РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

Павел Кюркчиев
Ас. към ПУ „Паисий Хилендарски“
@pkyurkchiev

РАЗПРЕДЕЛЕНИ АРХИТЕКТУРИ MICROSERVICES

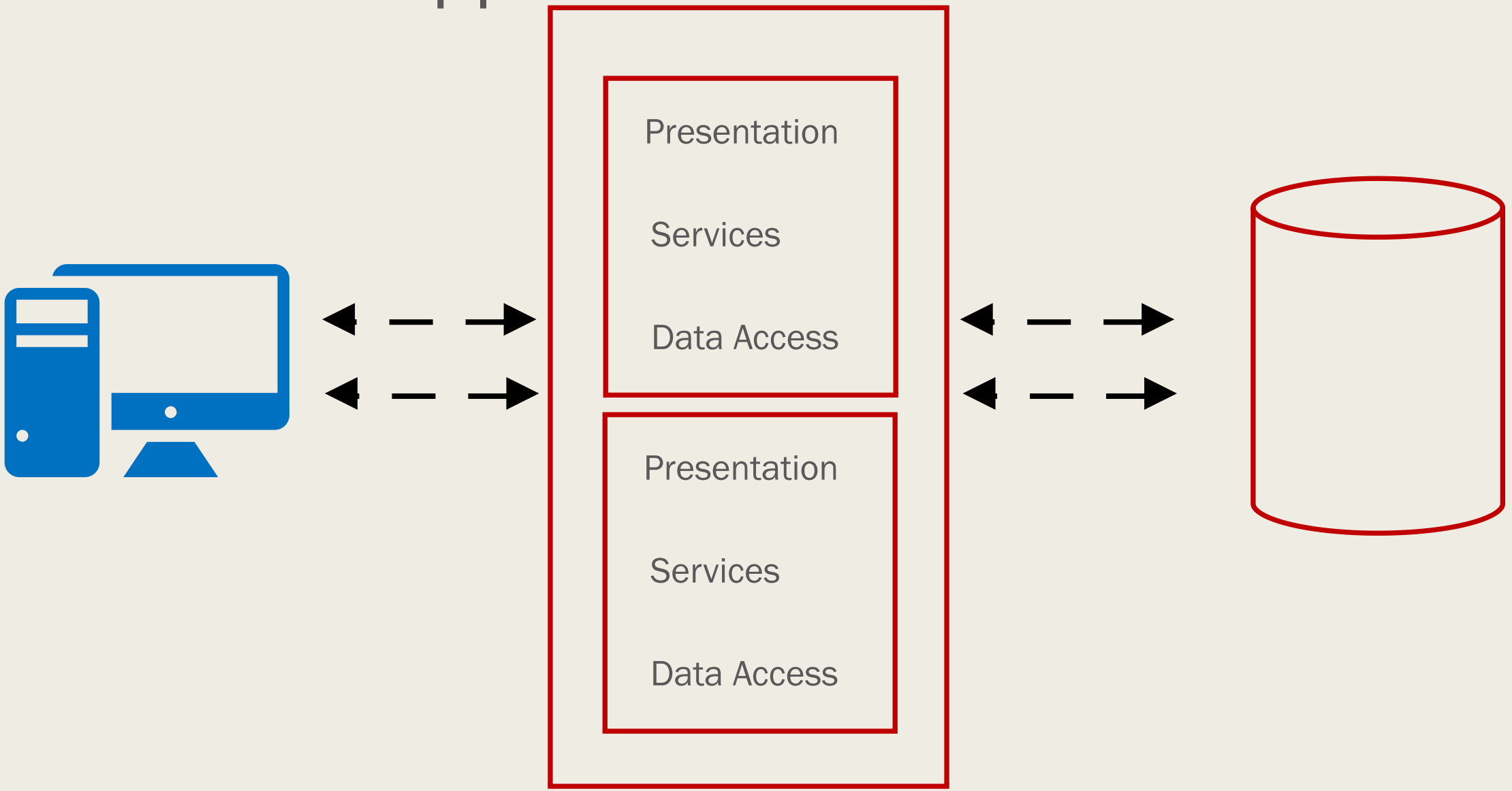
Традиционна Монолитна Архитектура (Monolithic architecture)

- Монолитният софтуер е проектиран като самостоятелно цяло. Програмните компоненти са тясно свързани и взаимозависими, за разлика от слабо свързаните компоненти в модулните софтуерни програми. При строго свързана архитектура, всички компоненти и техните зависимости трябва да са налични, за да може кодът да бъде изпълнен или компилиран.

