Real-time application con Blazor e Azure

Andrea Dottor

Sponsors


altitudo


beanTech
IT moves your business


OVERNET.
upgrade your digital skills


[stesi]
Powered by Innovation


ADAWEN
Add Value Group


AZUREBOX

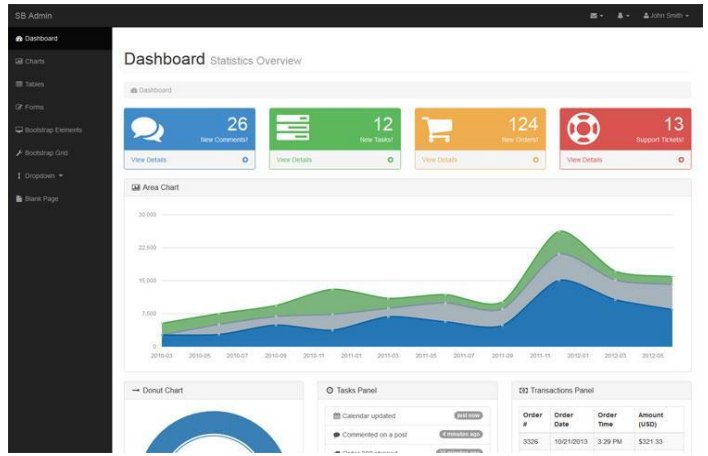

BETA 80 GROUP

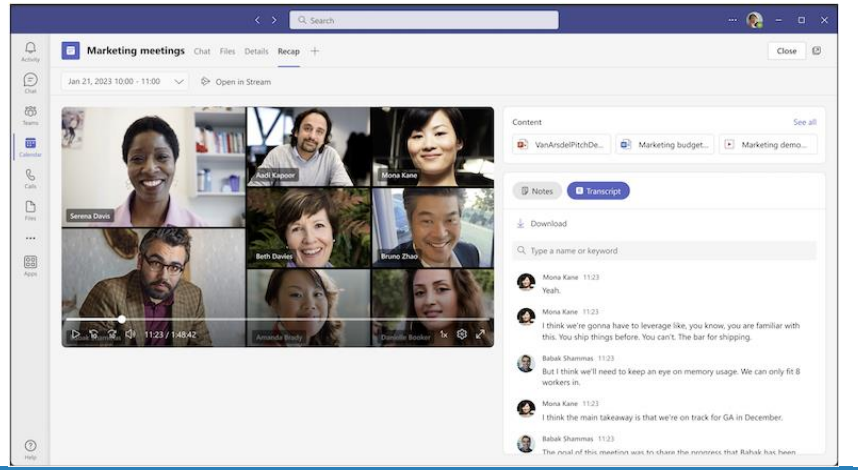

elettrone
intelligenza energetica

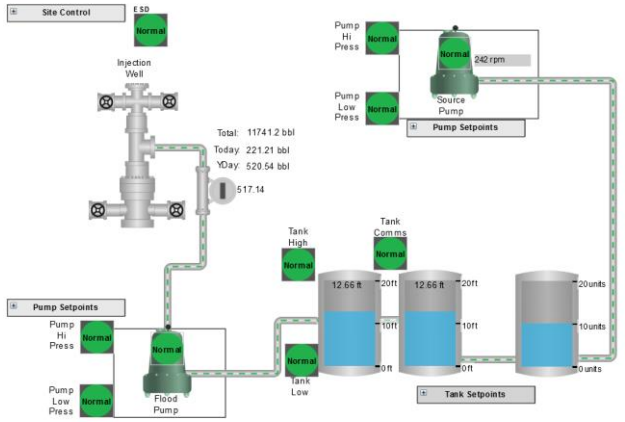

fairmat


JMATICA

Real-time application







polling vs pushing

Polling (Pull)

👎 Consumo maggiore di dati/traffico di rete

✔️ Di facile implementazione

HTTP-based APIs,
RESTful services, …

Pushing

👎 Aumento della complessità

✔️ Il client viene aggiornato solo se (e quando) è necessario

WebSocket,
Server Sent Events, …

Blazor Web App

InteractiveServer

InteractiveWebAssembly

InteractiveAuto

Possiamo far funzionare componenti sia come Blazor Server che Blazor Web Assembly, e tutto nella stessa applicazione

- @rendermode

Dependency Injection & Events

- Un servizio dichiarato come *scoped* in Blazor (interactive), equivale ad un *singleton* ma limitato però al singolo utente.
- Con InteractiveServer, un servizio dichiarato come *singleton* permette di condividere informazioni (ed eventi) per l'intera applicazione, con tutti gli utenti.

