



# Practical Blazor Server on Azure

Giampaolo TUCCI

# Un grazie agli sponsor



E alle community che ci hanno supportato

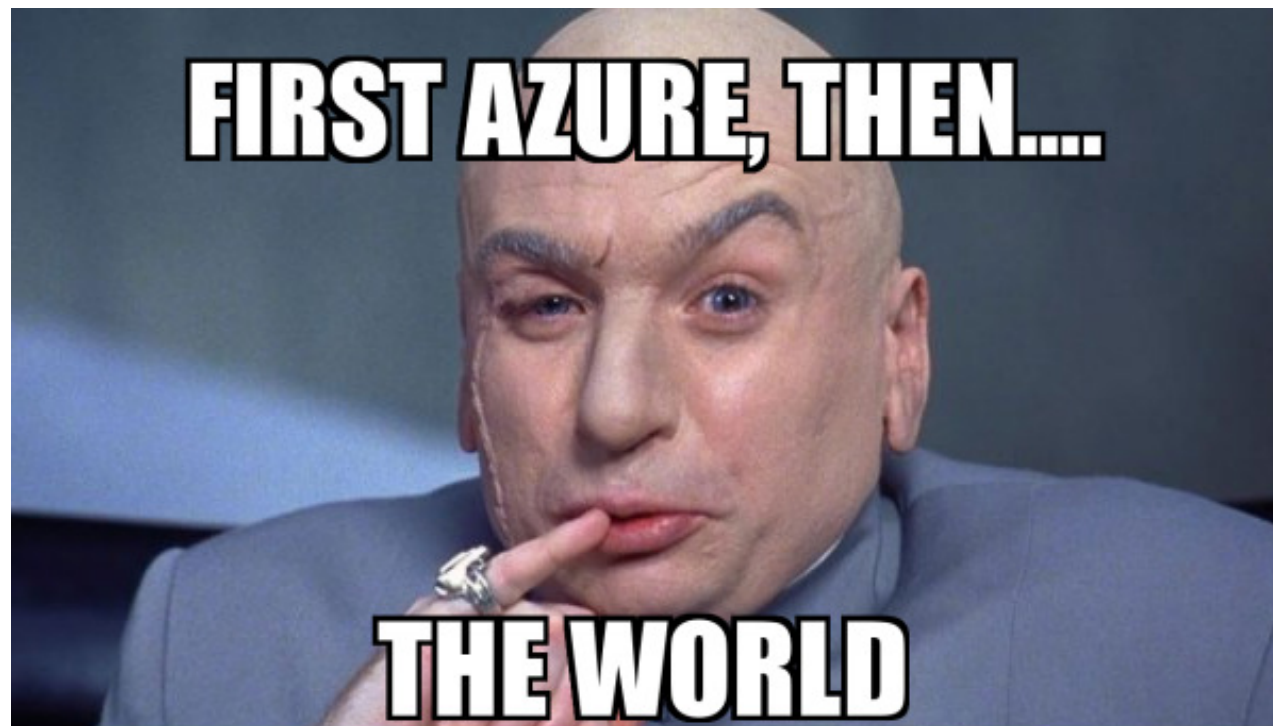


# Blazor Server vs Blazor WASM

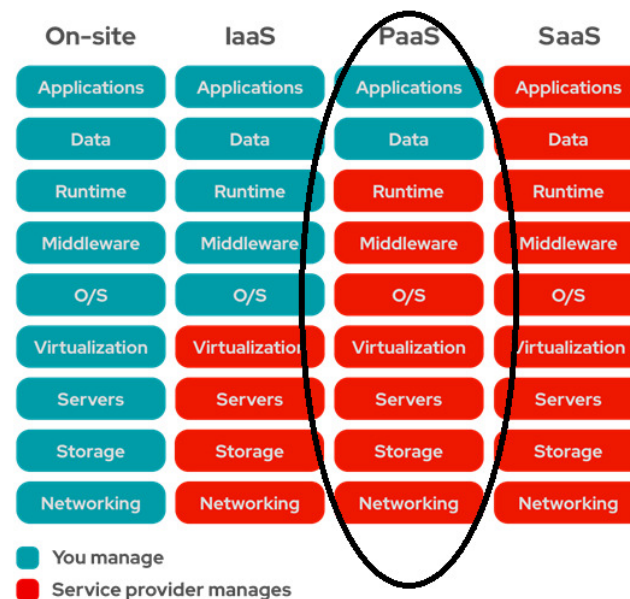
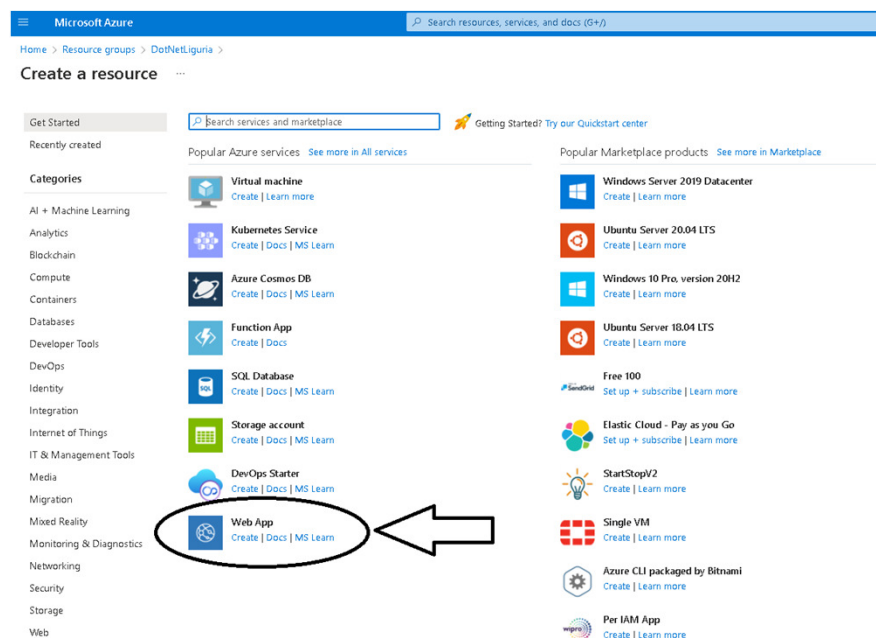


