

Xamarin and .net Core 2.0

The brave new world of Microsoft's modernized (UI-) development ecosystem



Tobias Scholze
Software Engineer, @tobonaut





„Mobile first, Cloud first“ – Satya Nadella



+

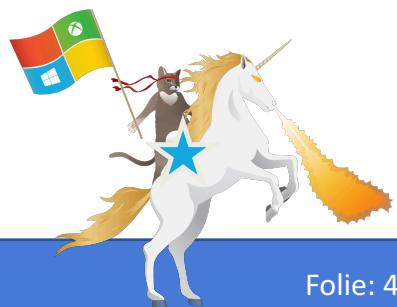
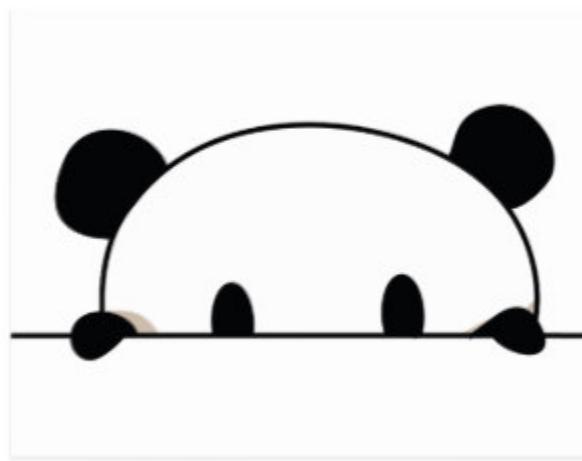


=

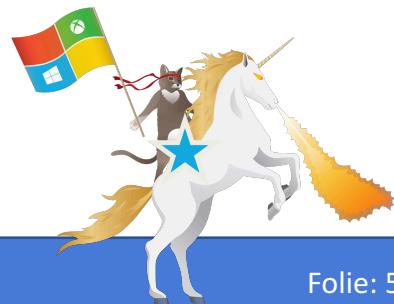


.net Standard 2.0 / .net Core 2.0

MOTIVATION – JUST WHY?



Conference recap



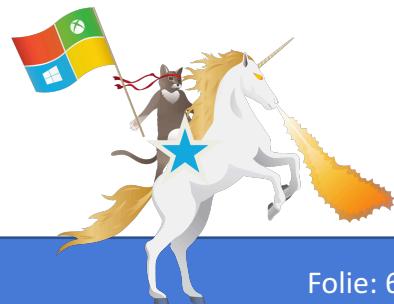
Personal Motivation



anfema

Matthias Redlin

xcode is echt so behindert ohne scheiß



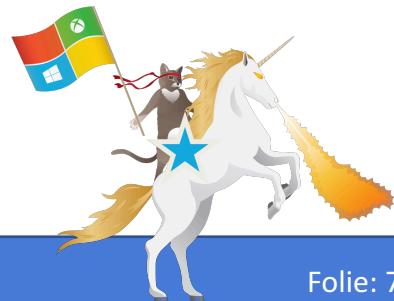
Personal Motivation

Frustration

- Swift evolution is still in progress (it's getting better)
- Xcode is very bad in backwards-compatibility
- Crashes, Black Screens, Beachball of Death
- Build times longer than a coffee break
- No end in sight
- Fast developer machines are too expensive for a community driven development

Ally against the hybrid app trend

- Everything is better than a hybrid app (imho)
- Be multi-skilled



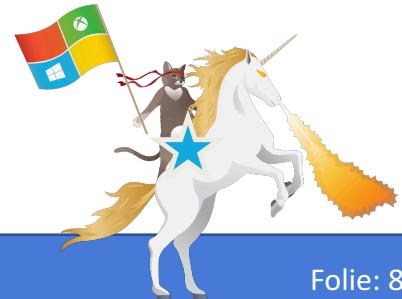
Technical Motivation

Write Once, Run Anywhere

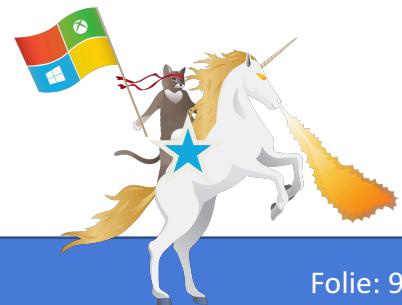
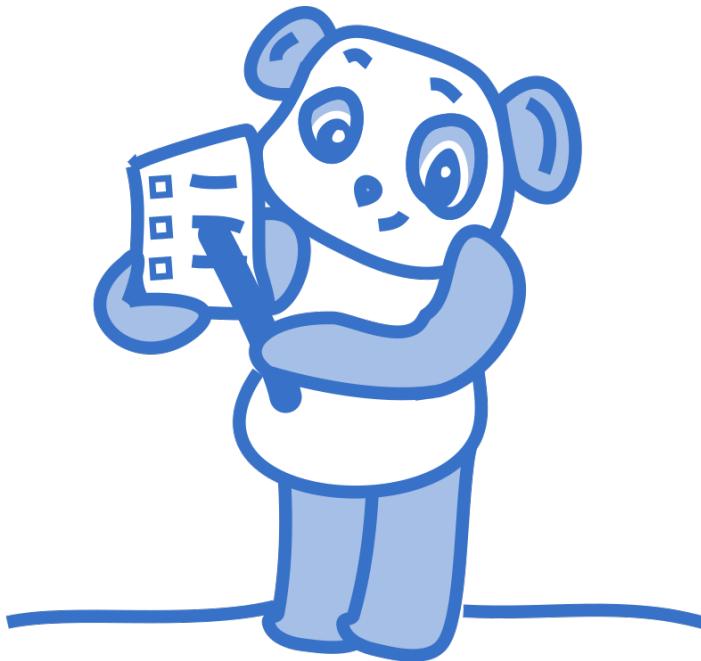
- The dream of „one for all, all for one“ development paradigm
- Working, type-safe alternative for existing types of app development
- Reduce time of development of multi-platform standard apps
- Unify employee skillset to be more exchangeable
- Enable developer to switch sites in the client / server architecture pattern
- Escape the „golden cage“ of vendor lockin’ed development ecosystems

