Pratical Blazor Server on Azure

Giampaolo TUCCI

Un grazie agli sponsor

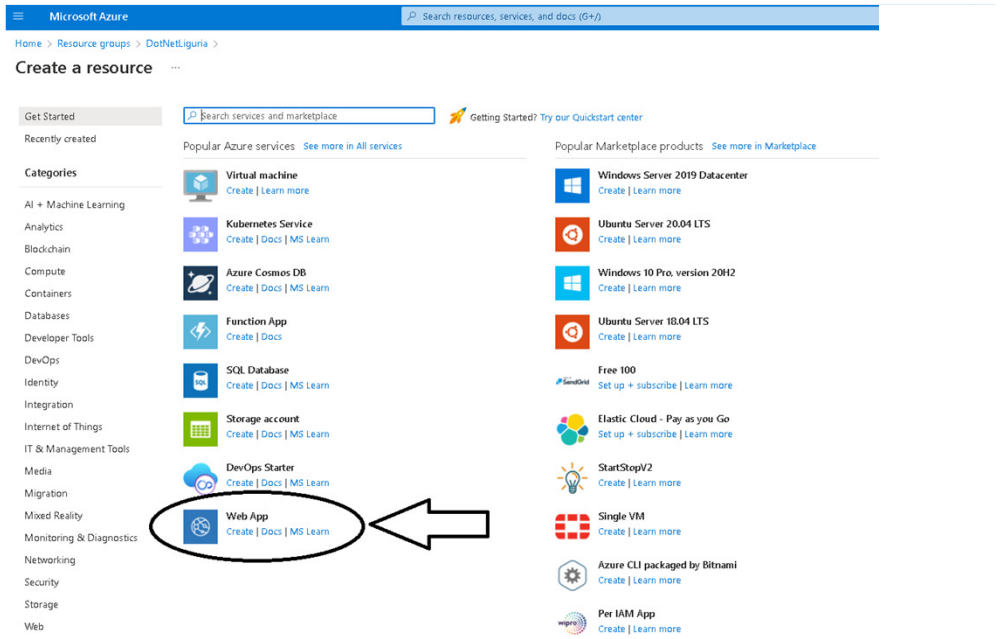
E alle community che ci hanno supportato

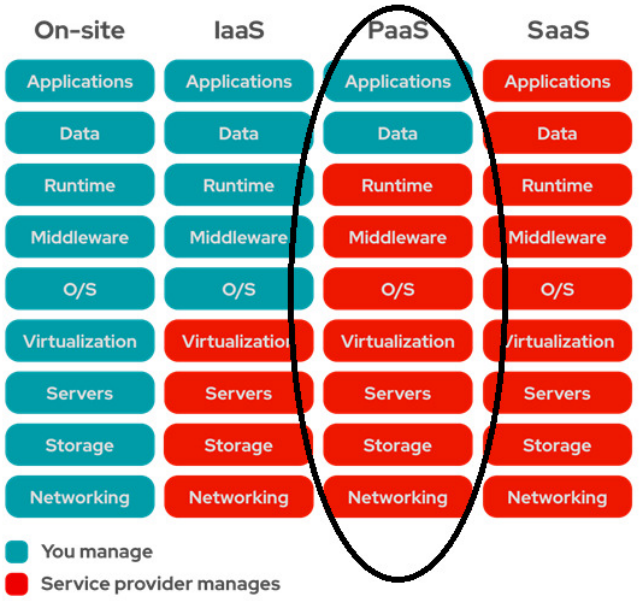
Blazor Server vs Blazor WASM

- Perchè abbiamo scelto Blazor Server ? Perfomance !
- 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

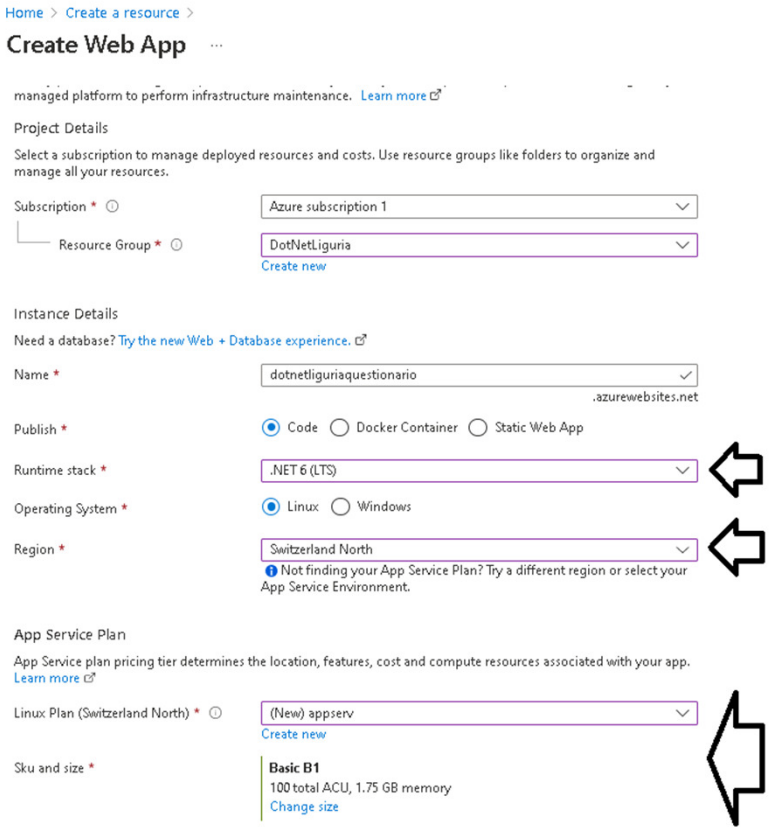
Perché proprio Azure ???

