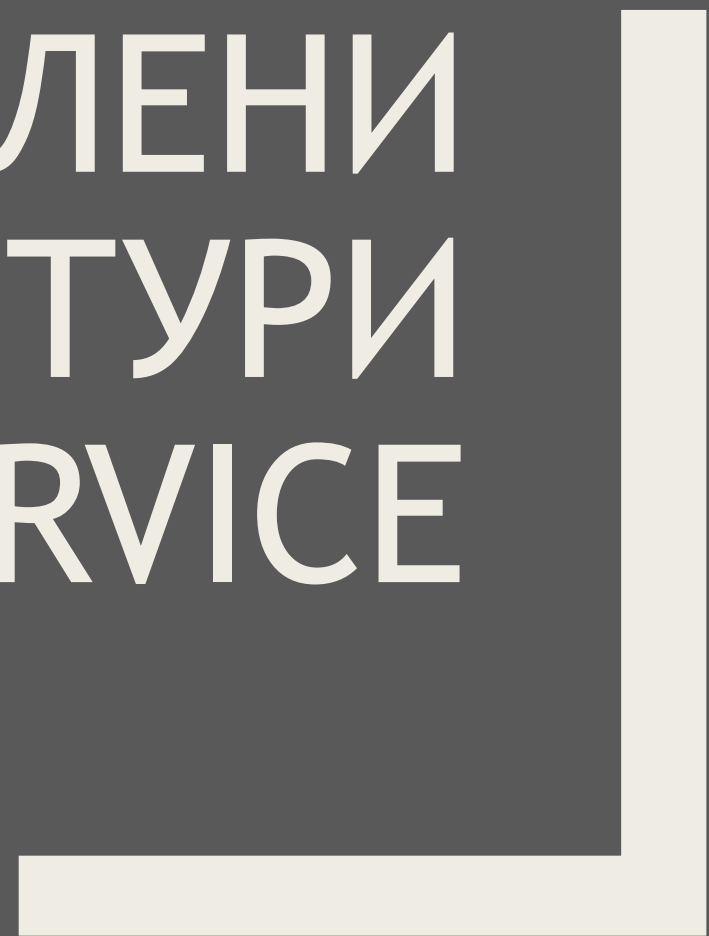


РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

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

РАЗПРЕДЕЛЕНИ АРХИТЕКТУРИ MICROSERVICE



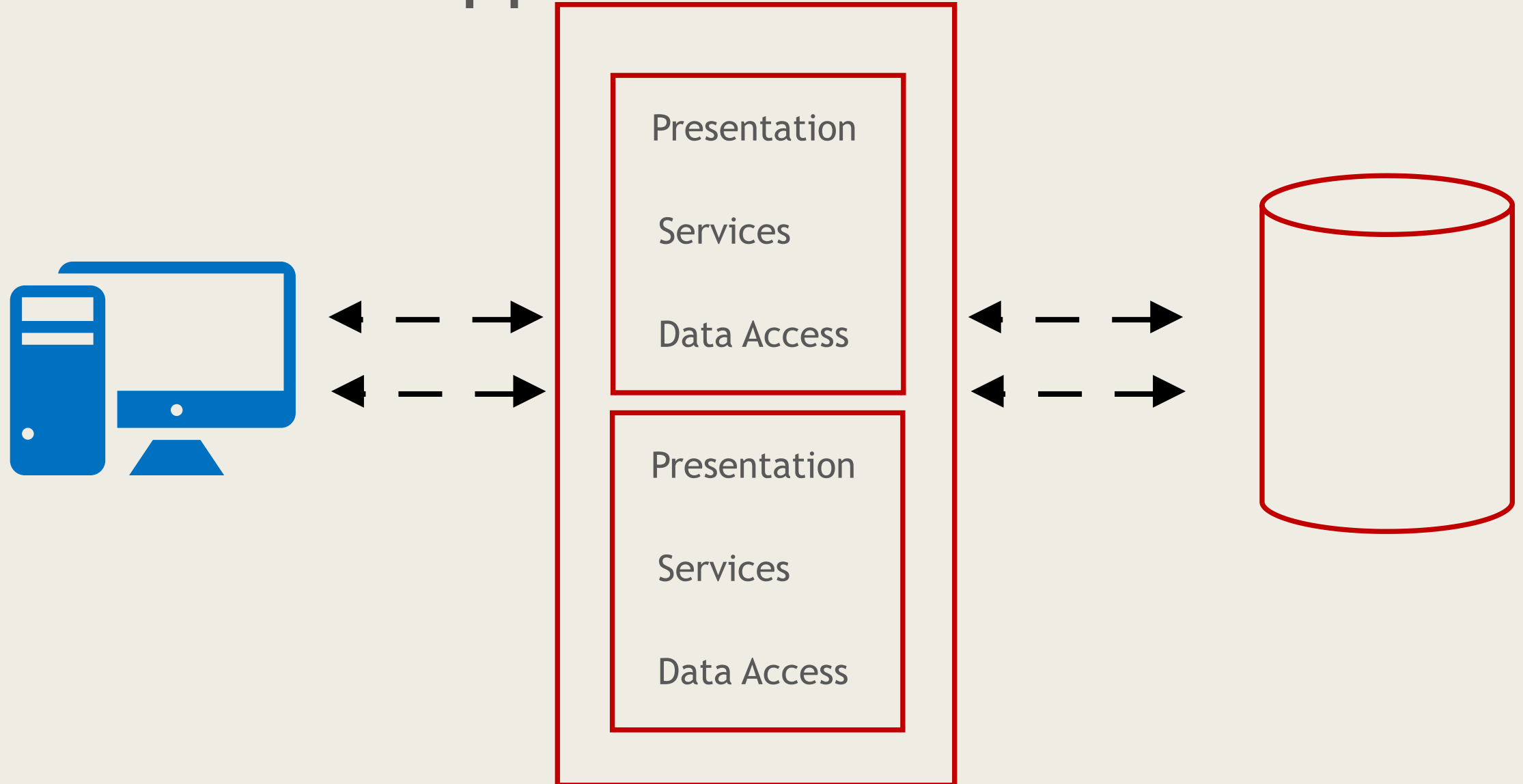
Традиционна Monolithic architecture

- Монолитният софтуер е проектиран да бъде самостоятелен; компонентите на програмата са взаимосвързани и взаимозависими, а не слабо свързани, както е случаят с модулните софтуерни програми. В строго свързана архитектура, всеки компонент и свързаните с него компоненти трябва да присъстват, за да може кодът да бъде изпълнен или компилиран

Нужна ли е промяна?

- Монолитните приложения могат да се превърнат в "Мега приложение". Ситуация, в която никой от разработчик не познава пълната функционалност на приложението
- Ограничена презиползваемост
- Разширението на монолитно приложение е голямо предизвикателство
- По дефениция монолитните приложения са разработвани само от точно дефиниран технологичен стек. Това от своя страна може силно да лимитира разработката