Opens new doors into enterprises

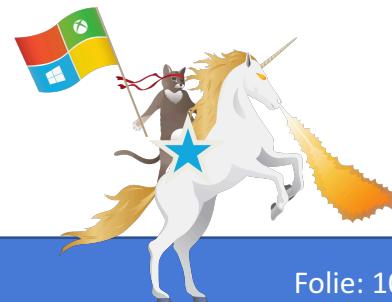
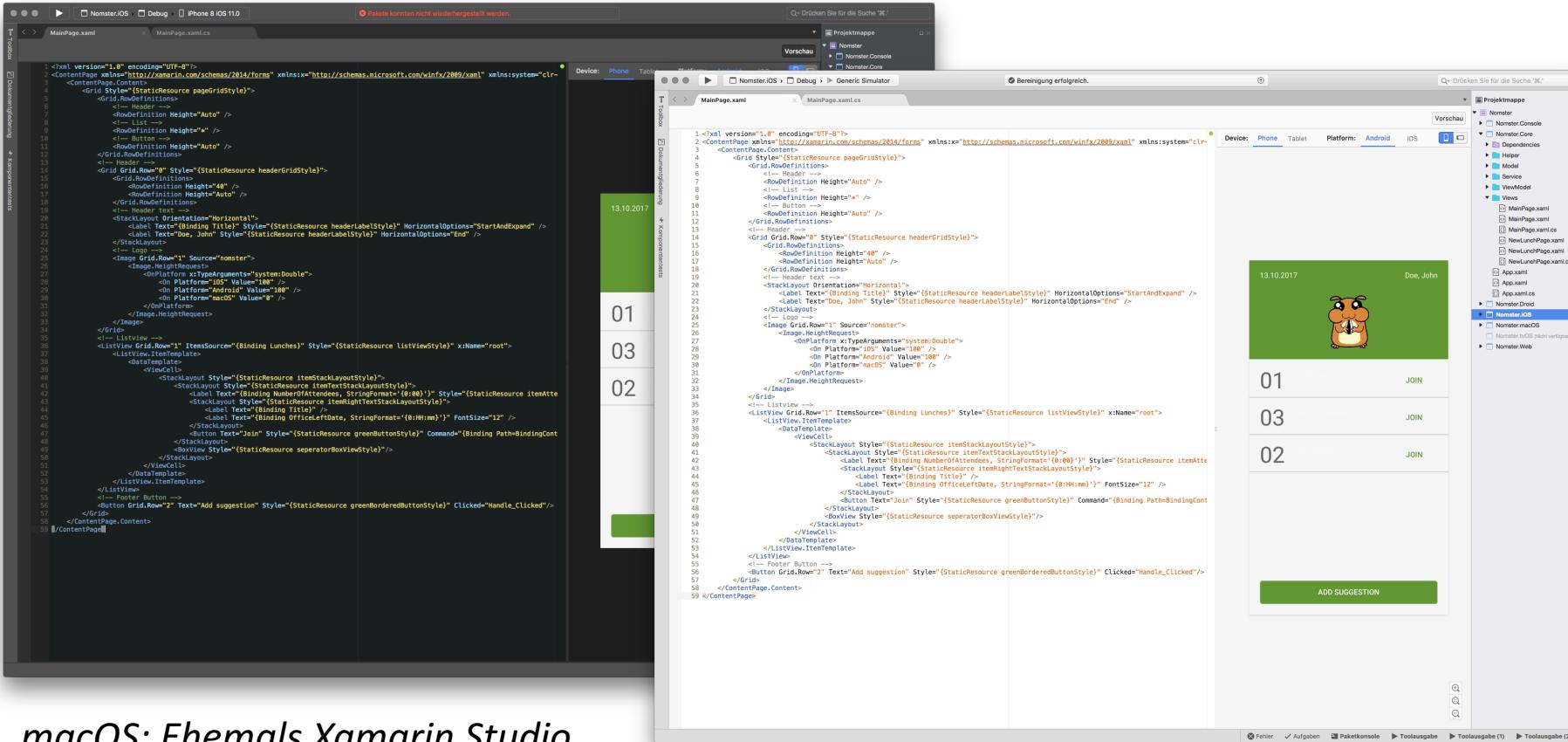
- C# and .net are very common and sometimes required for enterprise-grade development
- Easy access and usage of other Microsoft related products like Azure or Office