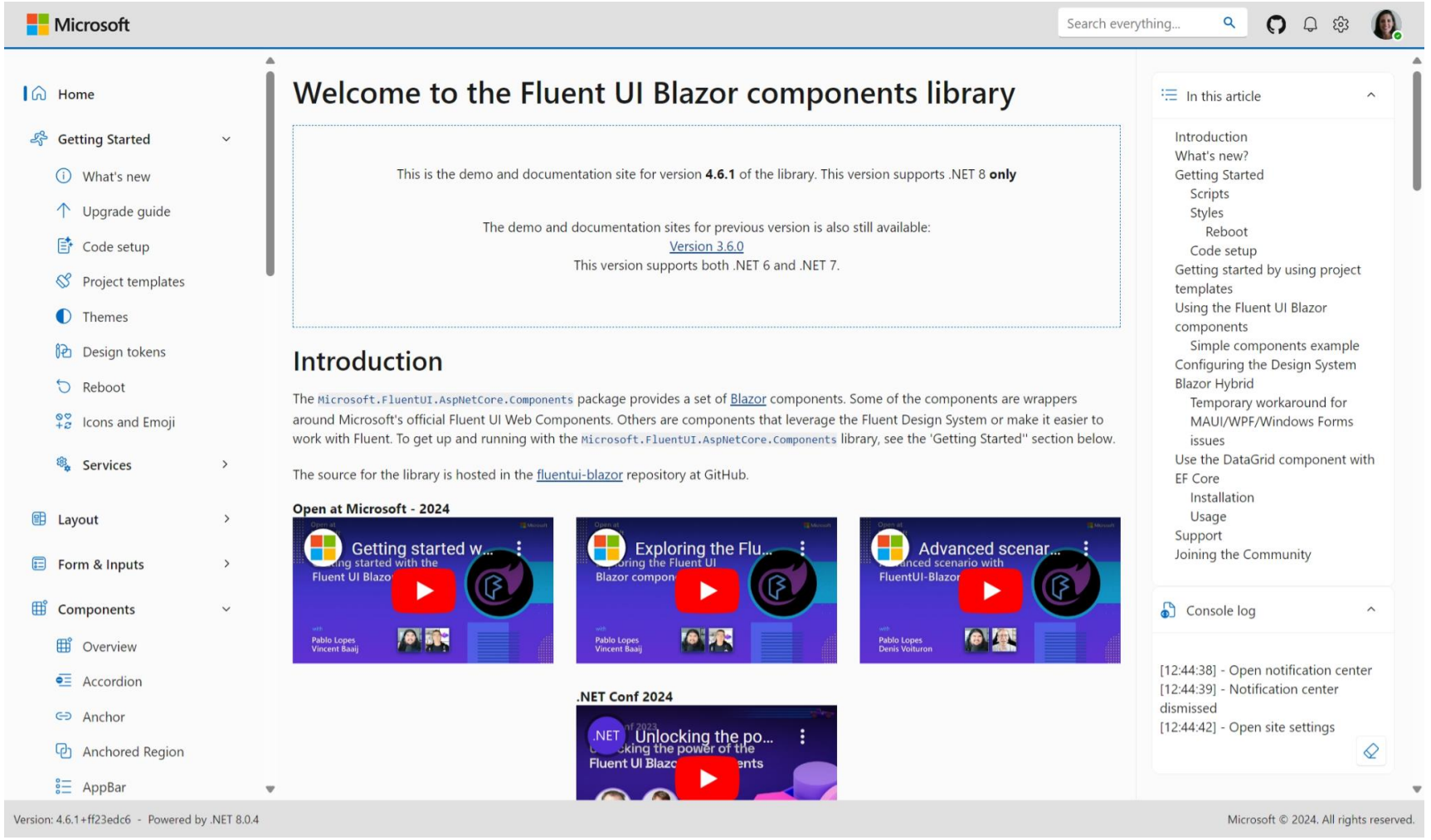
- Gli *eventi* permettono di poter far notificare ad un componente quando abbiamo un aggiornamento dei dati.

BackgroundService, IHostedService

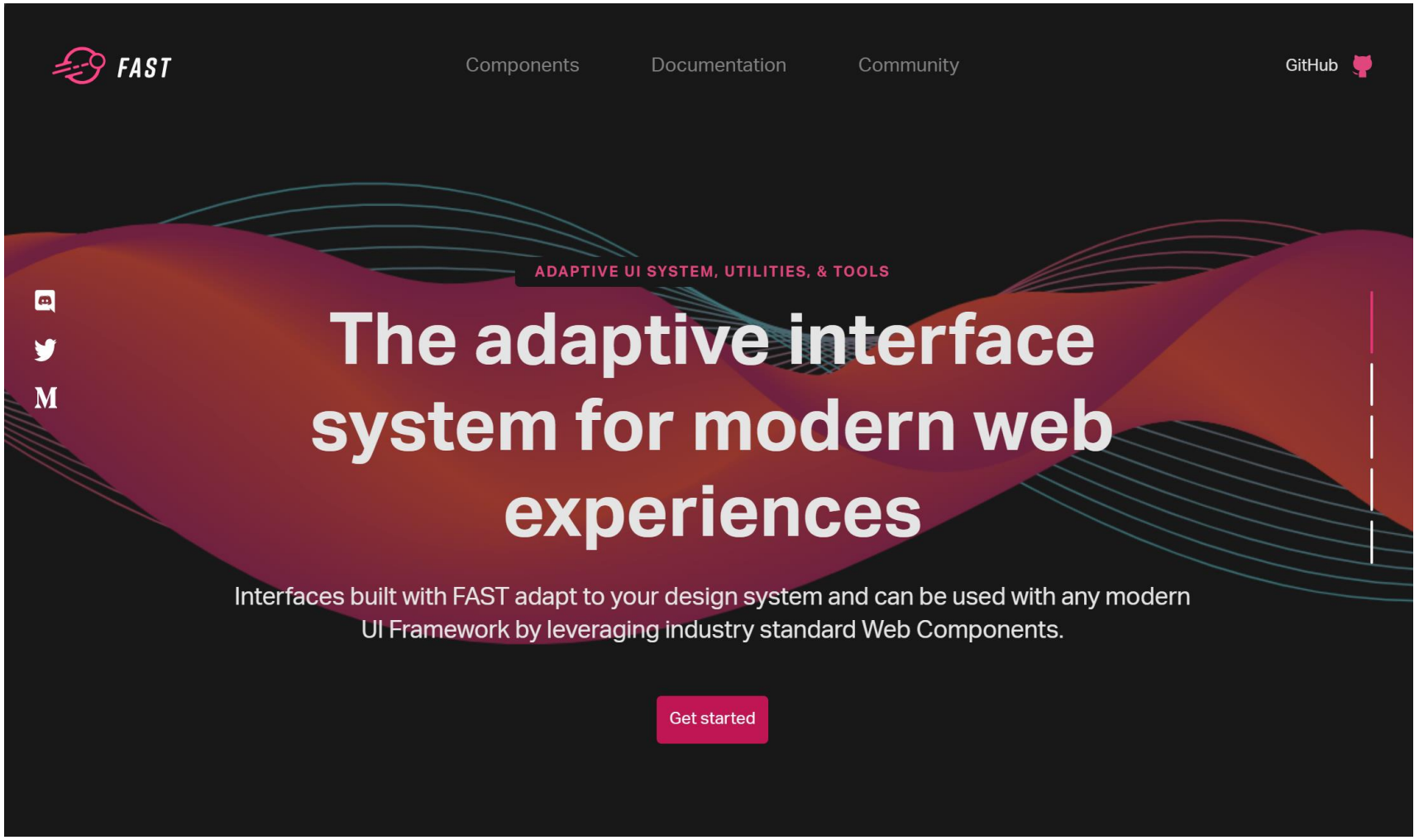
All'interno dell'applicazione (server) possiamo eseguire del codice in background, utile per sottoscriversi alla ricezione di messaggi da servizi esterni (es. ServiceBus, RabbitMQ, …)

