

# РАЗПРЕДЕЛЕНИ ПРИЛОЖЕНИЯ

Павел Кюркчиев

Ас. към ПУ „Паисий Хилендарски“

@rkyurkchiev

# АРХИТЕКТУРА „БЕЗ СЪРВЪР“

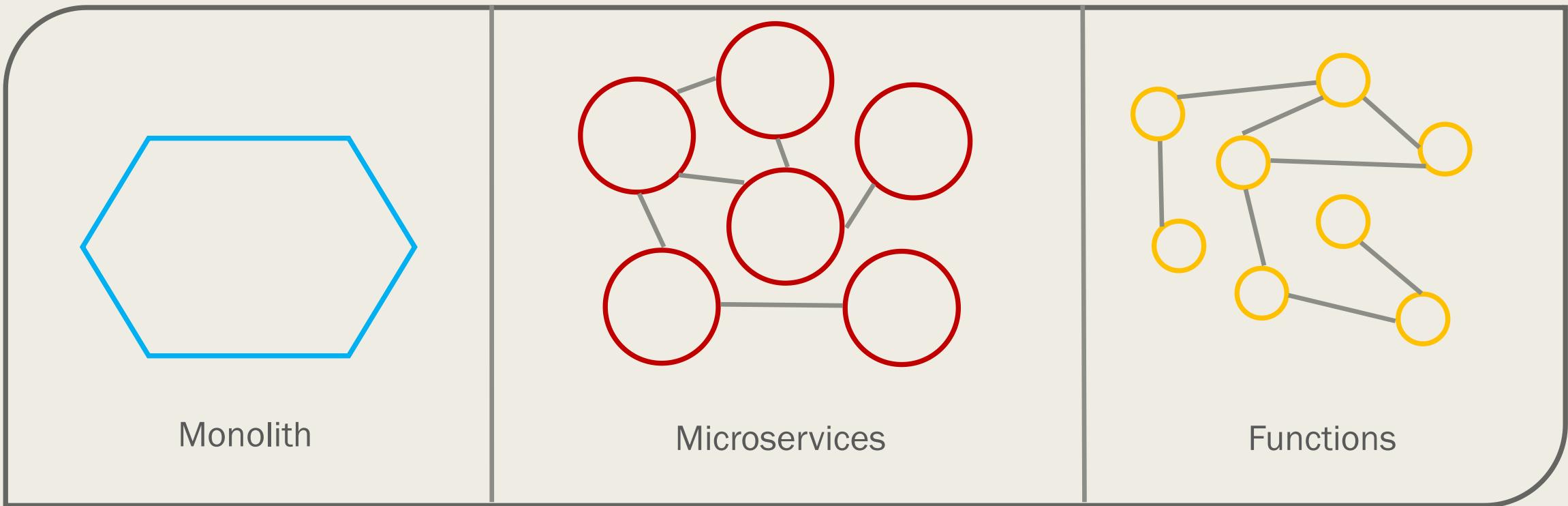
# Традиционна архитектура

# Традиционна архитектура

## 3 tier client-oriented



# Еволюцията на бизнес логиката



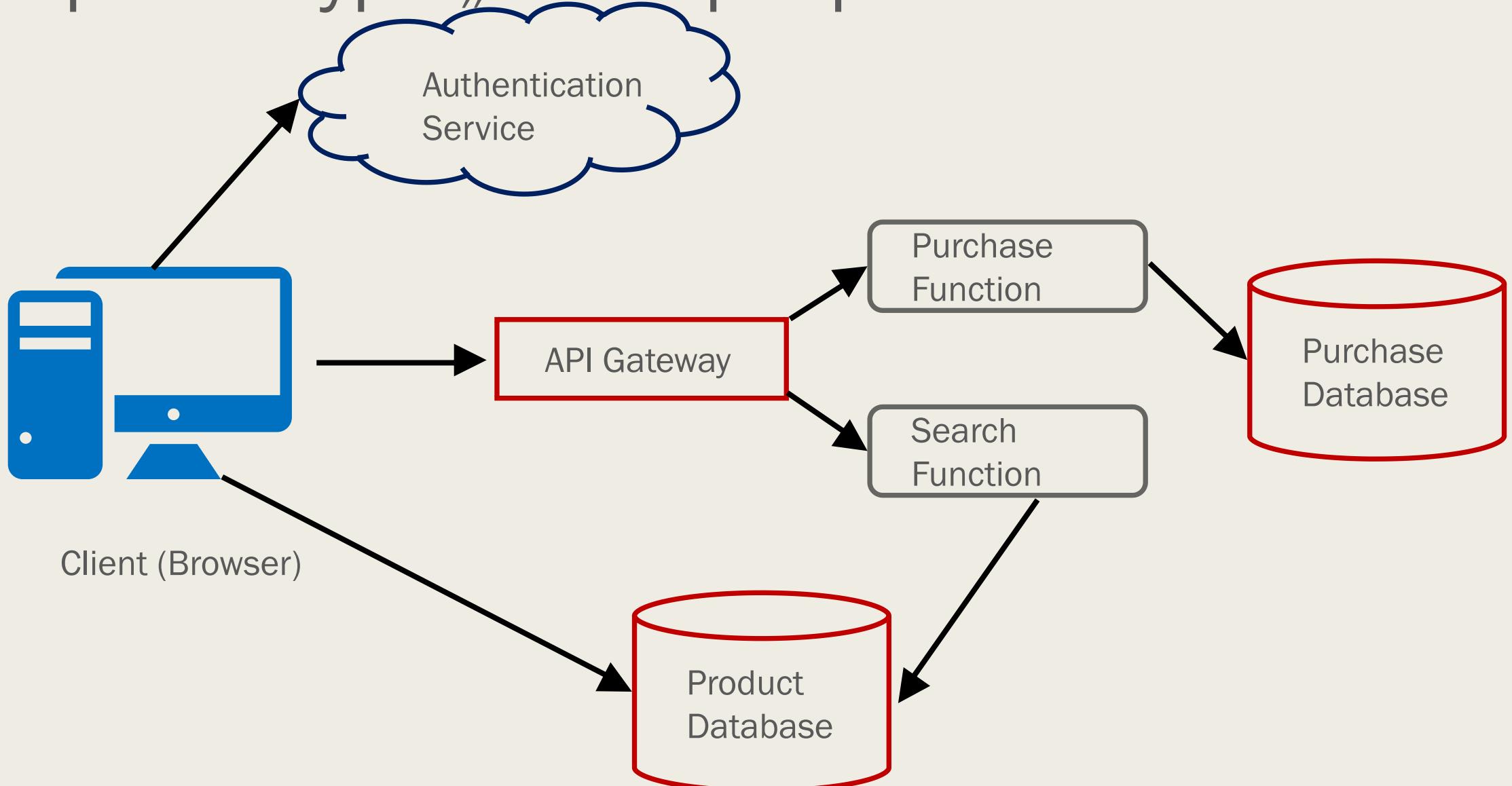
# Какво е архитектура "без сървър"?

- Архитектурата "без сървър" наричаме приложенията, които в голямата си част или изцяло зависят от трета страна за обработката и обслужването на техните нужди. Този тип услуги се описва като BaaS ((Mobile) Backend as a Service).



- „Без сървър“ архитектура може да означава и приложения, на които голяма част от логиката е написана от програмисти, но е качена на отдалечени контейнери, които се извикват и изпълняват при определени събития. Тези контейнери са напълно поддържани от трета страна. За тях можем да мислим като за (Functions as a service / FaaS).

# Архитектура „без сървър“



# Примери за Functions

- Microsoft Azure
  - *Azure Functions - C#, F#, Node.js, Java, PHP ...*
- Amazon AWS
  - *Lambda Functions – Node.js, Java, C#, Go, Python ..*
- Google Firebase
  - *Firebase Functions – Node.js*

## „Без сървър“ означава

- Без нужда от собствен сървър или от управлението му
- Плащаме само това, което използваме
- Разширение само според нашите потребности
- Лесен достъпност и толерантност към средата

# Регионална услуга



# Добре практики за Functions

- Да се минимизира големината на пакетите.
- Да се разделя handler от основната логика на функцията.
- Да се използват Environment Variables за модифициране на поведението.
- Да се възползваме от “Max Memory Used” за да определим правилно големината на function.
- Да се премахнат големите, неизползвани функции.