TECHNICAL BACKGROUND



IDE: Visual Studio (for Mac)



macOS: Ehemals Xamarin Studio

IDE: Alternativen



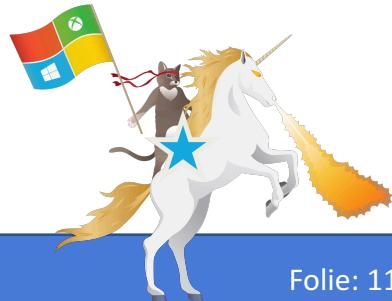
JetBrains Rider



Visual Studio Code
+ Extensions

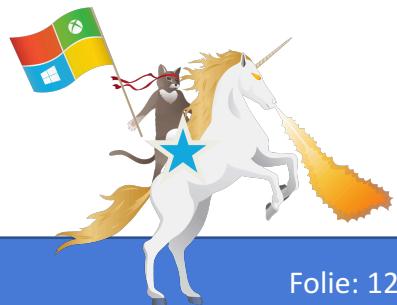


Everything else
with CLI support



Shared Code: .net Core 2.0

“.NET Core is a blazing fast, lightweight and modular platform for creating web applications and services that run on Windows, Linux and Mac”

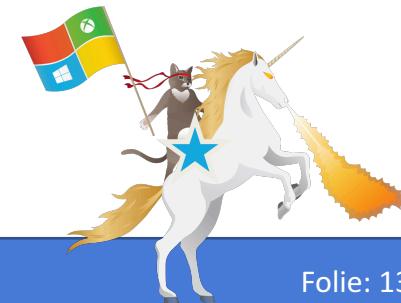


Shared Code: .net Core 2.0



- Windows, macOS, iOS, Android, Linux, etc.
- Open Source'ed on Github
- Useable in C#, F# and VB.net Bindings
- Referenceable in frameworks like asp.net Core, Xamarin and UWP
- Command line usage simplified
- Streamlined for async and block-based code

*Not **that** bad anymore*



Mobile: Xamarin

Basics

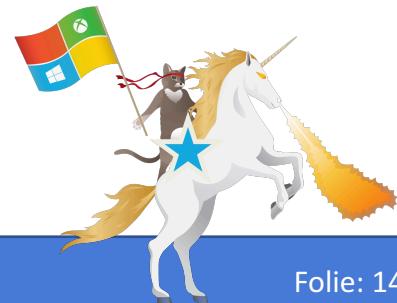
- Write mobile applications in C# or F# with the help of .net
- Share source between all platforms (write once, use often)
- Compiles to platform-specific code and binaries
- Based on the Mono Project (founded by Miguel de Icaza)
- Founded in 2011, bought by Microsoft in 2016

It's no niche

- Highly used in enterprise grade software development
- Requested by customers that are into the MSFT ecosystem
- Works perfectly with Azure Services
- Example Applications: iCircuit, Touchdraw, CodeHub (!)

Active community

- Tons of community build packages
- A lot of meetups and user groups



Mobile: Xamarin

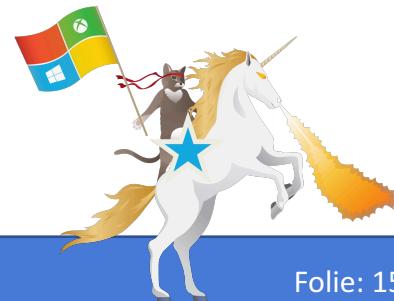


iOS / watchOS / tvOS

- Uses AOT (Ahead Of time) compilation
- 100% API coverage
- No difference in performance to Swift, Objc

Android / Android Wear / Android TV

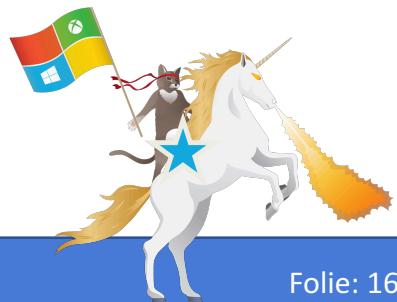
- Uses JIT (Just in time) compilation
- 100% API coverage
- No difference in performance as Java or Kotlin



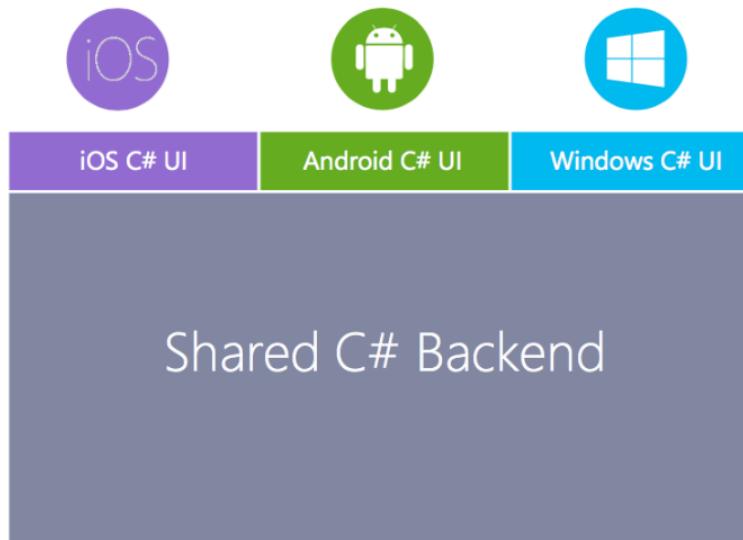
Mobile / Shared UI: Xamarin.Forms

F: What's better than shared source?

