

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

Дипломна робота на тему
«Дослідження командно-орієнтованої мікросервісної архітектури
корпоративних застосунків»

Виконавець:

студент групи КІТ-М1196
Варьєва А.О.

Керівник роботи:

ст. викладач Молчанов Г.І.

ОБ'ЄКТ, ПРЕДМЕТ ТА МЕТА ДОСЛІДЖЕННЯ

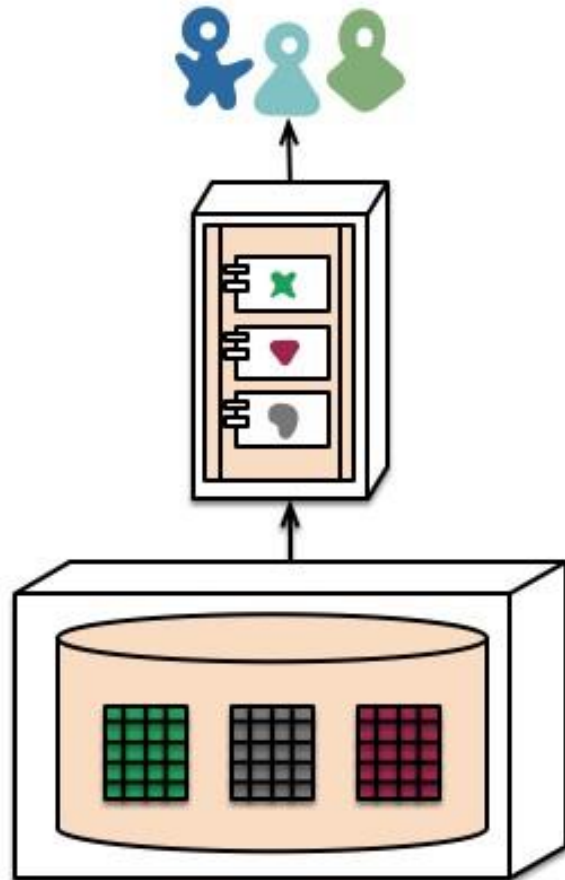
Об'єкт дослідження – мікросервісна архітектура.

Мета – Дослідити використання командно-орієнтованої мікросервісної архітектури у корпоративних застосунках .

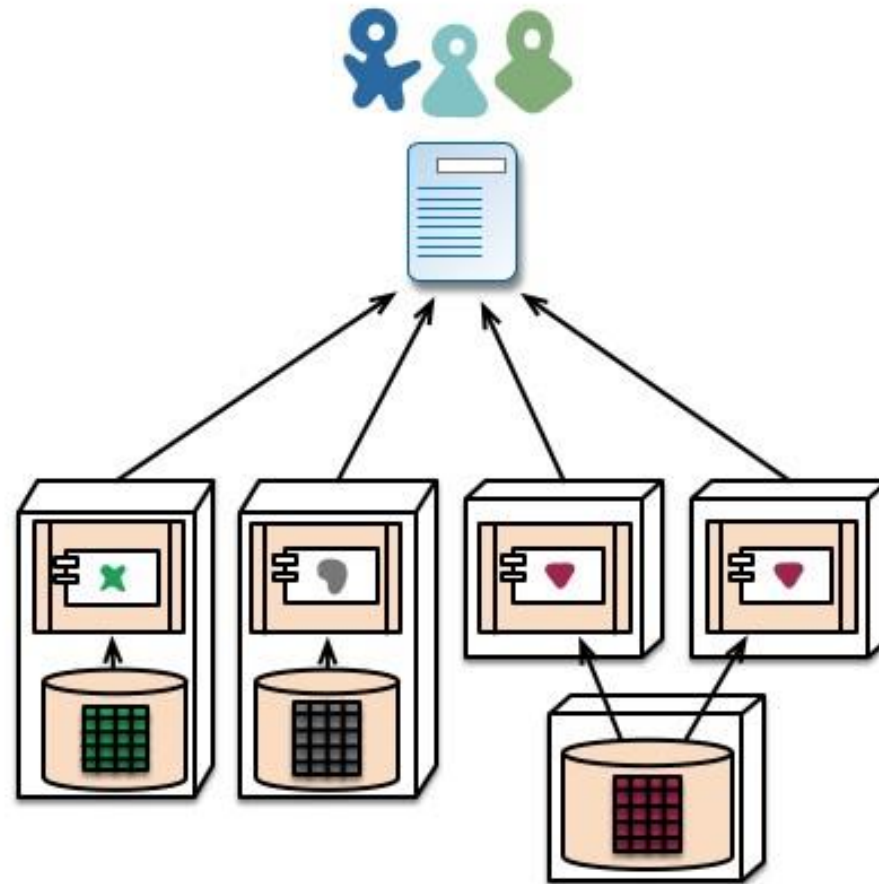
Задачі

- Порівняння підходів до розробки ПЗ
- Виділити необхідні компоненти мікросервісної архітектури
- Сформулювати нефункціональні вимоги
- Проектування архітектури ПЗ
- Виділення обмежень використання архітектури