# AWS Lambda functions

# Анатомия на Lambda Functions

- Handler() function
  - Функцията да бъде извикана при възникването на някакъв обект
- Event object
  - Изпращаната информация по време на извикване на функцията
- Context object
  - Достърен метод отговарящ при изпълнението на функцията

```
exports.myHandler = function(event, context, callback) {  
    console.log("value1 = " + event.key1);  
    console.log("value2 = " + event.key2);  
    callback(null, "some success message");  
}
```

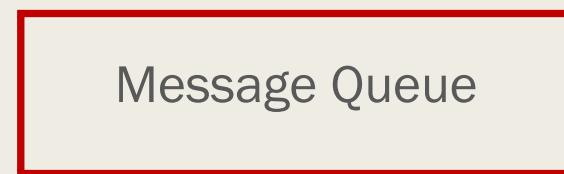
## AWS Lambda function

# Functions модел на изпълнение

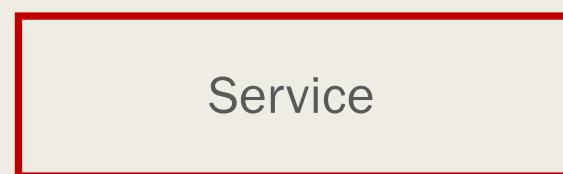
Synchronous (push)



Asynchronous (event)



