

Xamarin

DevDays : Azure Services

Vojtěch Mádr

<https://madrvojt.wordpress.com>

Twitter : @madrvojt



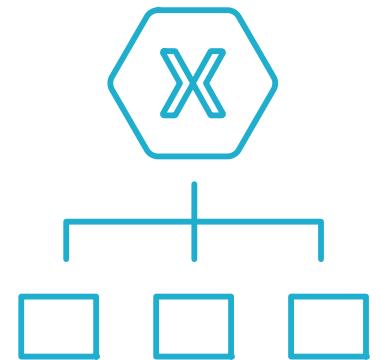
Xamarin Developers Group

Vize

- Spojení všech se zájmem o Xamarin do jedné komunity
- Stát se oficiální českou komunitou pro vývoj s Xamarin
- Společně učit a školit Xamarin
- Pomáhat při hledání projektů, či pracovních příležitostí
- Pořádat a podílet se na nejlepších Xamarin konferencích, přednáškách a komunitních setkáních
- Stát se občanským sdružením
- Pomoci společně tvořit lepší Xamarin

Plans

- Organizovat meetupy, přednášky, či workshopy
- Účastnit se větších konferencí jako mDevFest, či MSFest
- Tweetovat, tvořit výukové materiály a podcasty
- Zviditelnňovat málo známé osobnosti a firmy
- Nabízet prostor pro nové projekty a skupiny
- Tvorba výukových materiálů
- Podcast



První česká výuka

- Vytvořit první výukové materiály v češtině, učebnici
- V psané formě, podcasty, univerzitní přednášky, workshopy
- Zařazení do výuky na školách
- Propojit Best practices a výuku
- Sdružování nabídek stáží a praxí pro studenty