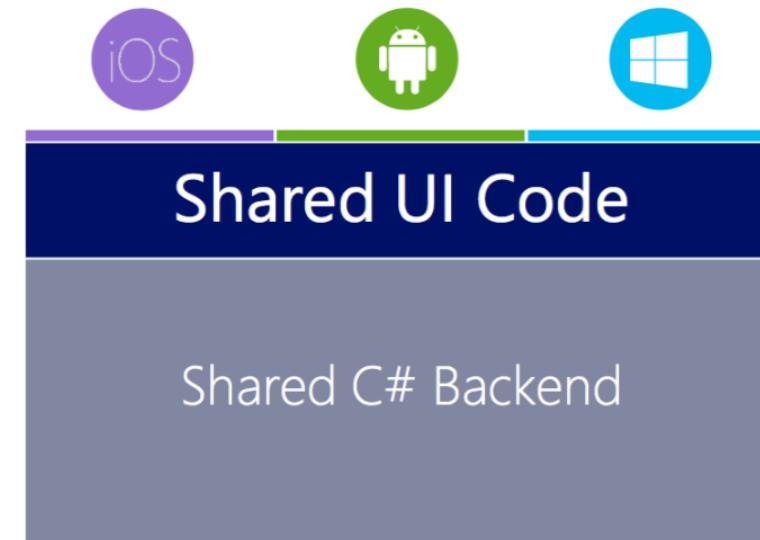
A: Shared UI definitions!



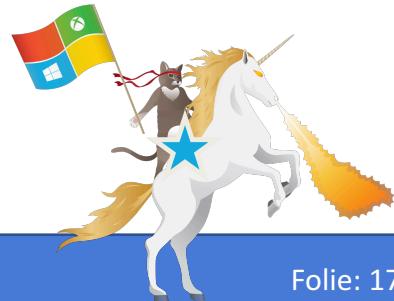
Mobile / Shared UI: Xamarin.Forms



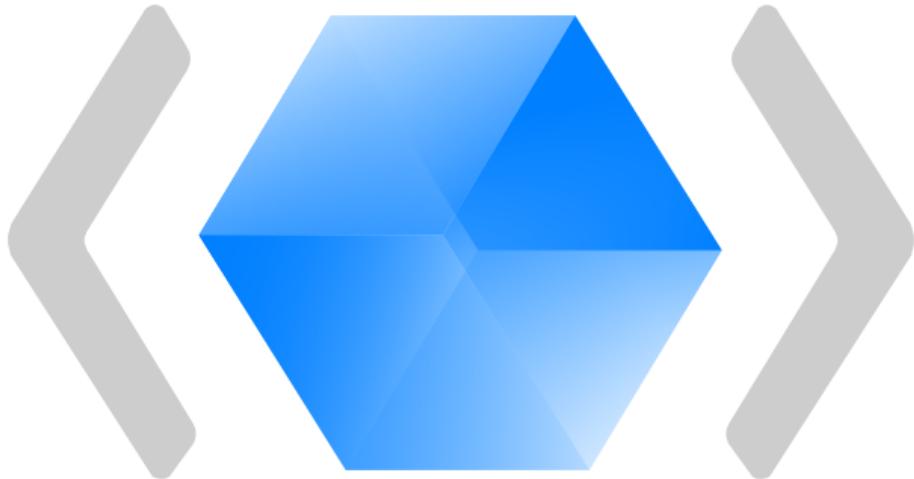
Traditional Xamarin
Approach



With Xamarin.Forms:
More code-sharing, all native

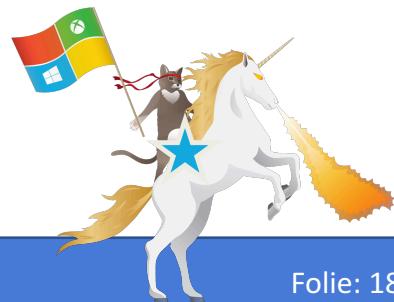


Mobile / Shared UI: Xamarin.Forms



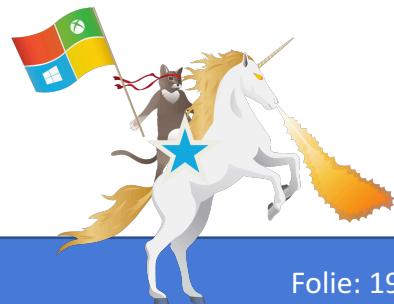
- Xaml-based UI declaration
- Works on iOS, Android, macOS and Windows
- Keep in mind: Lowest common denominator of controls
- Mixable with platform-specific code in sub projects
- 1M+ NuGet packages for custom controls and extension
- Future plans: Support for Xaml Standard 2.0 (UWP support)

