Lo strano caso del dottor Xamarin e di mister Blazor

Aprile Salvatore
Solution Architect @ Fincons Group S.p.A

DOTNETSIDE

27 aprile 2020

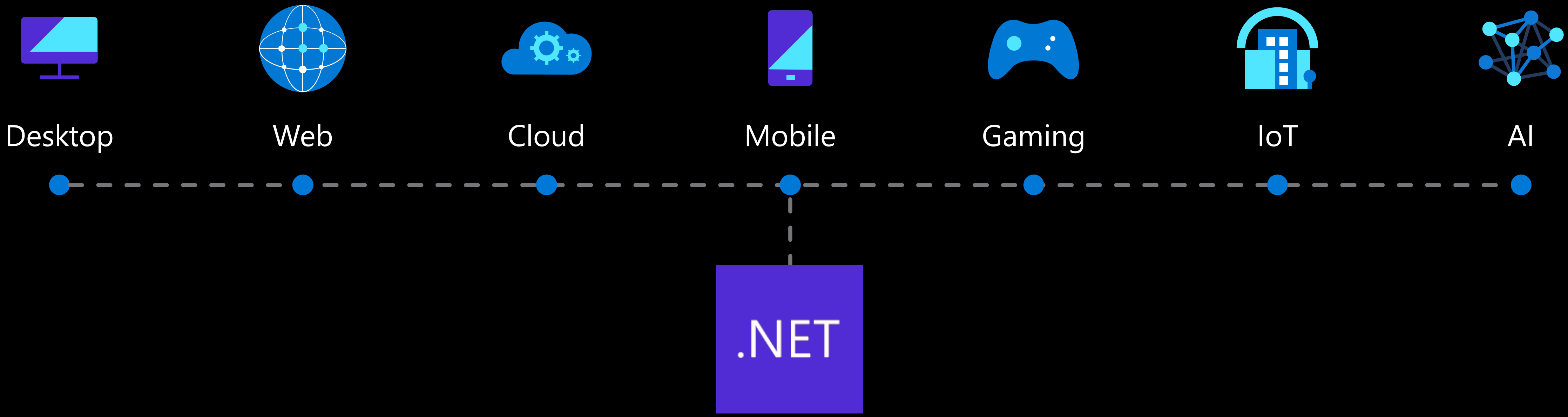
Chi sono

Solution Architect - Fincons Group S.p.A.

- MCPD - Designing and Developing Web Applications
- MCPD - Designing and Developing Windows Azure Applications
- MCTS ASP.NET, ADO.NET e WCF

.NET

Your platform for building anything



Desktop

Web

Cloud

Mobile

Gaming

IoT

AI

.NET

Xamarin
VS
blazor

Le novità Xamarin Forms 4.5

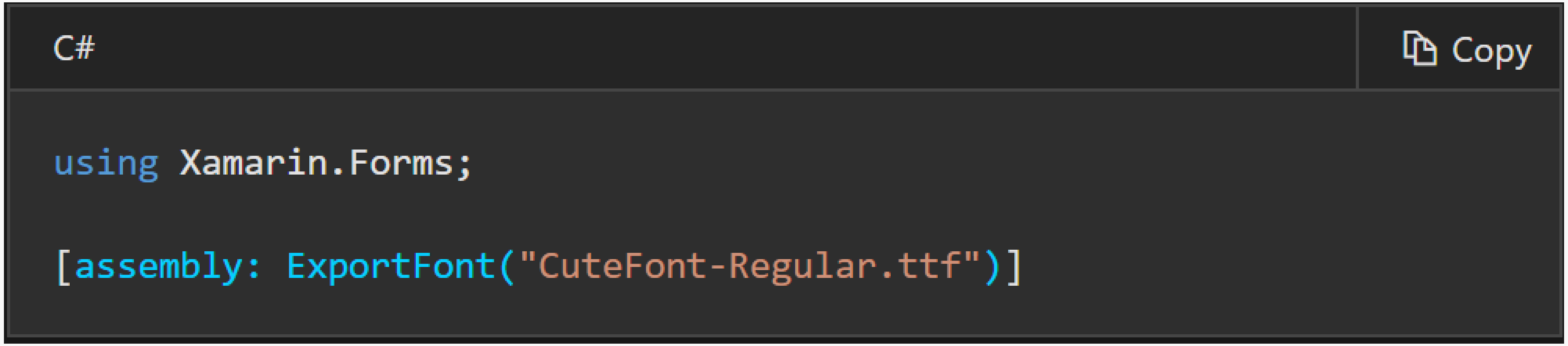
Ultima release disponibile: **Xamarin.Forms 4.5.0.617 (4.5.0 Service Release 4)**

1- At the end : AndroidX

Xamarin.Forms now uses the latest AndroidX libraries from Google. These replace the Android Support libraries which are no longer being updated.

2- Embedded Fonts

Now to use fonts you only need to add them to your Xamarin.Forms shared library as embedded resources and reference them with an assembly tag, rather than adding them to multiple platforms, each with their own implementation rules. To your AssemblyInfo.cs add:

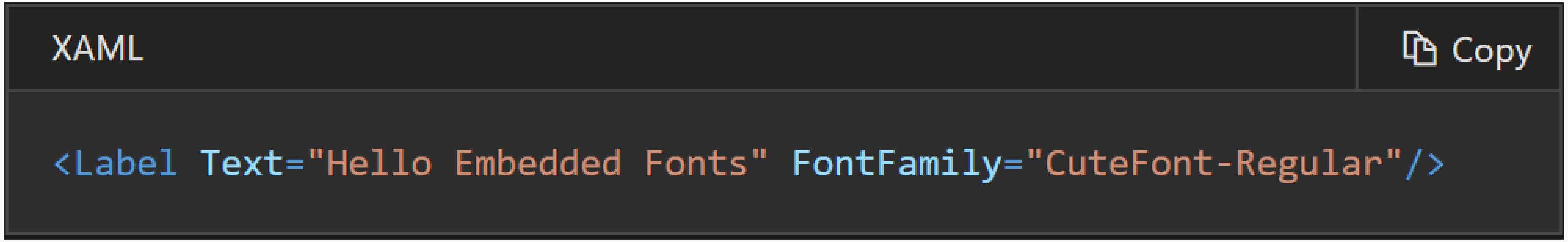


```csharp
using Xamarin.Forms;

[assembly: ExportFont("CuteFont-Regular.ttf")]
```