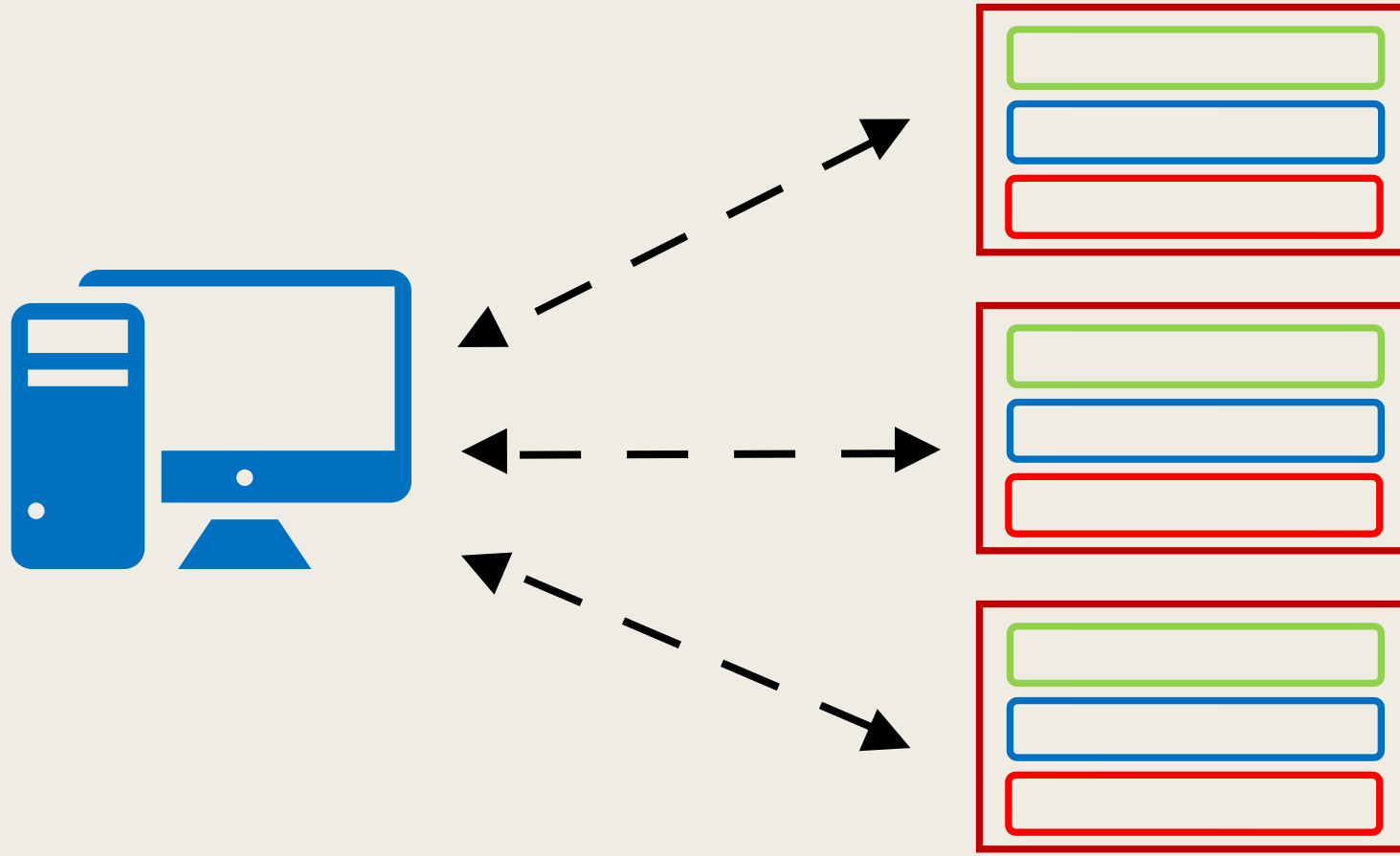
Monolithic application



Microservice

- Microservice architecture е архитектурен стил, който структурира приложение като съвкупност от свободно свързани услуги, които представят бизнес логиката на системата