It feels like web development

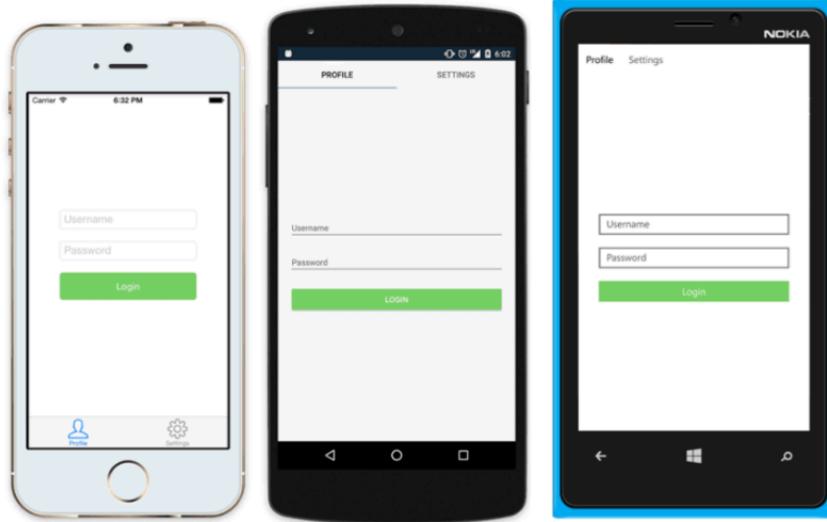


Mobile / Shared UI: Xamarin.Forms

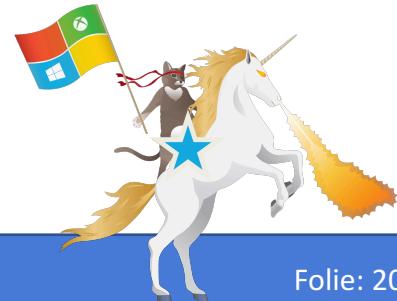
ActivityIndicator	BoxView	Button	DatePicker	Editor
Entry	Image	Label	ListView	Map
OpenGLView	Picker	ProgressBar	SearchBar	Slider
Stepper	TableView	TimePicker	WebView	EntryCell
ImageCell	SwitchCell	TextCell	ViewCell	



Mobile / Shared UI: Xamarin.Forms



```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MyApp.MainPage">
<TabbedPage.Children>
<ContentPage Title="Profile" Icon="Profile.png">
    <StackLayout Spacing="20" Padding="20"
                 VerticalOptions="Center">
        <Entry Placeholder="Username"
              Text="{Binding Username}"/>
        <Entry Placeholder="Password"
              Text="{Binding Password}"
              IsPassword="true"/>
        <Button Text="Login" TextColor="White"
               BackgroundColor="#77D0E5"
               Command="{Binding LoginCommand}"/>
    </StackLayout>
</ContentPage>
<ContentPage Title="Settings" Icon="Settings.png">
    <!-- Settings -->
</ContentPage>
</TabbedPage.Children>
</TabbedPage>
```



Web: asp.net Core 2.0

Create a new ASP.NET Core app:

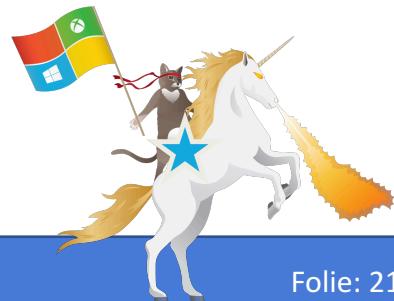
```
dotnet new razor -o aspnetcoreapp  
cd aspnetcoreapp
```

Run the app

```
dotnet run
```

- **Razor page**-based
- Ready for **Ember.js** (Community contribution)
- Full featured CLI available
- Deployable to nearly everything
- Out of the box Azure compatible
- Official Docker support
- Combineable with bower, gulp, etc. pp.

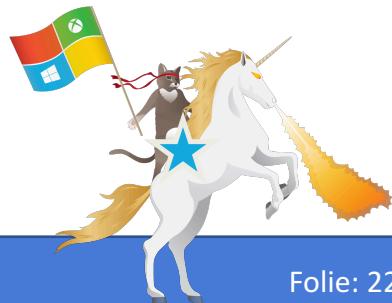
Personal winner: Razor Pages



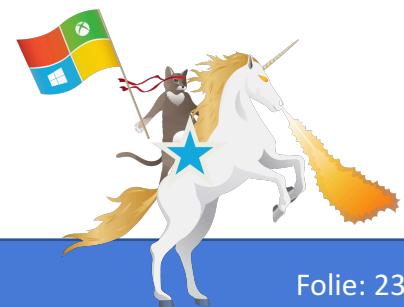
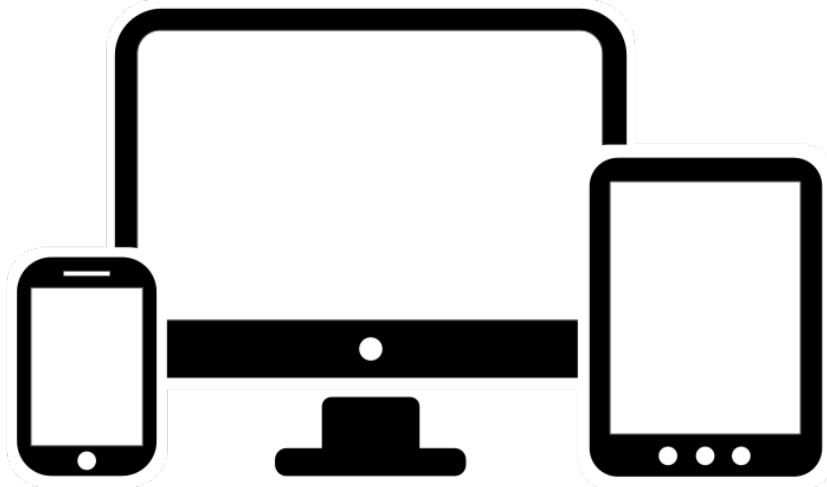
Web: asp.net Core 2.0

```
@foreach (var l in Model.viewModel.Lunches) {  
    <li>  
        <table>  
            <tr><td rowspan="3" valign="middle"><span class="attendees">@l.NumberOfAttendees</span></td></tr>  
            <tr><td></td><td><span class="title">@l.Title</span></td></tr>  
            <tr><td></td><td><span class="time">@(l.OfficeLeftDate.ToString("HH:mm"))</span></td></tr>  
        </table>  
        <hr />  
    </li>  
}
```