Нужна ли е промяна?

- Монолитните приложения могат да се превърнат в "Мега приложения", където никой разработчик не познава пълната функционалност.
- Има ограничена преизползваемост.
- Разширяването на монолитно приложение е голямо предизвикателство.
- По дефиниция, монолитните приложения се разработват само с точно определен технологичен стек, което може силно да ограничи разработката.

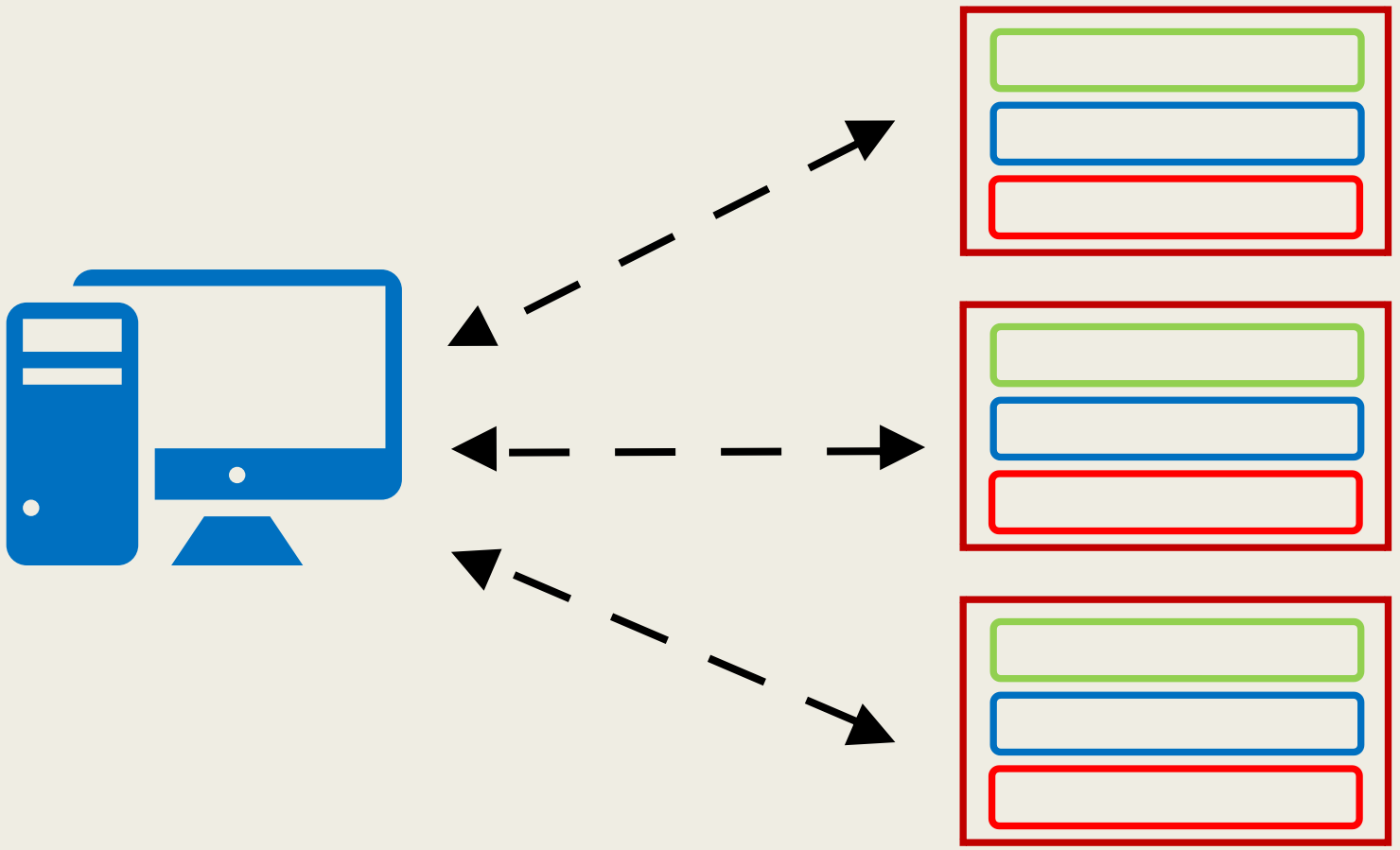
Monolithic application



Microservices

- Microservices architecture е архитектурен стил, който структурира приложение като съвкупност от слабо свързани услуги. Тези услуги заедно предоставят бизнес логиката на системата.