Gli *Hosted-Service* permettono di avere della logica che viene avviata in automatico allo start dell'applicazione

Fluent UI


Microsoft
Search everything...
Home
Getting Started
What's new
Upgrade guide
Code setup
Project templates
Themes
Design tokens
Reboot
Icons and Emoji
Services
Layout
Form & Inputs
Components
Overview
Accordion
Anchor
Anchored Region
AppBar
Welcome to the Fluent UI Blazor components library
This is the demo and documentation site for version 4.6.1 of the library. This version supports .NET 8 only
The demo and documentation sites for previous version is also still available:
Version 3.6.0
This version supports both .NET 6 and .NET 7.
Introduction
The Microsoft.FluentUI.AspNetCore.Components package provides a set of Blazor components. Some of the components are wrappers around Microsoft's official Fluent UI Web Components. Others are components that leverage the Fluent Design System or make it easier to work with Fluent. To get up and running with the Microsoft.FluentUI.AspNetCore.Components library, see the 'Getting Started'' section below.
The source for the library is hosted in the fluentui-blazor repository at GitHub.
Open at Microsoft - 2024
Getting started w...
Exploring the Flu...
Advanced scenar...
.NET Conf 2024
Unlocking the po...
In this article
Introduction
What's new?
Getting Started
Scripts
Styles
Reboot
Code setup
Getting started by using project templates
Using the Fluent UI Blazor components
Simple components example
Configuring the Design System
Blazor Hybrid
Temporary workaround for MAUI/WPF/Windows Forms issues
Use the DataGrid component with EF Core
Installation
Usage
Support
Joining the Community
Console log
[12:44:38] - Open notification center
[12:44:39] - Notification center dismissed
[12:44:42] - Open site settings
Version: 4.6.1+ff23edc6 - Powered by .NET 8.0.4
Microsoft © 2024. All rights reserved.

