



#GlobalAzure

#CloudGenVerona

@cloudgen_verona

Thanks to all the sponsors



PREMIUM SPONSOR



BASIC SPONSOR



CODICEPLASTICO





TOPIC

Reactive Programming

Gestiamo una flotta di veicoli in movimento

Who I am



RICCARDO ZAMANA



@riccardozamana



<https://github.com/zama202>



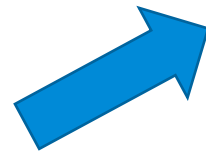
<https://www.linkedin.com/in/riccardozamana/>



..where are we going?..

- Why reactive architecture is important?
- What problem it's trying to solve?
- How are Reactive Systems related to Reactive Programming?

... Is "wrong" the answer, really?



The past and the present



Key factor	Yesterday	Today
Size of Installation	Large inst - Few nodes	Small inst – many nodes
Amount of data	Gbs	Tbs/Pb
Data change	Nightly (batches)	Constantly change
Maintenance Window	3 hours ... Comeback later!	101% SLA
User patience	< 30sec	< 3sec

Whats happen



- In case of failure, is difficult to recover data stream
- 100% SLA is achievable
- Small maintenance, no more hours
- Micro Deployments are born
- User patience is less then 3seconds
- Software must respond... otherwise user search for alternatives



PROVIDE THE SAME EXPERIENCE IN EVERY KIND OF
CONDITION

The Reactive Manifesto

Published on September 16 2014. (v2.0)



PRINCIPLES



RESPONSIVE (faster then ever)

- Generate USER CONFIDENCE

RESILIENT (remain responsive in case of failure)

- Replication, Isolation, Containment, Delegation
- Failures isolated in every single component
- Only recovery may be delegated to external components

ELASTIC (active during deployments or scaling)

- responsive during scale out or scale down
- predictive auto scaling technique
- cost effectiveness

MESSAGE Driven (async, non blocking)

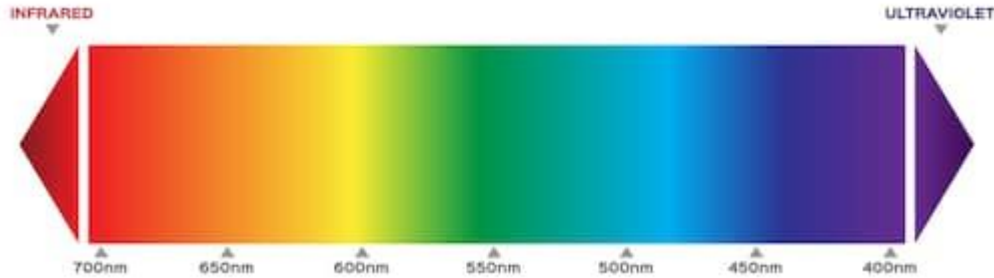
- Decouple everything, isolation, location transparency
- Resource Consumption only when are «active and «GREEN LIGHT»



Reactive Programming VS Reactive Systems/Architecture

- Reactive SYSTEMS: Apply principles to Architecture
- Reactive Programming: Used to implement Reactive Systems
- RX.NET is only a programming technique
- You can do a ReactiveSystem WITHOUT ReactiveProgramming
- REACTIVE Systems (based on Reactive Microservice) are separated throw Async mimic and boundaries

REACTIVE MICROSERVICES



My Opinion: No Black, nor white.
But a spectrum of capabilities between Monolith
and MicroServices

Remember the SOA dream?

Entities/Domains with dedicated
services, with different Database,
using other services throw API.

MicroServ is a SOA subsystem... but the difference?

- Independent Govern
- Async communication
- Continuous deployment
- Independent team delivery

SINGLE RESPONSABILITY PRICIPLE <| |> BOUNDED CONTEXT

KEYWORD n.1: ISOLATION



ISOLATION IN STATE

No backdoor straight to other DBs

Evolve internally throw API version

Health State API embedded

ISOLATION IN SPACE

Discovery or Gateway to find out the MS Location – not hard coded

Independent scale out/down

ISOLATION IN TIME

No wait for other MS. Async message Driven mimic and Not Blocking also.

