## Serverless

TRADE-OFFS

Laura Pernía Blanco Miguel Cuesta Martínez Nicolás Lozano García



# INTRODUCTION

#### What is serverless?

## Serverless = No servers <u>to manage</u>

## FAAS

Serverless

# FaaS vs PaaS

Platform as a Service



## MAIN PROVIDERS



Azure Functions



Google Cloud Functions





Launch of AWS Lambda

**3.5x calls** 

In 2021 than in 2019



Main advantages of serverless computing

## TWO TYPES OF ADAVANTAGES

#### FROM THE TECHNICAL POV

Architectural concerns

#### FROM THE BUSINESS POV

Economic savings

## **Less Operational Cost**

Lots of similar services

## **ADVANTAGES OF FaaS**

#### **Built-in Elasticity**

#### Optimization

#### **Event Driven Models**

Language Independence

## **ADVANTAGES OF BaaS**

#### Reduce development costs

Easier to manage

Use existing functionality

Components you can ignore

#### Isolation

Improved security

# Serverless can help the planet

Environmental impact

## **BUSINESS ADVANTAGES**



**Focus** 

the design

time to market

**Good** for small business



Despite the advantages of serverless, there are some points to take into account if we plan to use it

## PRINCIPAL DISADVANTAGES

The main inconveniences a project might find when implementing serverless architecture are:



- Virtually impossible to test / debug
- Complexity increases
- Cold Start Problem
- Service Limitations
- API Hell

## LEARNING CURVE OF MAPPING LANGUAGES

#### NO PREVIOUS EXPERIENCE

It's possible that no one in the team has used it before

#### LACKING USUAL FEATURES

Like typechecking and import (basic reutilization of code)

## VIRTUALLY IMPOSSIBLE TO TEST / DEBUG

Lacking easy to use

- Debugging
- Logging

The cloud vendor might make access to the information difficult!



## **COMPLEXITY INCREASES**

MULTIPLE NETWORKS

#### COORDINATION

IS IT WORTH IT?

### COLD START PROBLEM



## **SERVICE LIMITATIONS**

Not an actual limitation?

Due to the serverless philosophy:

- Processes include small functionality
- Low usage of memory per service

You have to take your project into account



## **API HELL**

#### OVERCOMPLICATED GATEWAYS

Ambitious design that

difficults their use

# Any Juestions ?

Thanks for attending!