DATA MANAGING APPROACHES

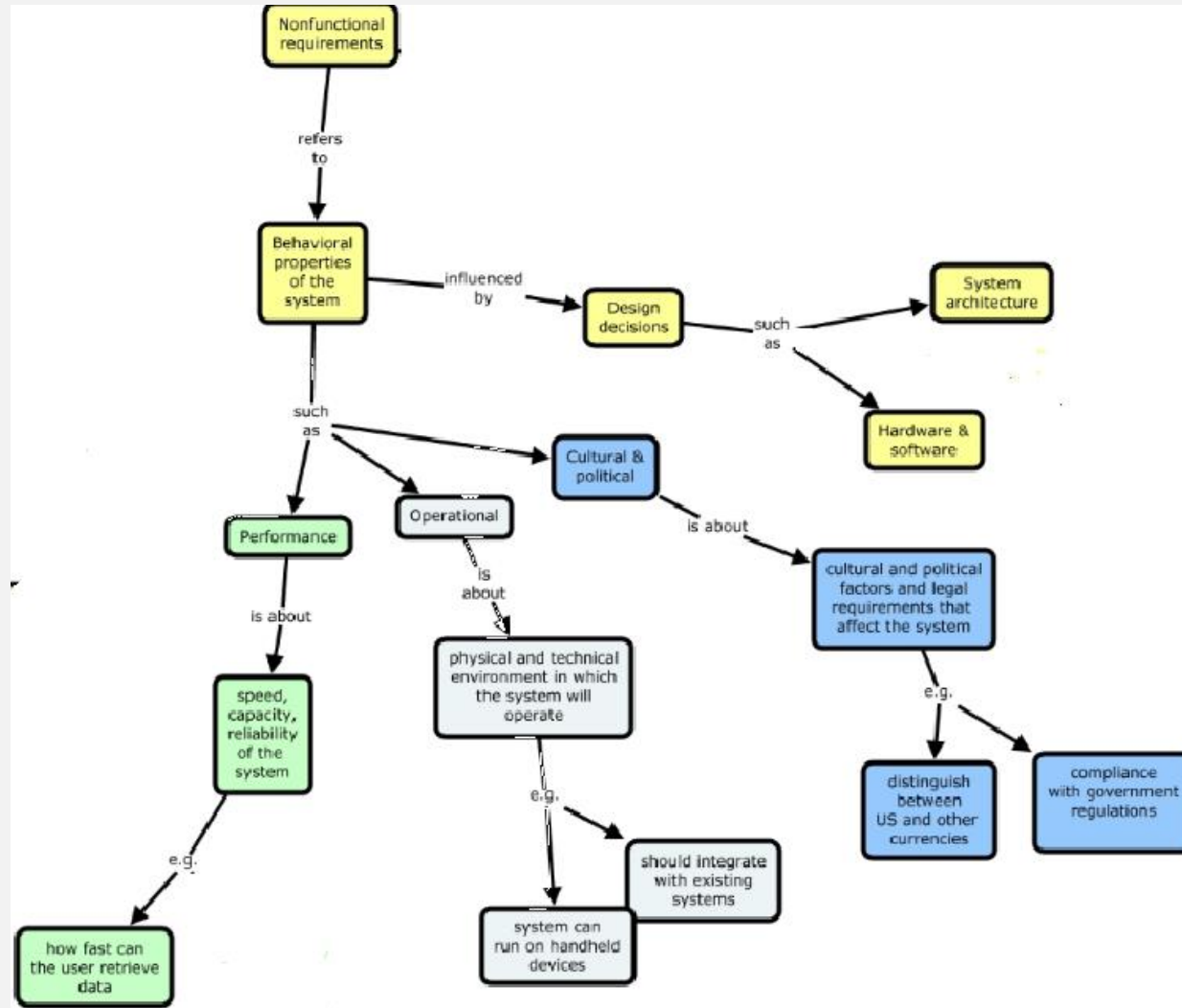


monolith - single database

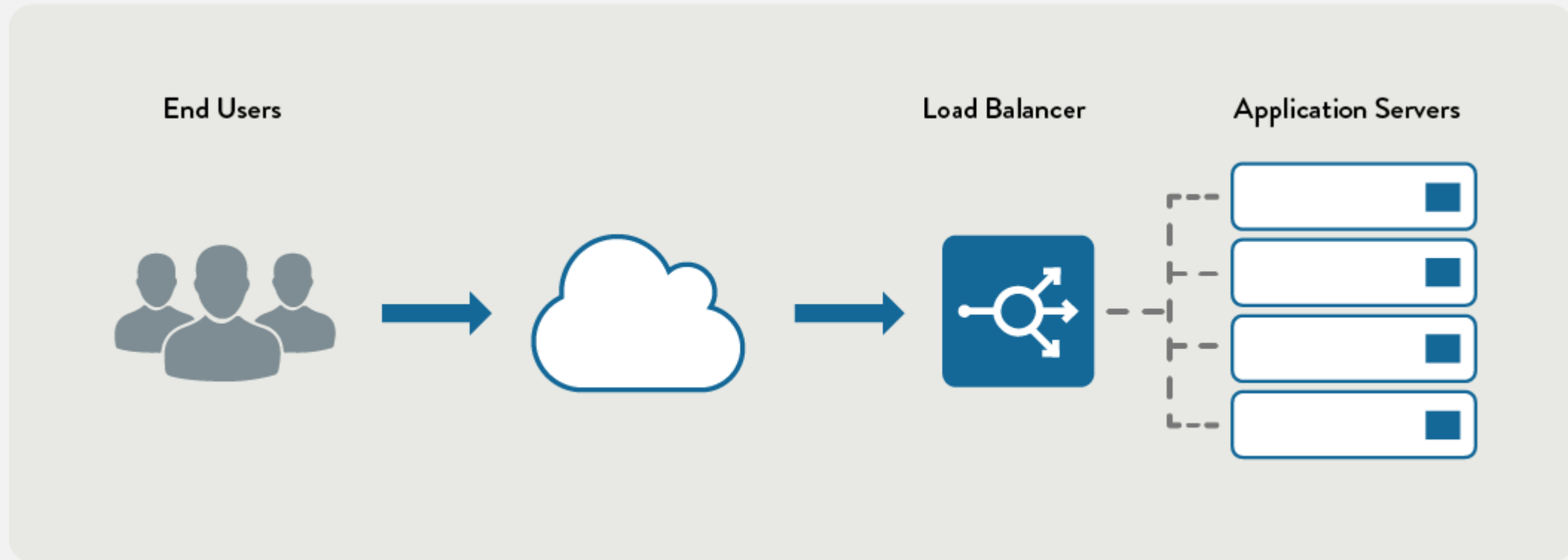


microservices - application databases

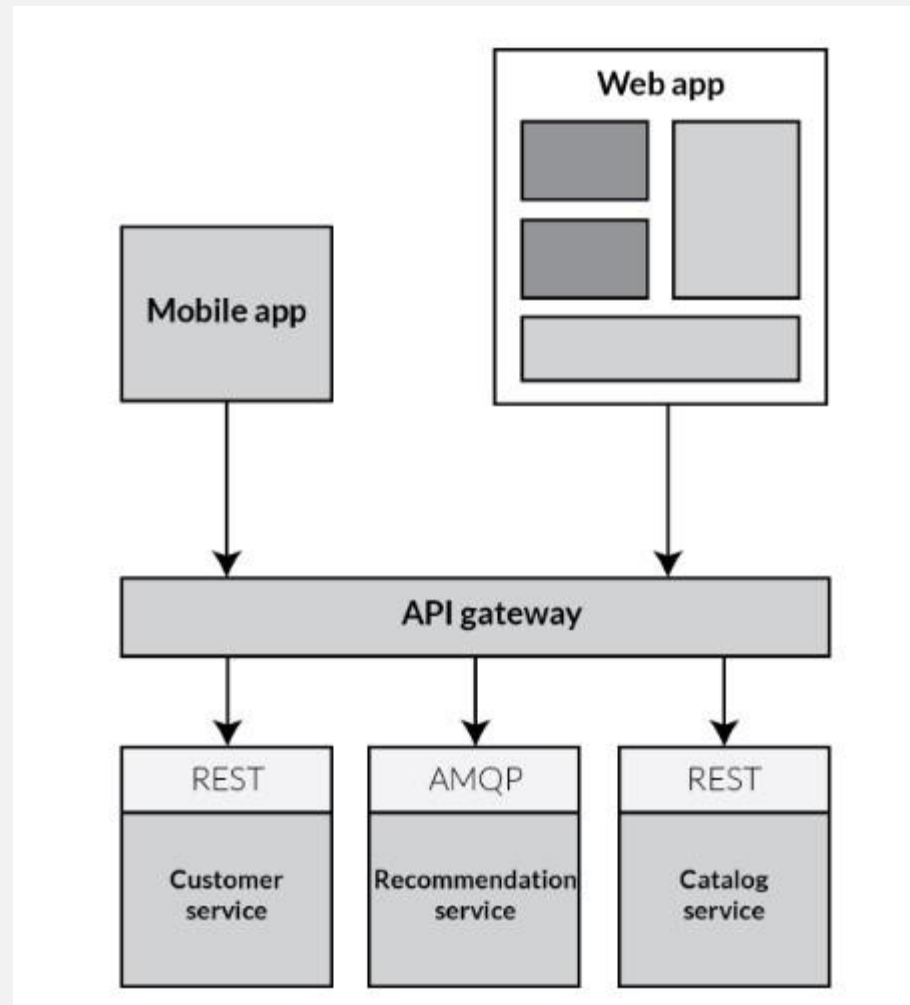
NON-FUNCTIONAL REQUIREMENTS



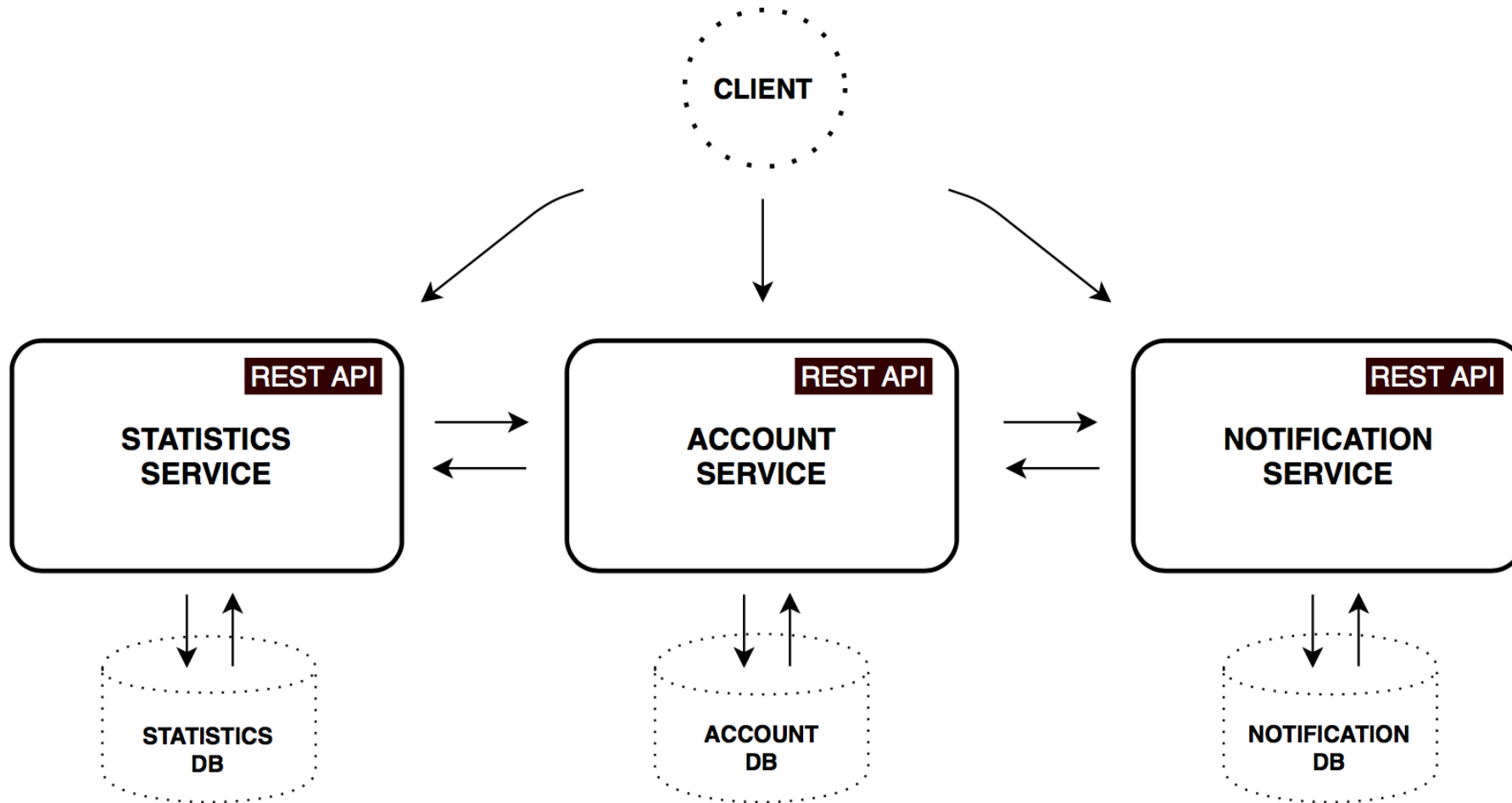
LOAD BALANCER



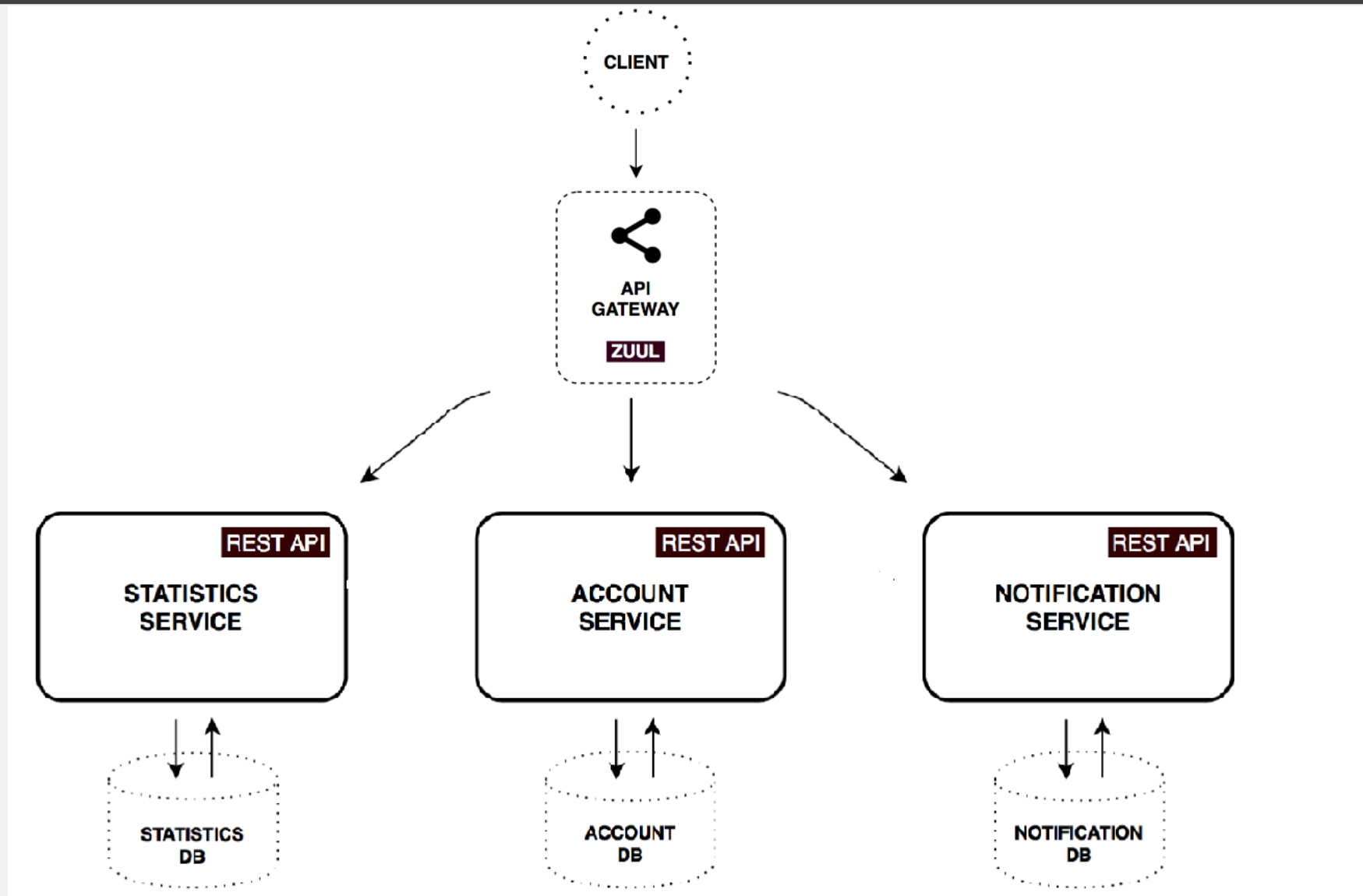
INTERACTION OF CLIENTS AND MICROSERVICES



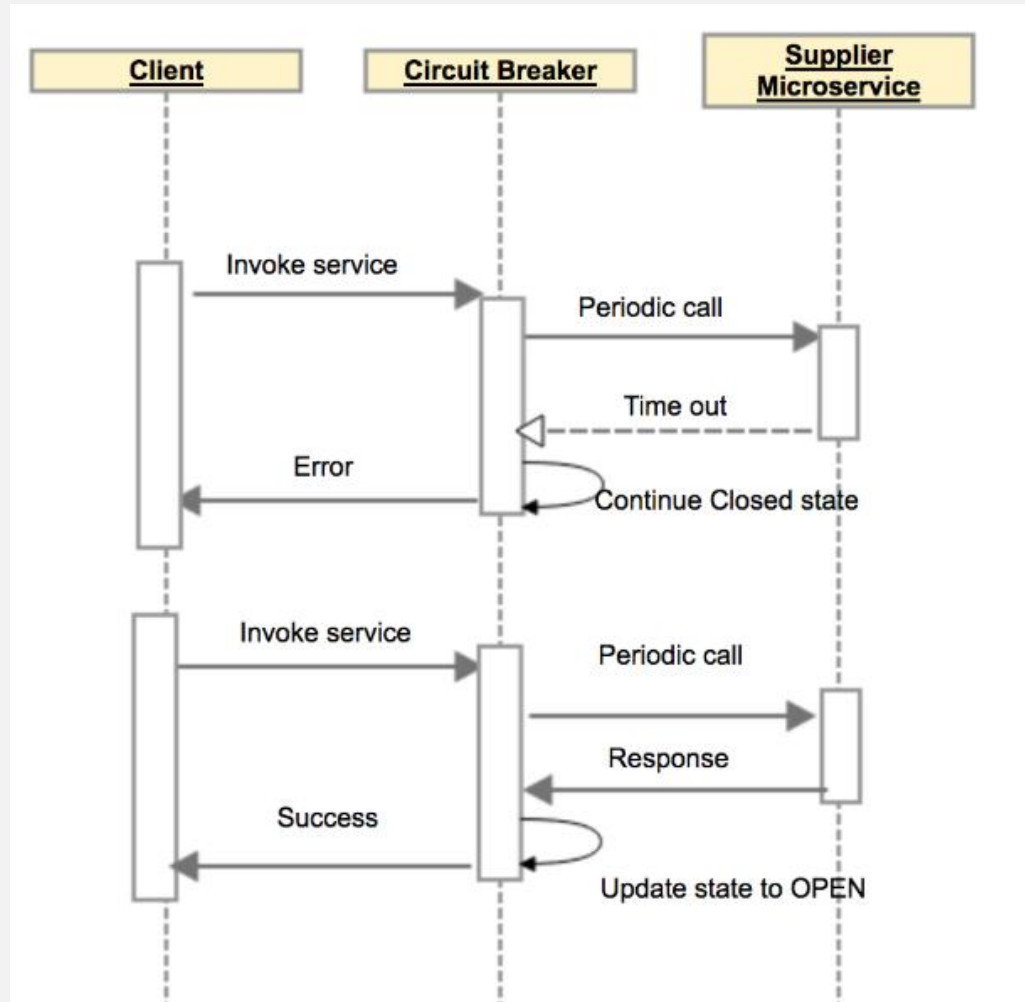