Eventual consistency is a MUST

ISOLATION IN FAILURE

Other consequences on other MS if there is a failure



BULK HEADING

Failure is isolated in failure zones (made by more ms)

Failure is not propagated

It admits a degraded flow but not dead flow => doubling the way or Health API

CIRCUIT BREAKER

Problem: OverLoaded Services? pMaybe other MS could retry and worse the situation.

With a 3 state semaphore, I can send the retry time or predict it !!

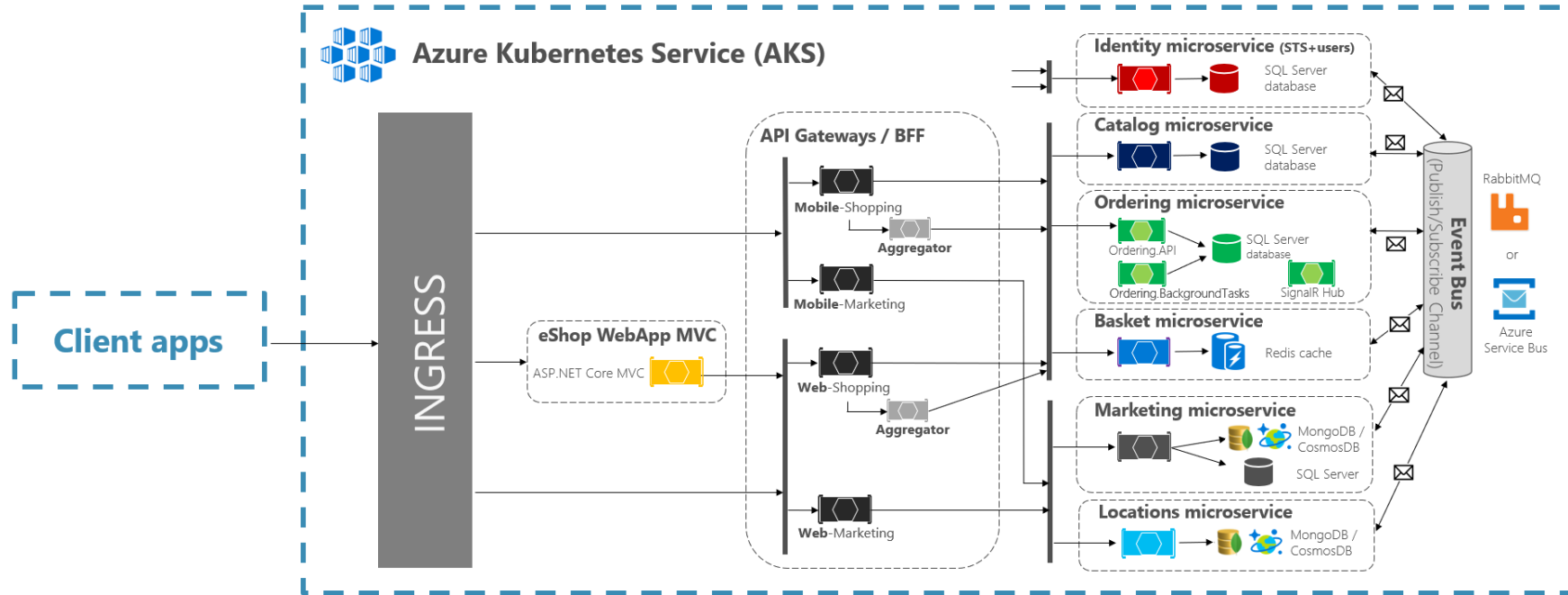
COMPETING CONSUMER

Message Driven: Enqueue, Exchange, Dequeue with Async messaging

ISOLATION TECHNIQUES



eShopOnContainers (Deployment into Kubernetes environment)



GATEWAY SERVICES

Managing complexity: Who aggregates MS.API in macro services? Should client know how to aggregate MS.API? i client hanno bisogno di gestire loro i casini di disponibilità dei micro servizi? PROXY/GATEWAY: not only binding! But aggregation also.

AUTONOMY:

Every service must have strategies to solve its own problem (maintaining internal state, with eventual consistency, avoiding direct dependencies on external services)



Small discrete steps

async based, with CALLBACK:

- FUTURE/PROMISES
- STREAMS



Is it possible to have ReactiveProgramming without a Reactive System?