- Perchè abbiamo scelto Blazor Server ?  
Performance !
- Lo switch nel nostro caso è avvenuto “in corsa”
- Strutturando il progetto in modo efficace è possibile fare il cambio di Blazor hosting model con un modesto refactoring
- Aree oggetto dello switch hosting model.
  - Autenticazione differente
  - Accesso ai dati: controller vs accesso diretto al db

Perché proprio Azure ???



# Deploy Blazor Server on Azure – Step 0



# Deploy Blazor Server on Azure – Step 1

[Home](#) > [Create a resource](#) >

## Create Web App

managed platform to perform infrastructure maintenance. [Learn more](#)

**Project Details**  
Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ☐ Azure subscription 1

Resource Group \* ☐ DotNetLiguria  
[Create new](#)

**Instance Details**  
Need a database? [Try the new Web + Database experience.](#)

Name \* dotnetliguriaquestionario  
azurewebsites.net

Publish \* ☒ Code ☐ Docker Container ☐ Static Web App

Runtime stack \* .NET 6 (LTS)

Operating System \* ☒ Linux ☐ Windows

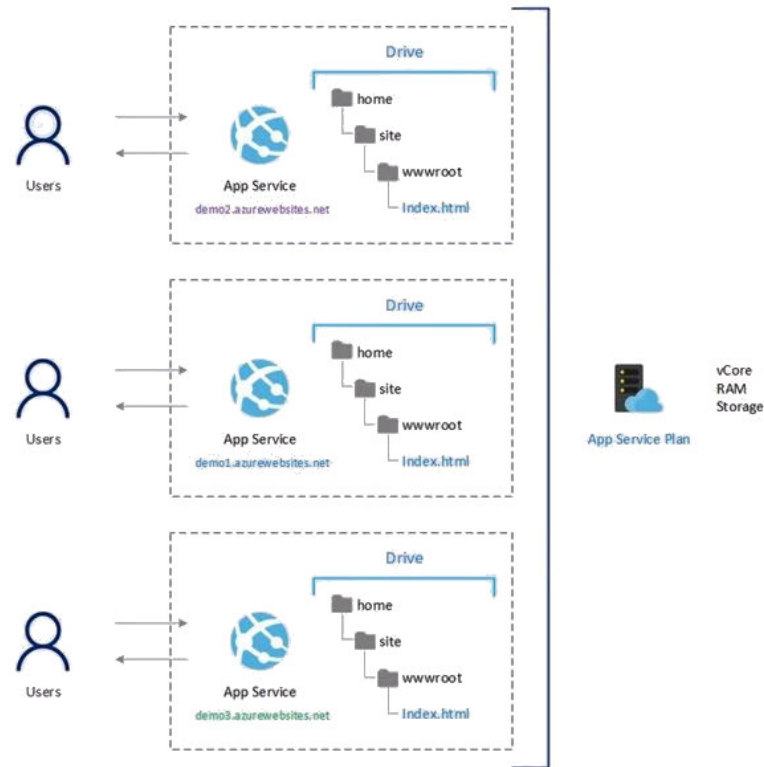
Region \* Switzerland North  
Not finding your App Service Plan? Try a different region or select your App Service Environment.

**App Service Plan**  
App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Linux Plan (Switzerland North) \* ☐ (New) appserv  
[Create new](#)

Sku and size \* **Basic B1**  
100 total ACU, 1.75 GB memory  
[Change size](#)

# App Server vs App Service Plan





# Configurazioni notevoli per Blazor Server

Microsoft Azure

Home > dotnetliguriaquestionario

dotnetliguriaquestionario | Configuration

App Service

Search (Ctrl+J)

Refresh Save Discard Leave Feedback

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems  
Security  
Events (preview)  
Deployment  
Quickstart  
Deployment slots  
Deployment Center  
Settings  
Configuration  
Authentication  
Application Insights (preview)  
Identity  
Backups  
Custom domains  
TLS/SSL settings  
TLS/SSL settings (preview)  
Networking  
Scale up (App Service plan)  
Scale out (App Service plan)  
WebJobs  
Push  
MySQL In App  
Service Connector (Preview)  
Properties