Deploy Blazor Server on Azure – Step 0

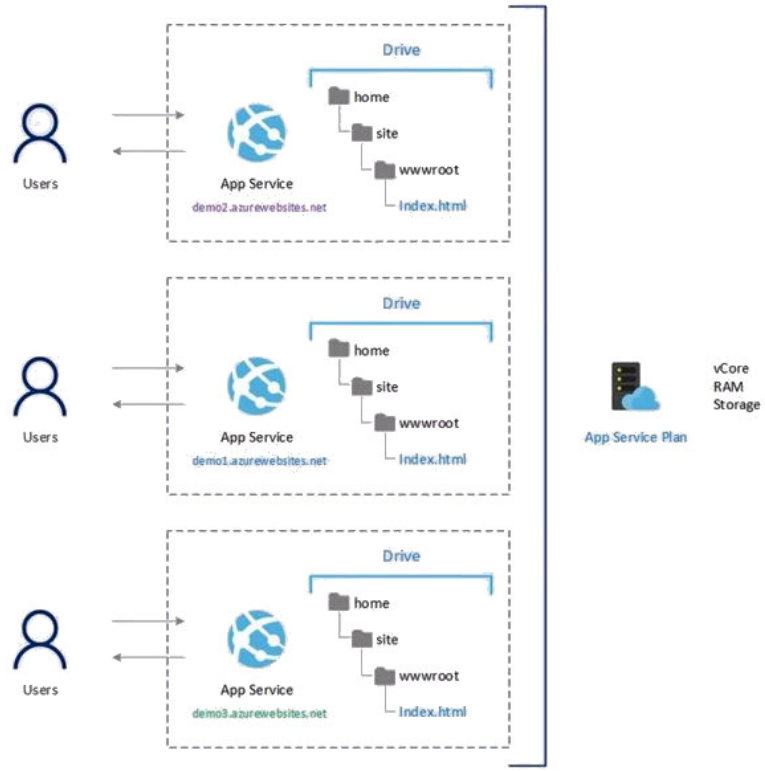




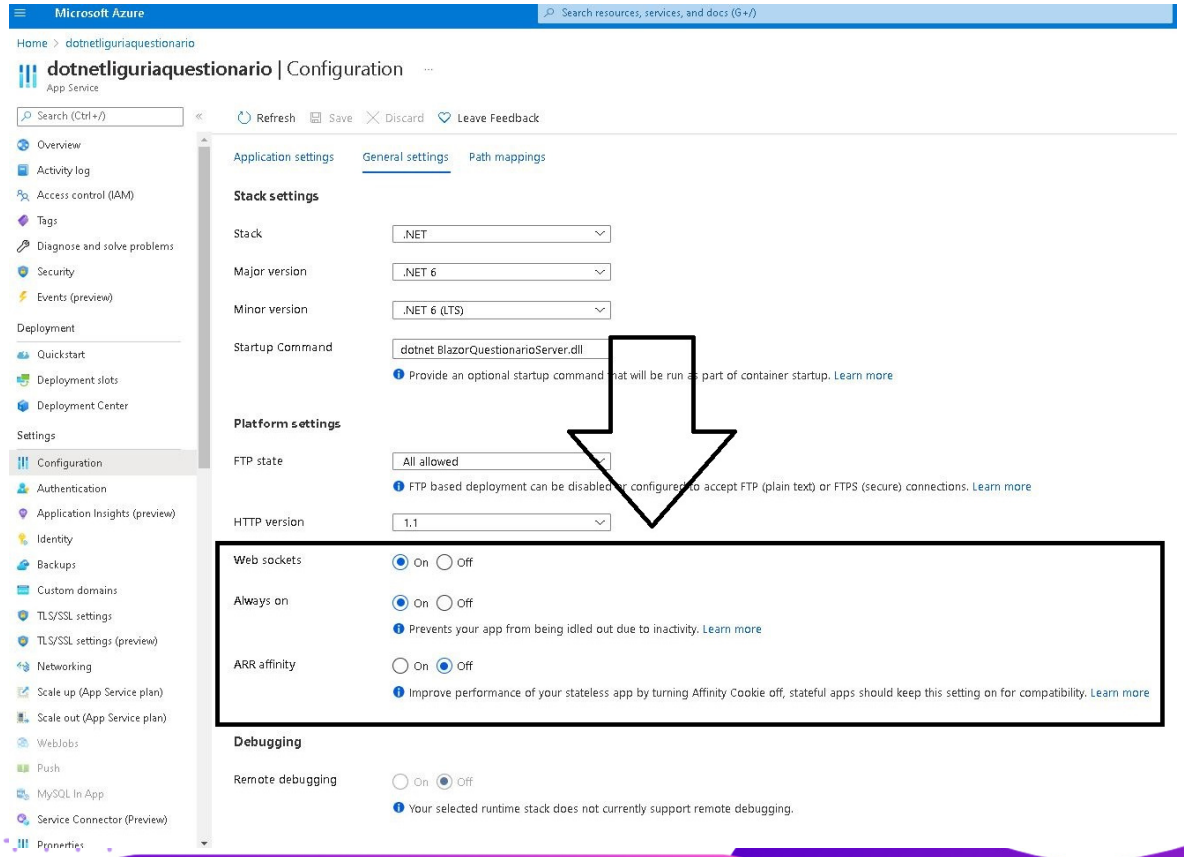
Deploy Blazor Server on Azure – Step 1