YES. Examples:

- 1) One unique node => No Resilience
- 2) Local cache and no way to sync cache with other nodes
=> NO Elasticity

ACTOR MODEL



- An Actor is a primitive unit of computation
- Actors are completely isolated from each other
- They will never share memory.
- Actor can maintain a private state
- They come in systems, never alone
- They need to have addresses
- It will just execute one at a time
- Concurrence means Actor of Actors
- Communicate with asynchronous messages

When an actor receives a message, it can do one of these 3 things:



- A) Create more actors
- B) Send messages to other actors
- C) Designate what to do with the next message



LOCATION TRANSPARENCY:

Local or remote, it doesn't matter.

They can be ROUTEE, routed by a ROUTER

YES. I can send msgs to a Router, without knowing about the subscribers.

WE HAVE TO COMBINE REACTIVE Systems, message driven, in order to have Location Transparency. Location TRANSPARENCY = Elasticity + Resilience

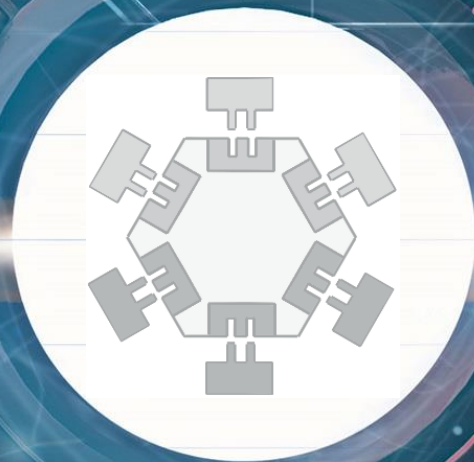
REACTIVE SYSTEM WITHOUT ACTORS?



Could be...

1. Service Registry
2. Load Balancer
3. Message BUS
4. Service that interchange msgs using a BUS

IT's a reactive system... But not inside the Microservices, because they lack of an Observer, and there isnt an entity alive.