Microservices application

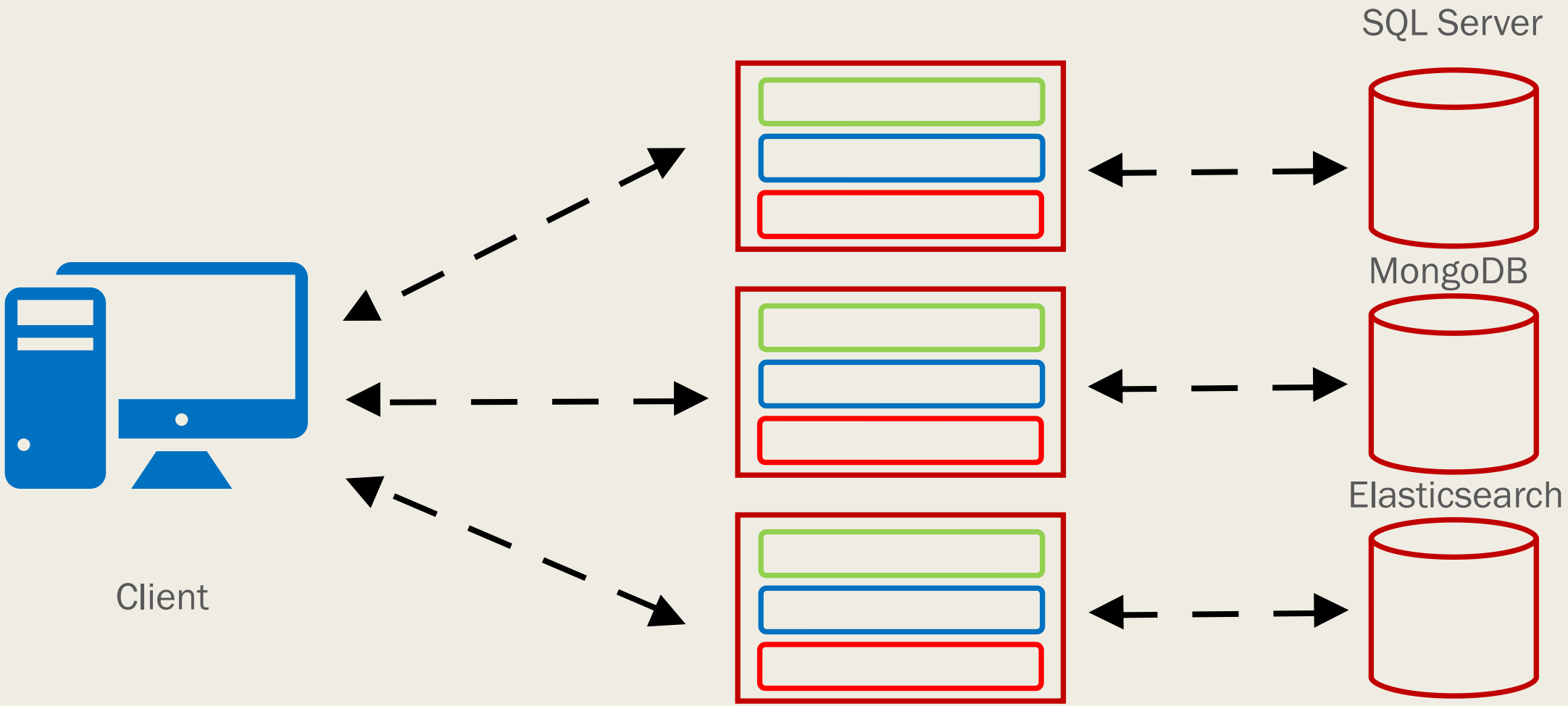


Психология на Microservices

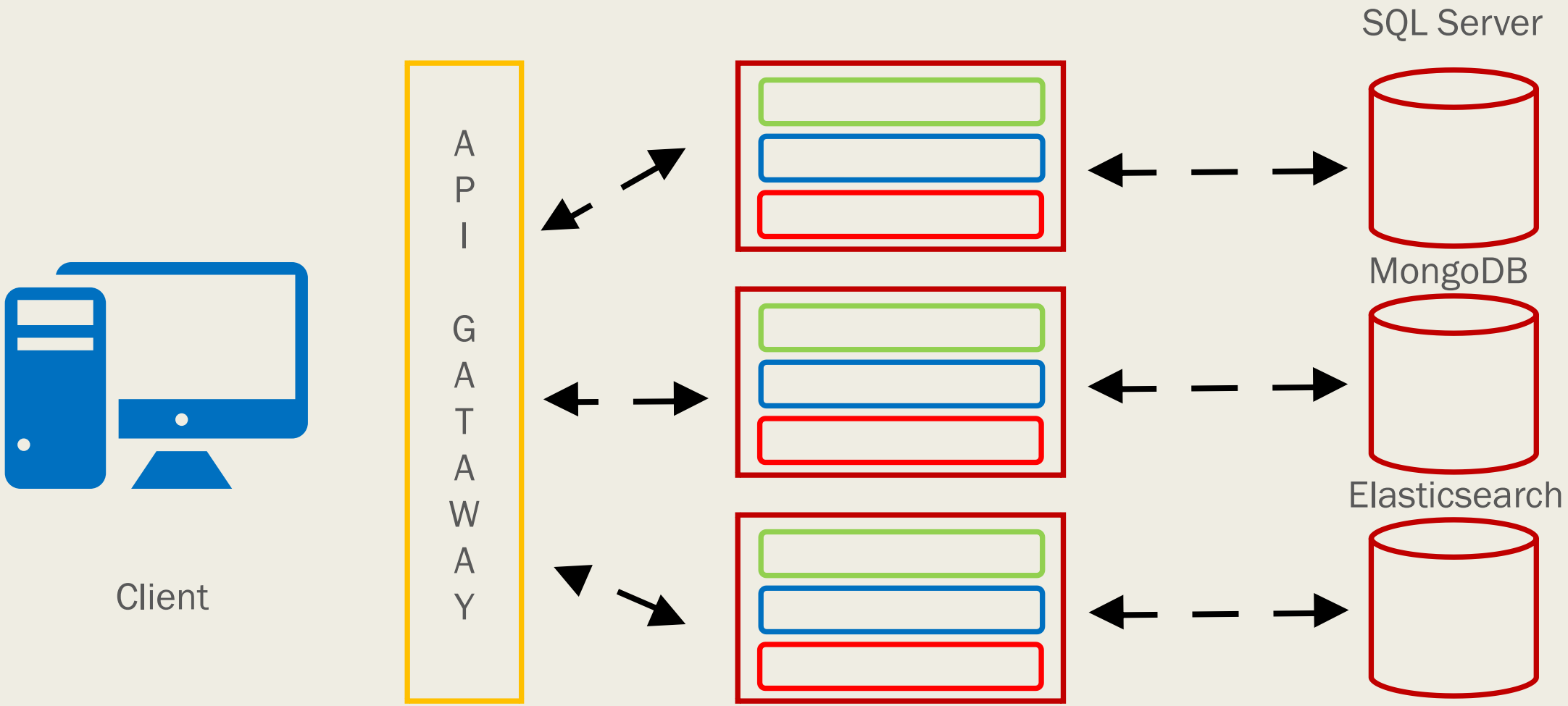
- Услугите трябва да бъдат малки и добре структурирани, за да изпълняват само една функция.
- Архитектурата следва да обхваща автоматизираното тестване и внедряване.
- Всяка услуга е еластична, композитна, минимална и завършена.