FAST's Adaptive UI technology

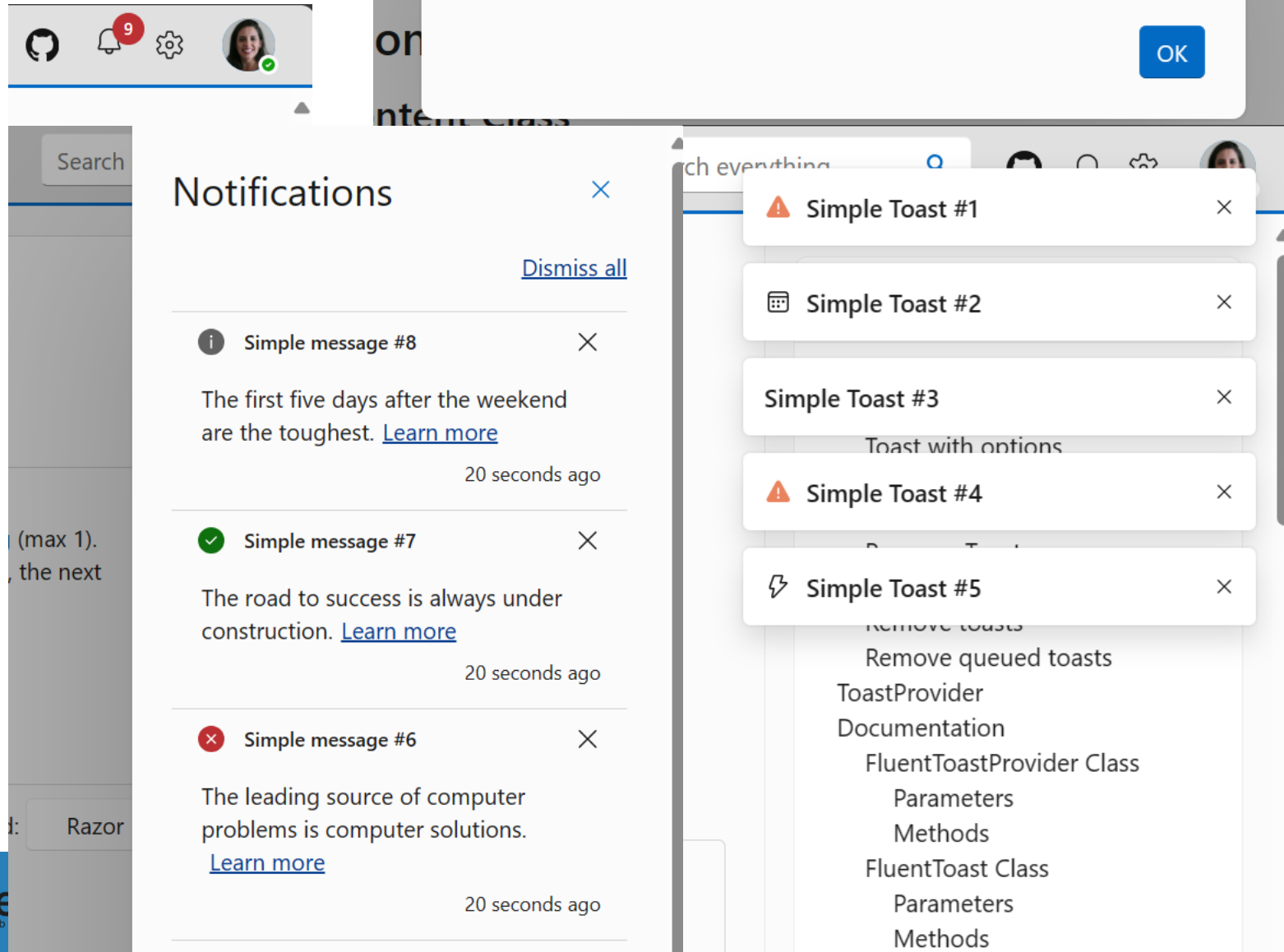


Fluent UI - Services

DialogService

MessageService

ToastService



Azure Service Bus

Azure Service Bus

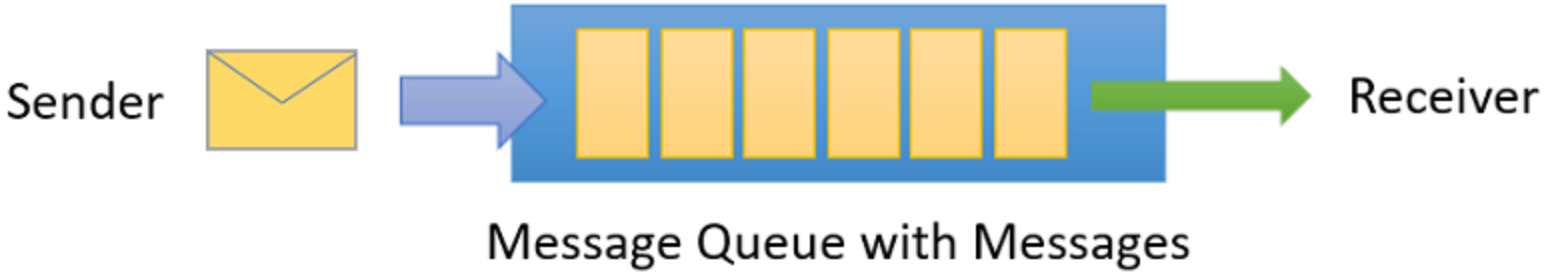
Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics. Service Bus is used to decouple applications and services from each other, providing the following benefits:

- Load-balancing work across competing workers
- Safely routing and transferring data and control across service and application boundaries
- Coordinating transactional work that requires a high-degree of reliability

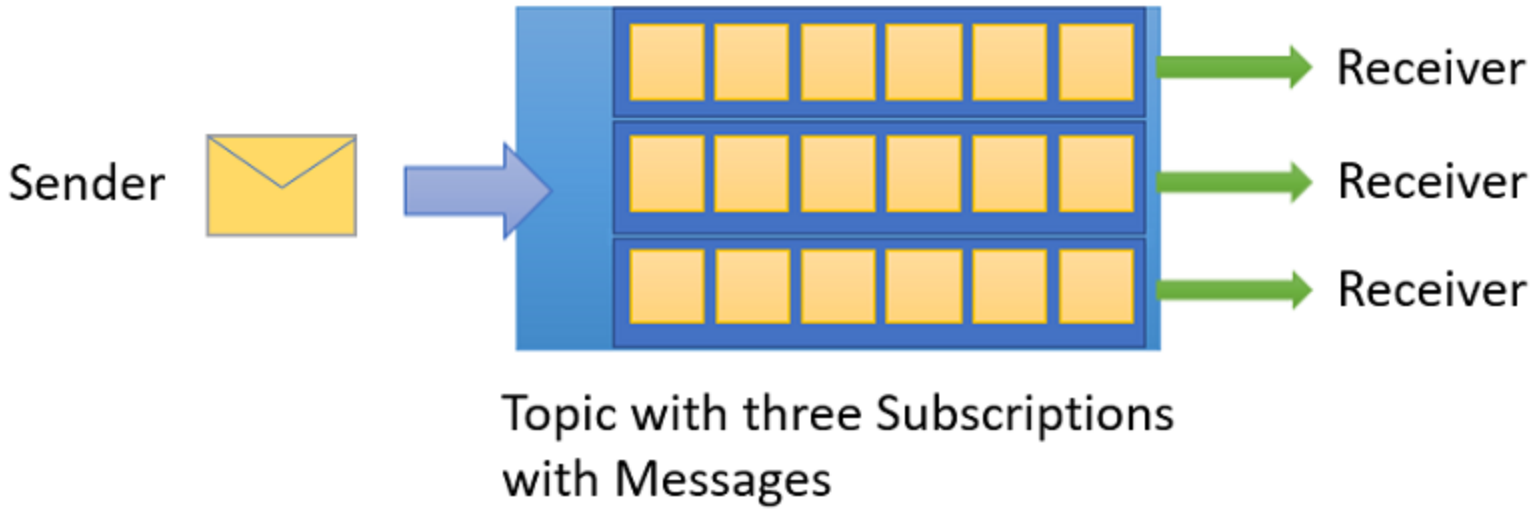
Azure Service Bus

> Azure Service Bus is a fully managed enterprise message broker with message **queues** and publish-subscribe **topics**. Service Bus is used to decouple applications and services from each other, providing the following benefits:
>
> - Load-balancing work across competing workers
> - Safely routing and transferring data and control across service and application boundaries
> - Coordinating transactional work that requires a high-degree of reliability