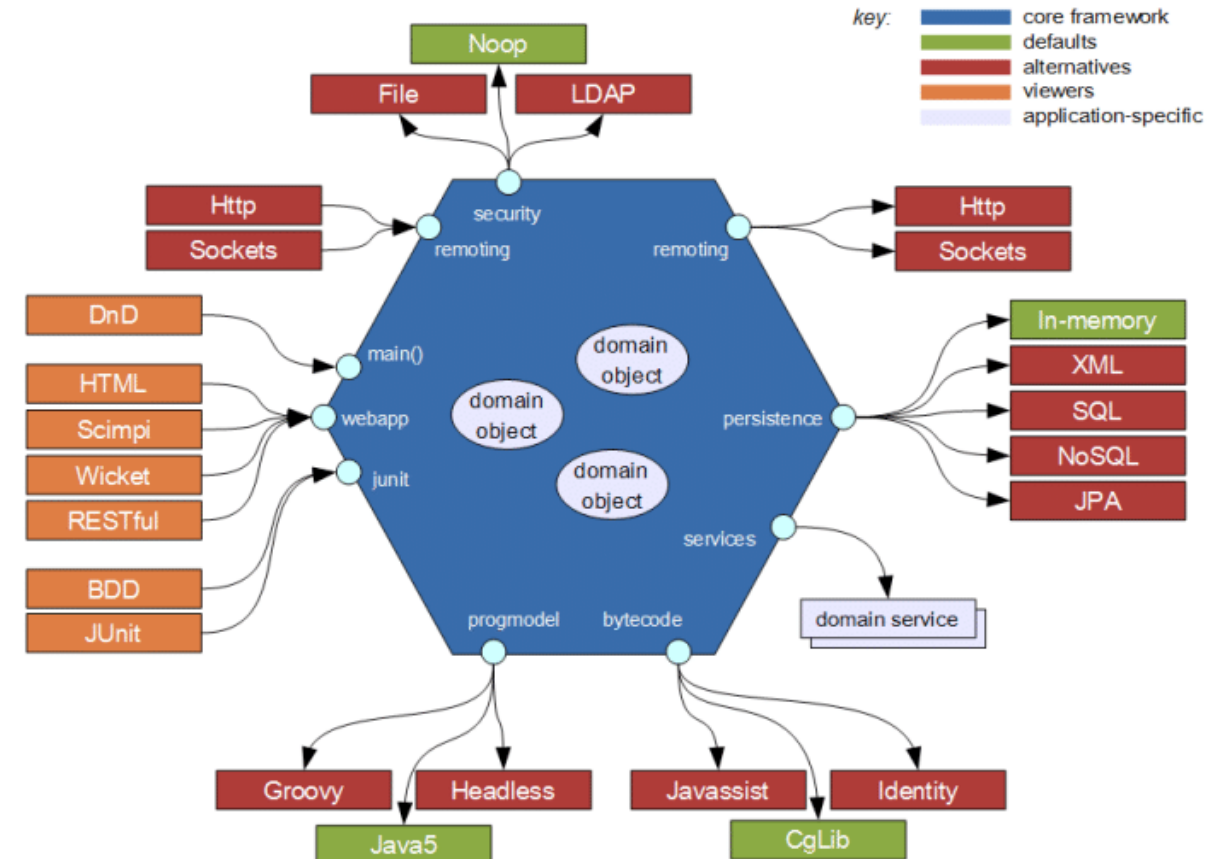


HEXAGONAL ARCHITECTURE

WHAT'S ABOUT THE REAL DESIGN?



- **HEXAGON Fulcrum is DOMAIN**
- **2 SIDES, RIGHT & LEFT**
 - **FIRST SIDE:** DATA SIDE API with ADAPTERS to DBs, Fs, etc. consumed by DOMAIN
 - **SECOND SIDE:** USER SIDE API, where PORTS (they are API) that serve Domain data to USER (interface, API, message broker)
- **INFRASTRUCTURE:** where Adapters maps PORTS