Stream-based

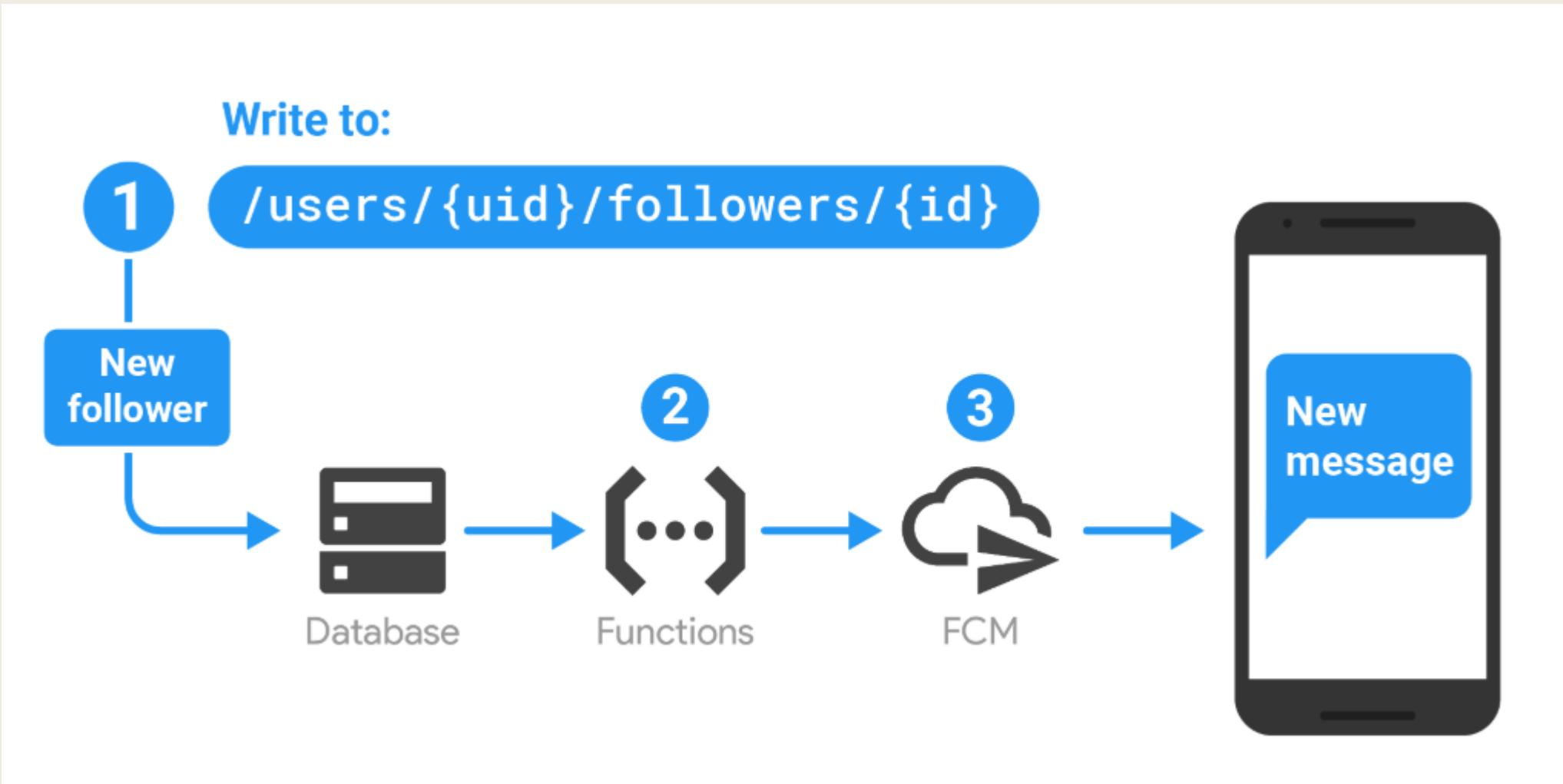


# Google Firebase Functions

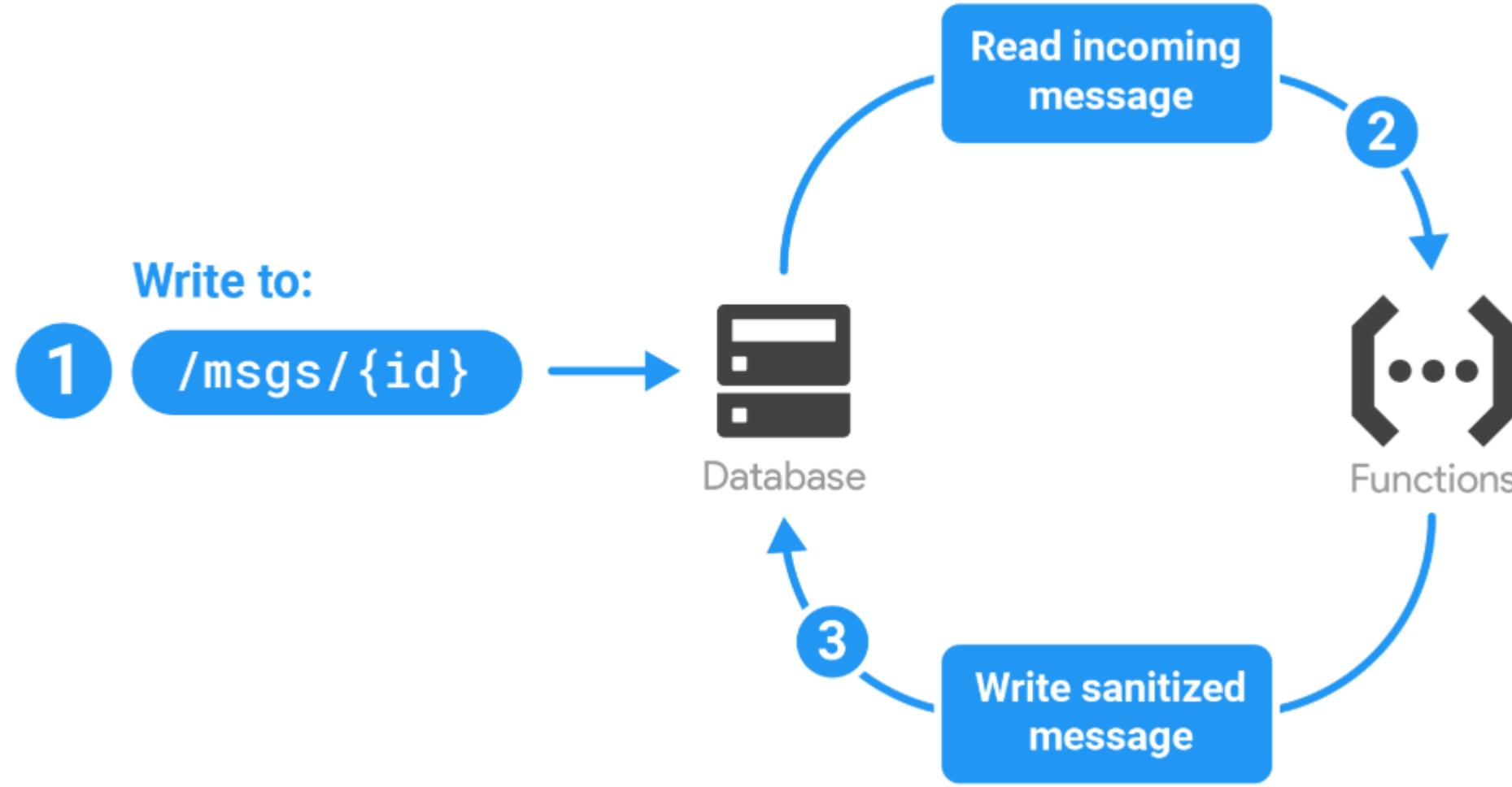
# Интеграция на Firebase platform

- Cloud Firestore Triggers
- Realtime Database Triggers
- Remote Config Triggers
- Firebase Authentication Triggers
- Google Analytics for Firebase Triggers
- Crashlytics Triggers
- Cloud Storage Triggers
- Cloud Pub/Sub Triggers
- HTTP Triggers
- ....