Application settings General settings Path mappings

**Stack settings**

Stack: .NET  
Major version: .NET 6  
Minor version: .NET 6 (LTS)  
Startup Command: dotnet BlazorQuestionarioServer.dll  
Provide an optional startup command that will be run as part of container startup. [Learn more](#)

**Platform settings**

FTP state: All allowed  
FTP based deployment can be disabled or configured to accept FTP (plain text) or FTPS (secure) connections. [Learn more](#)

HTTP version: 1.1

**Web sockets**

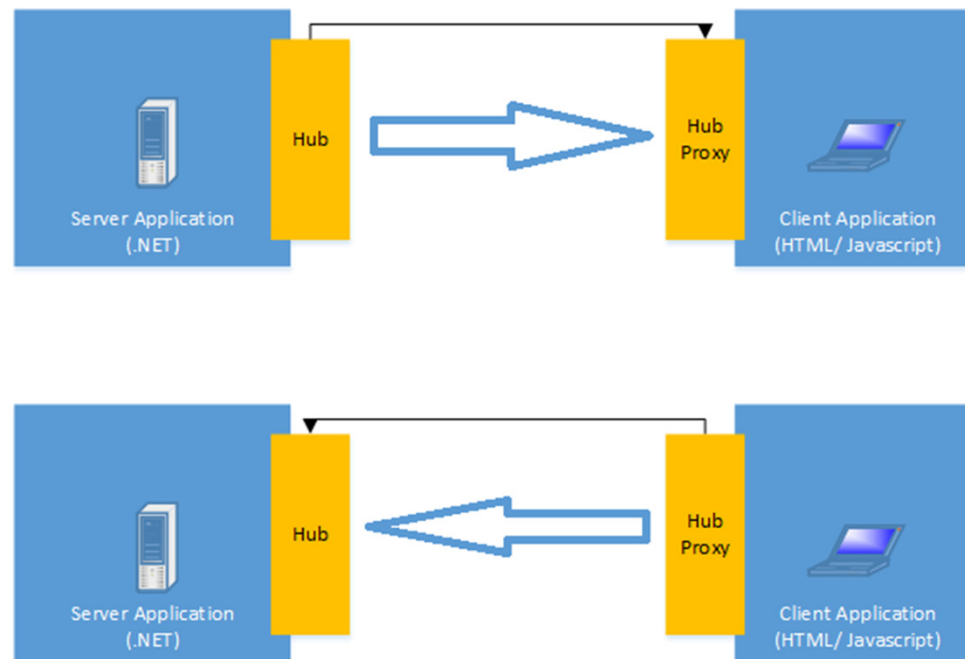
Always on: ☒ On ☐ Off  
Prevents your app from being idled out due to inactivity. [Learn more](#)

ARR affinity: ☐ On ☒ Off  
Improve performance of your stateless app by turning Affinity Cookie off, stateful apps should keep this setting on for compatibility. [Learn more](#)

**Debugging**

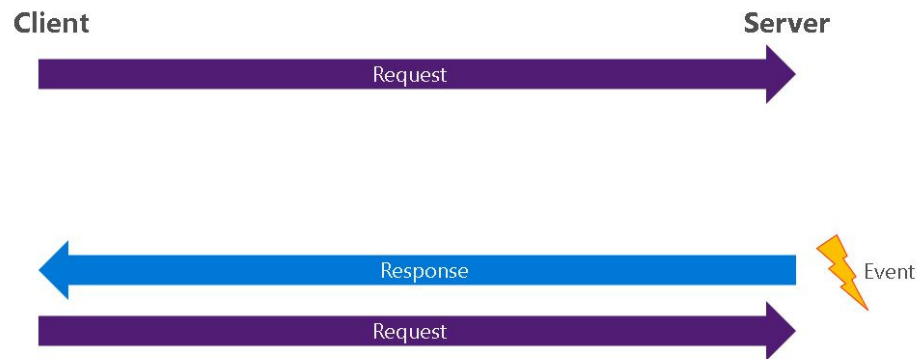
Remote debugging: ☐ On ☒ Off  
Your selected runtime stack does not currently support remote debugging.