2- Embedded Fonts

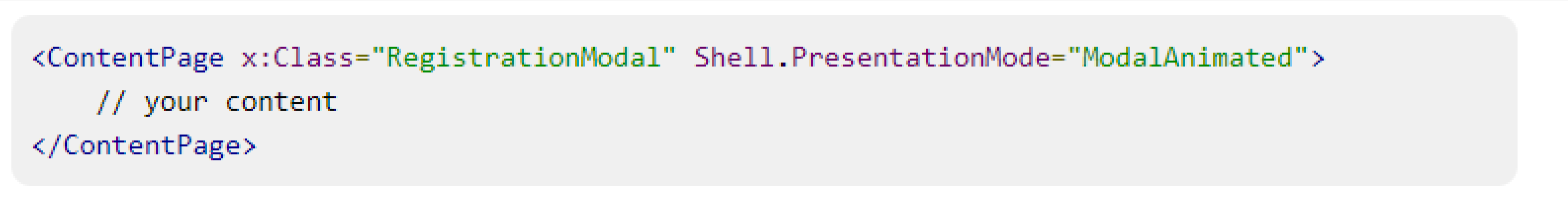
Then to use the font in XAML, reference that name:



```xml
<Label Text="Hello Embedded Fonts" FontFamily="CuteFont-Regular"/>
```

3- Shell Modals

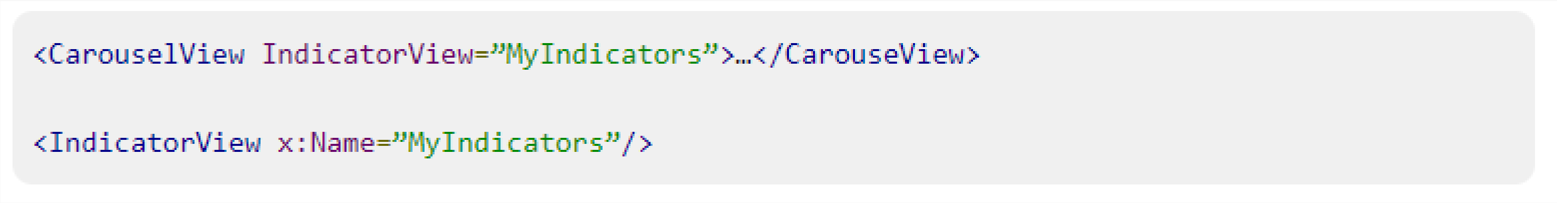
Whether you are using tabs or a fly-out menu, Shell is the easiest way to get started with your cross-platform mobile application. A favorite aspect of using Shell is the URI based navigation service that is omnipresent. Now, the ability to display modal pages uses that very same URI method. By using the new *Shell.PresentationModel* attached property, can describe your intention for any page you want to present as a modal.



```xml
<ContentPage x:Class="RegistrationModal" Shell.PresentationMode="ModalAnimated">
    // your content
</ContentPage>
```

4 - CarouselView and IndicatorView (Preview)

Now are available CarouselView and IndicatorView



```
<CarouselView IndicatorView=”MyIndicators”>…</CarouseView>

<IndicatorView x:Name=”MyIndicators”/>
```

The carousel also has implemented four visual states to make it easier to style. The following represent the visual state for:

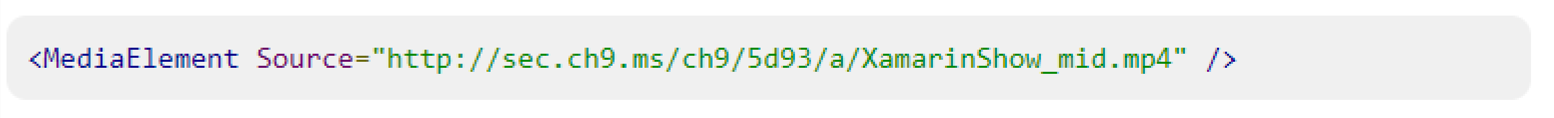
1. **CurrentItem** – currently displayed item.
2. **PreviousItem** – previously displayed item.
3. **NextItem** – the next item.
4. **DefaultItem** – remainder of the items.

5 – Swipe View (Preview)

Reveal contextual content when swiping in specific directions by wrapping any control with a SwipeView. The community has been very inspired by this control and have been using it in many different ways! We are working hard to stabilize this control by implementing many requested improvements.

6 – Media Element (Preview)

Brand new in this release is MediaElement, a base control for playback of audio and video. See this control featured in our XamarinTV sample app, which was featured during the Microsoft Surface Dual-Screen live event from Redmond. The control is extremely easy to use! Add the control to your layout, then set the source to any secure media URI.



```
<MediaElement Source="http://sec.ch9.ms/ch9/5d93/a/XamarinShow_mid.mp4" />
```

7 – State Triggers (Preview)

We have added state triggers in support of new dual-screen experiences, the new MediaElement, and generally to be more useful.

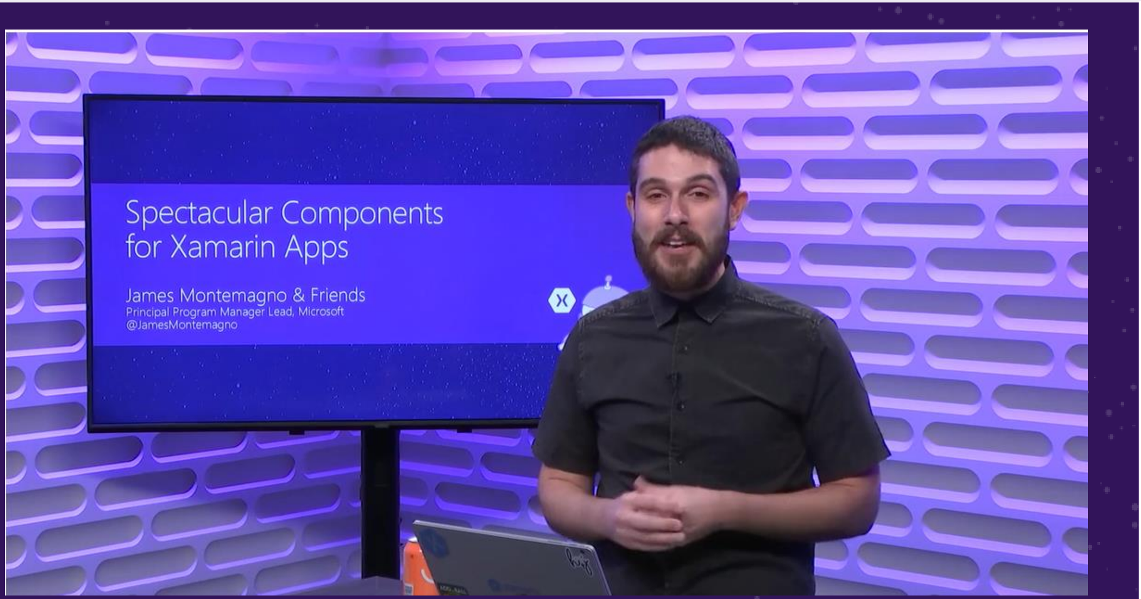
1.Adaptive Trigger – reacts to changes in the width and height of an application window.