ARCHITECTURE WITHOUT API GATEWAY



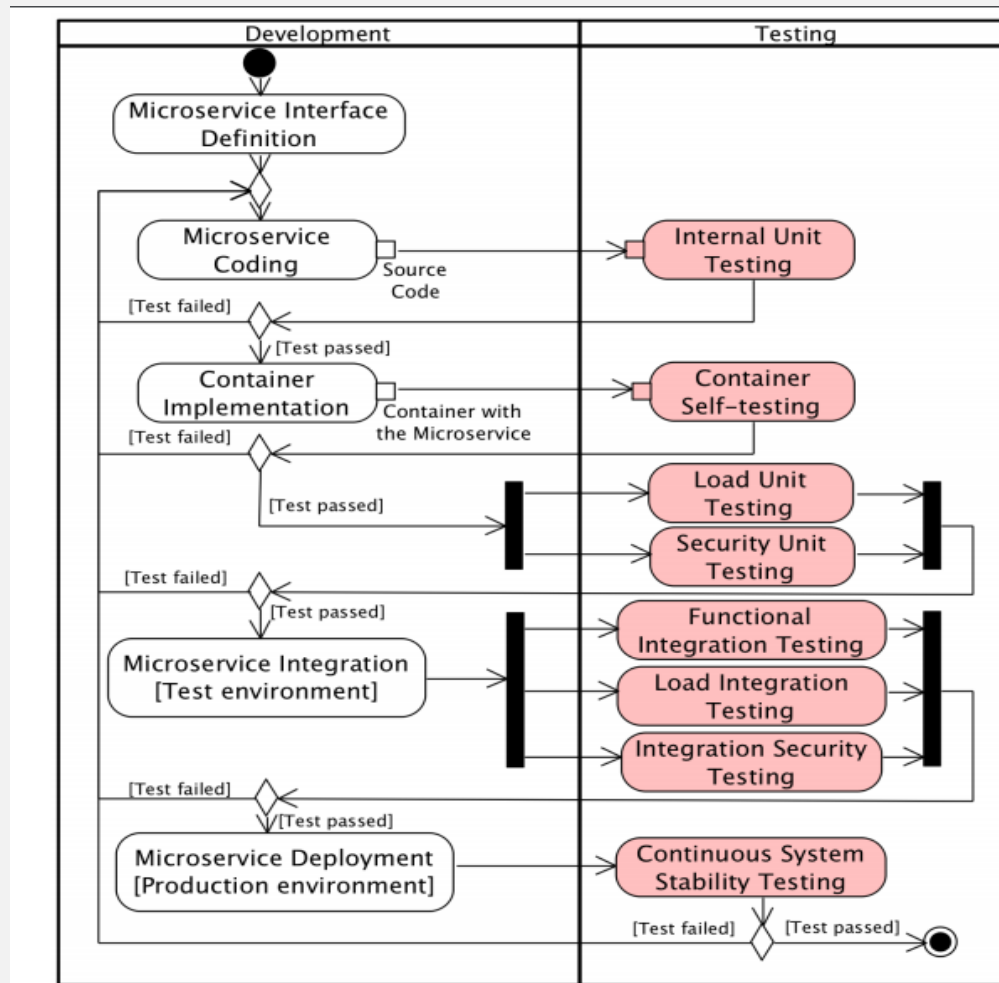
APXITEKTYPА 3 API GATEWAY



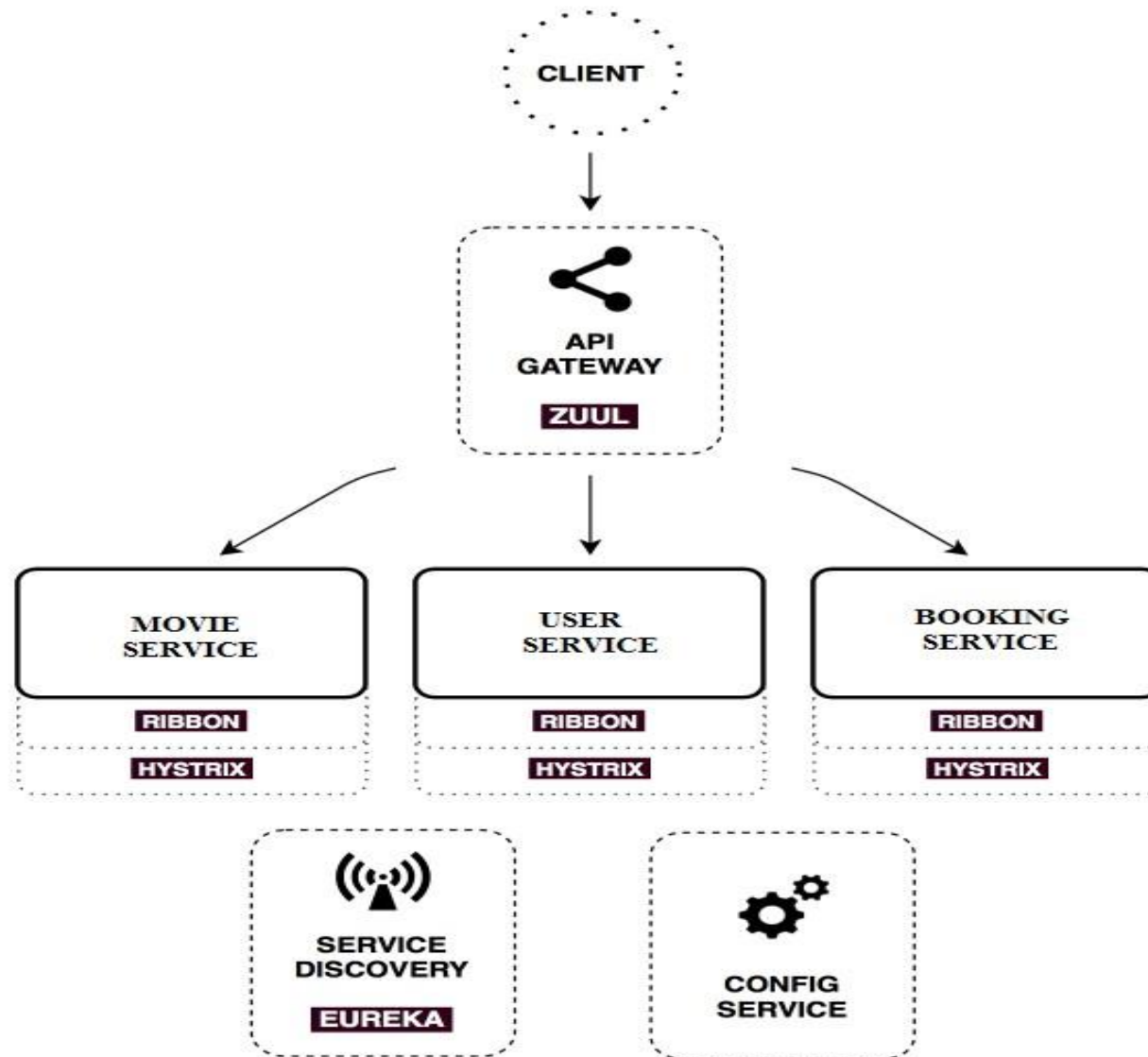
CIRCUIT BREAKER



TESTING PROCESS



GENERAL APP ARCHITECTURE



NETFLIX EUREKA AND HYSTRIX DASHBOARD

← → ↺ 192.168.59.103:8761 🔍 ☆ 🖨️ 📢 📧 ☰

springx HOME LAST 1000 SINCE STARTUP

System Status

| | | |
|-------------|--------------------------|---------------------------|