REAL DECOUPLING || REAL PORTABILITY || REAL SCALING

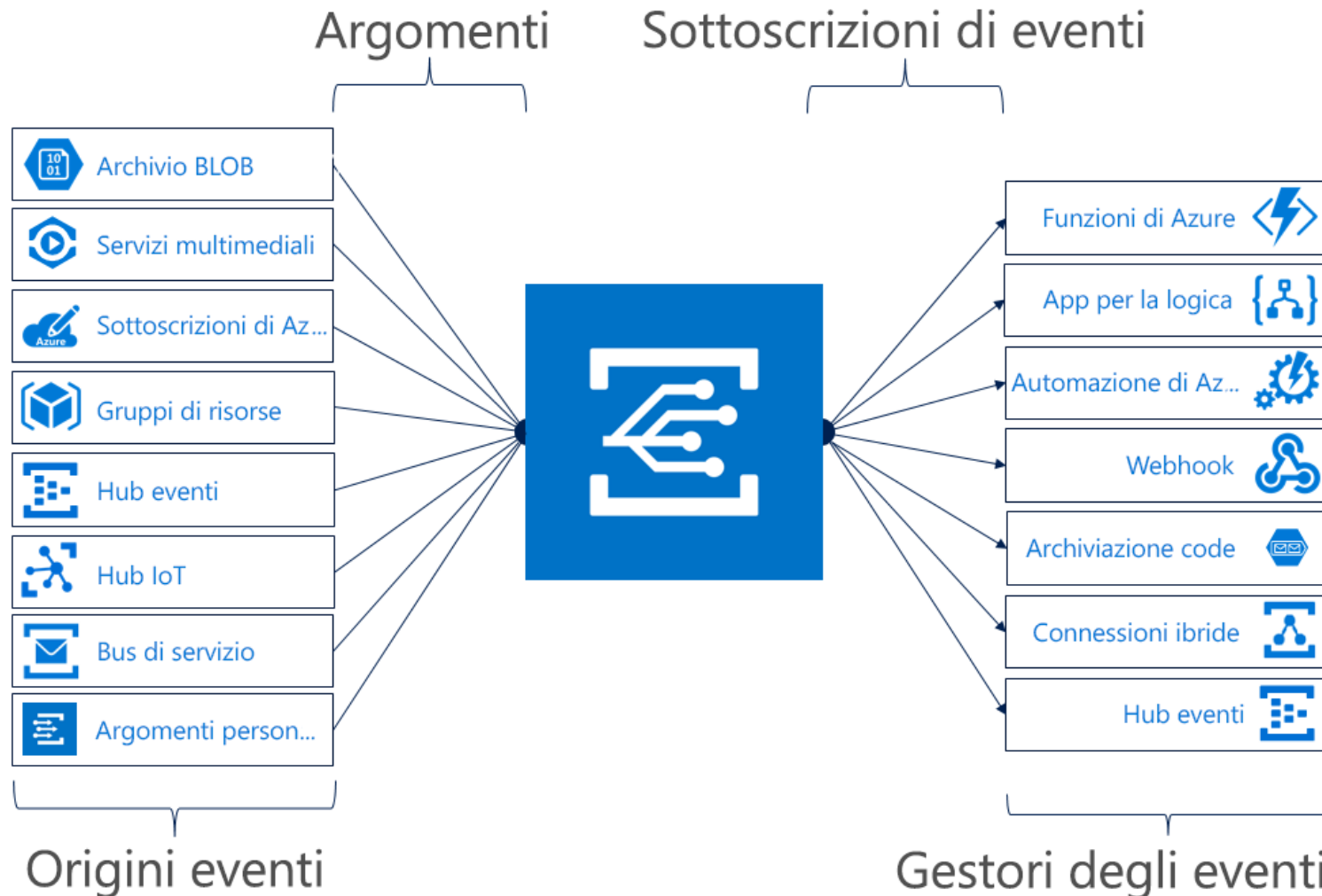


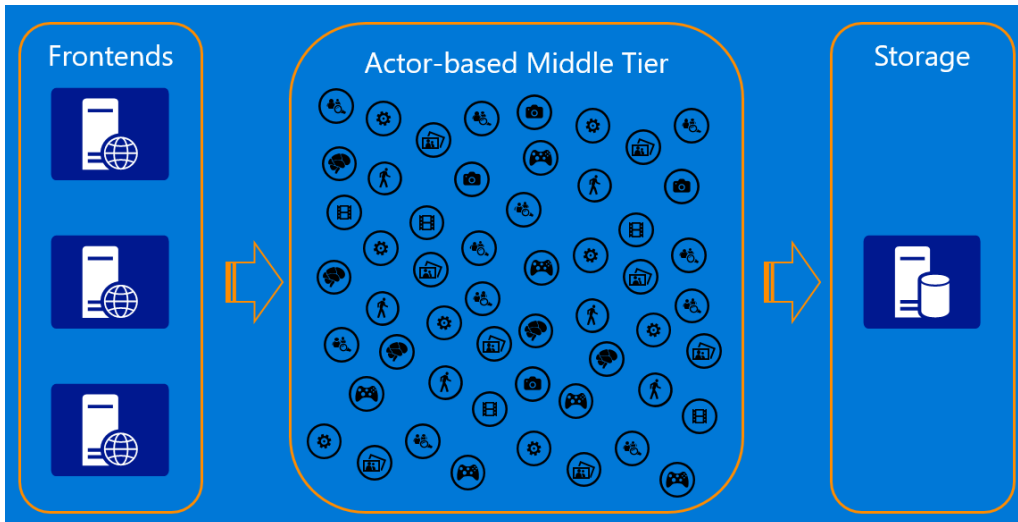
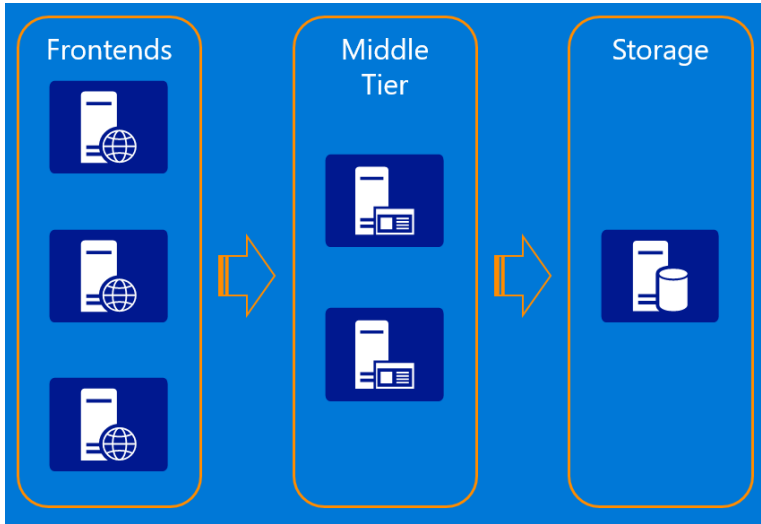
“My 2 cents ..