2.Compare Trigger – occurs when two values are compared.

3.Device Trigger – occurs when running on the specified device.

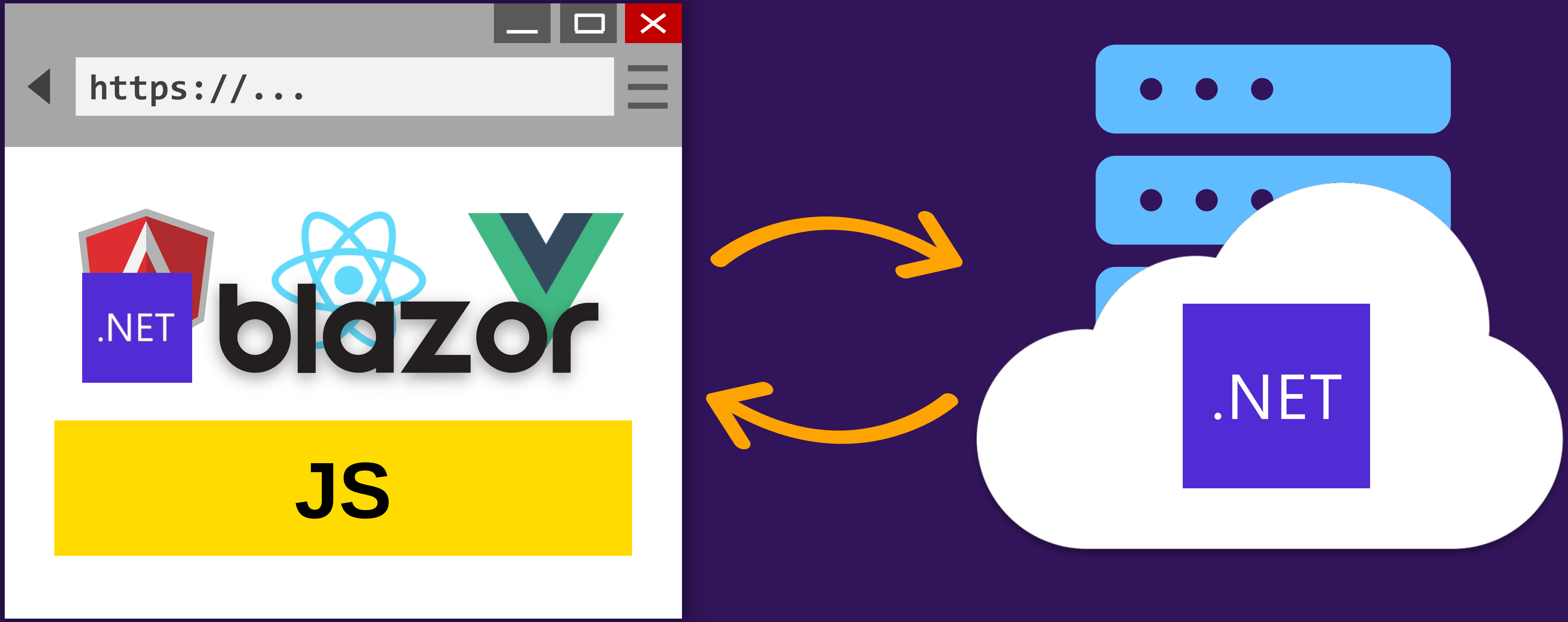
4.Orientation Trigger – occurs when the device orientation changes.

Spectacular Components for Xamarin Apps


Spectacular Components for Xamarin Apps
James Montemagno & Friends
Principal Program Manager Lead, Microsoft
@JamesMontemagno

https://channel9.msdn.com/Events/dotnetConf/Focus-on-Xamarin/Spectacular-Components-for-Xamarin-Apps

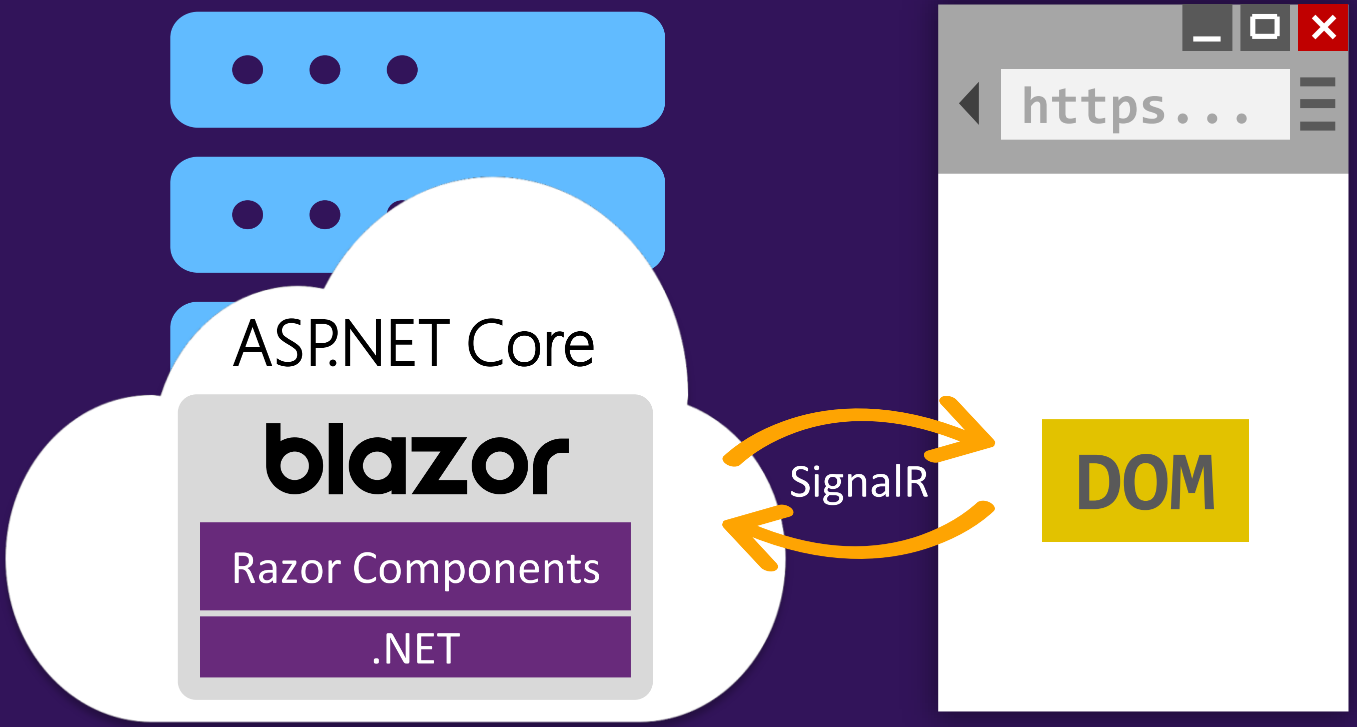
Che cos'è Blazor?