Feels a bit like the mixed styled in php



EXAMPLE



Example

Disclaimer

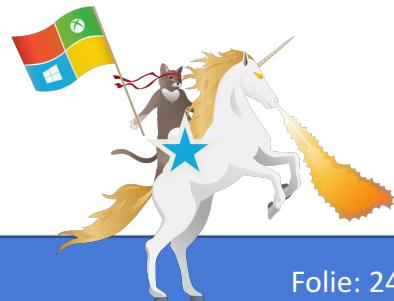
- Focus on shared UI and possibilities instead of functional logic and data management
- Ignored the „native“ Xamarin UI development
- Not very trained in the „niceness“ of C# or Xaml

Open Source

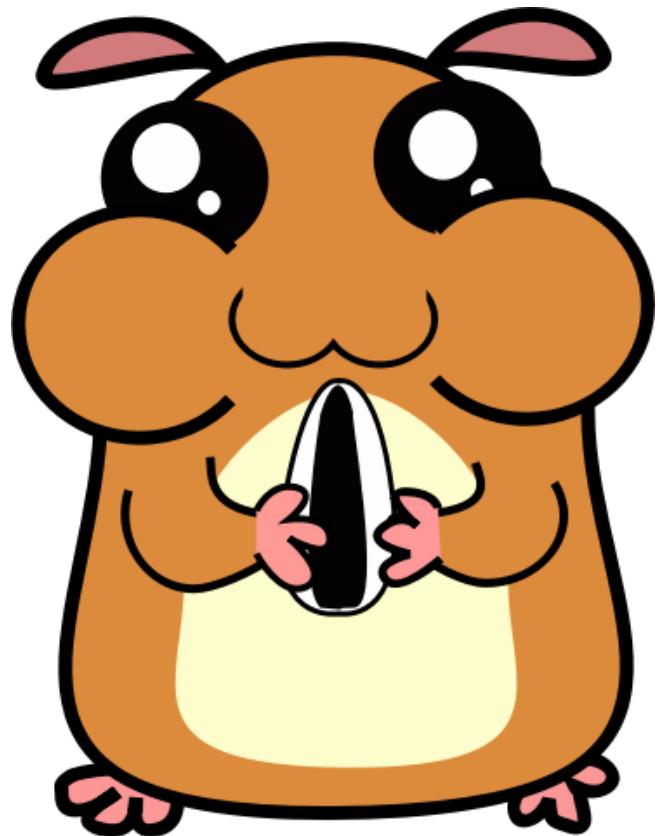
- Code is hosted on Github ([Link](#))
- No proprietary, external dependencies are used