Microservice application

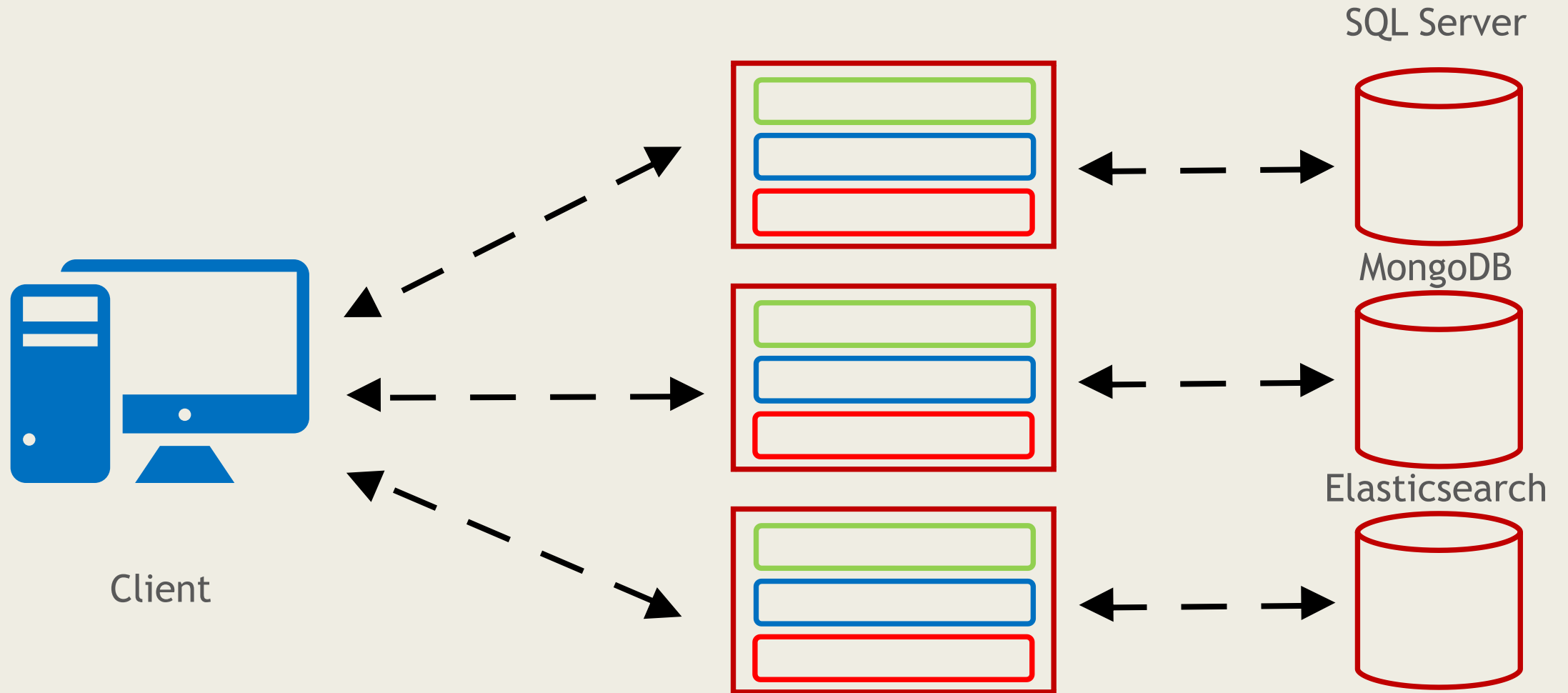


Психология на microservice

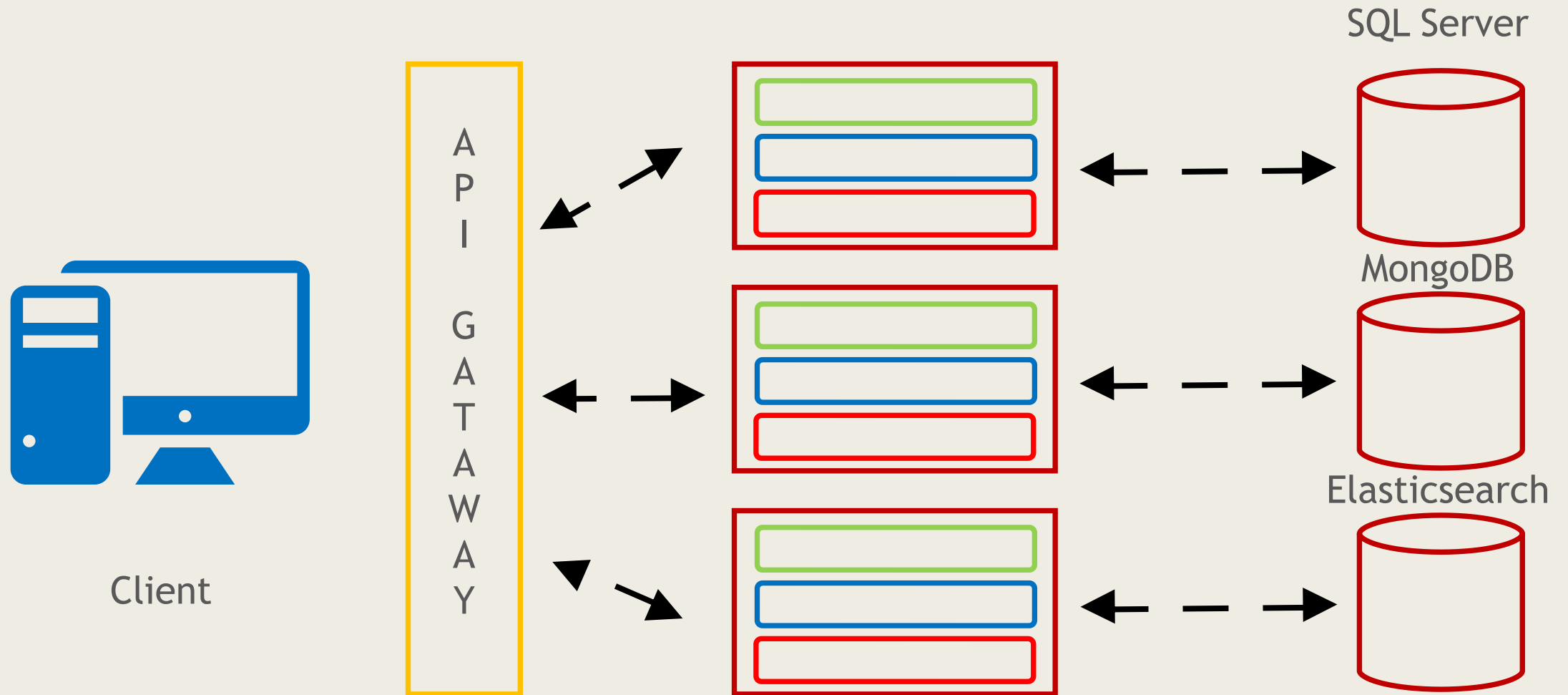
- Услугите трябва да са малки, добре структурирани за да могат да изпълняват само една функция
- Архитектурата трябва да обхваща автоматизираното тестване и внедряване
- Всяка услуга е еластична, композиционна, минимална и пълна.