https://...
.NET
blazor
JS
.NET

Come funziona Blazor

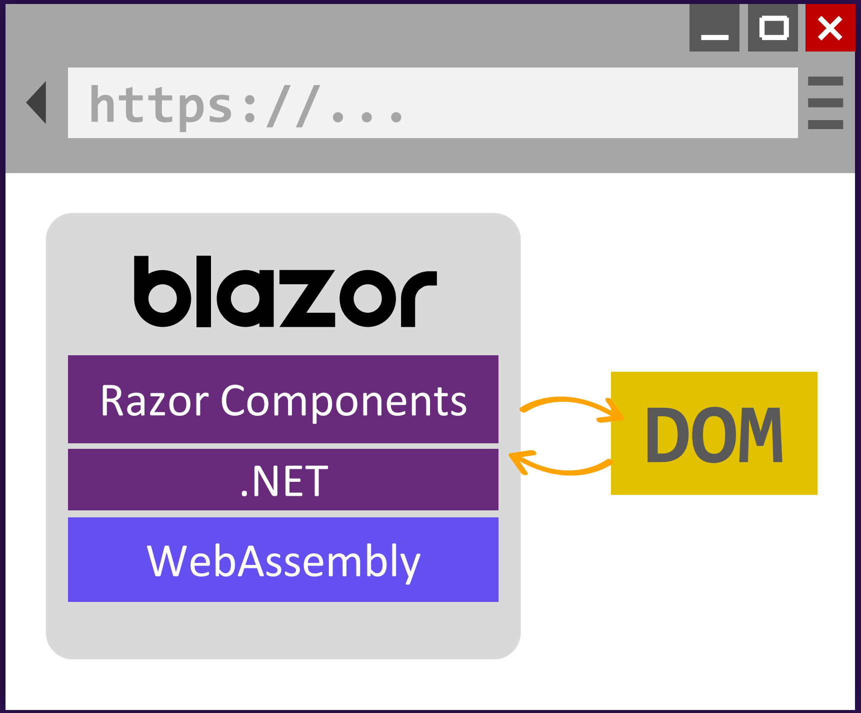
Blazor Server



.NET Core 3.1 LTS

Come funziona Blazor

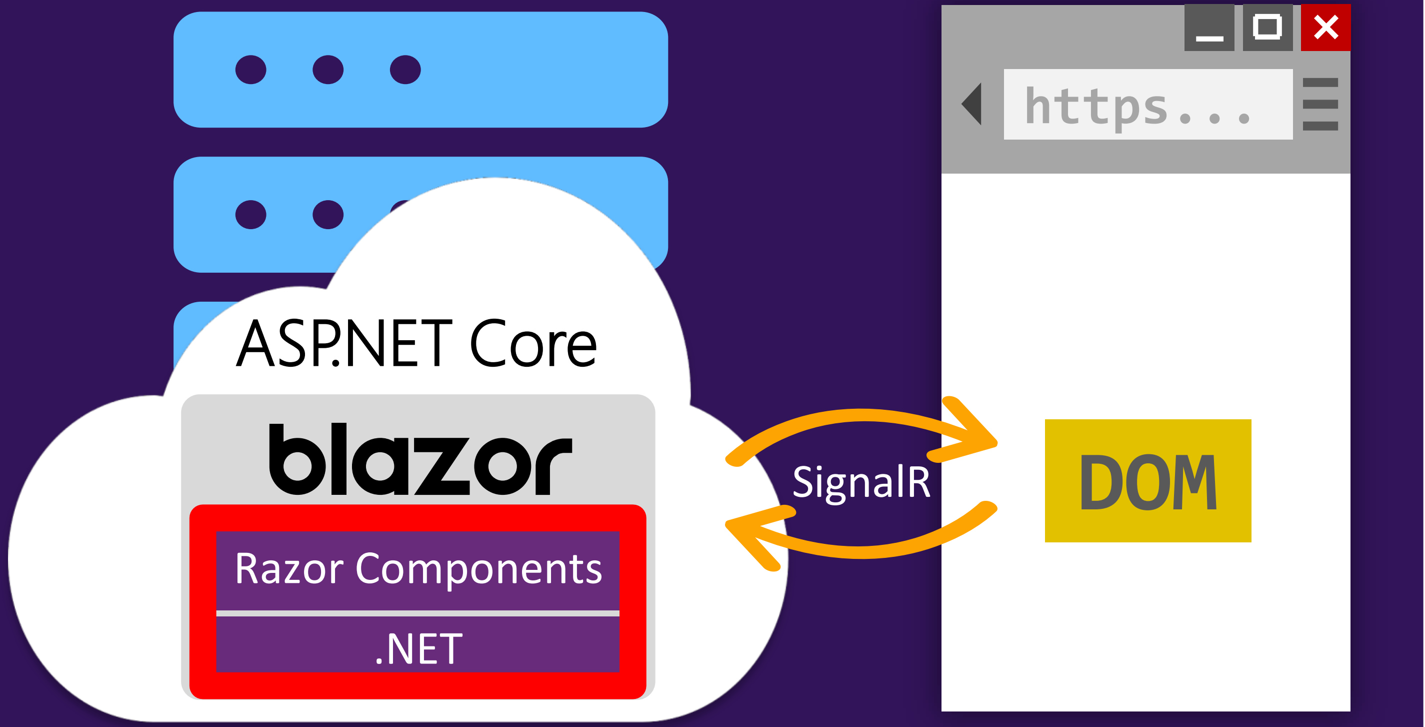
Blazor WebAssembly



Preview

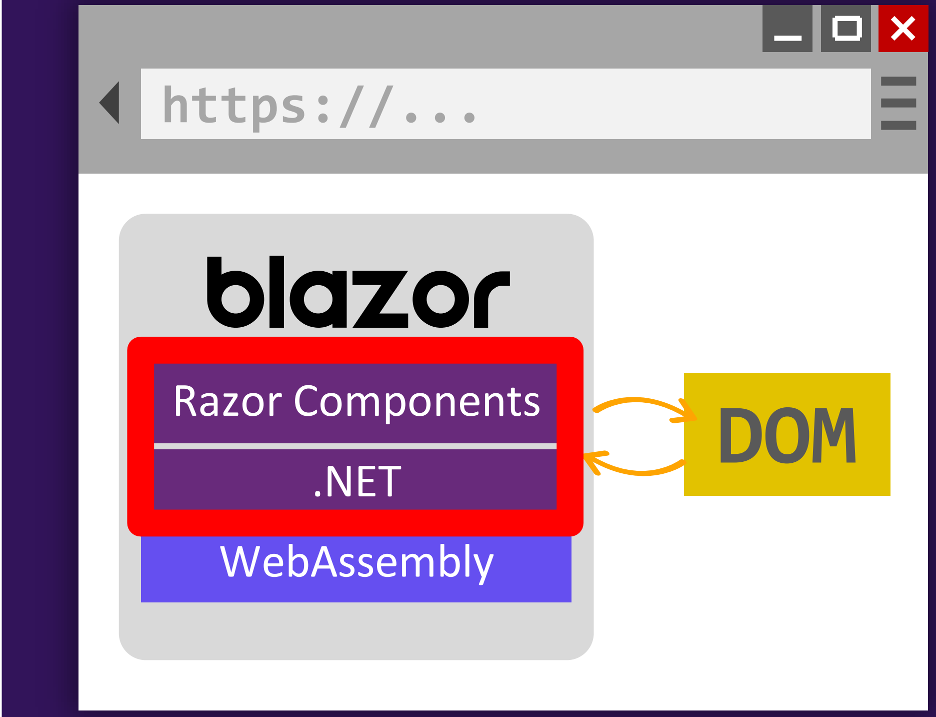
Shared component model

Blazor Server

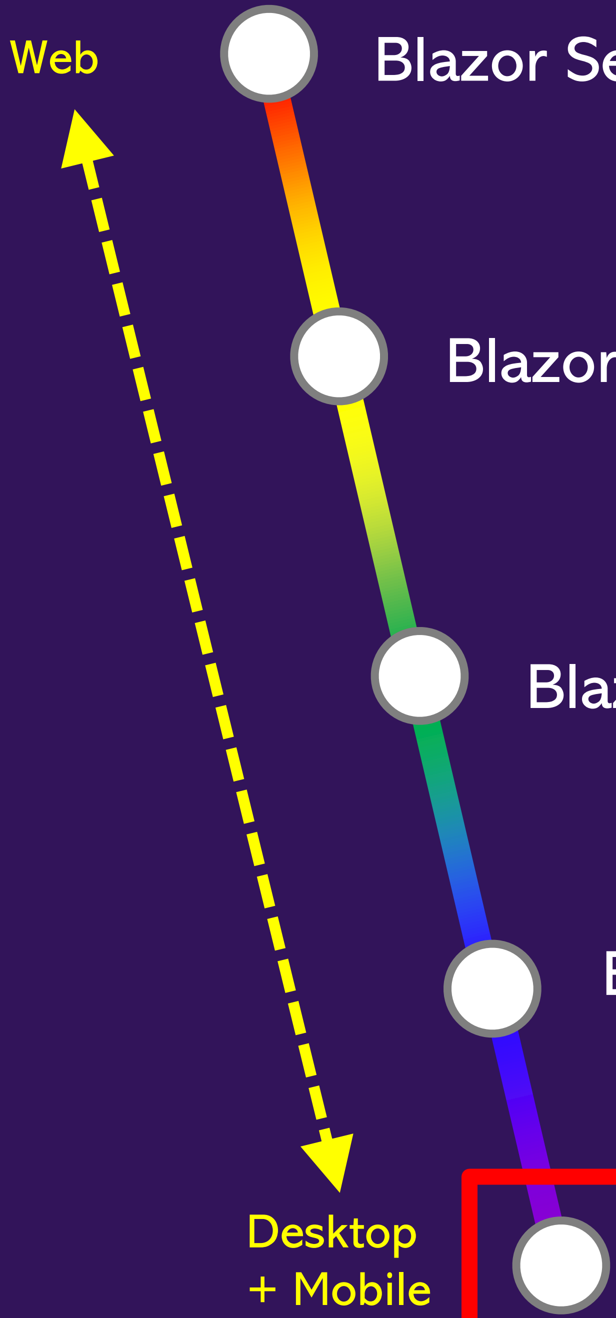


.NET Core 3.1 LTS

Blazor WebAssembly



Preview



Blazor Server

Web app
Tutte le interazioni vengono gestite sul server
HTML Prerenderizzato

Blazor WebAssembly

Web app con esecuzione sul client
Caricata dal web server
Può funzionare offline con i Service Worker

Blazor PWA – OS installed

Si presenta come una app nativa (con una finestra propria)
Funziona sia offline che online

Blazor Hybrid