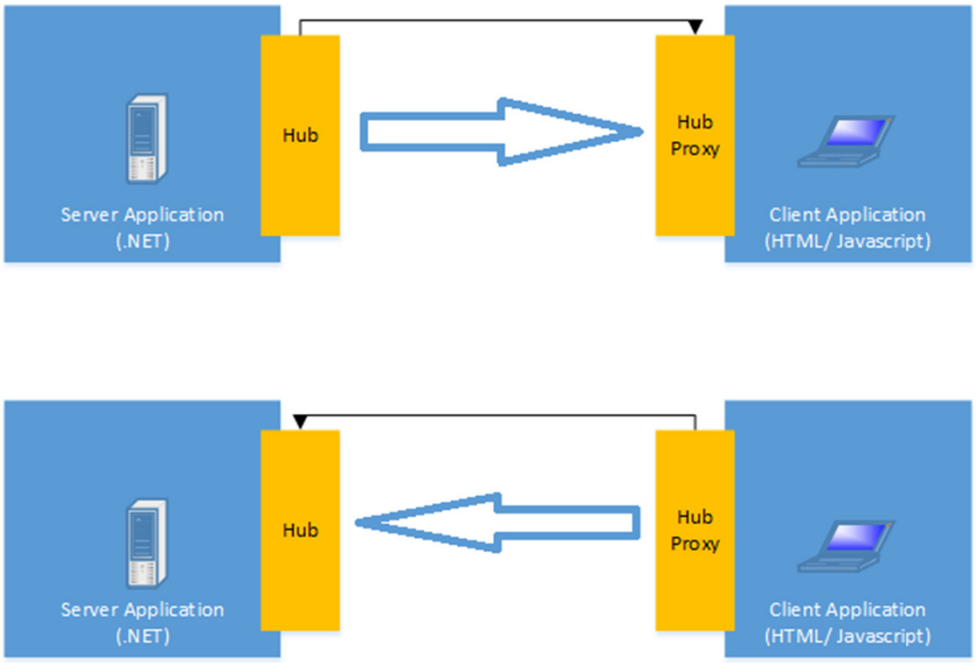
App Server vs App Service Plan



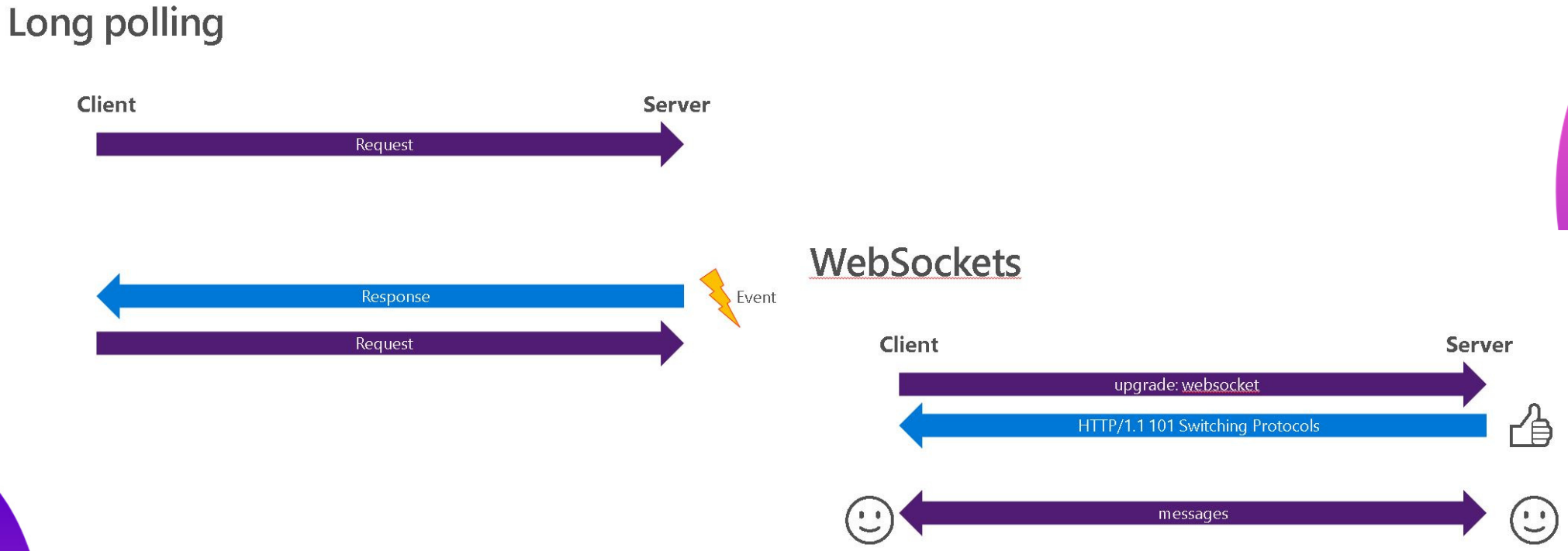
Configurazioni notevoli per Blazor Server