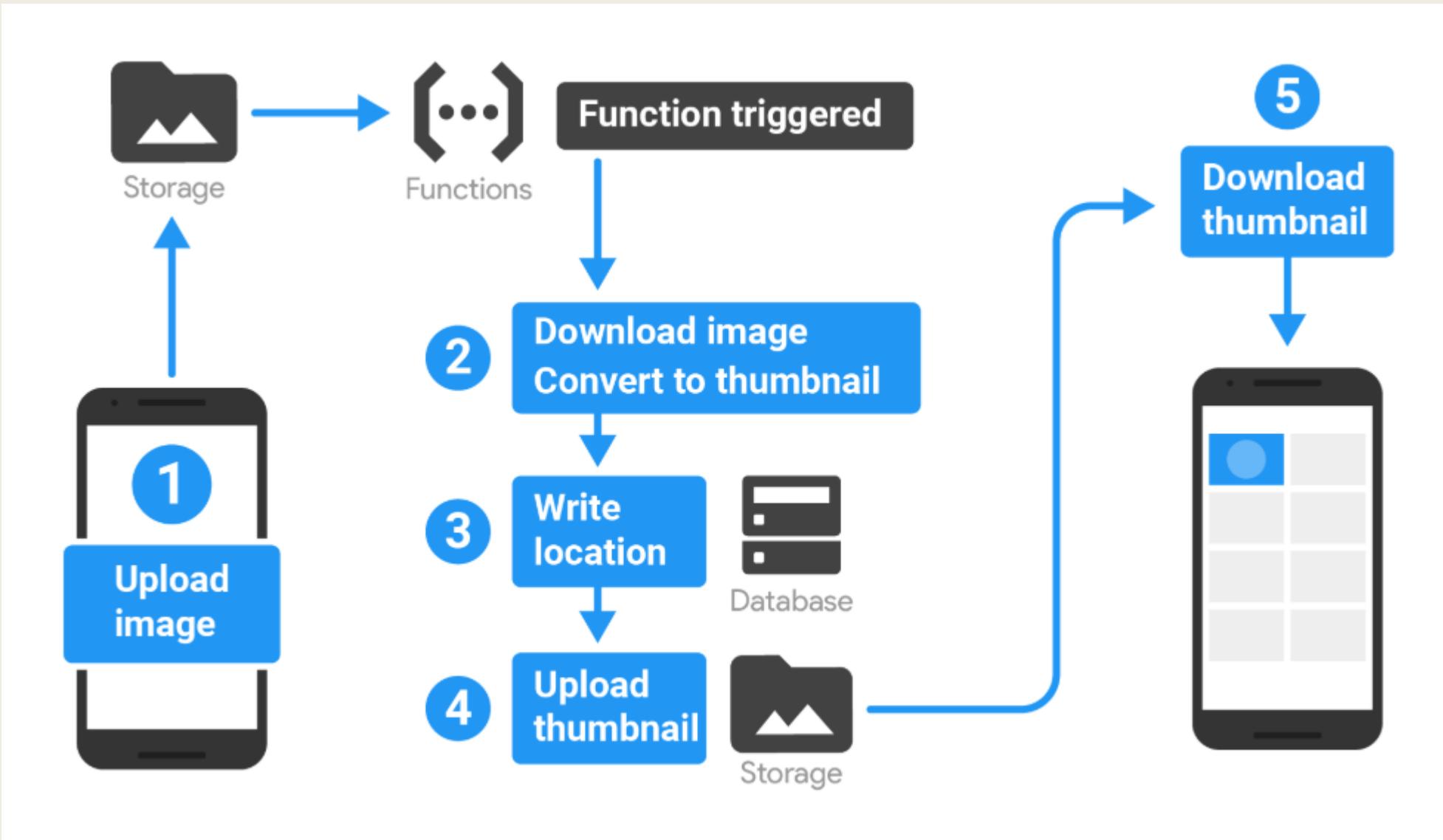
# Уведомяване на потребител



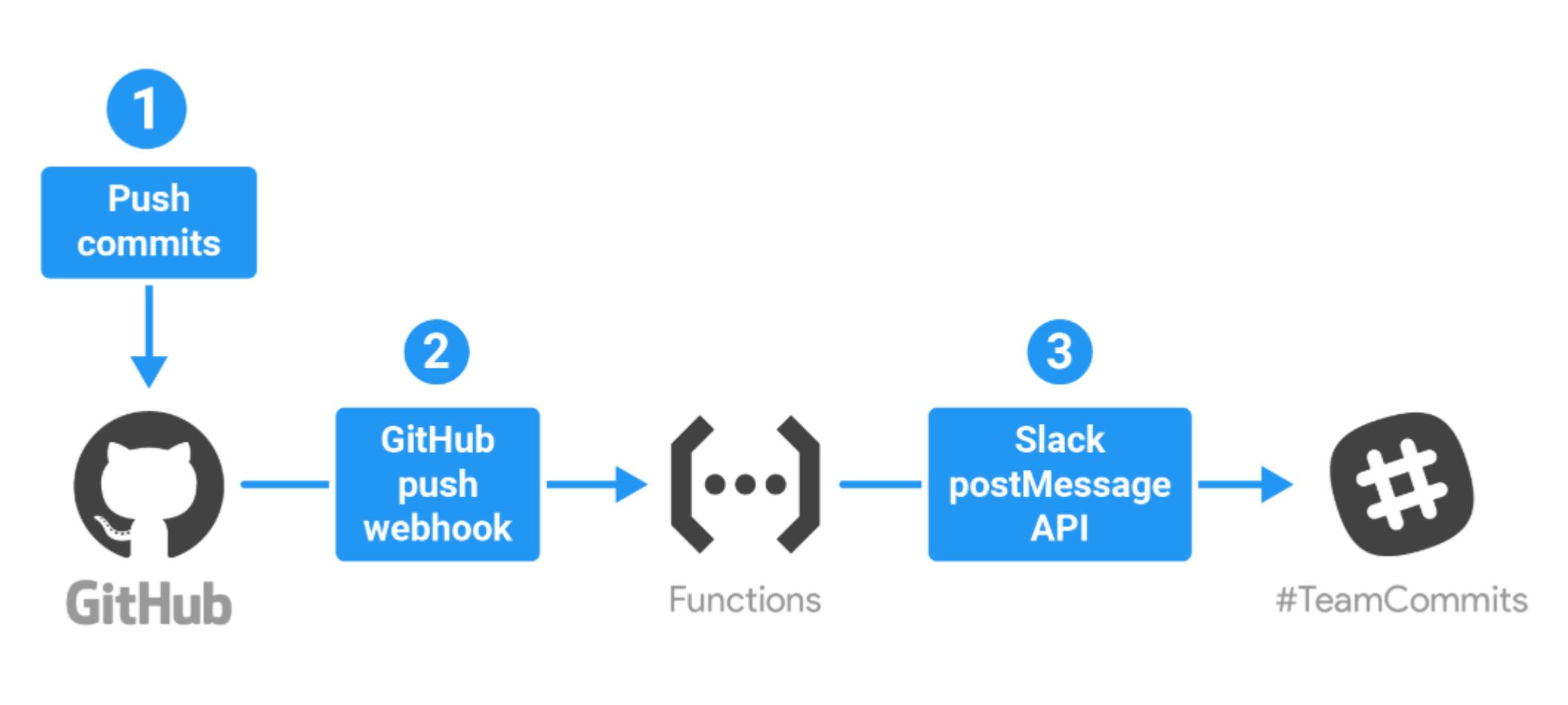
# Realtime Database интеграция



# Обработка на сложни задачи



# Интеграция с трети услуги



# DEMO FIREBASE FUNCTION 1

examples/helloWorld-firebase-function-ts

# DEMO FIREBASE FUNCTION 2

examples/translate-firebase-function-js

# Azure Functions

# Интеграция на Azure function

Type	1.x	2.x and higher <sup>1</sup>	Trigger	Input	Output
Blob storage	✓	✓	✓	✓	✓
Azure Cosmos DB	✓	✓	✓	✓	✓
Azure SQL (preview)		✓		✓	✓
Dapr ↗ <sup>3</sup>		✓	✓	✓	✓
Event Grid	✓	✓	✓		✓
Event Hubs	✓	✓	✓		✓
HTTP & webhooks	✓	✓	✓		✓
IoT Hub	✓	✓	✓		
Kafka ↗ <sup>2</sup>		✓	✓		✓
Mobile Apps	✓			✓	✓
Notification Hubs	✓				✓
Queue storage	✓	✓	✓		✓
RabbitMQ <sup>2</sup>		✓	✓		✓
SendGrid	✓	✓			✓
Service Bus	✓	✓	✓		✓