Utilizza Electron / WebView con un rendering nativo .NET presentandosi come una app nativa (con una finestra propria)
Funziona sia offline che online

Blazor Native

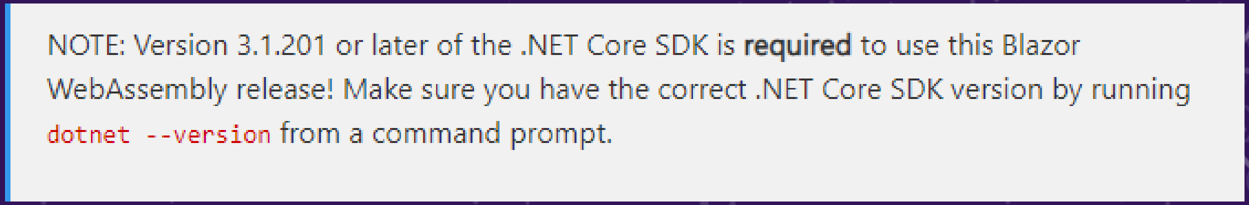
Stesso modello di programmazione, ma con un rendering della UI non HTML

Blazor WebAssembly 3.2.0 Preview 5 release now available

A new preview update of Blazor WebAssembly is now available! Here’s what’s new in this release:

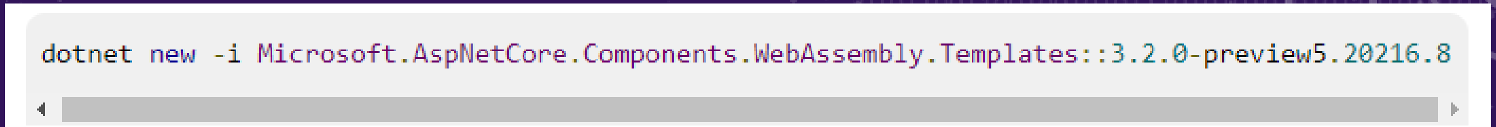
- Read configuration during startup
- Configure HTTP fetch request options
- Honor existing web.config when publishing
- Attach tokens to outgoing requests
- Support for time zones

To get started with Blazor WebAssembly 3.2.0 Preview 5 install the latest .NET Core 3.1 SDK.