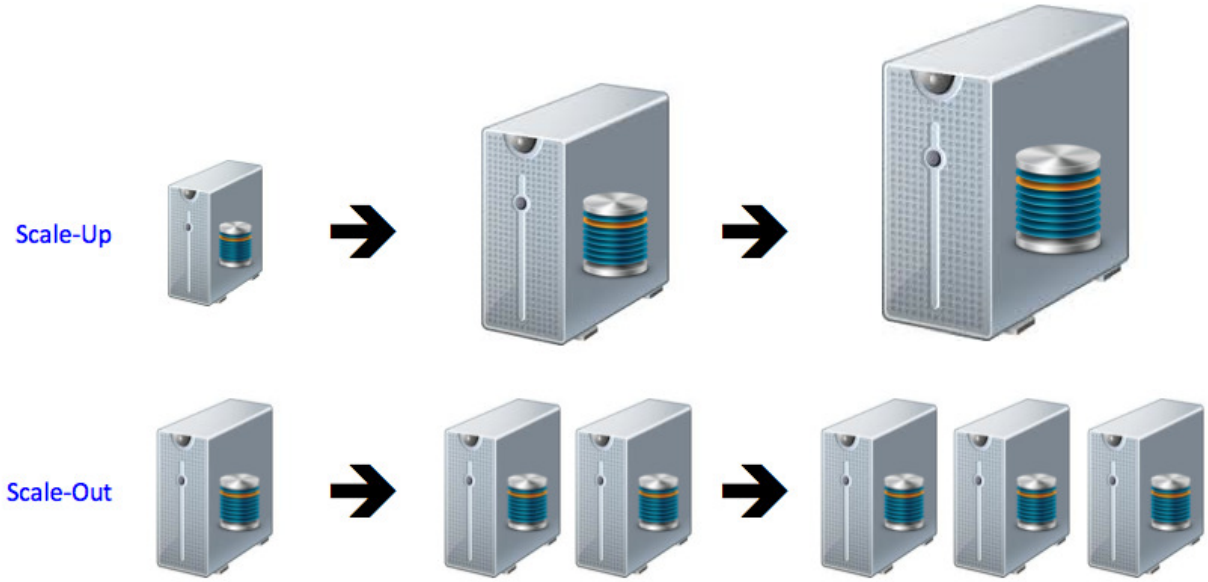
Blazor Server & SignalR



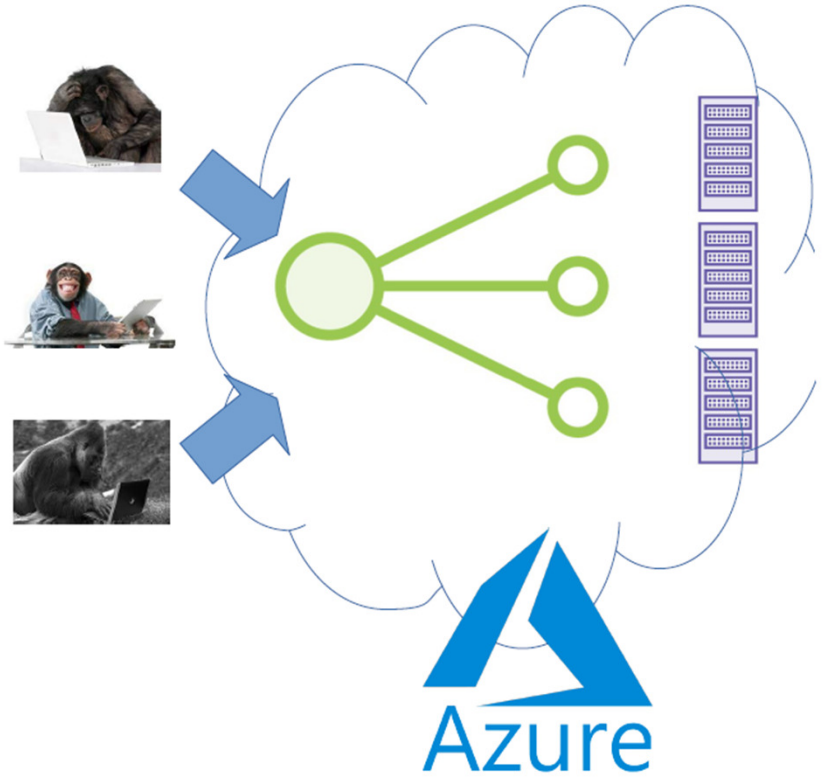
SignalR: Long Polling vs Web Sockets



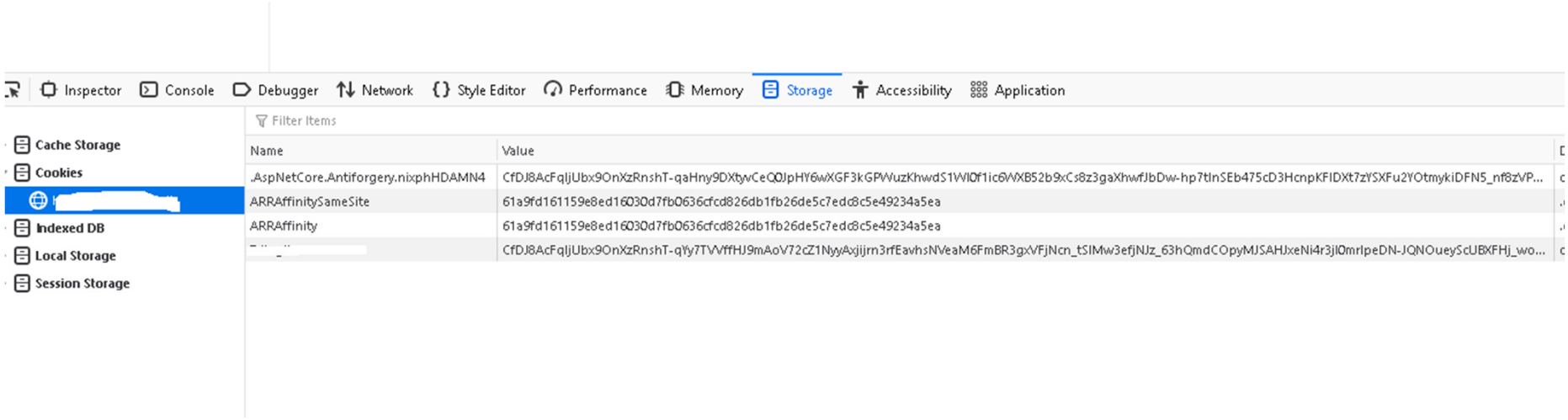
Scale-Up & Scale-Out: All about dimensions...



Blazor Server e lo Scale-Out



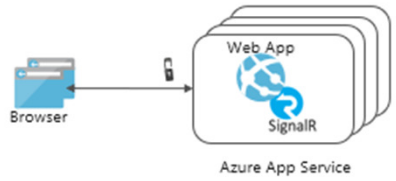
Blazor Server e lo Scale-Out



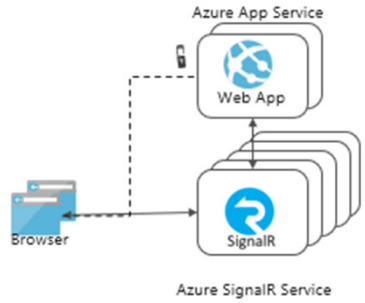
Application State/UI State

- 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 Dependecy 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 ?