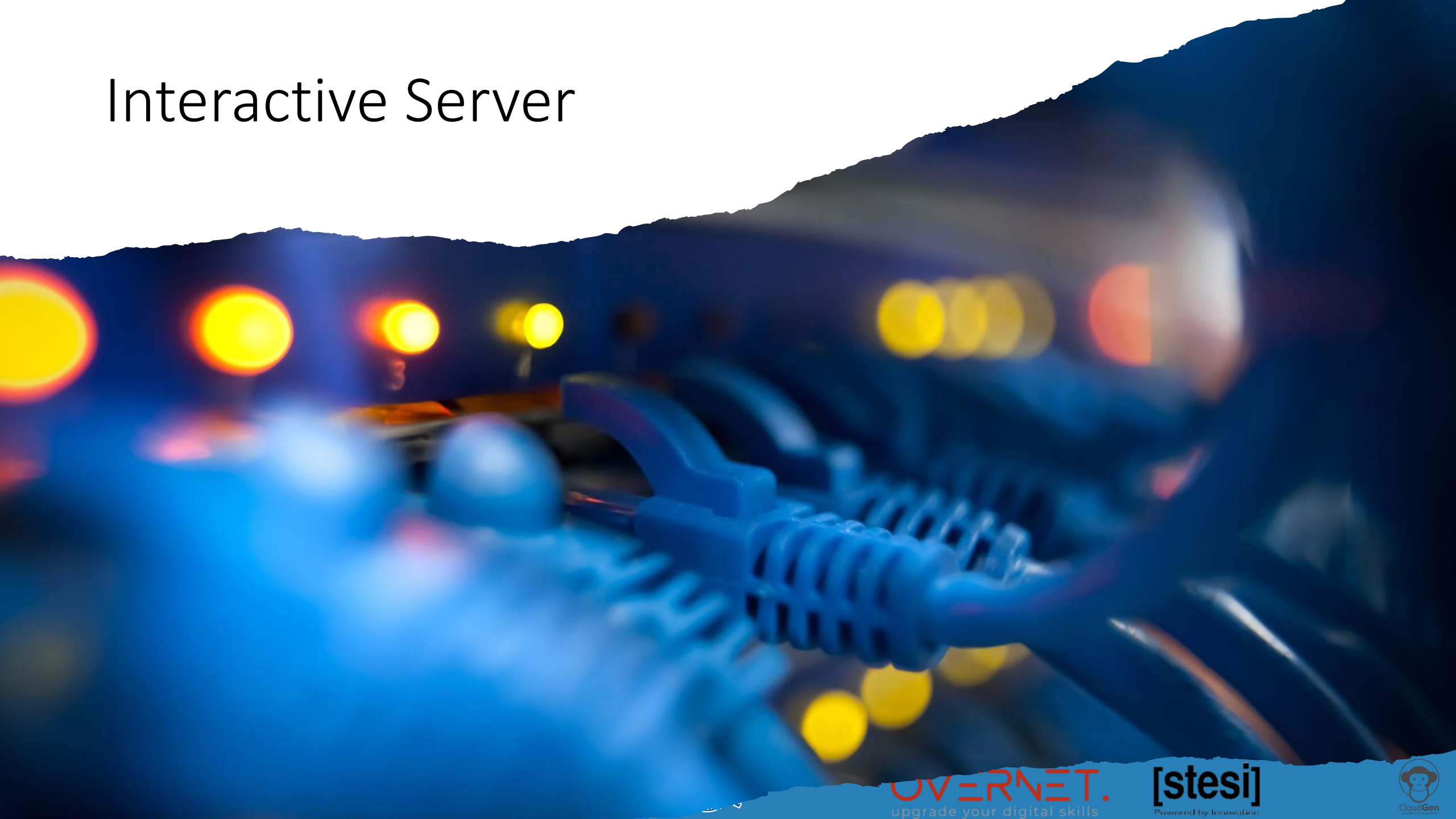
Azure Service Bus - queue



Azure Service Bus - topic

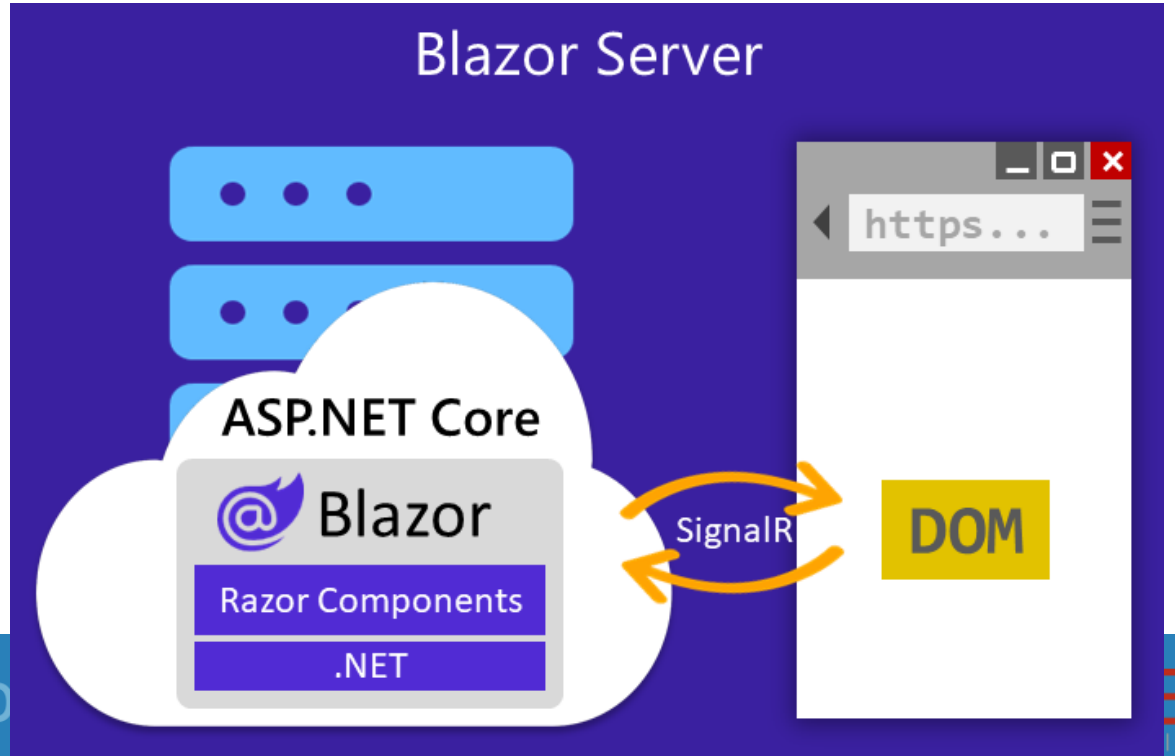


Interactive Server

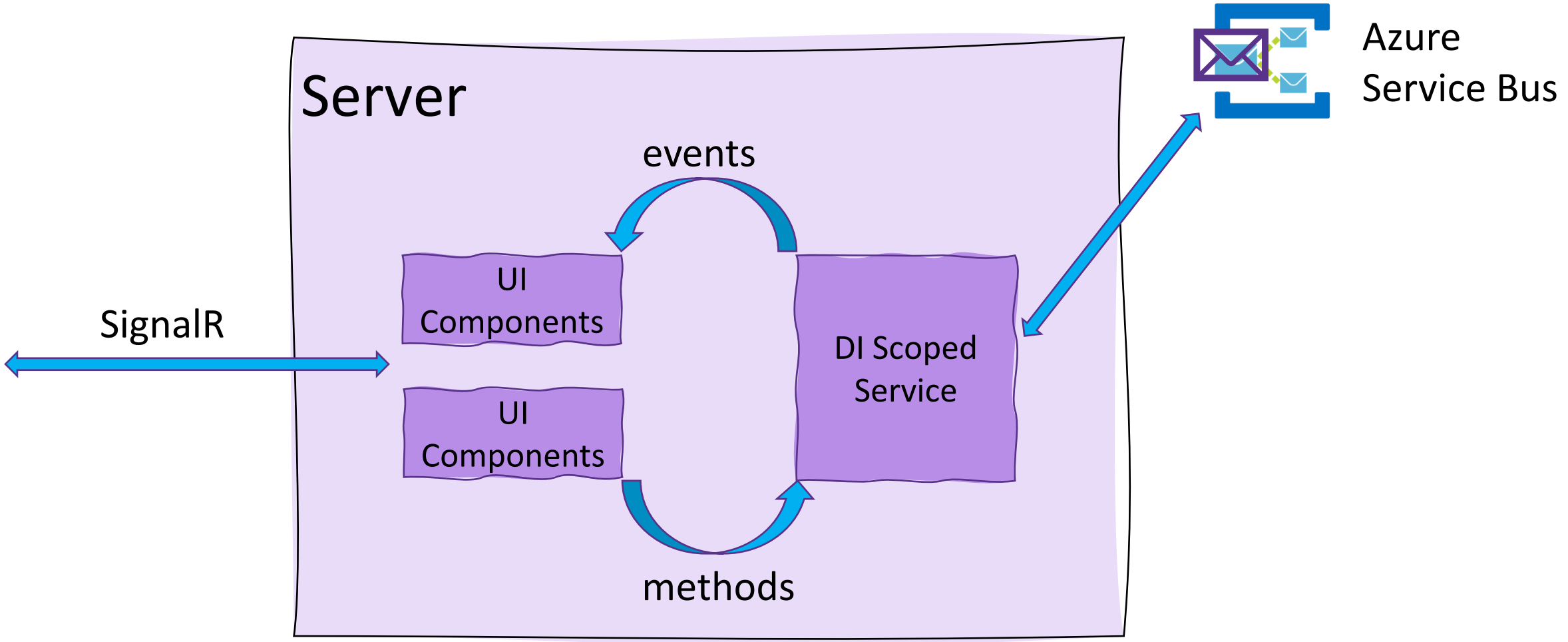


Interactive Server

- Siamo già sul server
- Il client è collegato con SignalR



DI Scoped + Events



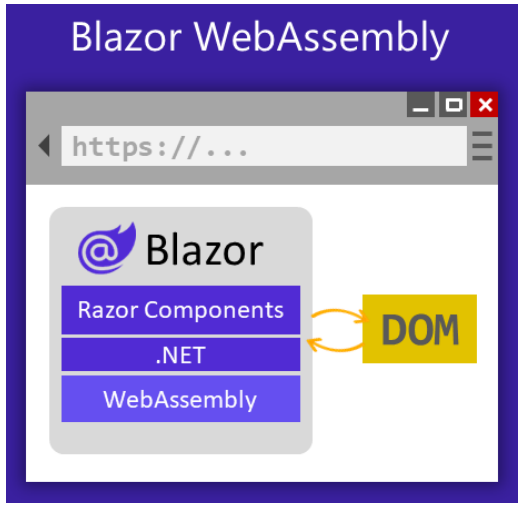