# Blazor Server & SignalR

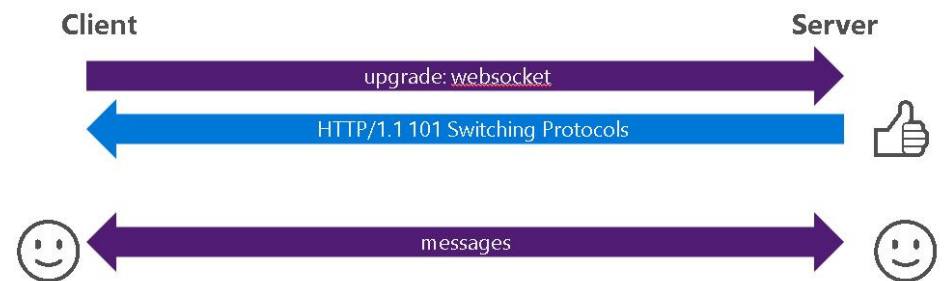


# SignalR: Long Polling vs Web Sockets

## Long polling



## WebSockets

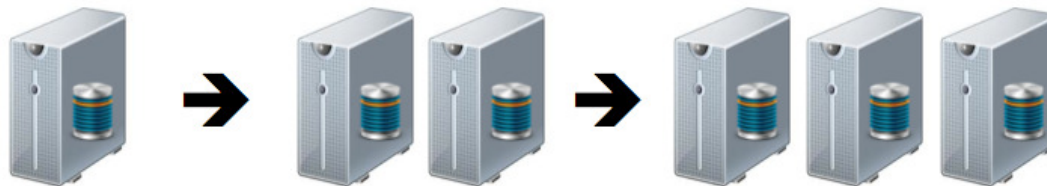


# Scale-Up & Scale-Out: All about dimensions...

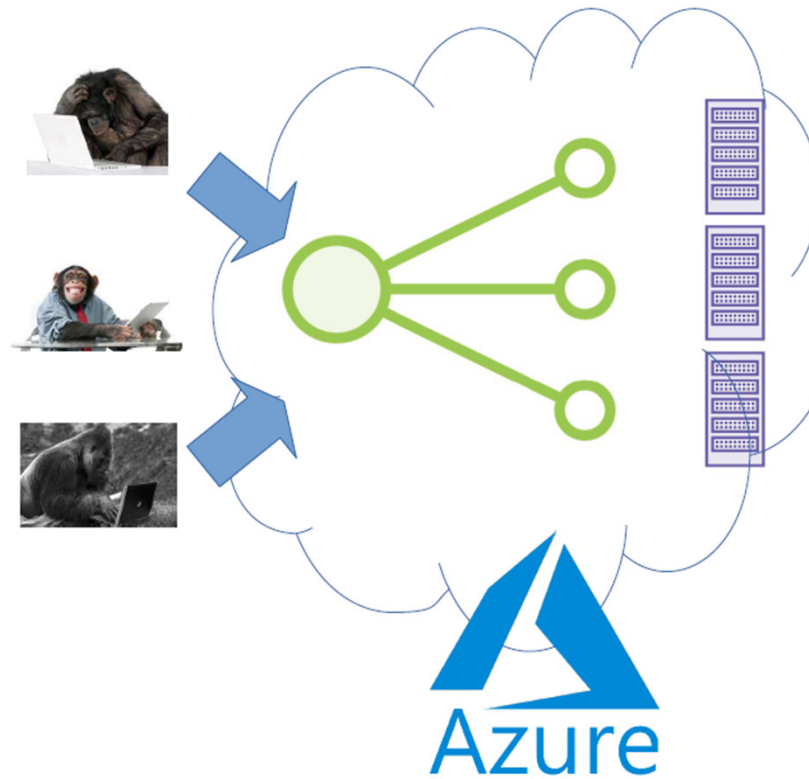
Scale-Up



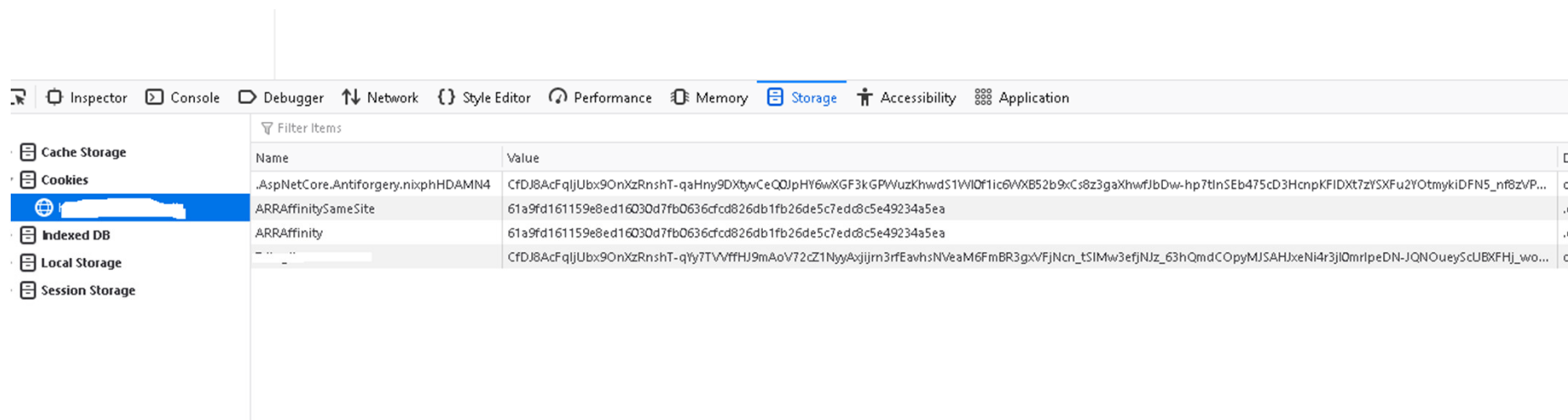
Scale-Out



# Blazor Server e lo Scale-Out



# Blazor Server e lo Scale-Out

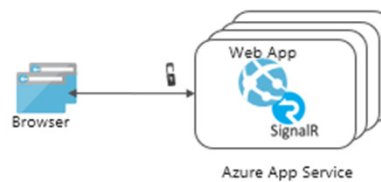


# Application State/UI State

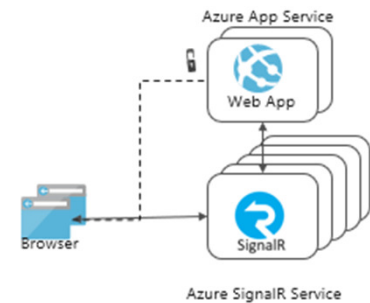
- Blazor Server è un framework stateful: UI State e Application State
- Lo stato è mantenuto lato server
  - UI State
    - In generale: stato dei controlli visualizzati (Es: controlli abilitati/disabilitati)
    - In Blazor: DOM della pagina visualizzata
  - Application State
    - Contenuto del Dependency Injection
    - Stato di ogni oggetto (Es.: valore assunto dalle proprietà)
- Preservare lo stato tra circuiti
  - Database Server-Side
  - URL
  - Browser storage (localStorage/sessionStorage)
- UI State
  - E' il vero punto debole di Blazor Server !!!
  - Ad oggi non esiste alcun «metodo semplice per preservarla».
  - In .Net 7 dovrebbero essere inclusi notevoli miglioramenti su questo aspetto.