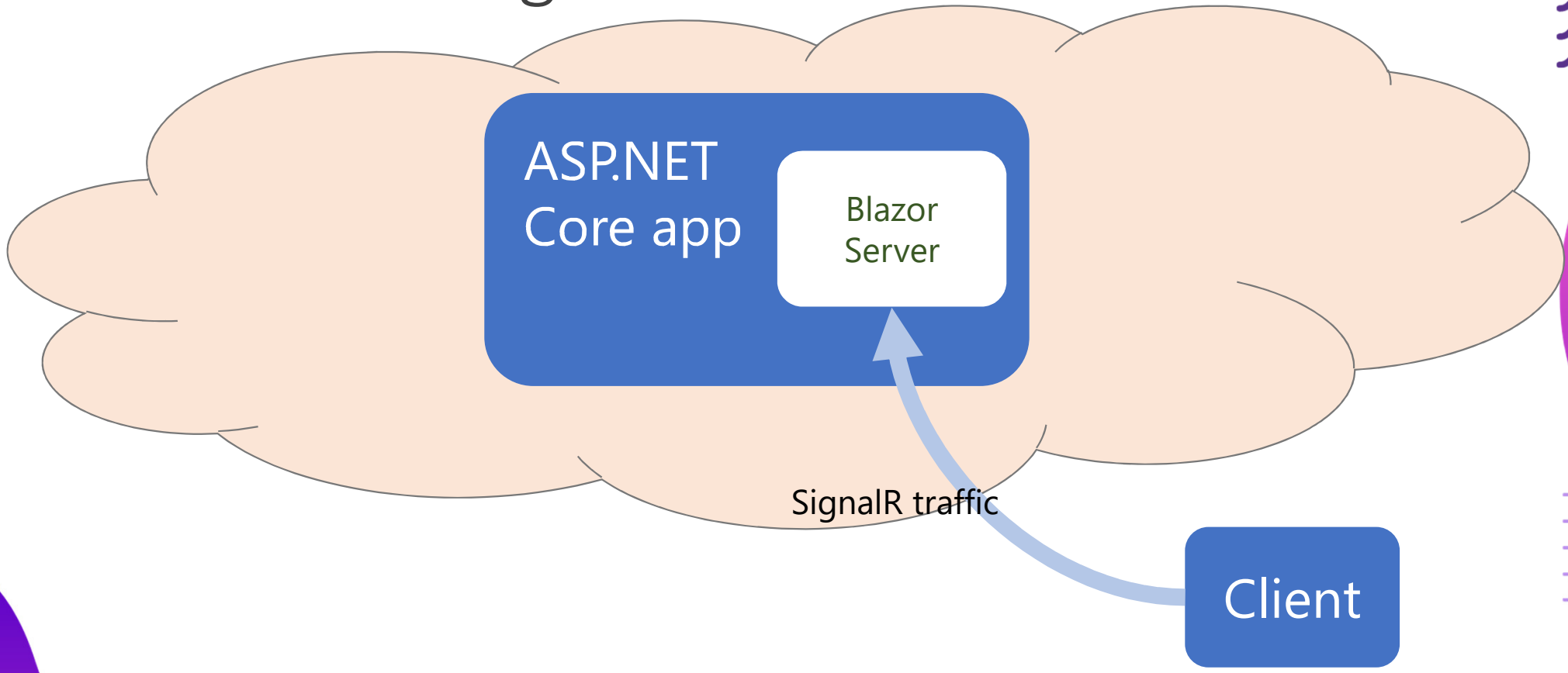


Self-hosted SignalR

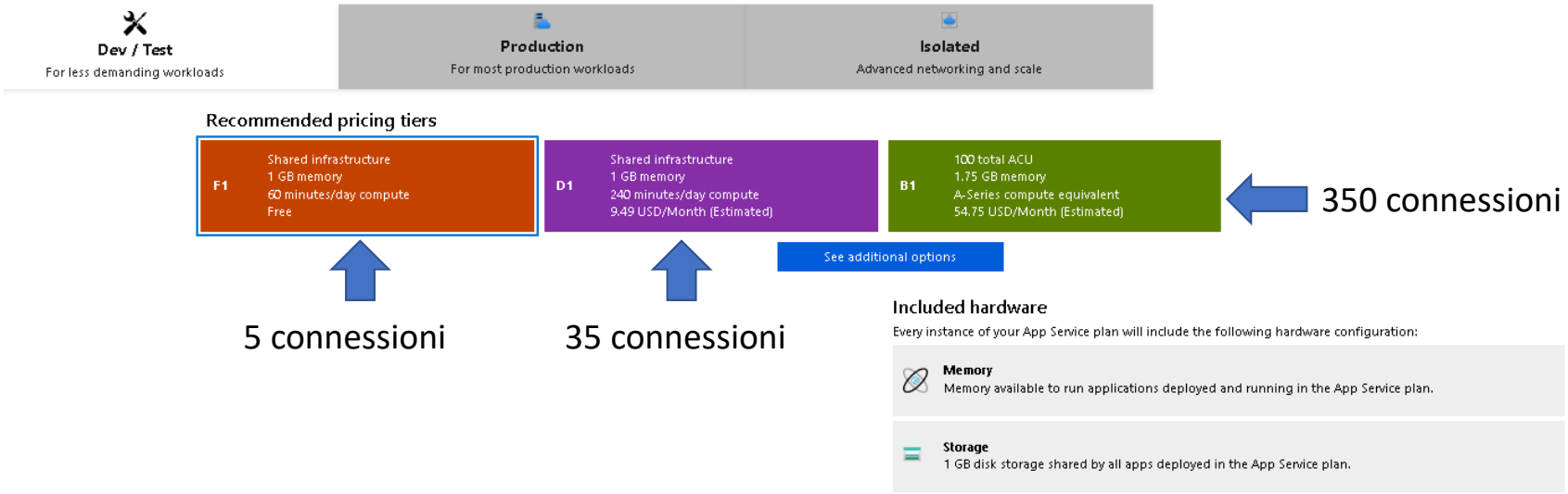


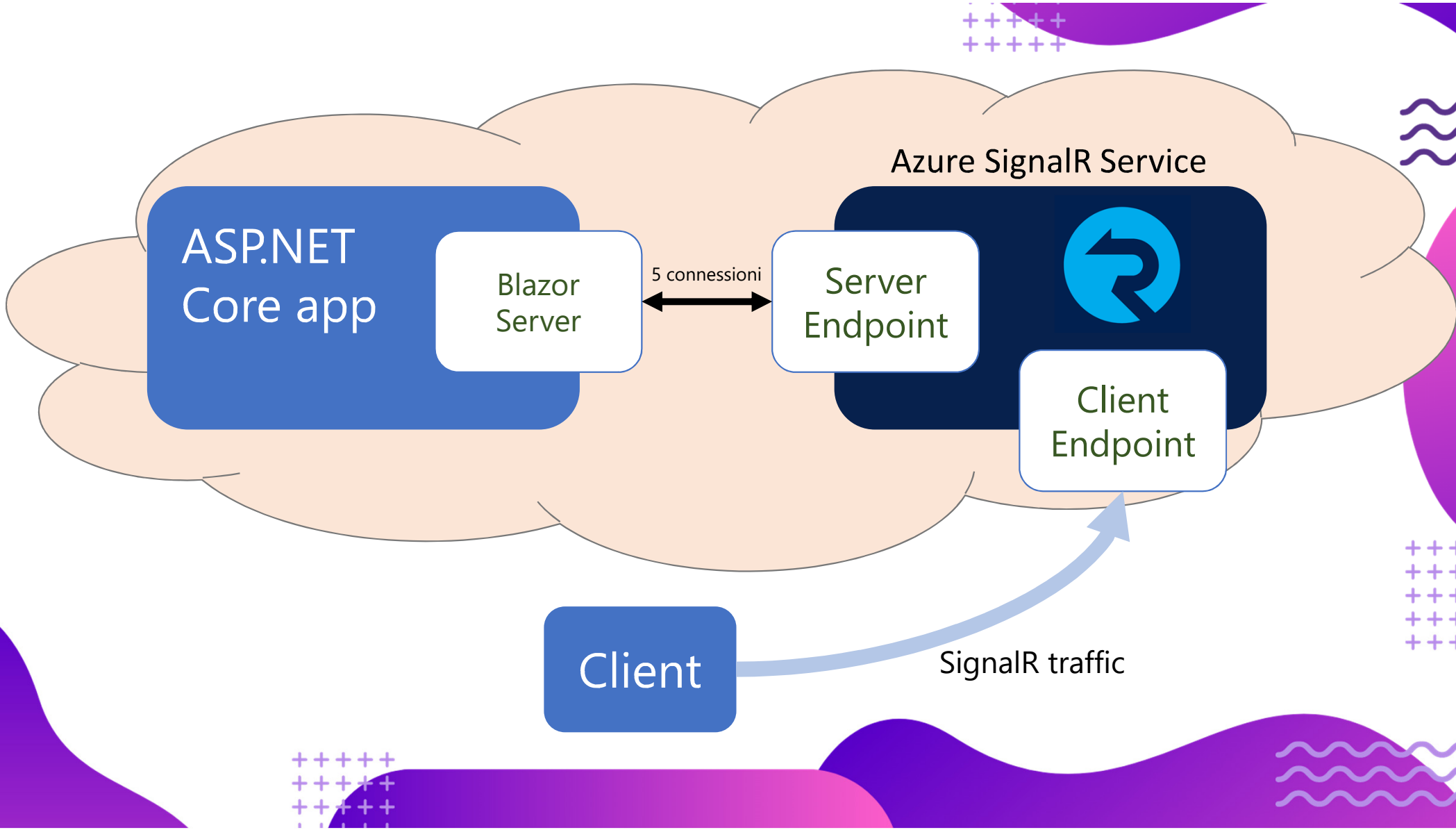
