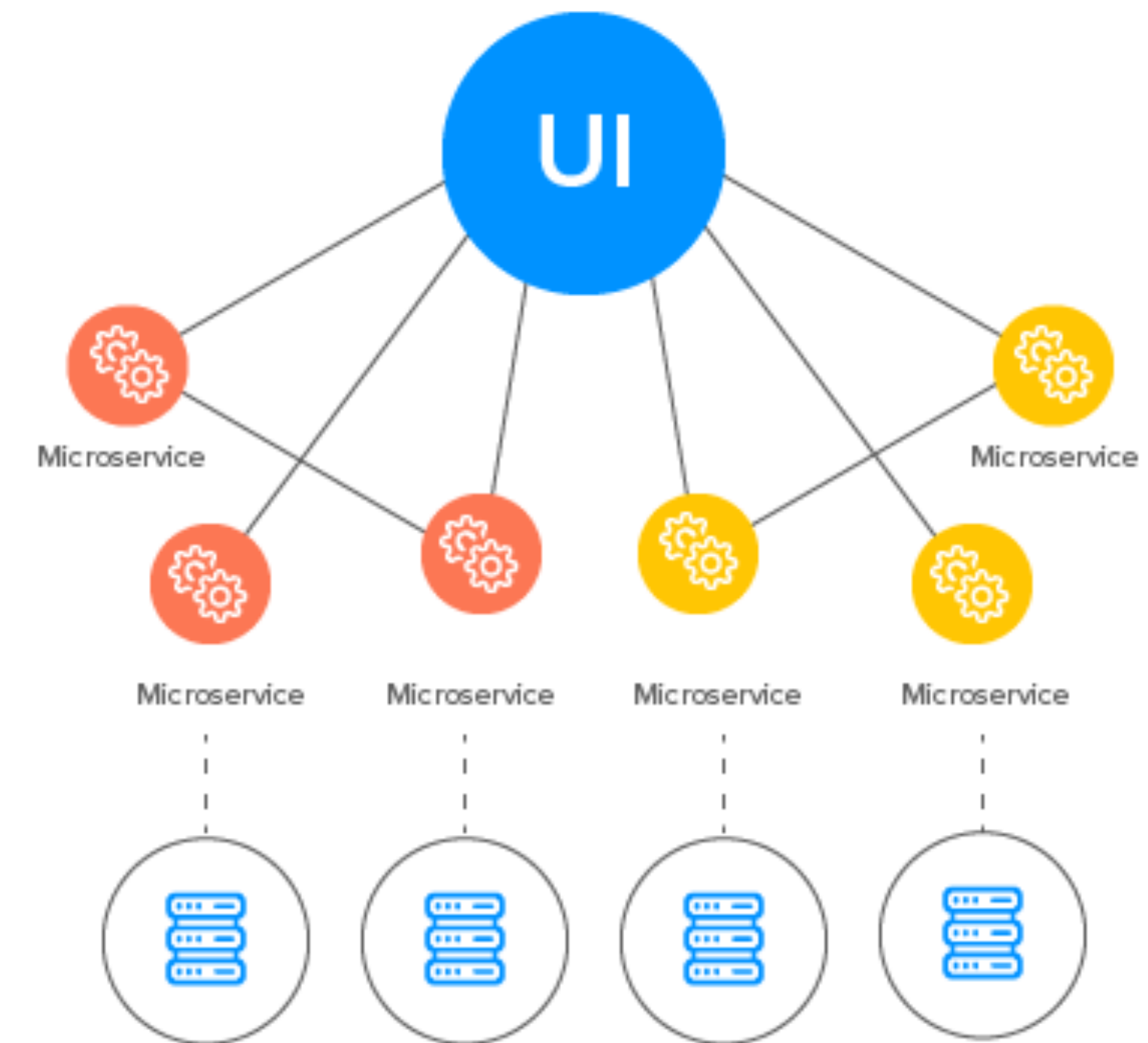


Microservices vs. Monolith

При монолитной архитектуре все контексты приложения запускаются на сервере с помощью внутрипроцессных взаимодействий.



Микросервисная архитектура — это подход, позволяющий инкапсулировать определённые контексты приложения.