Demo

Interactive Server

Interactive WebAssembly

Interactive WebAssembly

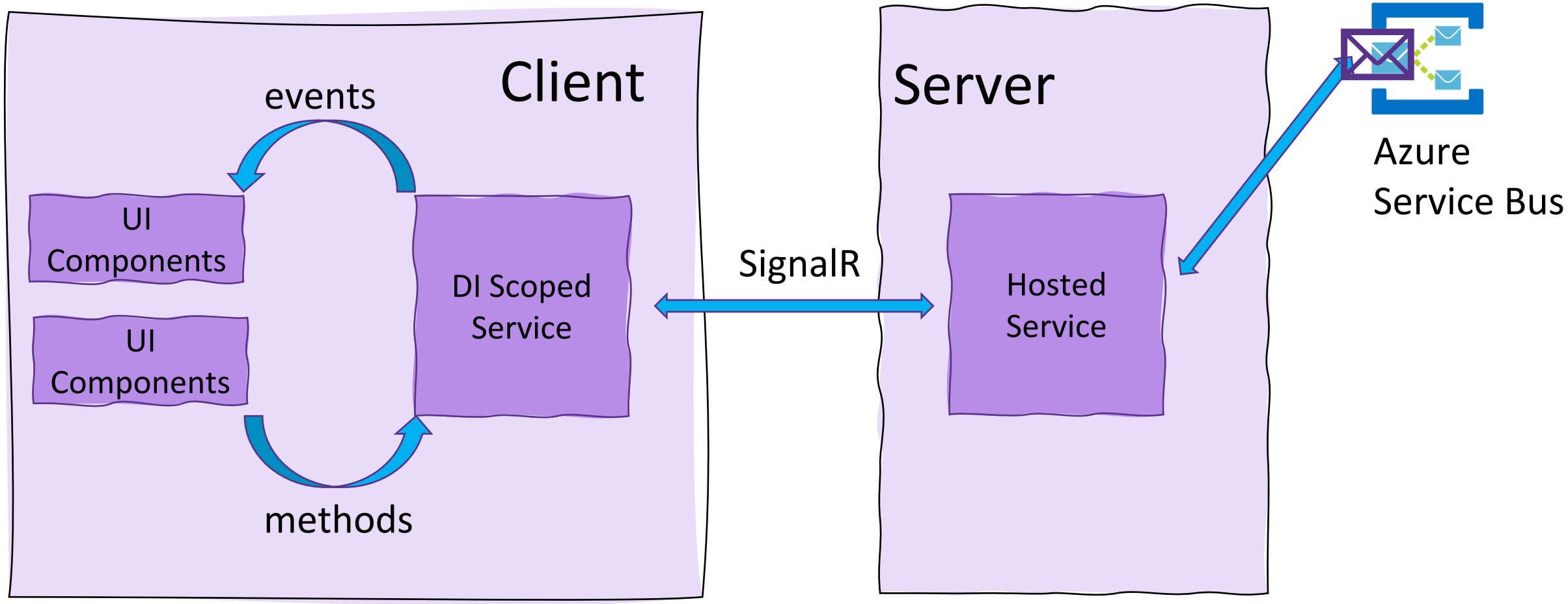
- L'applicazione viene eseguita nel browser
- Possiamo utilizzare SignalR per ricevere gli eventi dal server
  - Sta a noi gestire la connessione
  - Sta a noi gestire correttamente l'invio e la ricezione dei messaggi