# Azure SignalR: yes or not ?



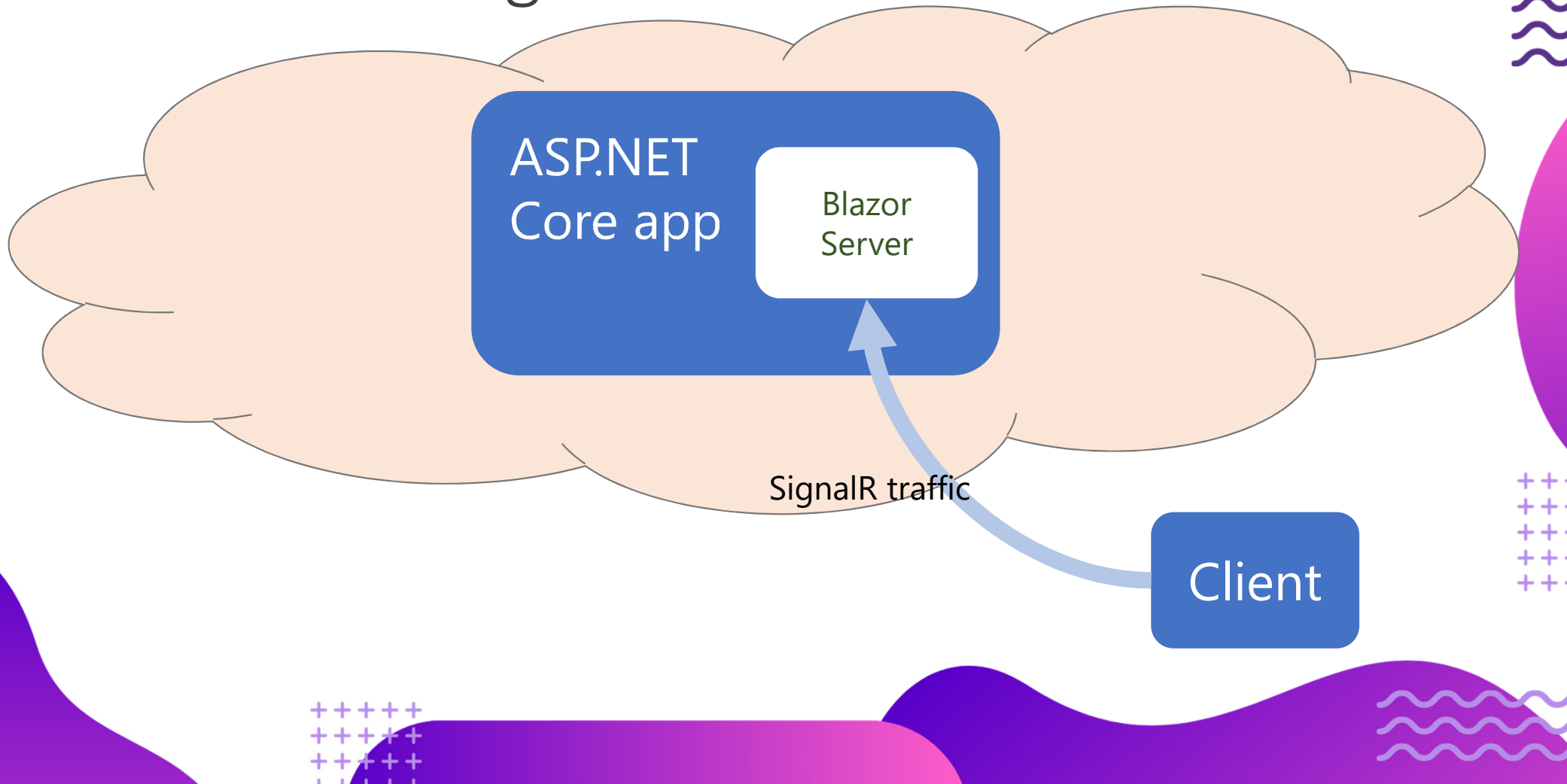
Self-hosted SignalR



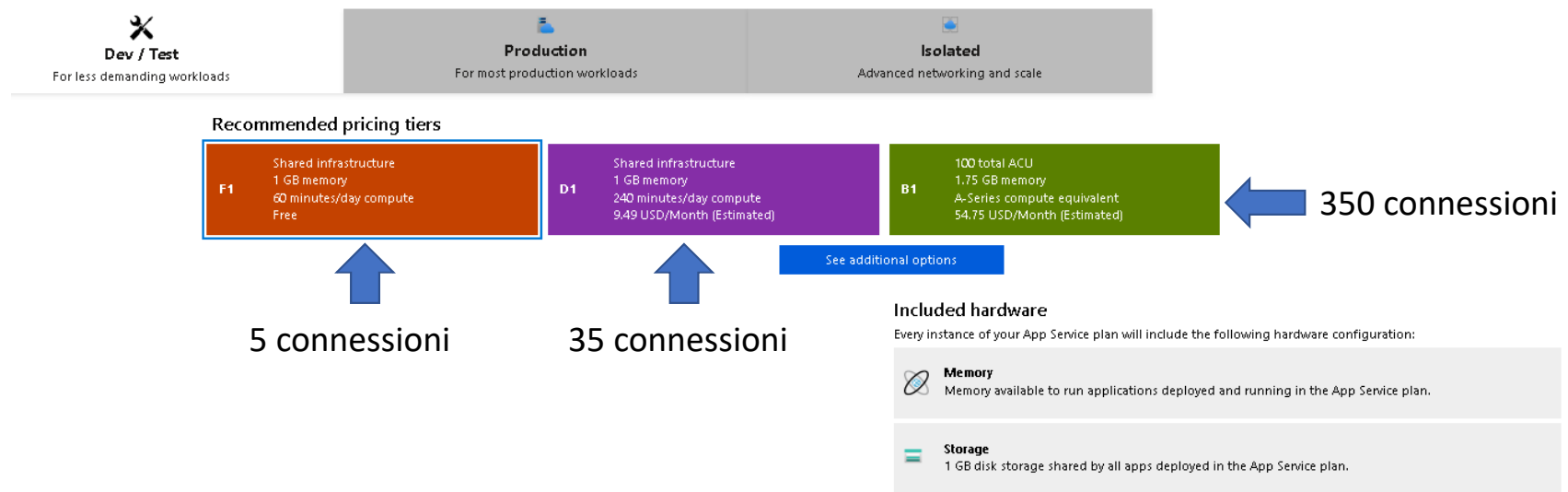
Managed SignalR Service

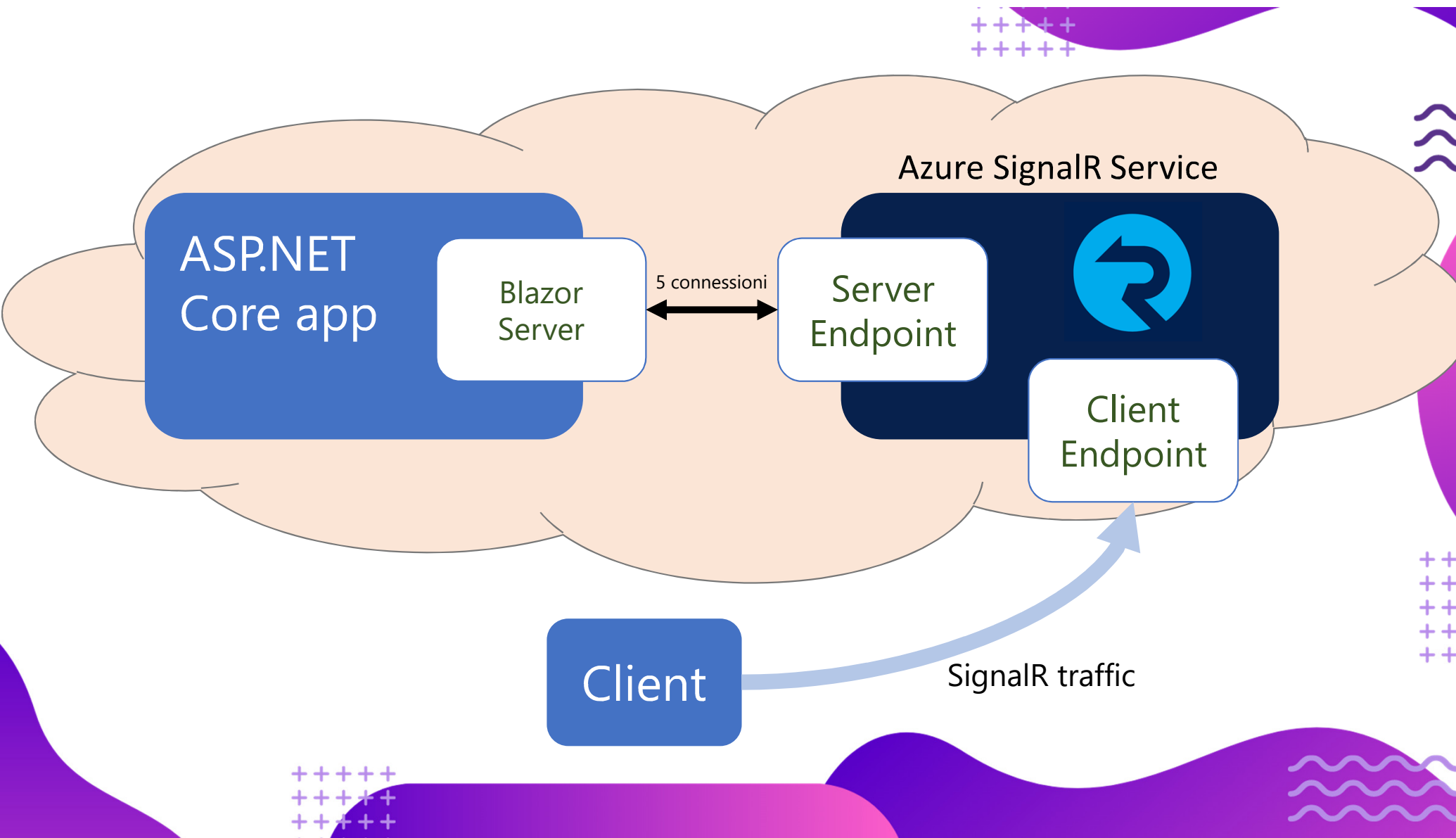