Direct client communication versus API Gateway

Microservice implementation 1



Microservice implementation 2

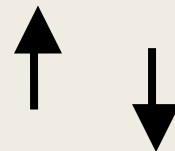


Synchronous versus Asynchronous Microservice communication

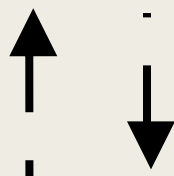


Директна microservice комуникация е възможна но трябва да се внимава

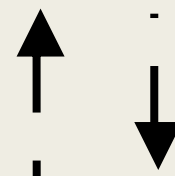
Http Sync



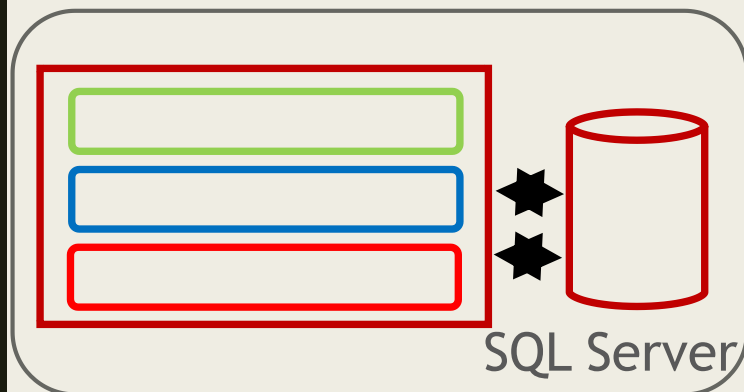
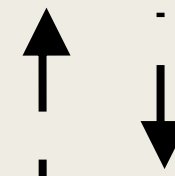
API GATAWAY



