# TUGA IT 2017

LISBON, PORTUGAL

# THANK YOU TO OUR SPONSORS

## PLATINUM

Microsoft

Rumos

## GOLD

oh22information services

## SILVER

everis

an NTT DATA Company

TUGA IT

# PARTICIPATING COMMUNITIES

agileconnect®

Office 365 Portugal

CLOUD PRO PT

SQLPort | Comunidade Portuguesa de SQL Server

sharept

PASS CHAPTER PORTO.DATA

netponto comunidade

OUGPT

DEVOPS PORTO

TUGA IT

# TUGA IT 2017

LISBON, PORTUGAL

# MIXED ARCHITECTURE

Andrea Giunta
Azure MVP
Senior architect @OrangeDev

# Hi there!

- Former MSP
- Founder of DotNetSicilia
- Azure MVP since 2015
- Used to work on cloud projects
- Used to work as fullstack developer
- I have an open issue with UI

@andreagiunta63

TUGA IT

# Agenda

- How our software architecture look alike?
  - Comparison
  - Microservices vs Serverless
- There is a standard strategy?
- How and what should i know?
- Q&A

# How our software architecture look alike?

- We are used to write software for 3 kinds of scenario
  - Desktop
  - Web
  - Mobile

- And usually this is done with a trustly and true friend
  - Server / Client
  - Maybe with a decomposition on 3 layers
- But is this really modern in a cloud first reality?

# Server / Client

- We all should know the classic pattern with servers and clients
- Could be implementd in different ways
- An often used solution is a multi-tier software architecture

- These kind of «monolith» can be deployed to cloud but there are sever architectural issue

# Server/Client Cloud Deployment issue

- A few side effects of Server/Client architecture
  - Horizontal scaling
  - No granularity
  - Less control on costs

- At the same time an old 3layer server client architecture can be «hosted» into the cloud both via IaaS or PaaS
- But remain really distant from how a cloud based application should look like

# What about SOA and ROA?

- Both service and resource oriented architecture can be implemented in different ways.

- Both are usually cloud-friendly
- Not if implemented with a server/client architecture, why?

# Microservices

“Microservices is a variant of the service-oriented architecture (SOA) architectural style that structures an application as a collection of loosely coupled services. In a microservices architecture, services should be fine-grained and the protocols should be lightweight. The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to understand, develop and test”

-Wikipedia

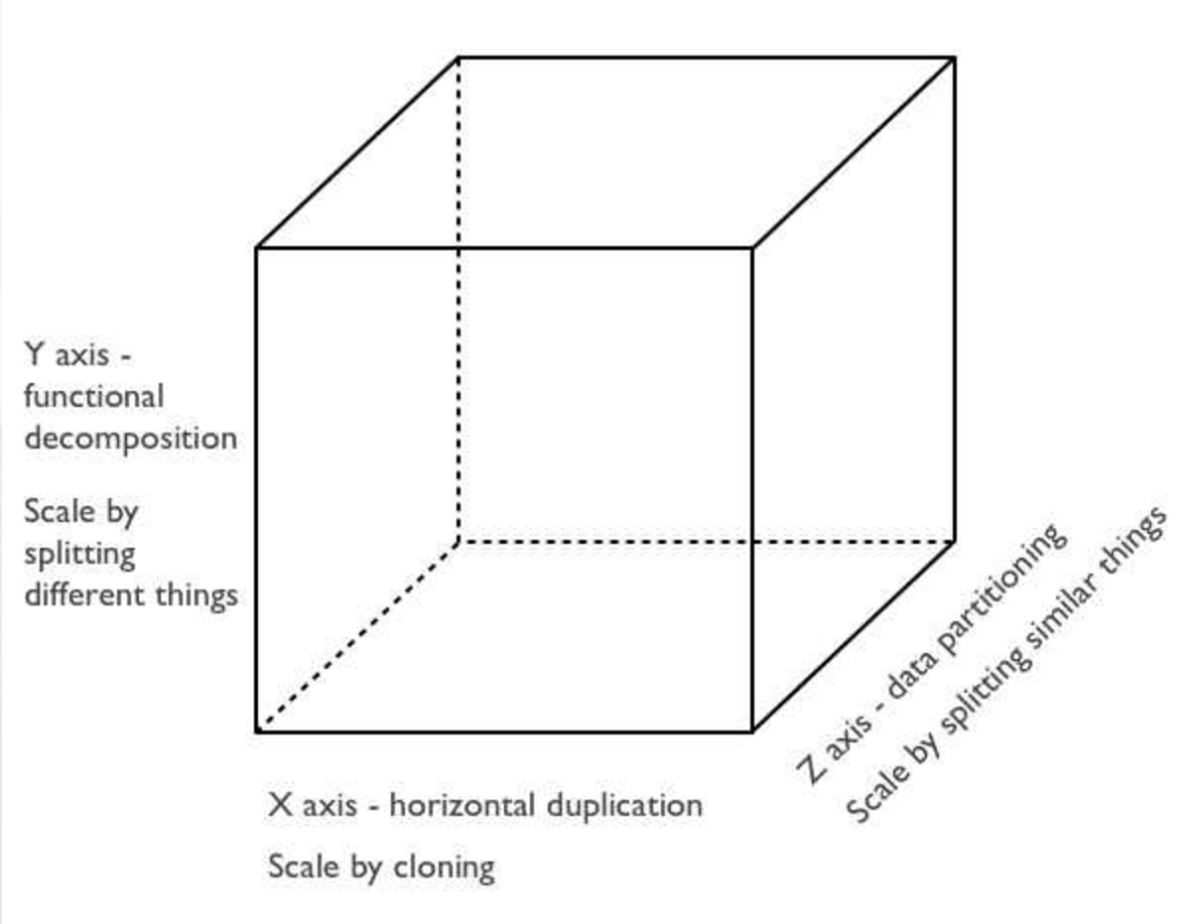
# How microservices work?