| Environment | Current time | 2015-07-12T23:42:07 +0000 |
| Data center | Uptime | 02:32 |
| | Lease expiration enabled | true |
| | Renews threshold | 1 |
| | Renews (last min) | 12 |

DS Replicas

Instances currently registered with Eureka

| Application | AMIs | Availability Zones | Status |
|----------------|---------|--------------------|--------------------------------------|
| CONFIGSERVER | n/a (2) | (2) | UP (2) - bde84c0f8653 , configserver |
| GATEWAY | n/a (2) | (2) | UP (2) - 099791d000bd , gateway |
| MOVIE | n/a (1) | (1) | UP (1) - 172.17.0.15 |
| MOVIESUI | n/a (1) | (1) | UP (1) - 172.17.0.14 |
| RECOMMENDATION | n/a (1) | (1) | UP (1) - 172.17.0.12 |
| USER | n/a (1) | (1) | UP (1) - 172.17.0.13 |

General Info

| Name | Value |
|----------------------|-------------|
| total-avail-memory | 739mb |
| environment | |
| num-of-cpus | 8 |
| current-memory-usage | 219mb (29%) |
| server-uptime | 02:32 |
| registered-replicas | |
| unavailable-replicas | |
| available-replicas | |

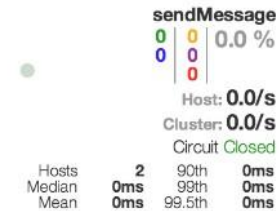
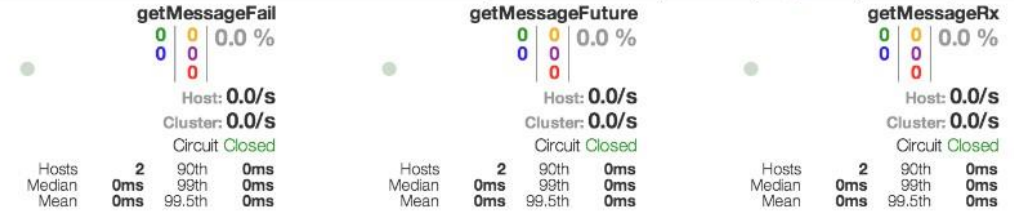
Instance Info

| Name | Value |
|--------|-------------|
| ipAddr | 172.17.0.10 |
| status | UP |

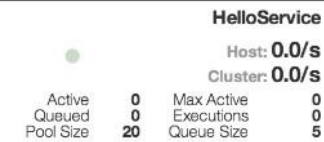
Hystrix Stream: Sample Apps