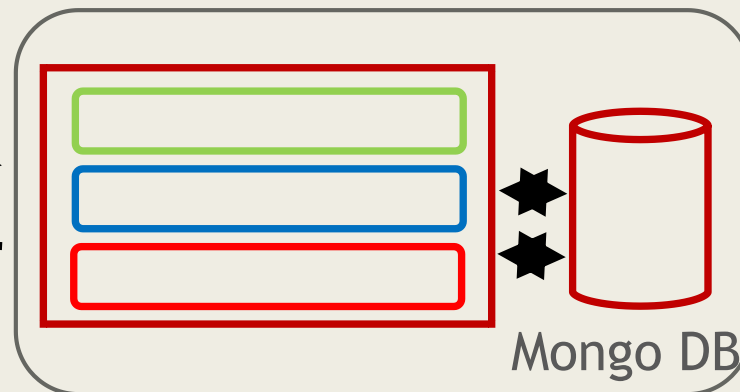
Http Sync



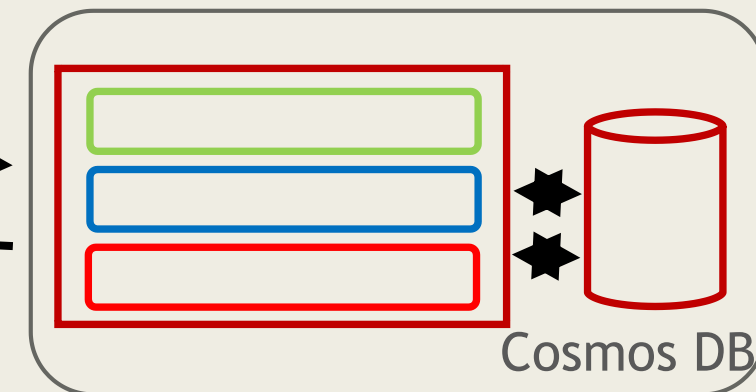
Http Sync



SQL Server



Mongo DB



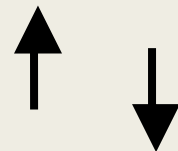
Cosmos DB

Synchronous communication

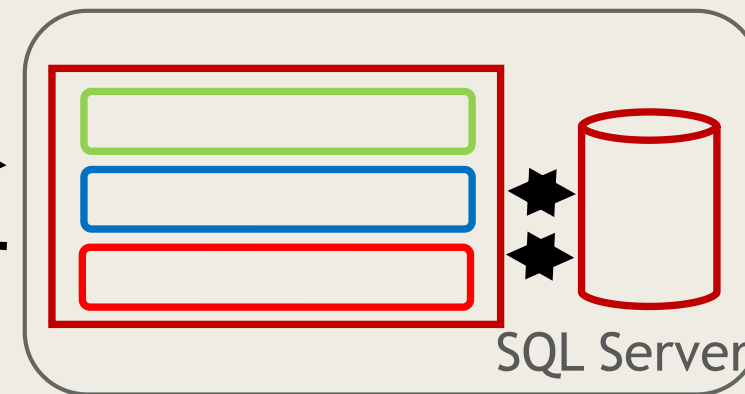
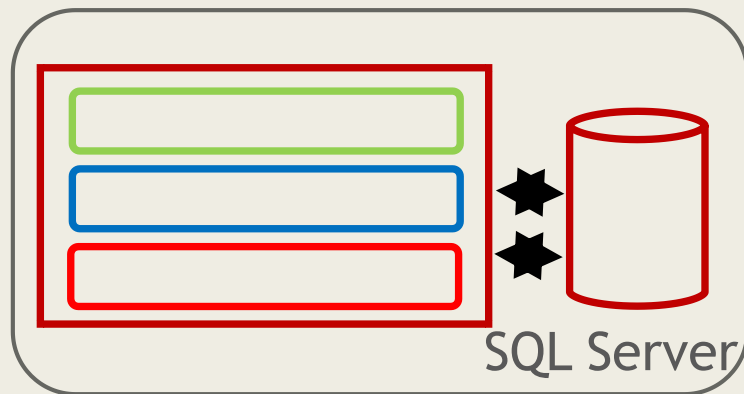
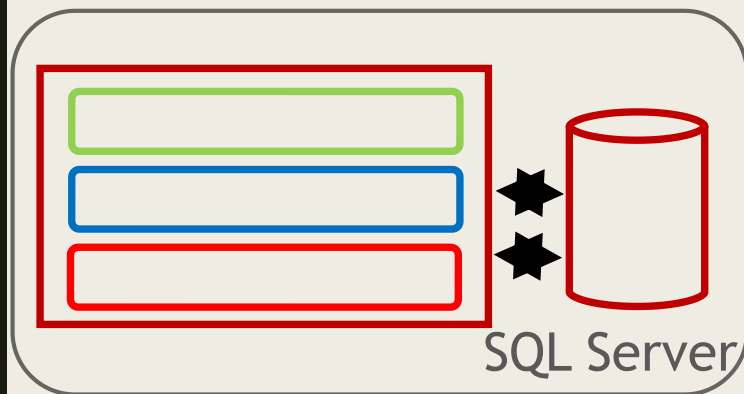
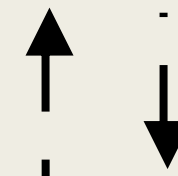
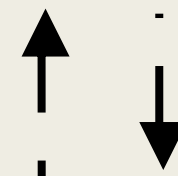
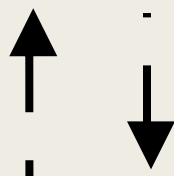


Асинхронната
комуникация използва
асинхронни съобщения

Http Sync



API GATAWAY



Event Bus - Publish/Subscriber channel

Основни предимства

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

Недостатъци

- Тестването и разпространението на кода е по - трудно
- Асинхронната вътрешната комуникация между отделните услуги е значително по - тежка, от комуникацията между услуги изградени на оновата на monolithic architecture
- Преместването на отговорностите(логика) между услуги е по-трудно
 - *Това може да включва комуникация между различни екипи, пренаписване на функционалност на друг език или поставянето на логиката в друга инфраструктура*

Недостатъци

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

DEMO MICROSERVICE

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