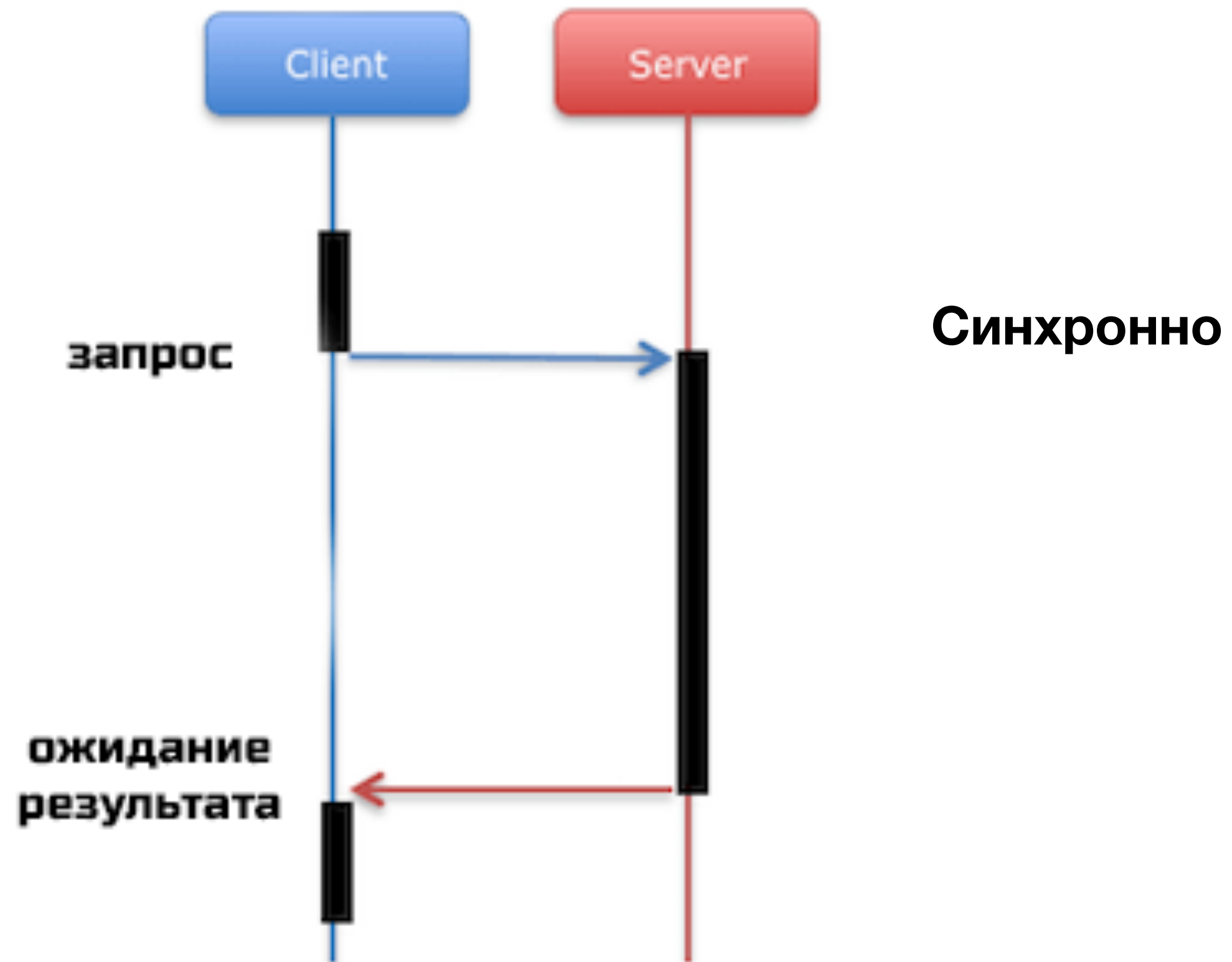


Подходы к организации взаимодействия

Подходы к организации взаимодействия



Подходы к организации взаимодействия



Подходы к организации взаимодействия



Синхронно

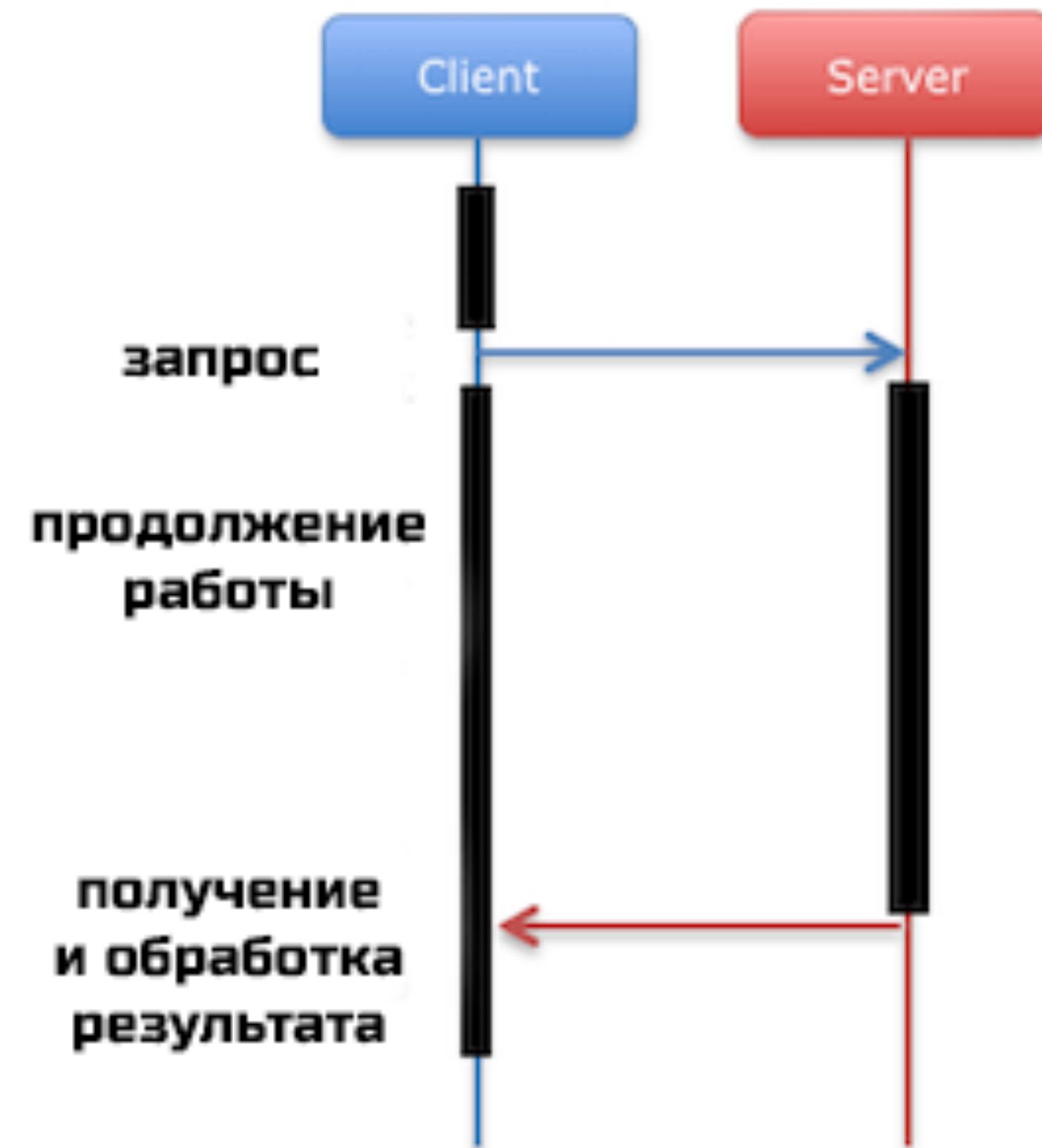
Один сервис обращается к другому и ожидает ответа. Для организации синхронного взаимодействия используется протокол *HTTP* или *HTTPS*. Разработка и отладка просты, однако сервис должен быть постоянно доступен, в противном случае обмен сообщениями остановится.

Подходы к организации взаимодействия



Синхронно

Один сервис обращается к другому и ожидает ответа. Для организации синхронного взаимодействия используется протокол *HTTP* или *HTTPS*. Разработка и отладка просты, однако сервис должен быть постоянно доступен, в противном случае обмен сообщениями остановится.