NOTE: Version 3.1.201 or later of the .NET Core SDK is **required** to use this Blazor WebAssembly release! Make sure you have the correct .NET Core SDK version by running `dotnet --version` from a command prompt.

To install Blazor WebAssembly template:



```
dotnet new -i Microsoft.AspNetCore.Components.WebAssembly.Templates::3.2.0-preview5.20216.8
```

ANNUNCIO

Experimental Mobile Blazor Bindings

Puoi creare app mobile native con *Blazor*

Puoi riutilizzare le tue skill web: Razor, C#, CSS

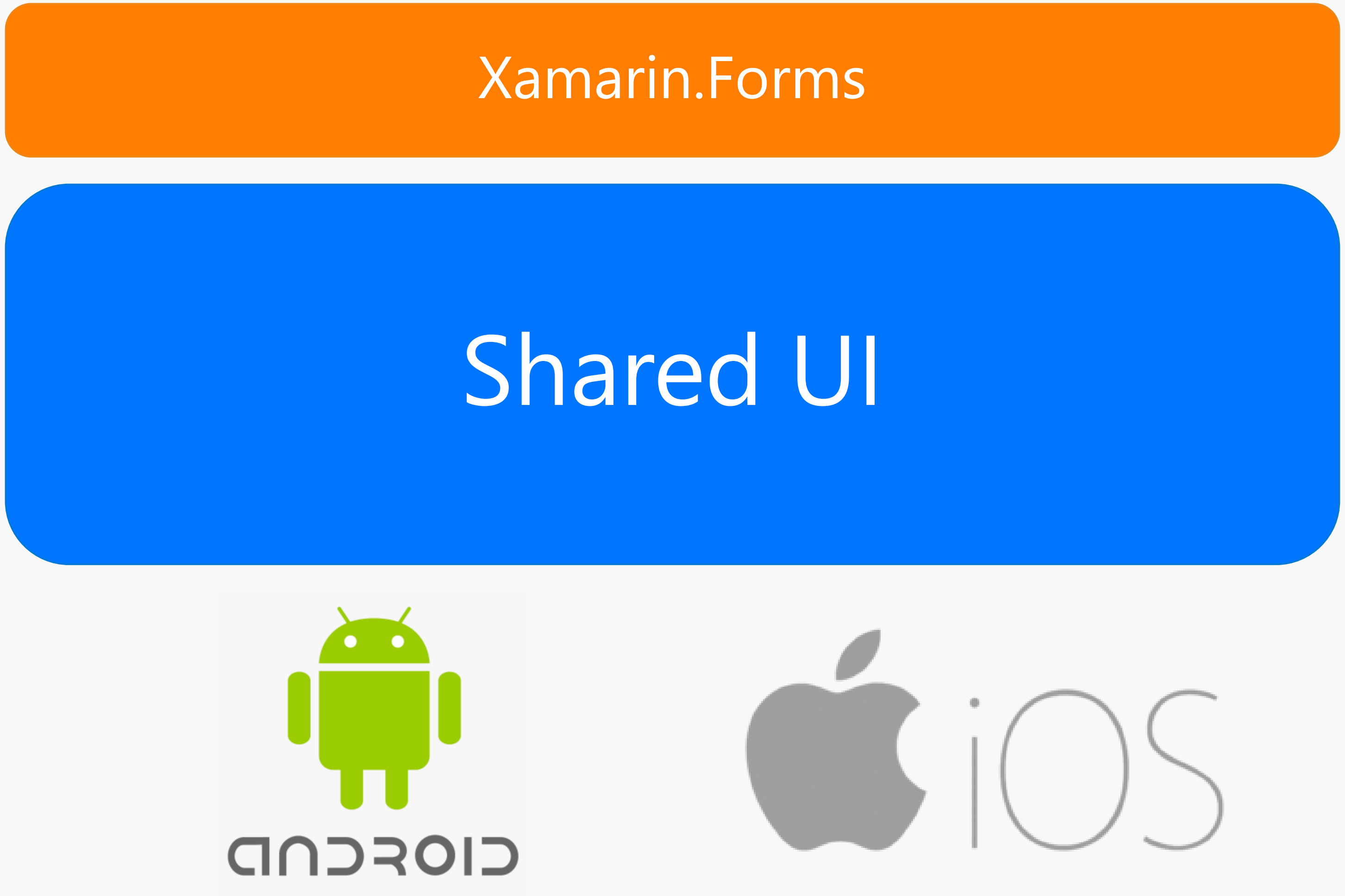
Hai un ricco set di componenti mobile nativi