Direct client communication vs API Gateway

Microservices implementation 1

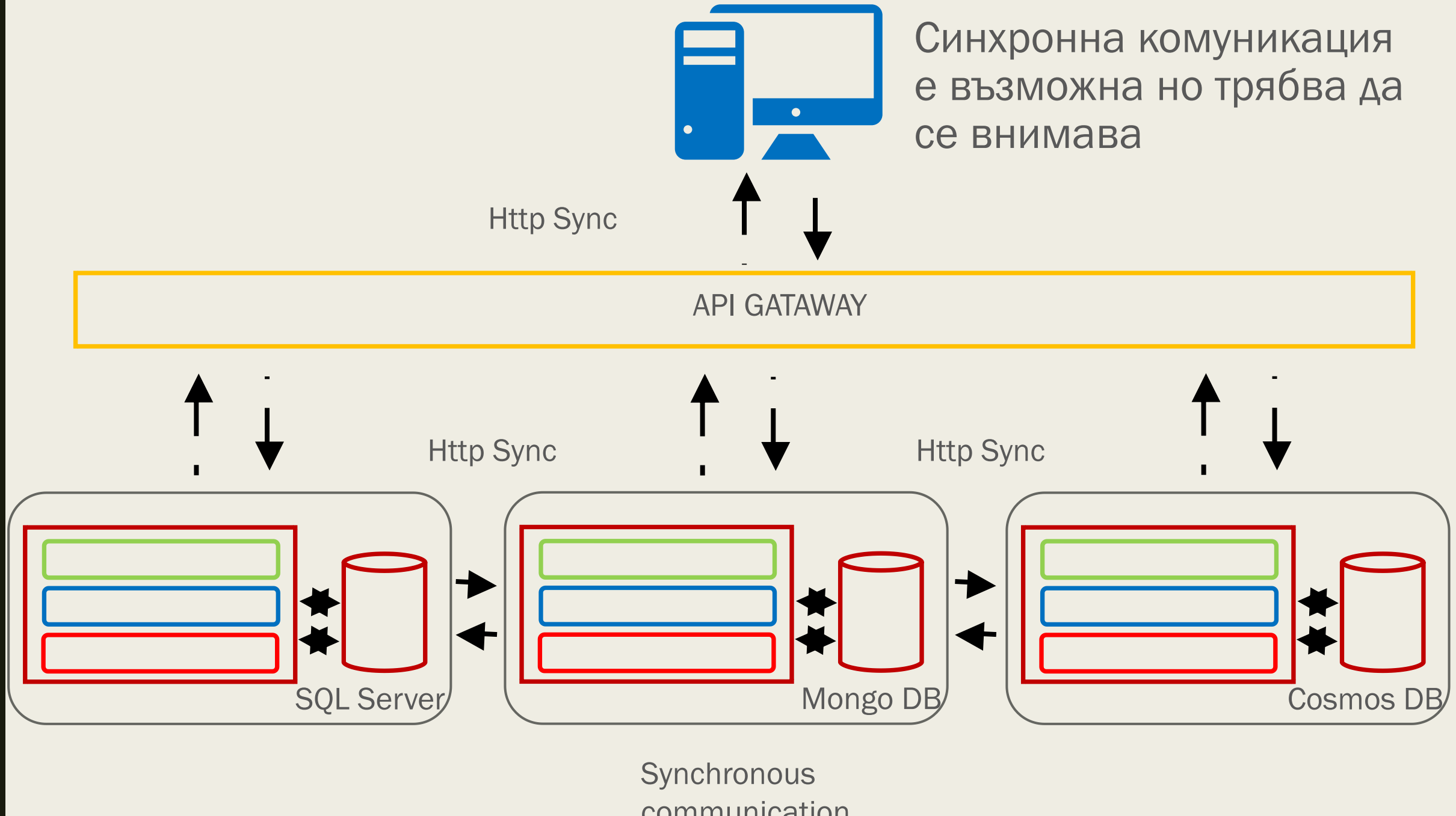


Microservices implementation 2



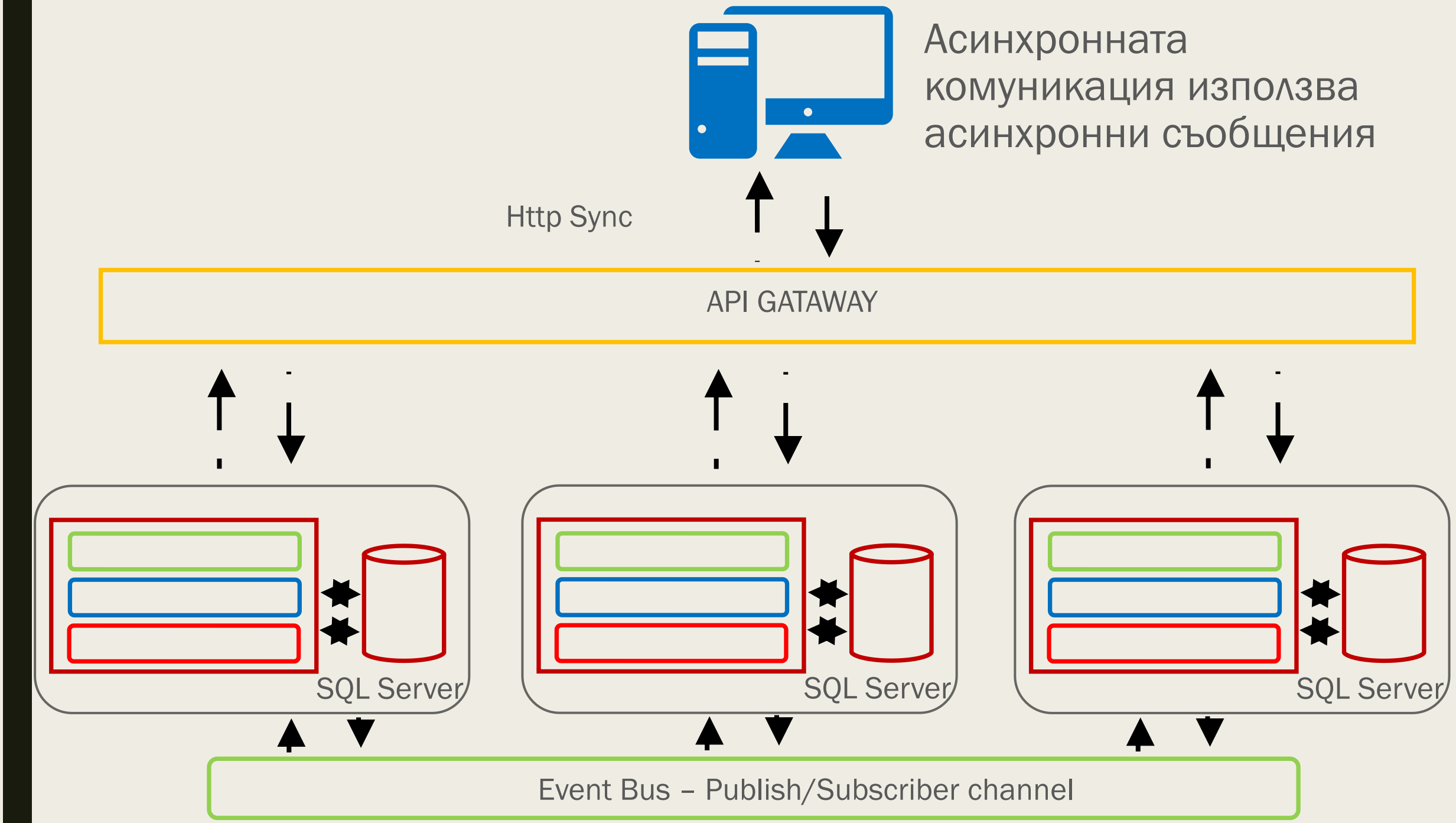
Synchronous vs Asynchronous Microservices communication

Синхронна комуникация е възможна но трябва да се внимава



Synchronous communication

Асинхронната комуникация използва асинхронни съобщения



Основни Предимства

- Възможност за разширение – както хоризонтално, така и вертикално.
- Модулна структура.
- Осигурява процес на непрекъснато обновяване.
  - *DevOps (CI/CD)*

Недостатъци

- Тестването и разпространението са по-трудни.
- Асинхронната вътрешна комуникация между отделните услуги е значително по-сложна, отколкото комуникацията между услуги, изградени на базата на monolithic architecture.
- Преместването на отговорности (логика) между услуги е по-трудно.
  - *Това може да включва комуникация между различни екипи, пренаписване на функционалност на друг език или преместване на логиката в друга инфраструктура.*

- Разглеждането на размера на услугите като основен структуриращ механизъм може да доведе до твърде много услуги, а алтернативата на вътрешната модулация може да доведе до по - опростен дизайн.

DEMO MICROSERVICES

https://github.com/pkyurkchiev/microservices_skeleton_net-core

ВЪПРОСИ ?