DEMO

Event GRID





Everything is an actor.

Computational entity that, in response to a message it receives, can concurrently:

- send a finite number of messages to other actors;
- create a finite number of new actors;
- designate the behavior to be used for the next message it receives.

Asynchronous communication and control structures

Recipients of messages are identified by address, sometimes called "mailing address". Thus an actor can only communicate with actors whose addresses it has.



QUESTA E' la chiamata da mettere nel WebHook del EventGrid

<https://{functionappname}.azurewebsites.net/runtime/webhooks/eventgrid?functionName={functionname}&code={systemkey}>

COME SI FA ad ottenere la system key? QUESTA è la chiamata GET per ottenere la SystemKEY

http://{functionappname}.azurewebsites.net/admin/host/systemkeys/eventgrid_extension?code={masterkey}

DA DOVE PRENDI LA MASTER KEY? Pubblici la funzione, vai su MANAGE e trovi la _Master

ESEMPIO COMPLETO:

http://FuncAppEventTrigger20190422124458.azurewebsites.net/admin/host/systemkeys/eventgrid_extension?code=Sri9IKxvpbbpJdFn6dF3dIXZoAuk4DF6URWYf5g1yVGw1A0q0787AQ==

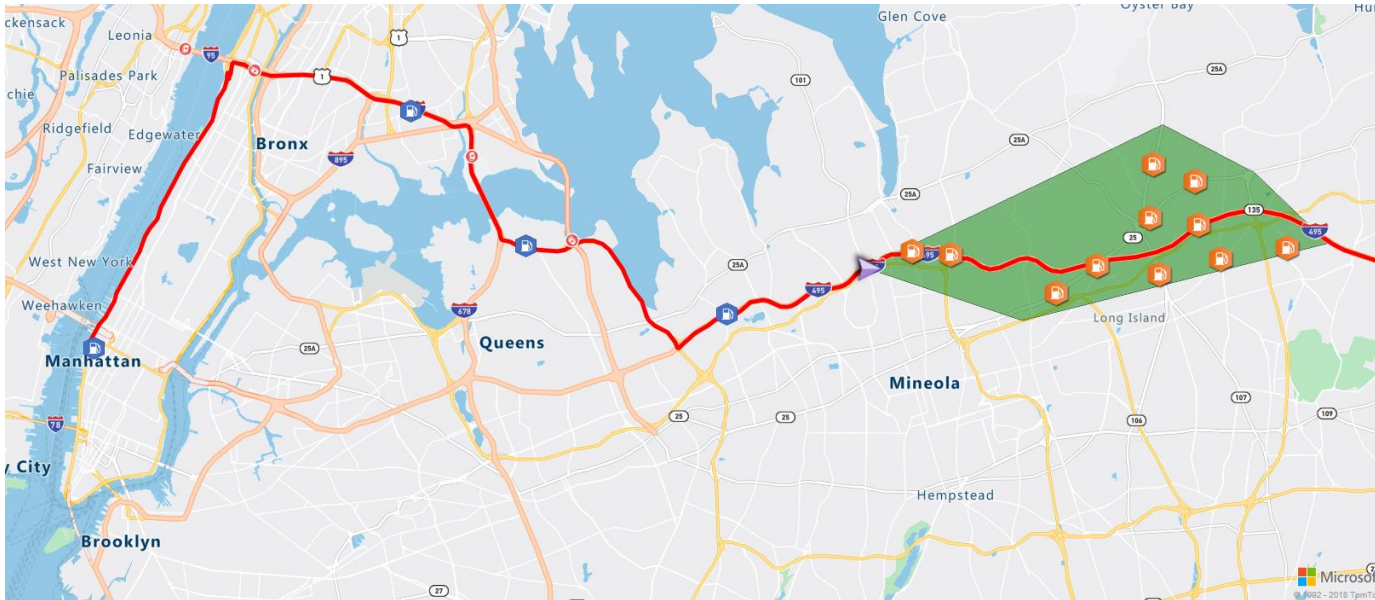
ECCO LA RISPOSTA:

```
{
  "name": "eventgrid_extension",
  "value": "Bks9oZKL9VS9aB2O/Poir073bW671UBZhGc34//vdK9ZtMqw2ZwQiQ==",
  "links": [{
    "rel": "self",
    "href": "https://funcappeventtrigger20190422124458.azurewebsites.net/admin/host/systemkeys/eventgrid_extension"
  }]
}
```