IHostedService + SignalR + DI Scoped + Events

- Un servizio in background può sottoscriversi alla ricezione di eventi da servizi esterni (es. Azure Service Bus, …)
- Utilizzare SignalR per inviare il messaggio ricevuto a tutti i client che ne hanno bisogno
- Utilizzare un servizio scoped per ricevere i messaggi da SignalR e scatenare l'evento (gestito dai component di UI)

IHostedService + SignalR + Scoped + Events



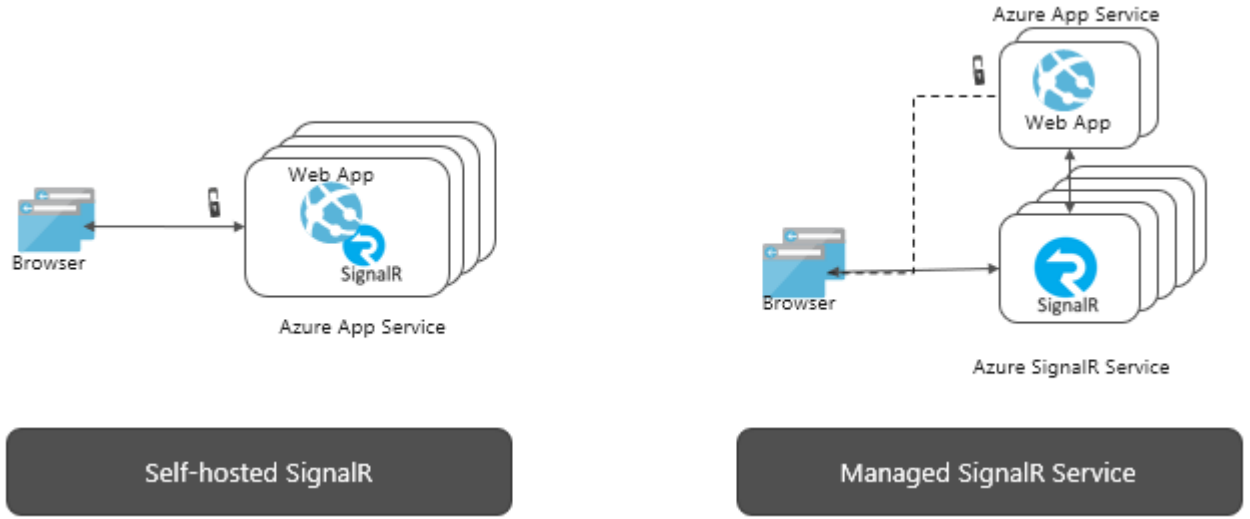
Demo

Interactive WebAssembly

Azure SignalR
Azure Web PubSub

Possiamo ottimizzare?

- Se riceviamo una grande quantità di dati, è possibile valutare di sostituire SignalR (verso il server) con servizi esterni all'applicazione.
- Azure SignalR o Azure Web PubSub sono una possibile scelta.



Azure SignalR

> Azure SignalR Service simplifies the process of adding real-time web functionality to applications over HTTP. This real-time functionality allows the service to push content updates to connected clients, such as a single page web or mobile application. As a result, clients are updated without the need to poll the server, or submit new HTTP requests for updates.

Azure SignalR

> Azure SignalR Service simplifies the process of adding real-time web functionality to applications over HTTP. This real-time functionality allows the service to push content updates to connected clients, such as a single page web or mobile application. As a result, clients are updated without the need to poll the server, or submit new HTTP requests for updates.

Azure SignalR – integrazione in Blazor Server

```
services.AddSignalR().AddAzureSignalR(config =>
    {
        config.ServerStickyMode = ServerStickyMode.Required;
        config.ConnectionString = "...";
    });
```

Azure Web PubSub

> The Azure Web PubSub Service helps you build real-time messaging web applications using WebSockets and the publish-subscribe pattern easily. This real-time functionality allows publishing content updates between server and connected clients (for example a single page web application or mobile application). The clients do not need to poll the latest updates, or submit new HTTP requests for updates.

Azure Web PubSub

> The Azure Web PubSub Service helps you build real-time messaging web applications using WebSockets and the publish-subscribe pattern easily. This real-time functionality allows publishing content updates between server and connected clients (for example a single page web application or mobile application). The clients do not need to poll the latest updates, or submit new HTTP requests for updates.