Partners



Střední průmyslová škola
na Proseku

Xamarin Core Approach

Android

iOS

Windows

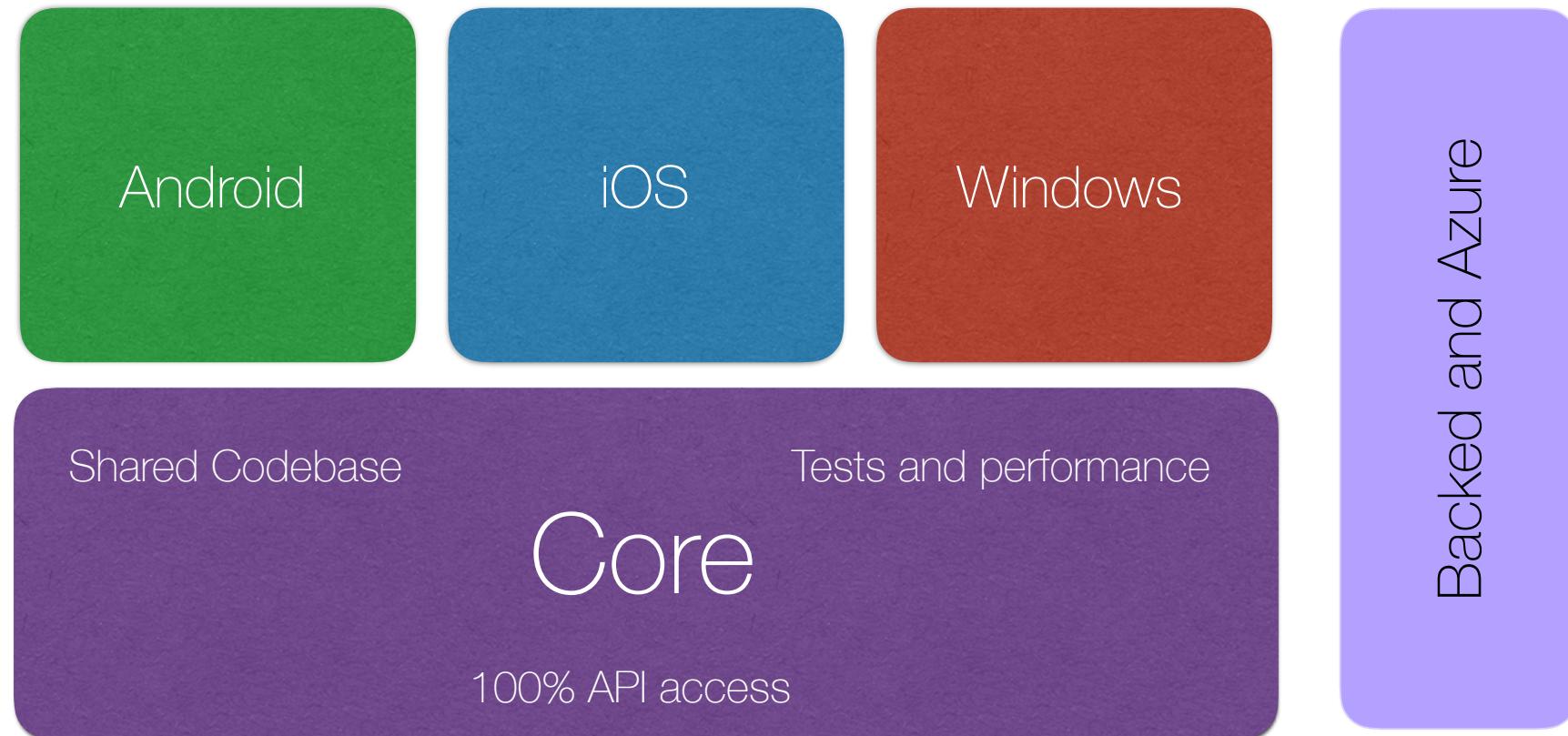
Shared Codebase

Tests and performance

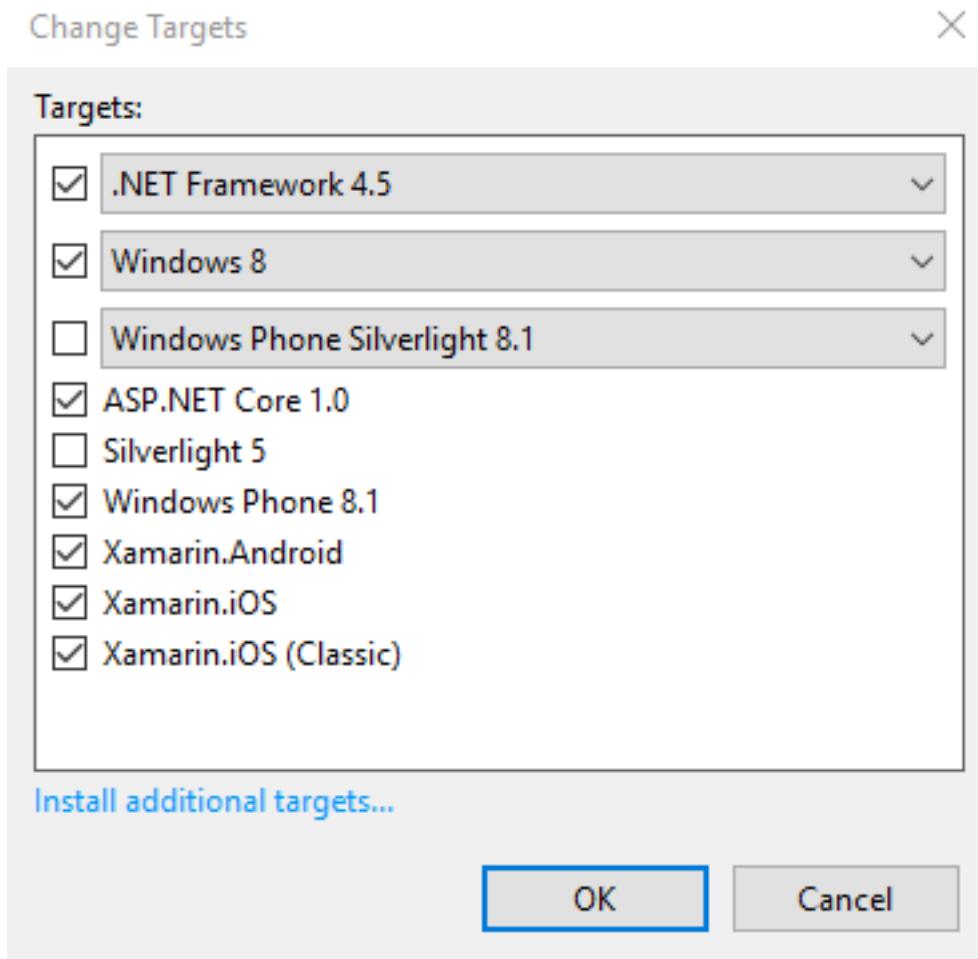
Core

100% API access

Xamarin Core Approach



Portable class libraries



One Assembly for Multiple Platforms

Inc. **Xamarin.iOS** and **Xamarin.Android**