Hai accesso al 100% alle API native: GPS, Media, & altro

Puoi usare l’ecosistema di librerie NuGet per il mobile

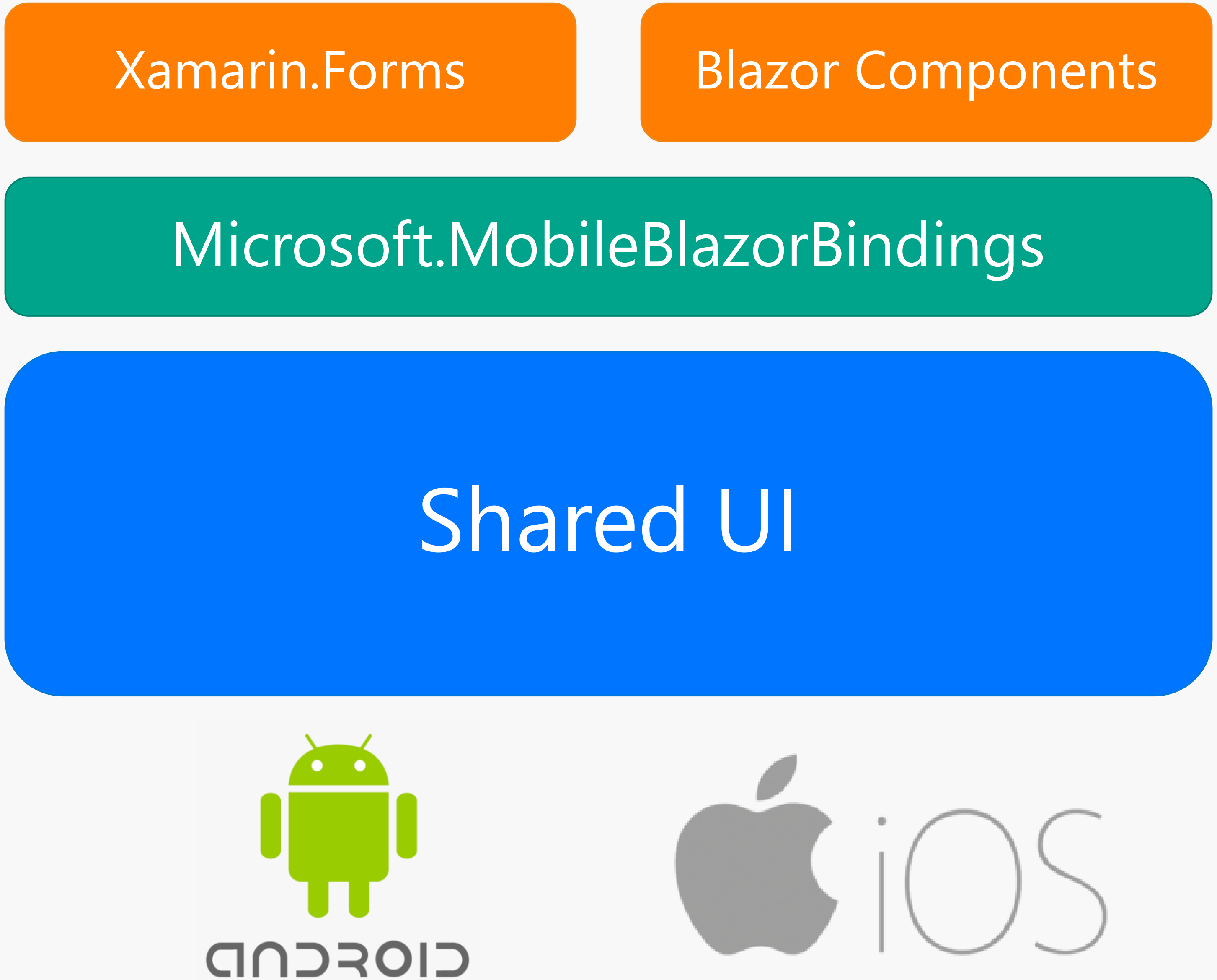
Tutto in .NET

aka.ms/mobileblazorbindings



Xamarin.Forms

Shared UI



Xamarin.Forms

Blazor Components

Microsoft.MobileBlazorBindings

Shared UI

Come partire con Blazor per il mobile

- Vai su https://blazor.net
- Installa .NET Core 3.1
- Installa il template Blazor Mobile Bindings

Visual Studio
2019 16.4

Visual Studio for Mac
8.4

Visual Studio Code
+ C# extension

Mobile Blazor Bindings in Action

Controlli disponibili

Page components

- ContentPage
- MasterDetailPage
- Page
- TabbedPage
- TemplatedPage

Layout components

- ContentView
- Frame
- Grid
- ScrollView
- StackLayout

View components

- ActivityIndicator
- BoxView
- Button
- CheckBox
- Image
- ImageButton
- Entry
- Label
- ProgressBar
- Slider
- Stepper
- Switch

Specialized components

- Application
- BaseMenuItem
- FormattedString
- GestureElement
- MenuItem
- Shell e ShellContent
- ShellGroupItem e ShellItem
- FlyoutItem
- Tab e TabBar
- ShellSection
- Span

Xamarin.Essentials

Open Files

Secure Settings

Flashlight

Clipboard

Geolocation

Phone Dialer

Email

Device Info

Email Attachments

Accelerometer

Share

Geocoding

Vibration

File System

Connectivity

Detect Shake

Share File

Sms

Preferences

Gyroscope

Magnetometer

Device Display Info

Battery

Compass

Screen Lock

Text to Speech

Demo

.NET

Cogito

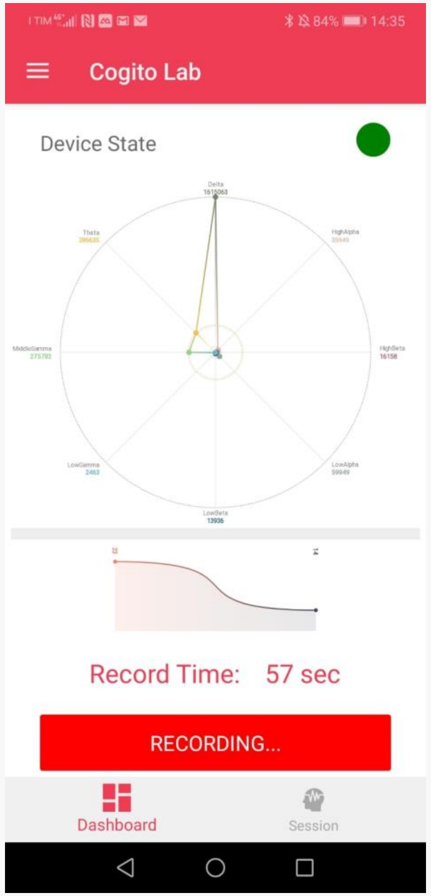
- Progetto di ricerca volto a creare una BCI (Brain Communication Interface) che sia in grado di associare un pensiero ad una azione
- Usa un device chiamato «neurosky mindwave» che è in grado di misurare le onde celebrali
- Questo device supporta il protocollo BLE per la comunicazione