Managed SignalR Service

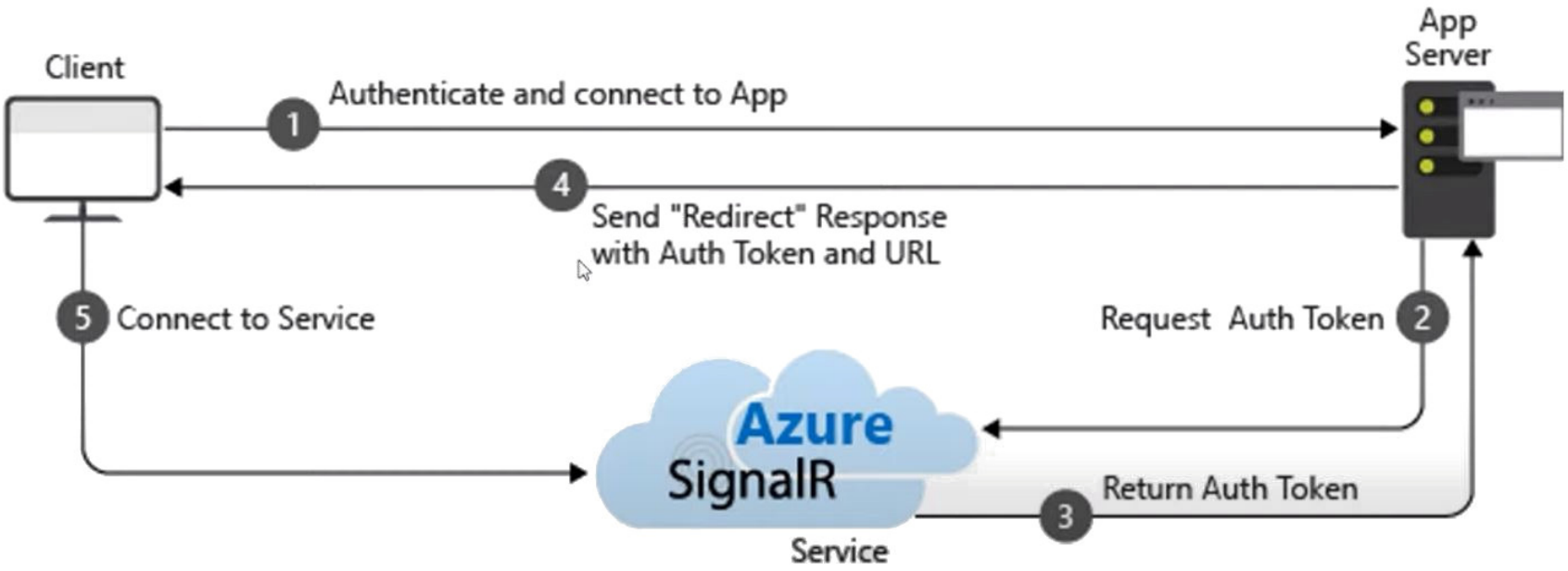
Senza Azure SignalR



Connessione diretta: limiti teorici




Azure SignalR Service
ASP.NET
Core app
Blazor
Server
5 connessioni
Server
Endpoint
Client
Endpoint
Client
SignalR traffic