Nevertheless, it's practical

- Maybe as an addition to our lunch trains



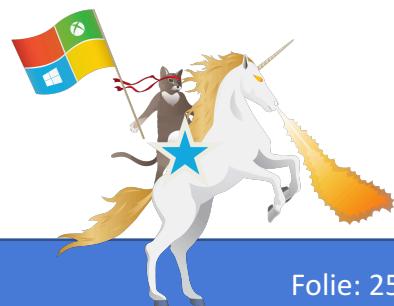
Example



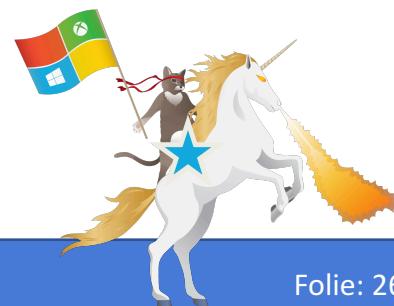
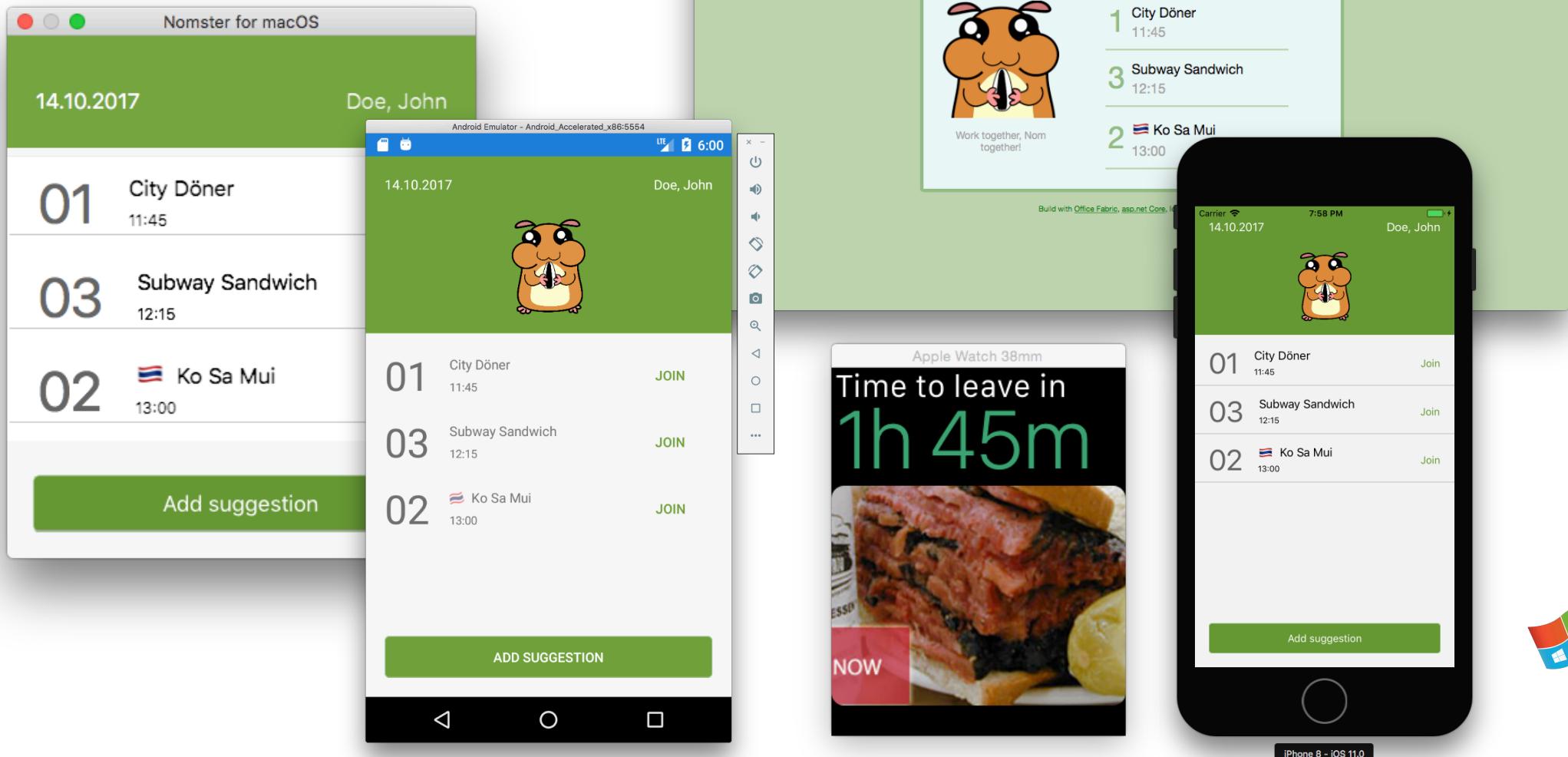
Nomster *Work together, Nom together*

- Suggest lunch options on a daily basis
- Let's you jump on existing lunch options
- Provides additional information for the lunch location

#nomnom

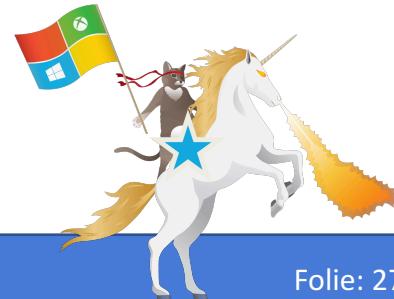
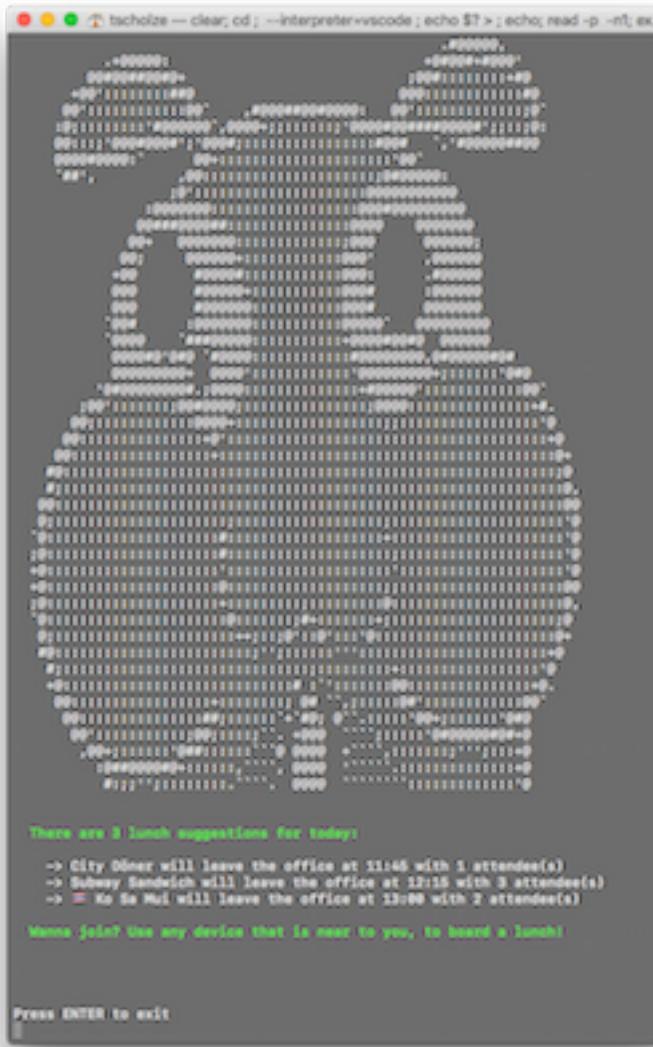