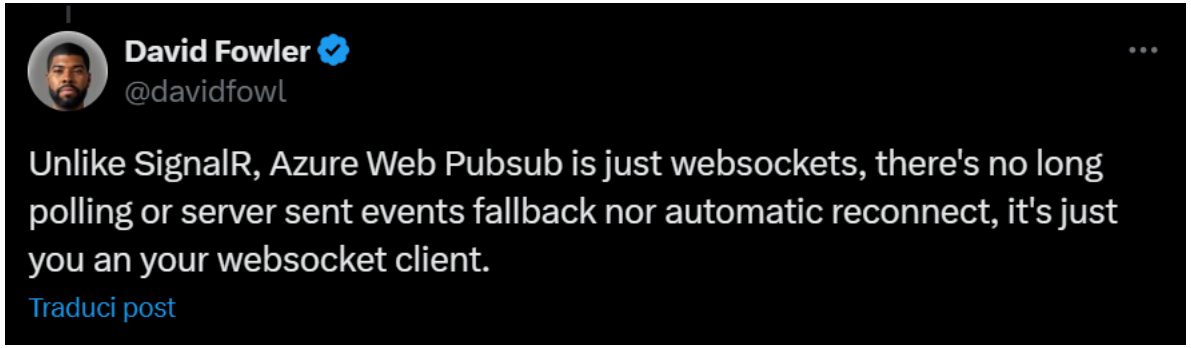
Azure Web PubSub - *standard WebSocket*

> Azure Web PubSub service works with a broad range of clients, such as web and mobile browsers, desktop apps, mobile apps, server process, IoT devices, and game consoles. Since this service supports the standard WebSocket connection with publish-subscribe pattern, it is easily to use any standard WebSocket client SDK in different languages with this service.

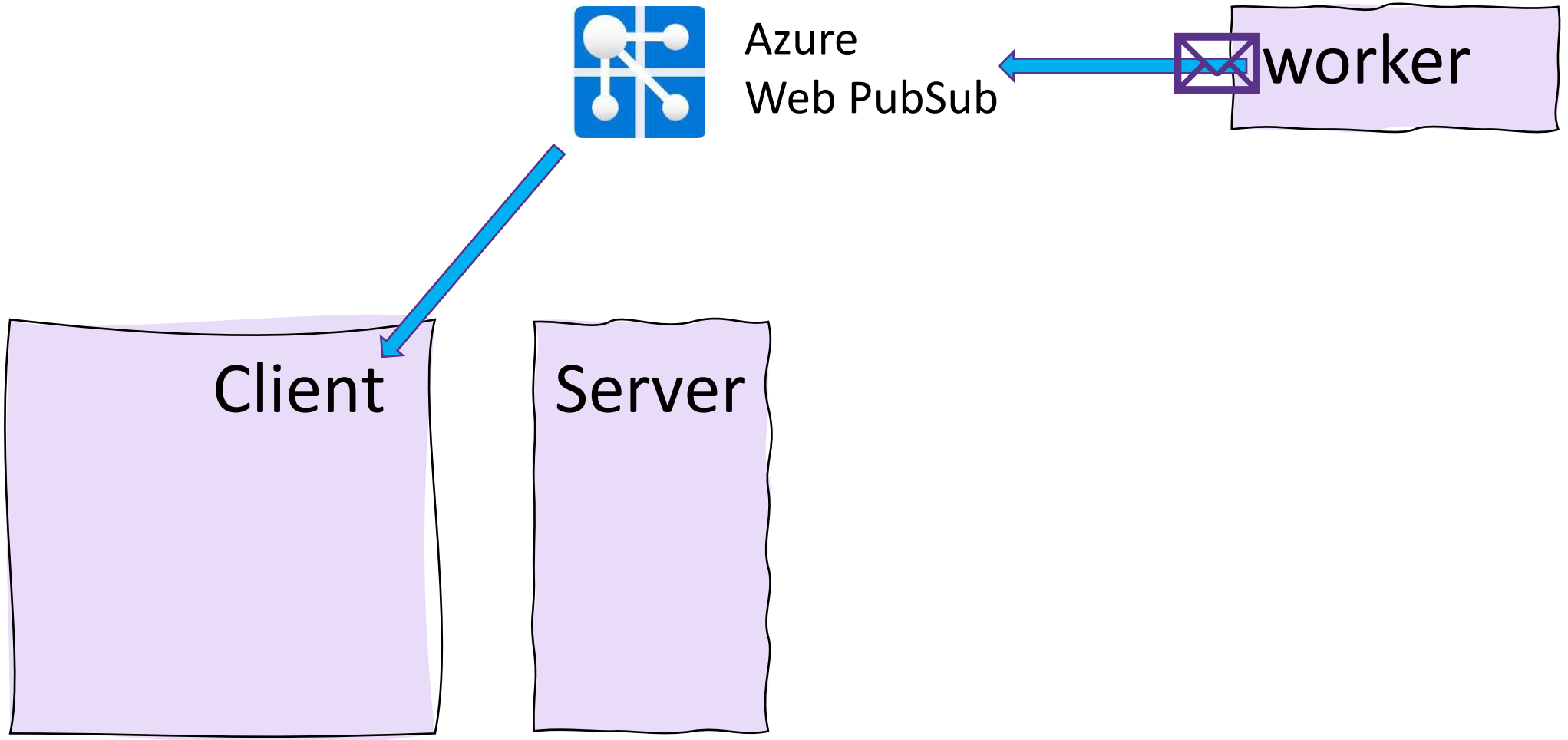
Azure Web PubSub - *standard WebSocket*

> Azure Web PubSub service works with a broad range of clients, such as web and mobile browsers, desktop apps, mobile apps, server process, IoT devices, and game consoles. Since this service supports the standard WebSocket connection with publish-subscribe pattern, it is easily to use any standard WebSocket client SDK in different languages with this service.

Azure Web PubSub vs Azure SignalR



Push messages from server/worker



Demo

Posizione gps in real-time

E ora?

- Interactive Server o Interactive WebAssembly?
- SignalR o gRPC?
- Azure Service Bus, Azure SignalR o Azure Web PubSub?

¯_(ツ)_/¯ dipende…

Codice e slide su github

- https://github.com/andreadottor/Umarell-ParentalControl

Grazie!

@andreadottor

{

name:	**Andrea Dottor**
email:	**andrea@dottor.net**
x:	**@dottor**
linkedin:	**@andreadottor**

}

.NET IN PILLOLE
il podcast

MVP Microsoft® Most Valuable Professional