QUINDI:

<https://FuncAppEventTrigger20190422124458.azurewebsites.net/runtime/webhooks/eventgrid?functionName=StartSession&code=Bks9oZKL9VS9aB2O/Poir073bW671UBZhGc34//vdK9ZtMqw2ZwQiQ==>

Azure Maps



- SDK WEB
- API REST

- Map controls
- Android SDK
- Azure Maps Services
- Rendering service
- Route planning service
- Research service
- Time zone service
- Traffic information service
- IP to location (country)
- Batch geocoding service
- Batch routing service

Azure Container Instances



Great solution for

- isolated containers
- including simple applications
- task automation
- jobs

Advantages:

- Start containers in seconds => NO NEED to provision VMs.
- Simple Public IP connectivity and DNS name
- Exact specifications of CPU cores and memory
- Get billed by the second

NEWS: Virtual network deployment (preview)

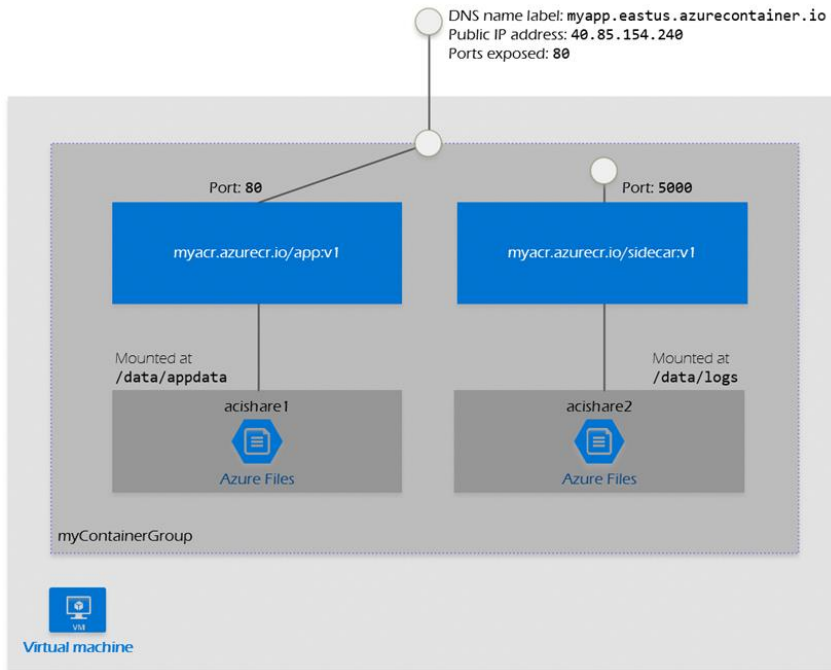
NOTES:

- Persist state with Azure Container Instances, we offer direct mounting of Azure Files shares.
- Azure Container Instances can schedule both Windows and Linux containers with the same API
- **Co-scheduled groups** that share a host machine, local network, storage, and lifecycle

Azure Container Instances



Co-scheduled groups that share a host machine, local network, storage, and lifecycle



```
az container create --resource-group myResourceGroup --file deploy-aci.yaml
```

YAML

```
apiVersion: 2018-10-01
location: eastus
name: myContainerGroup
properties:
  containers:
    - name: aci-tutorial-app
      properties:
        image: mcr.microsoft.com/azuredocs/aci-helloworld:latest
        resources:
          requests:
            cpu: 1
            memoryInGb: 1.5
        ports:
          - port: 80
          - port: 8080
    - name: aci-tutorial-sidecar
      properties:
        image: mcr.microsoft.com/azuredocs/aci-tutorial-sidecar
        resources:
          requests:
            cpu: 1
            memoryInGb: 1.5
  osType: Linux
  ipAddress:
    type: Public
    ports:
      - protocol: tcp
        port: '80'
      - protocol: tcp
        port: '8080'
  tags: null
  type: Microsoft.ContainerInstance/containerGroups
```



DEMO



USE CASE





Thanks

Questions?



<https://github.com/zama202>



@riccardozamana



<https://www.linkedin.com/in/riccardozamana/>