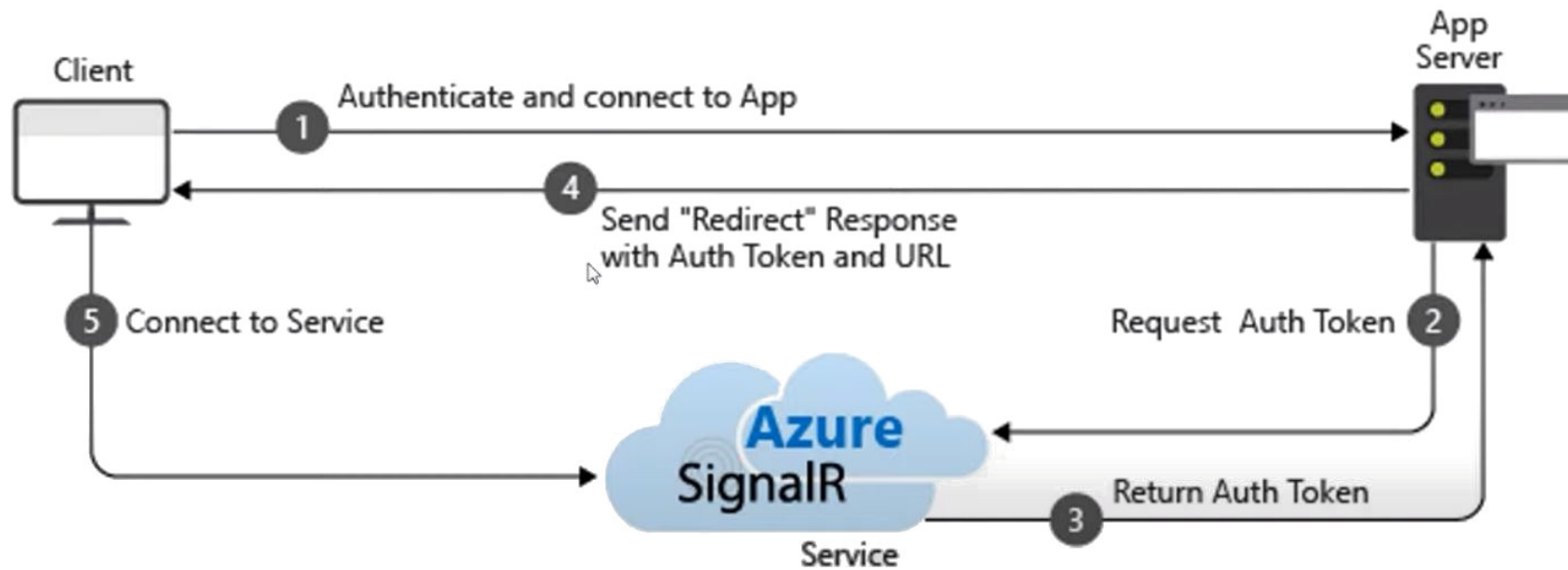
# Senza Azure SignalR



# Connessione diretta: limiti teorici







# Azure SignalR: yes or not ?

Offload ASP.NET Core SignalR verso un servizio esterno

















Free tier – Sino a 20 connessioni (in realtà 15)

Per l'utilizzo basta aggiungere un pacchetto NuGet e aggiungere la stringa di connessione: il tutto con 0 linee di codice