SignalR		✓	✓	✓	✓
Table storage	✓	✓		✓	✓
Timer	✓	✓	✓		
Twilio	✓	✓			✓

# Azure functions 1 vs 2 vs 3 vs 4

Language	1.x	2.x	3.x	4.x
C#	GA (.NET Framework 4.8)	GA (.NET Core 2.1 <sup>1</sup> )	GA (.NET Core 3.1) GA (.NET 5.0)	GA (.NET 6.0) Preview (.NET Framework 4.8)
JavaScript	GA (Node.js 6)	GA (Node.js 10 & 8)	GA (Node.js 14, 12, & 10)	GA (Node.js 14) GA (Node.js 16)
F#	GA (.NET Framework 4.8)	GA (.NET Core 2.1 <sup>1</sup> )	GA (.NET Core 3.1)	GA (.NET 6.0)
Java	N/A	GA (Java 8)	GA (Java 11 & 8)	GA (Java 11 & 8)
PowerShell	N/A	GA (PowerShell Core 6)	GA (PowerShell 7.0 & Core 6)	GA (PowerShell 7.0) Preview (PowerShell 7.2)
Python	N/A	GA (Python 3.7 & 3.6)	GA (Python 3.9, 3.8, 3.7, & 3.6)	GA (Python 3.9, 3.8, 3.7)
TypeScript <sup>2</sup>	N/A	GA	GA	GA

# DEMO AZURE FUNCTIONS

examples/ToDoOperations

ВЪПРОСИ ?