Portable class libraries

Profile 7

Profile 78

Profile 259

.NET 4.5
Windows Store 8+
Xamarin.iOS
Xamarin.Android

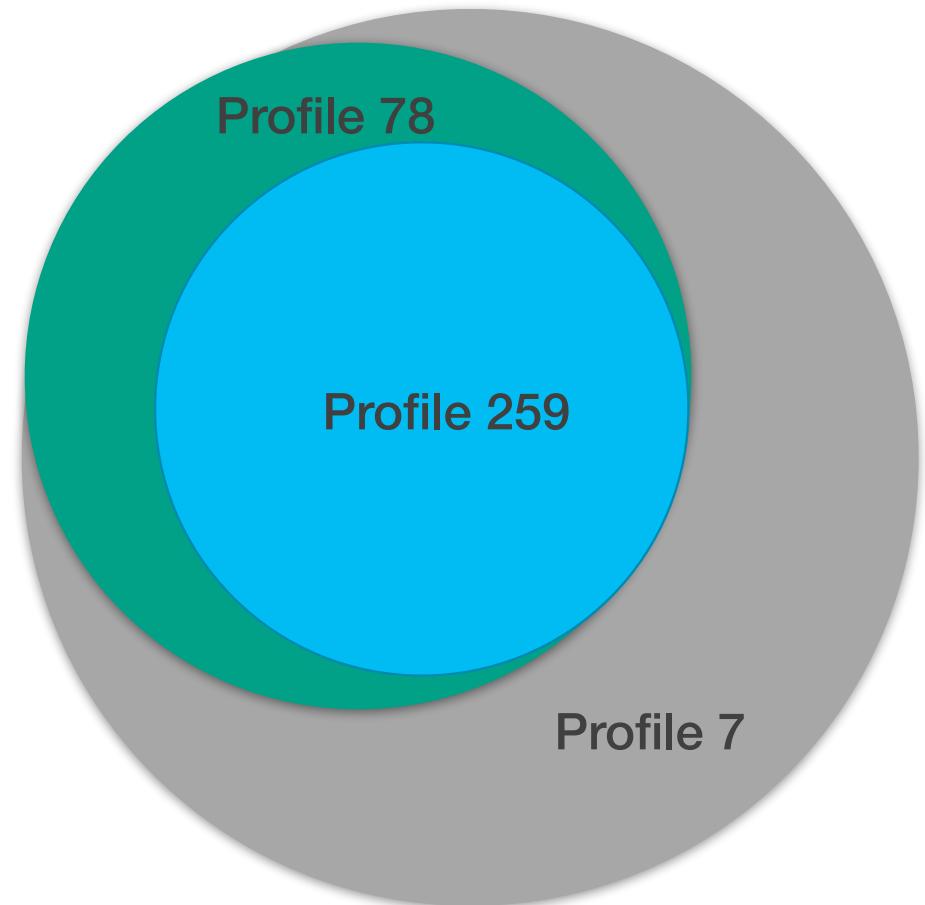
.NET 4.5
Windows Phone
Silverlight 8+
Windows Store 8+
Xamarin.iOS
Xamarin.Android

.NET 4.5
Windows Phone 8.1
Windows Phone Silverlight 8+
Windows Store 8+
Xamarin.iOS
Xamarin.Android