[Home](#) > [Microsoft.SignalRGalleryPackage](#) > [testsins](#) >

## Choose your pricing tier

Browse the available plans and their features

Free (Dev/Test Only)	Standard	Premium
 Connections Up to 20 connections	 Connections 1,000 connections/unit	 Connections 1,000 connections/unit
 Included Messages 20,000/Unit/Day	 Included Messages 1,000,000/Unit/Day	 Included Messages 1,000,000/Unit/Day
 Additional Messages Unsupported	 Additional Messages Unlimited	 Additional Messages Unlimited
 SSL	 SLA 99.9% of the time	 SLA 99.95%
	 SSL	 SSL With custom domains
		 Autoscale
		 Availability Zone In supported regions
0.00 FREE	41.41 EUR/MONTH/UNIT (ESTIMATED)	51.44 EUR/MONTH/UNIT (ESTIMATED)

Azure SignalR: è possibile farne senza ! (?)



# Connessione diretta: opzioni possibili

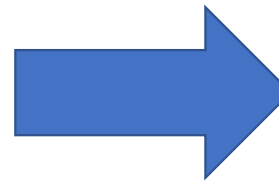
## Configurazioni notevoli

- Numero di circuiti disconnessi mantenuti in memoria  
(*DisconnectedCircuitRetentionPeriod*)
- Timeout per circuiti disconnessi  
(*DisconnectedCircuitMaxRetained*)

## Due approcci

- Preservare RAM
- Preservare UI State

# Telemetria per Blazor Server



Application Insights






# Azure Application Insight: modalità di abilitazione



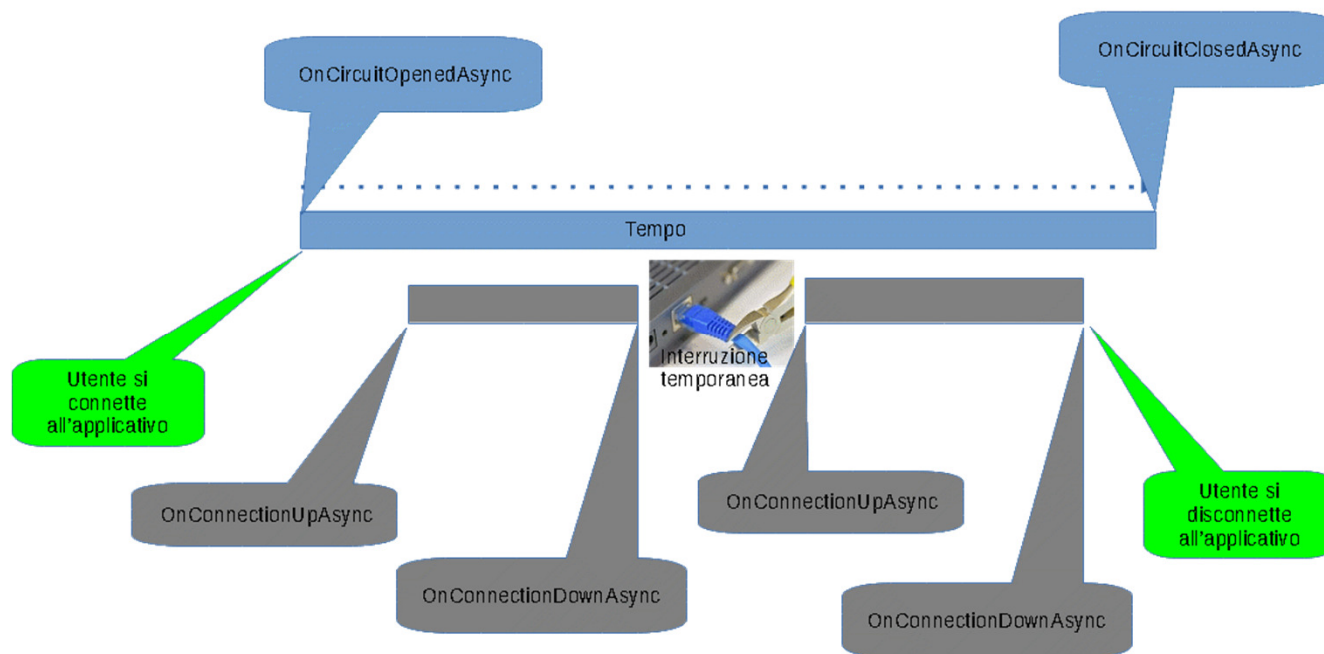
- **Auto-Instrumentation**

- Non occorre fare alcuna modifica al codice: basta abilitarla nella configurazione in Azure
- Esegue monitoring standard delle risorse utilizzate

- **Manually instrumenting the application through code**

- Occorre mettere mano al codice
  - Oltre al monitoring standard è possibile aggiungere altri valori da monitorare (pagine visitate, circuiti aperti,latenza)
- 

# Conteggio numero circuiti



Grazie !!! ....e ....W Blazor !!!!!



**GitHub DotNetLiguria:** <https://github.com/DotNetLiguria/dotnetliguriawebsite>

**GitHub Personale:** <https://github.com/gptucci>

**Youtube:** <https://www.youtube.com/dotnetliguria>

**Linkedin:** <https://www.linkedin.com/in/giampaolo-tucci-6a733b2a>