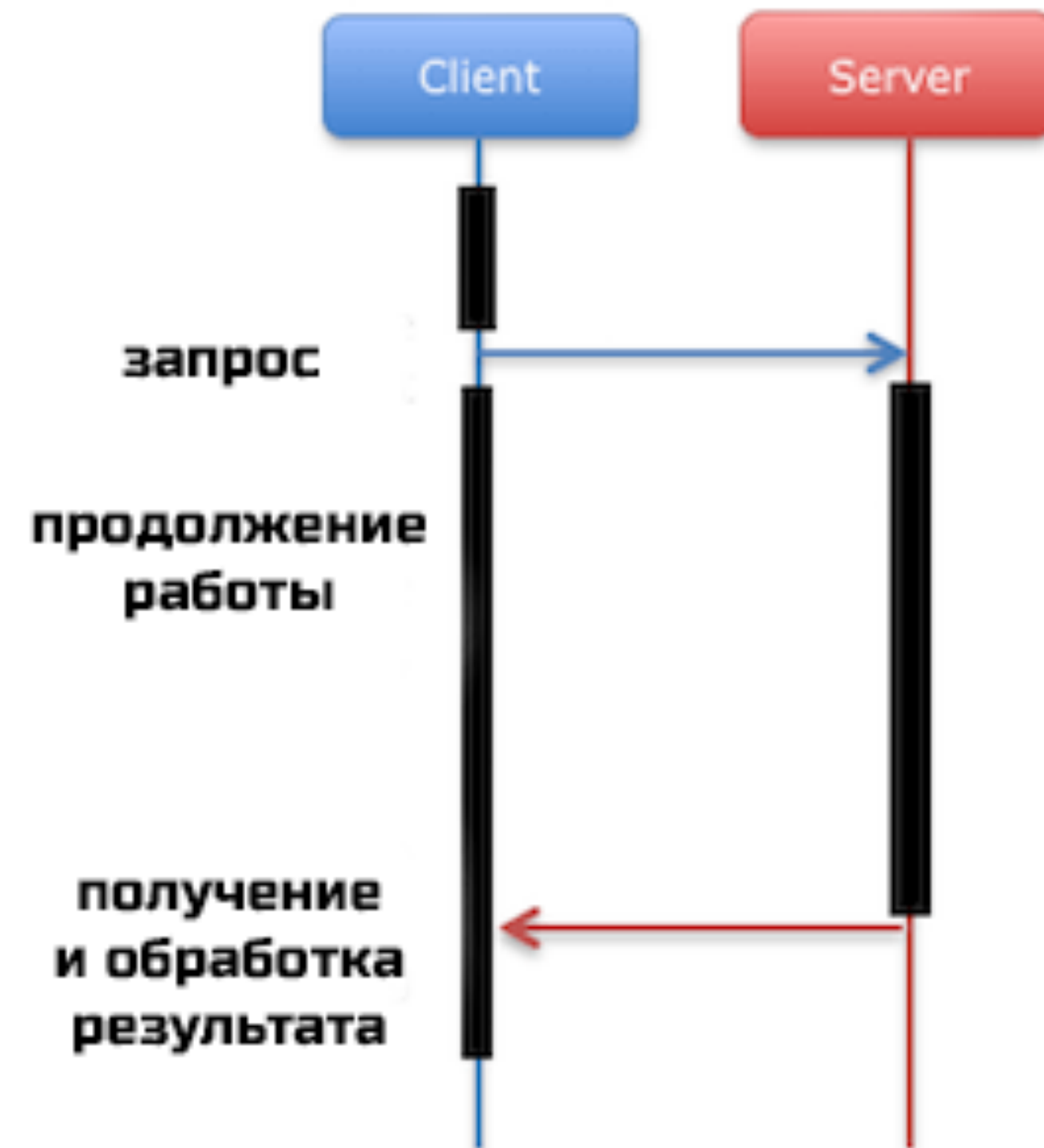


Подходы к организации взаимодействия



Синхронно

Один сервис обращается к другому и ожидает ответа. Для организации синхронного взаимодействия используется протокол *HTTP* или *HTTPS*. Разработка и отладка просты, однако сервис должен быть постоянно доступен, в противном случае обмен сообщениями остановится.