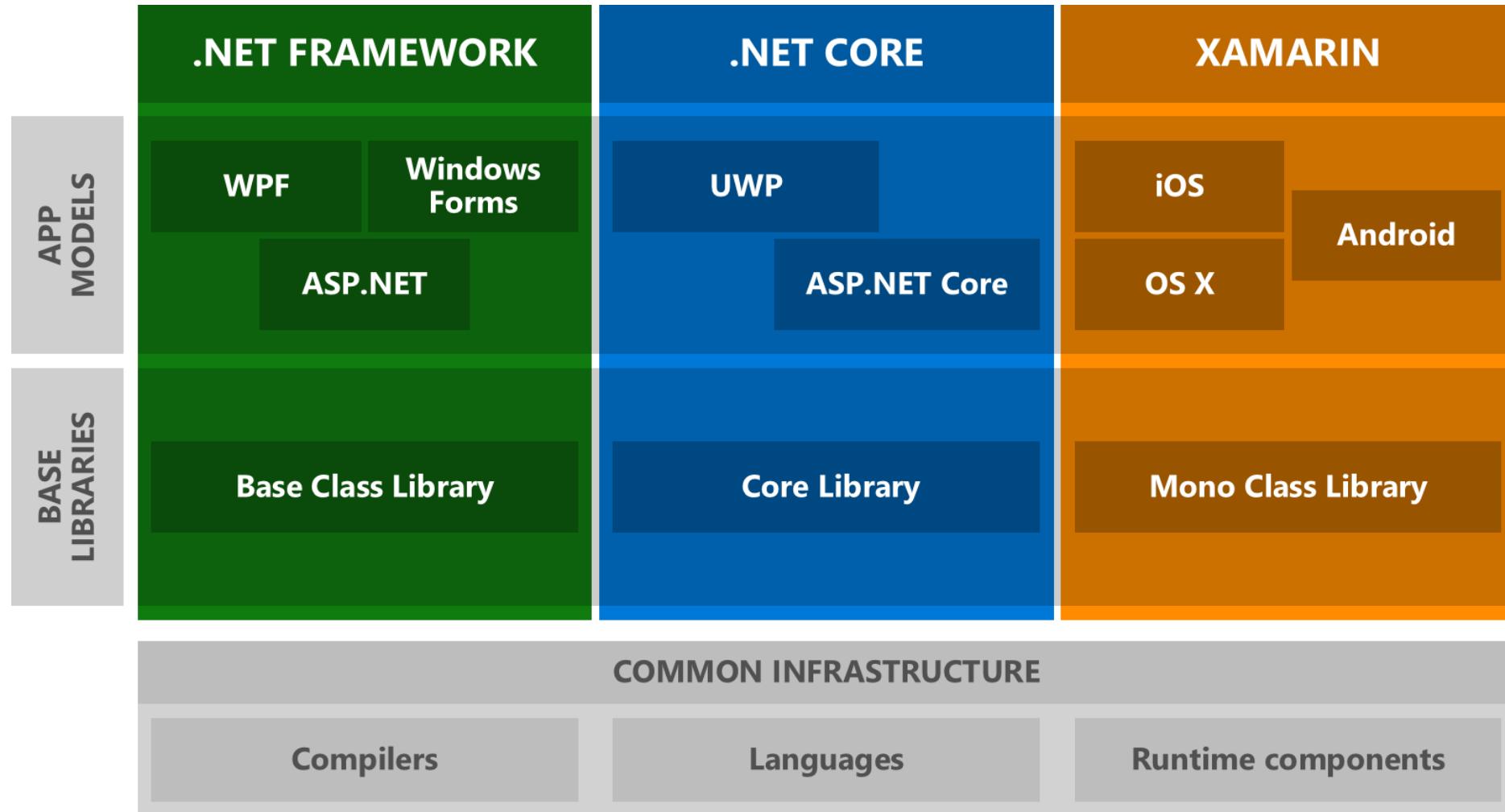
1.90 MB

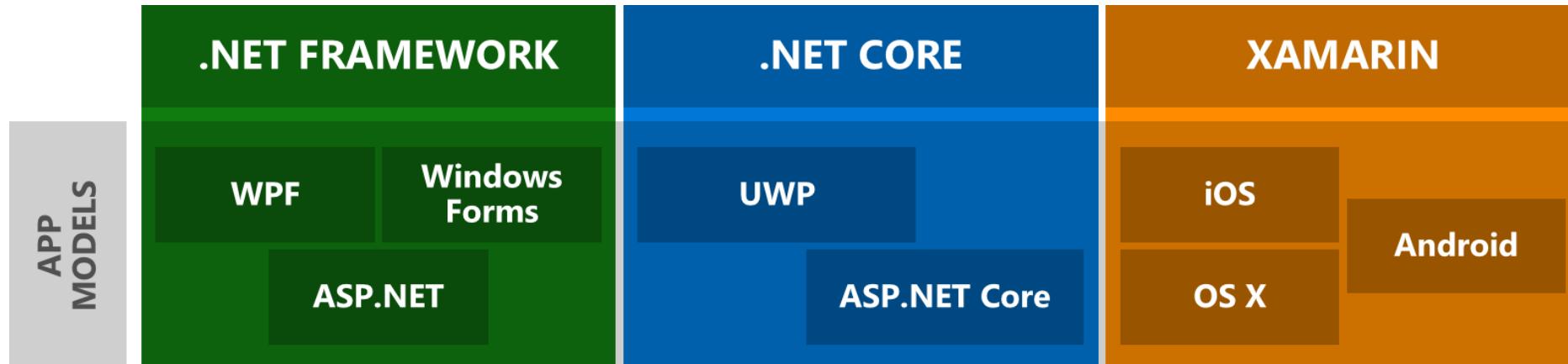
1.46 MB

1.32 MB



Welcome to .NET CORE





.NET STANDARD LIBRARY

One library to rule them all

COMMON INFRASTRUCTURE

Compilers

Languages

Runtime components

.NET Standard Platforms

PLATFORM NAME	ALIAS	1.0	1.1	1.2	1.3	1.4	1.5	1.6
.NET Standard	netstandard	1.0	1.1	1.2	1.3	1.4	1.5	1.6
.NET Core	netcoreapp	→	→	→	→	→	→	1.0
.NET Framework	net	→	4.5	4.5.1	4.6	4.6.1	4.6.2	4.6.3
Mono/Xamarin Platforms		→	→	→	→	→	→	*
Universal Windows Platform	uap	→	→	→	→	→	10.0	
Windows	win	→	8.0	8.1				
Windows Phone	wpa	→	→	8.1				
Windows Phone Silverlight	wp	8.0						