Cogito - Architettura

Neurosky

BLE

Cogito App



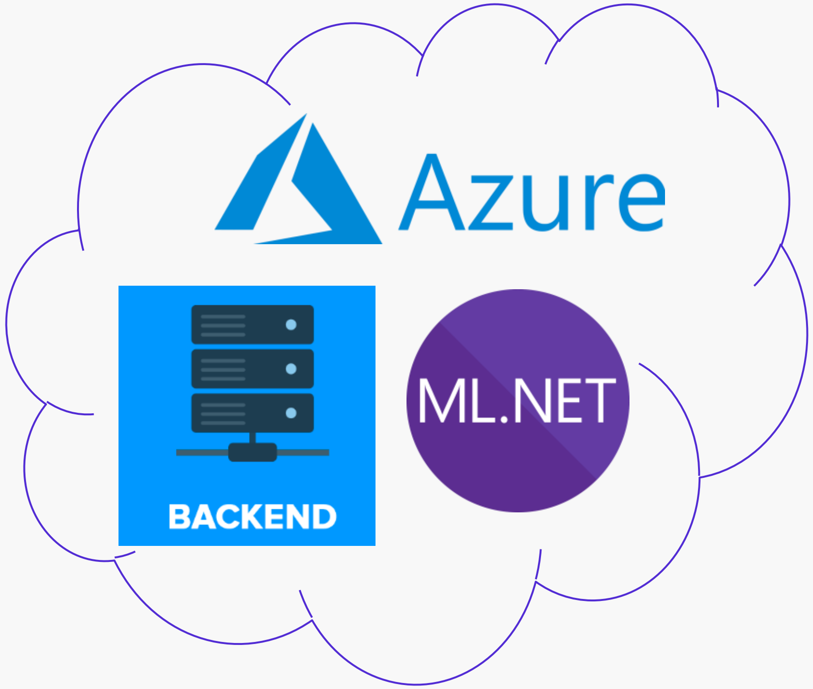
TIM 84% 14:35

Cogito Lab

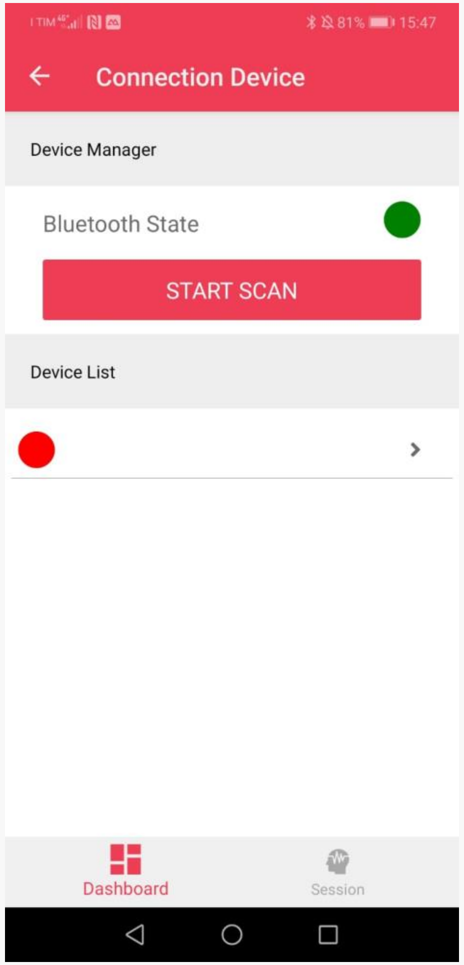
Device State

Record Time: 57 sec

RECORDING...



Cogito - App


Connection Device

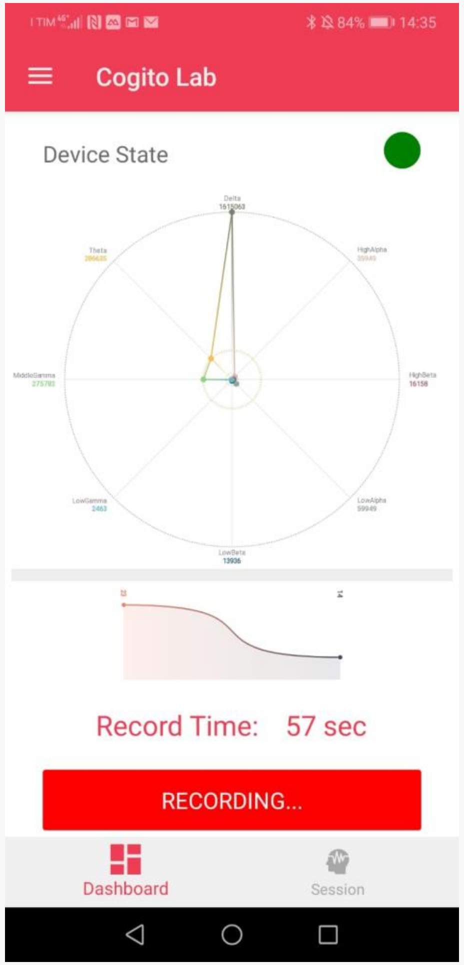
Device Manager

Bluetooth State ●

START SCAN

Device List

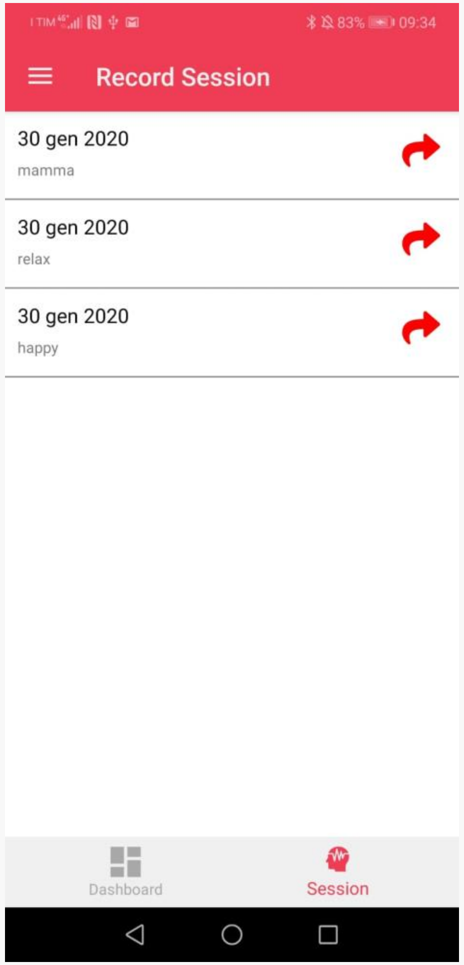
● ›


Cogito Lab

Device State ●

Record Time: 57 sec

RECORDING...


Record Session

30 gen 2020 mamma	➦
30 gen 2020 relax	➦
30 gen 2020 happy	➦

Cogito – Demo

Un po’ di link utili:

- Blazor: https://blazor.net
- Docs: https://blazor.net/docs
- .NET Core 3.1: https://dot.net/get-core3
- Workshop: https://aka.ms/blazorworkshop
- Community: https://aka.ms/awesomeblazor

Seguite la community!

- https://blazordev.it/
- https://twitter.com/blazordevita
- https://github.com/blazordevita
- https://fb.me/blazordeveloperitaliani

Contatti

- Twitter: @SalvoAprile
- Linkedin: https://www.linkedin.com/in/aprilesalvatore/it-it/
- Facebook: https://www.facebook.com/salvatore.aprile1
- Mail: aprile.salvatore@gmail.com
- Github: https://github.com/CodearoundHub