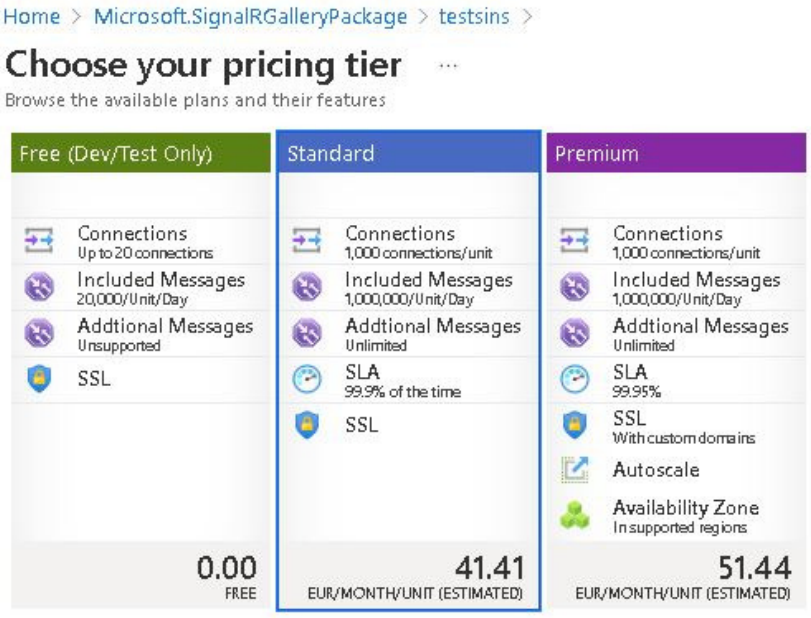

Client
App Server
1
Authenticate and connect to App
4
Send "Redirect" Response with Auth Token and URL
5
Connect to Service
Request Auth Token
2
Azure SignalR Service
3
Return Auth Token

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



Azure SignalR: è possibile farne senza ! (?)

Connessione diretta: opzioni possibili

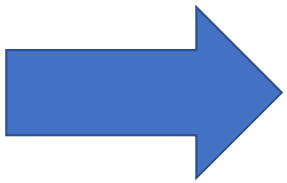
Configurazioni notevoli

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

Due approcci

- Preservare RAM
- Preservare UI State

Telemetria per Blazor Server



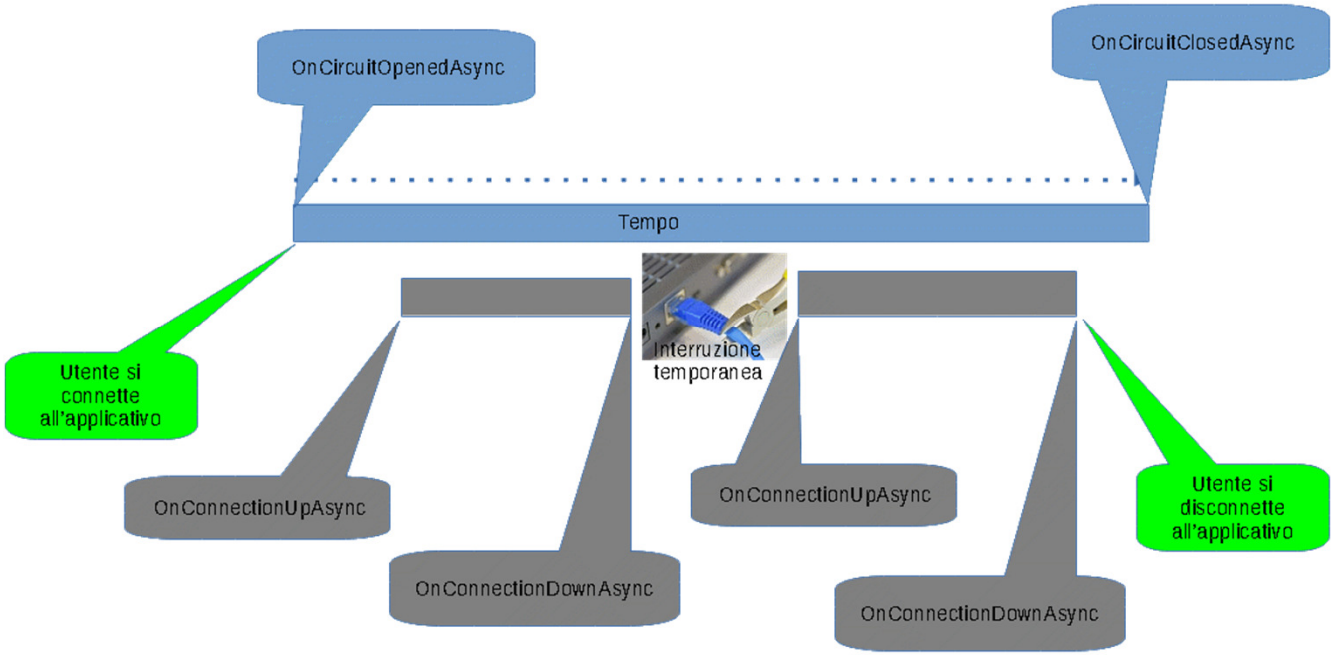
Application Insights

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 possible 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