Circuit Sort: [Error then Volume](#) | [Alphabetical](#) | [Volume](#) | [Error](#) | [Mean](#) | [Median](#) | [90](#) | [99](#) | [99.5](#)
[Success](#) | [Short-Circuited](#) | [Timeout](#) | [Rejected](#) | [Failure](#) | [Error %](#)

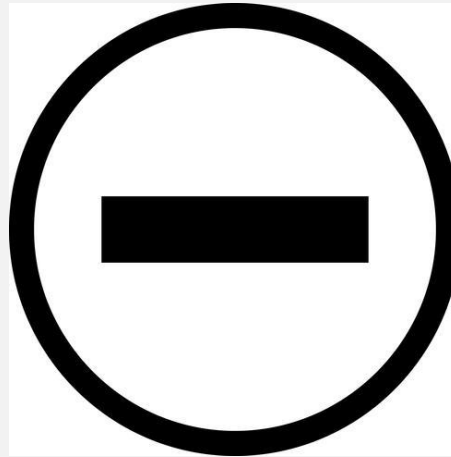


Thread Pools Sort: [Alphabetical](#) | [Volume](#)



MICROSERVICE ARCHITECTURE USAGE LIMITATIONS

Inconsistency



Distribution complexity

Operational complexity

CONCLUSIONS



Microservices are the most suitable for large-scale applications. Smaller apps are usually better off with a monolithic code base, though.



While it's easier to develop and maintain independent microservices, network management requires additional efforts. Container platforms, DevOps practices, and cloud computing can help a lot in adopting the microservices architecture.

THANK YOU FOR
YOUR ATTENTION