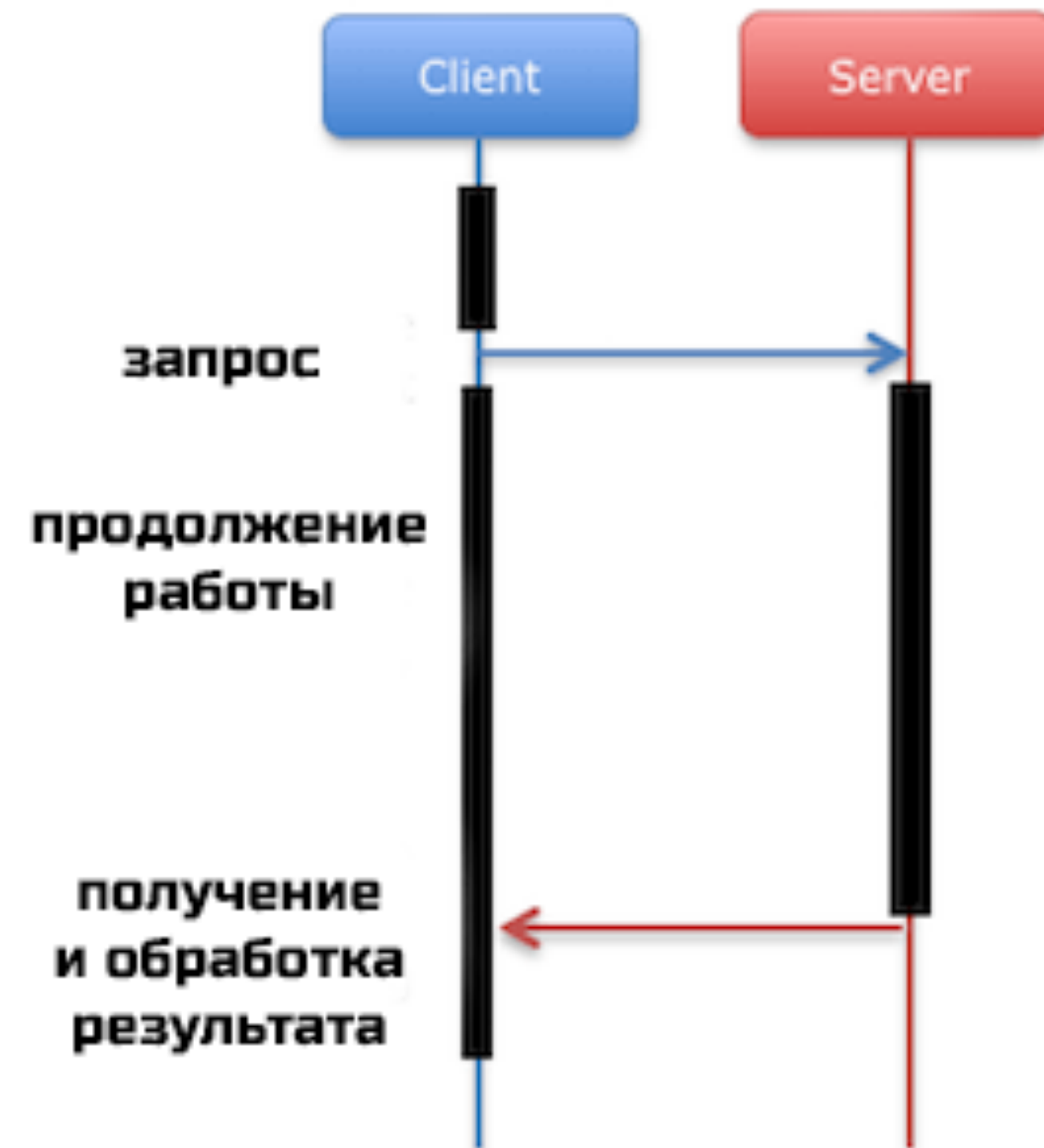
Асинхронно

Подходы к организации взаимодействия



Синхронно

Один сервис обращается к другому и ожидает ответа. Для организации синхронного взаимодействия используется протокол *HTTP* или *HTTPS*. Разработка и отладка просты, однако сервис должен быть постоянно доступен, в противном случае обмен сообщениями остановится.



Асинхронно

Сервисы взаимодействуют между собой путем передачи сообщений. Таким образом, сервис не ожидает ответ, а продолжает работу. Нужный сервис принимает сообщение и начинает его обработку.

Для организации асинхронного взаимодействия используются очереди сообщений.

Брокеры сообщений

Брокеры сообщений



Брокеры сообщений



Структура данных представлена в виде очереди:

- первым "читается" сообщение пришедшее первым
- новые элементы добавляются только в конец очереди
- чтение и удаление только с начала очереди

Как правило, в таких системах используется паттерн проектирования издатель-подписчик, при этом доступ к очереди должен быть асинхронным, чтобы обеспечить хранение сообщений в очереди до момента прочтения их получателем без одновременного взаимодействия с очередью.