Example



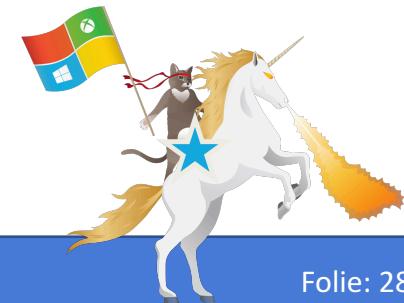
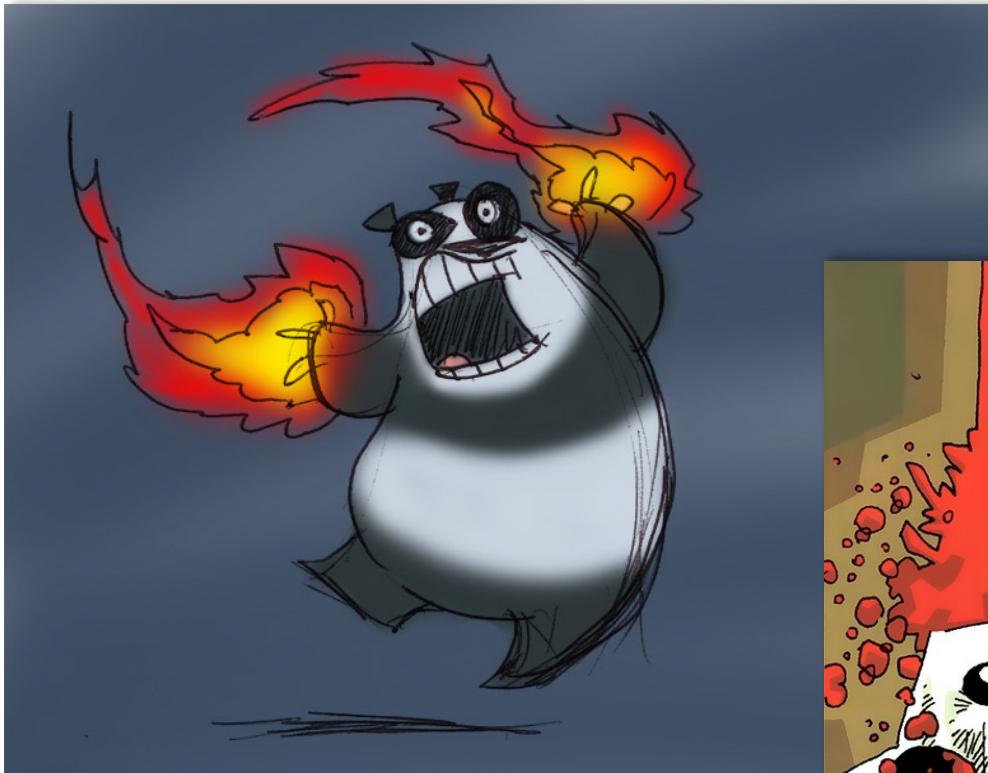
Example

... and of course
a Terminal version



LESSONS LEARNED

The very, very, very hard way



Lessons learned

Personal

- It was great to think outside my personal box
- Every ecosystem has its flaws and no one runs smooth as wished

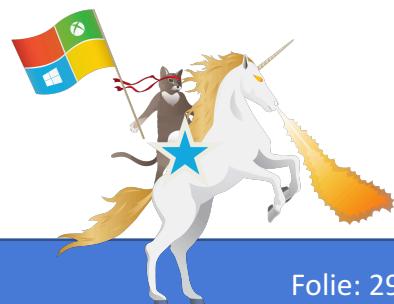
Technical

- Great to see MSFT achieved it to be a player in this game
- Looks promising (Roadmap to unify even more)
- Much more to learn to fully understand this ecosystem
- It still whets my appetite to dig deeper into this topic

Businesswise

- Simple apps with no platform specific UI could be published with a higher margin
- Developers are more interchangeable between platforms and front- or backend
- Nevertheless, needs an investment into knowledge building before even beginning of thinking to use it

„Stay hungry, Stay foolish“ – Steve Jobs



Questions & maybe some answers



- Feel free to rate me! ([Survey](#))
- Have a look at the code ([Github](#))
- German Xamarin TE: [@robinmanuelthiel](#)
- XUG Munich leader: [@stuebe2k14](#)
- Would you like to use Nomster for lunch planning?