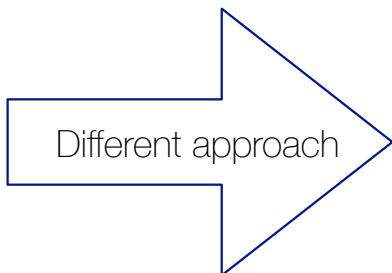
Shared project



Compile separately



PCL or .NET Standard
with own libraries



Compile with application

Shared Project

PCL or Shared Project?

Which is right for you?

Posted by Jason Smith on January 19, 2016

Since the dawn of time man has been faced with one question. Should I use PCL's libraries for my Xamarin.Forms projects, or should I use a Shared project? I'm here to tell you the answer is PCL, it is the way, the truth, and the light.

Friends don't let friends use shared projects.

Okay so that's a bit strong but in general if you don't know what you should do, go PCL. If you have a strong reason to use a shared project, sure, but otherwise go PCL, your lack of #ifdef and spaghetti code will thank me later. Among other things, PCL will ensure that code you write is going to be portable not just to all current platforms, but any future platforms we might support as well.

Also I want to make sure everyone knows PCL is pronounced Pickle. Thats all.

zdroj č.1

Shared Projects or PCL?

Posted on 22 Jan 2016 by Miguel de Icaza

My colleague Jason Smith has shared [his views](#) on what developers should use when trying to share code between projects. Should you go with a Shared Project or a Portable Class Library (PCL) in the world of Xamarin.Forms?

He hastily concludes that you should go with PCLs (pronounced Pickles).

For me, the PCL is just too cumbersome for most uses. It is like using a canon to kill a fly. It imposes too many limitations (limited API surface), forces you to jump through hoops to achieve some very basic tasks.

PCLs when paired with Nuggets are unmatched. Frameworks and library authors should continue to deliver these, because they have a low adoption barrier and in general bring smiles and delight to their users.

But for application developers, I stand firmly on the opposite side of Jason.

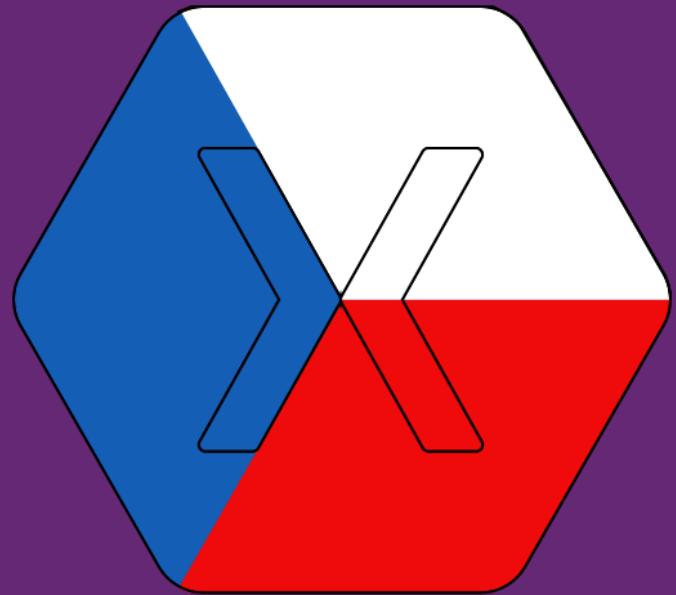
I am a fan of simplicity. The simpler the technology, the easier it is for you to change things. And when you are building mobile applications chances are, you will want to make sweeping changes, make changes continuously and these are just not compatible with the higher bar required by PCLs.

Jason does not like `#if` statements on his shared code. But this is not the norm, it is an exception. Not only it is an exception, but careful use of `partial` classes in C# make this a non issue.

Plugging a platform specific feature does not to use an `#if` block, all you have to do is isolate the functionality into a single method, and have each platform that consumes the code implement that one method. This elegant idea is the same elegant idea that makes the Linux kernel source code such a pleasure to use - specific features are plugged, not `#ifdefed`.

If you are an application developer, go with Shared Projects for your shared code.
And now that we support this for F#, there is no reason to not adopt them.

zdroj č.2



Join us

<https://www.meetup.com/xmdg-prague/>

Questions?