Брокеры сообщений



Структура данных представлена в виде очереди:

- первым "читается" сообщение пришедшее первым
- новые элементы добавляются только в конец очереди
- чтение и удаление только с начала очереди

Как правило, в таких системах используется паттерн проектирования издатель-подписчик, при этом доступ к очереди должен быть асинхронным, чтобы обеспечить хранение сообщений в очереди до момента прочтения их получателем без одновременного взаимодействия с очередью.



Брокеры сообщений

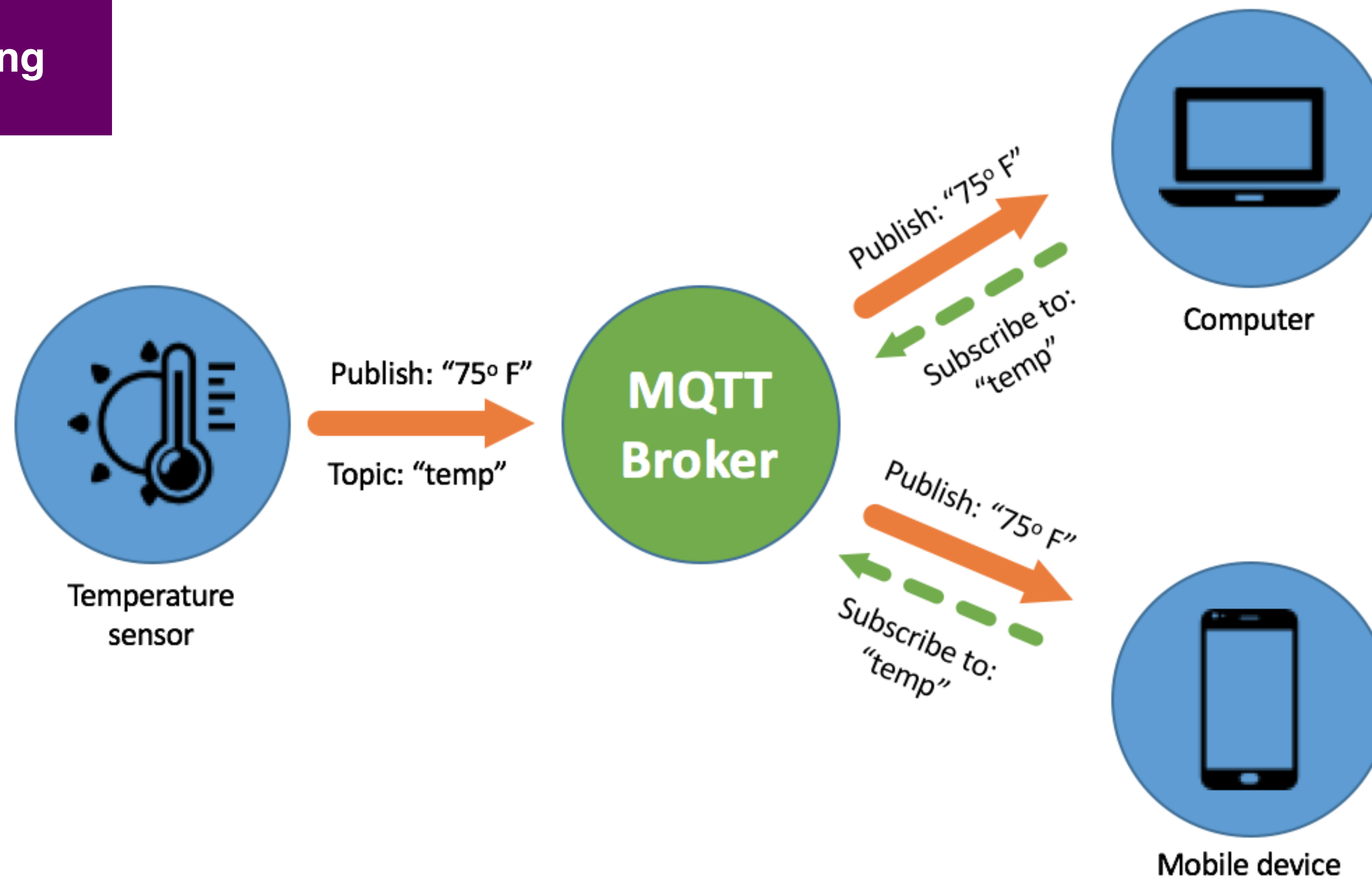


MQTT: The Standard for IoT Messaging

Брокеры сообщений



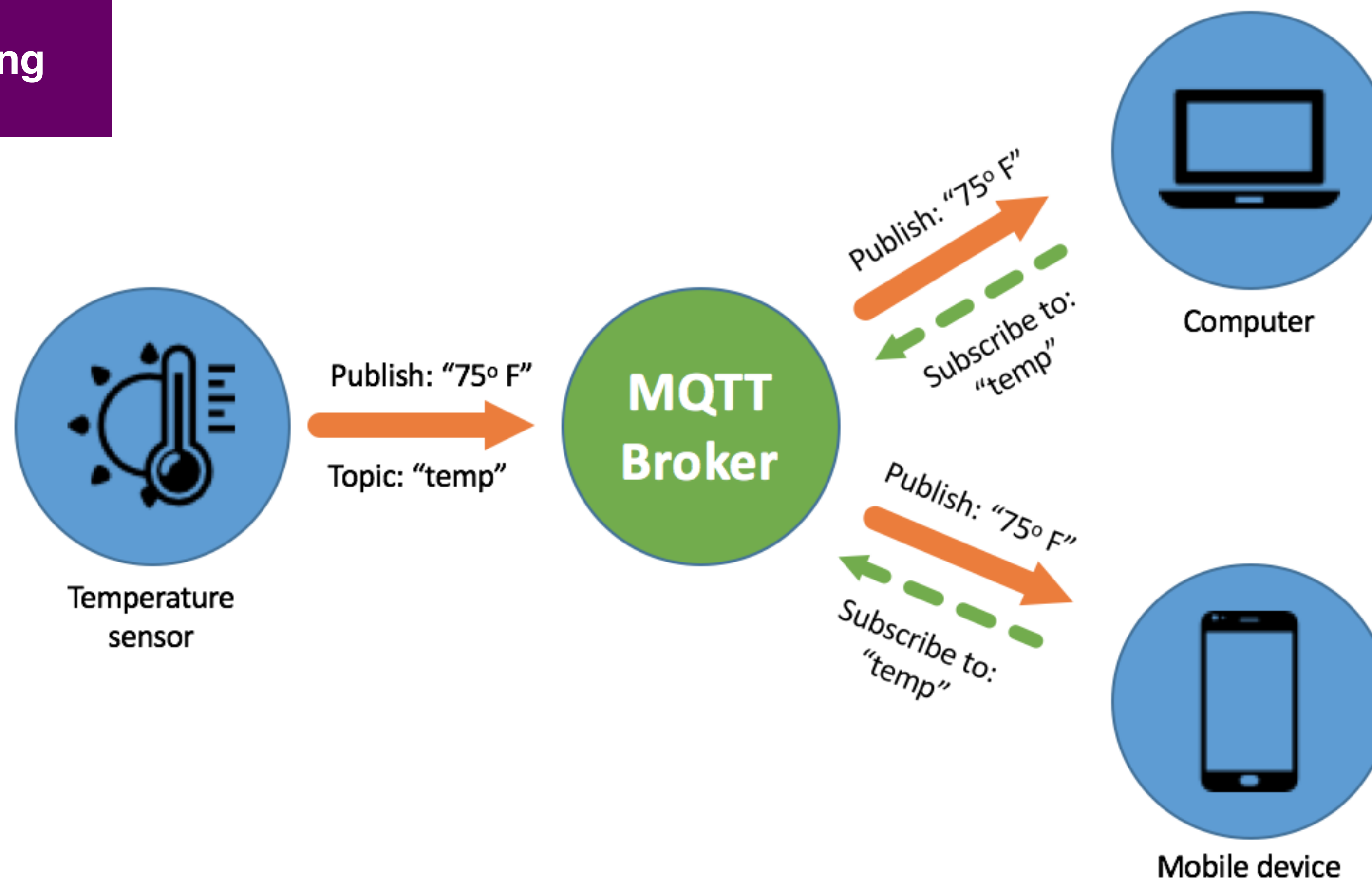
MQTT: The Standard for IoT Messaging



Брокеры сообщений



MQTT: The Standard for IoT Messaging



Подробнее:

<https://mqtt.org>

<https://habr.com/ru/post/463669/>