- Imagine a 3 layer architecture like a wall… Now consider buy an hammer
- Take the hammer, and apply a «light» decomposition on the 3layer wall

- You can apply a decomposition by Name or by Verb
  - Items obtained communicate via API
  - Items are in charge of a small job

# Scalecube

Consider reading on : http://microservices.io/articles/scalecube.html




TUGA IT

# Serverless

Serverless computing, also known as function as a service (FaaS), is a cloud computing code execution model in which the cloud provider fully manages starting and stopping of a function's container platform as a service (PaaS) as necessary to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request, rather than per virtual machine, per hour.

-Wikipedia

# How serverless work?

- The idea behind is quite simple : create decoupled small unit of application logic
- Focus on code, not on server
- Provisioning, maintenance, and scalability can be provided as a service

# So microservices is serverless?!?

- Well, this can be a «strange» question
- They are similar but… they aren’t the same
- The main difference is on how you think to manage your service.
- If you have to orchestrate your containers you are not in a Serverless scenario

# In medio stat virtus

- The real world is something different
- You have to fit with customer requirements
- Or customer IT platform availability

- “If you cannot have a single malt scotch, try at least to have a blended one” [cit.]

TUGA IT

# Evolve an old architecture

- Starts with a classic 3tier server/client application
- As said before, this can be hosted in the cloud
- We have two concurrent possibilities
  - Evolve existing code to cloud
  - Add new feature, born in the cloud

# Adding new features

- Shoud be easy but…
- I’ve to choose if this feature has to be deployed as a Microservice, or a function, or a webapi.
- This new feature MUST communicate with my multi tier application
  - Anyway i’ve to add a communication channel
    - Via webservice
    - A WCF service via tcp
    - Queue
    - Anything you can handle

# Evolve existing software

- Analyze
  - Application domain
  - Feature complexity and weight
  - Scaling scenario
- Define
  - How to make a distributed architecture
  - Born in the cloud service you will use
- Partition
  - Code porting to the new architecture

# Evolve existing software

- Carefull with cloud
  - Understimate the complexity would be an hell for your developer
  - Migrating to cloud could require several intermediate steps to be completed

# Delicate items

- Authentication
- Authorization
- Data Transfer Object
- Third Party components
- Application data workflows

# Why&When consider mixed architetures


TUGA IT

# The five pillars of software quality

- Scalability
  - The ability of a system to handle increased load.
- Availability
  - The proportion of time that the system is functional and working.
- Resiliency
  - The ability of the system to recover from failures and continue to function.
- Management
  - Operations processes that keep an application running in production.
- Security
  - Protecting applications and data from threats.

TUGA IT

# PLEASE FILL IN EVALUATION FORMS

FRIDAY, MAY 19th

https://survs.com/survey/cprwce7pi8

SATURDAY, MAY 20th

https://survs.com/survey/l9kksmlzd8

# YOUR OPINION IS IMPORTANT!

TUGA IT

# THANK YOU TO OUR SPONSORS

## PLATINUM

Microsoft

Rumos

## GOLD

oh22information services

## SILVER

everis

an NTT DATA Company

TUGA IT