



iOS, Android and Windows 10 Easy and Fast

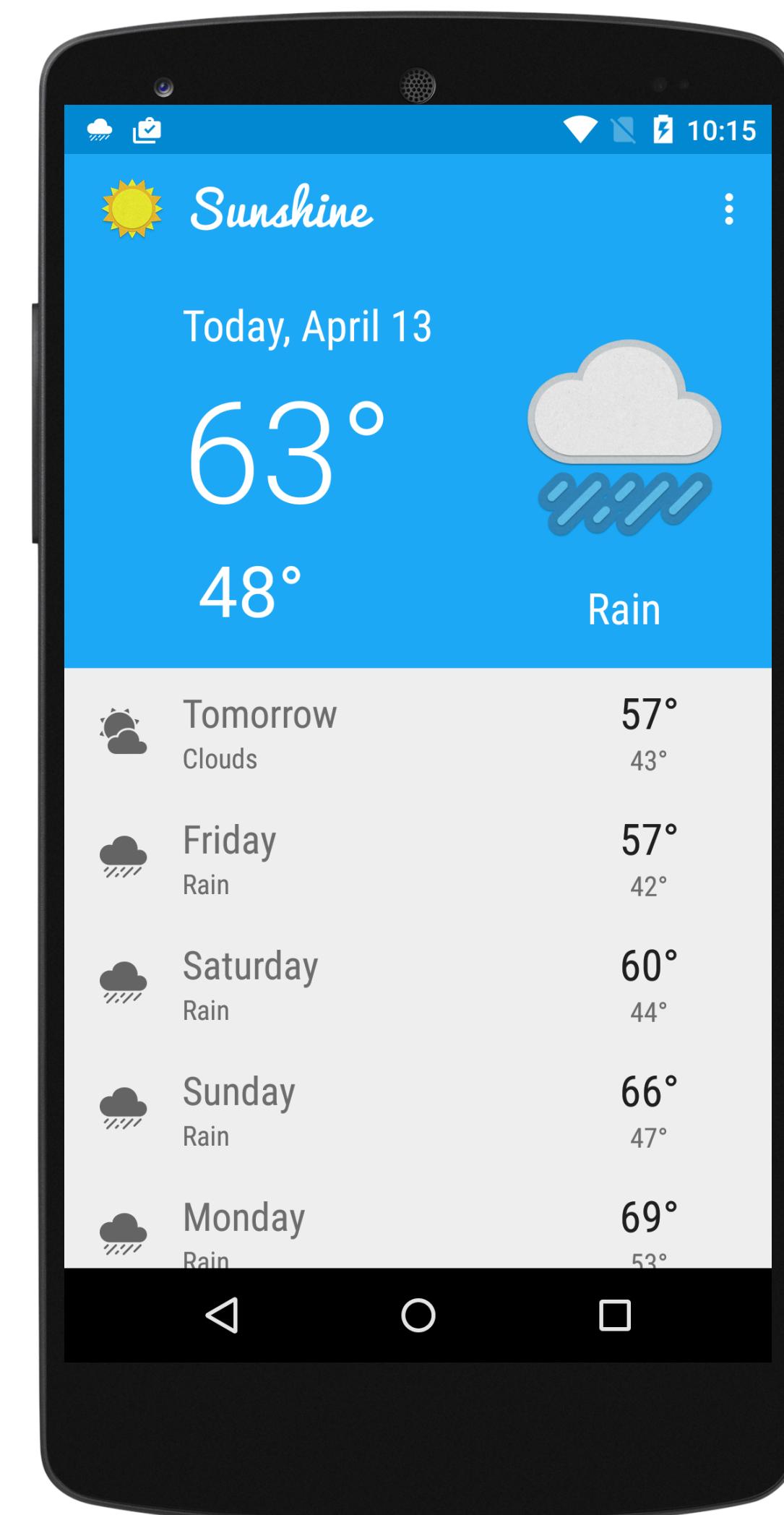
Vojtěch Mádr



Is Xamarin Native? Try Sunshine!

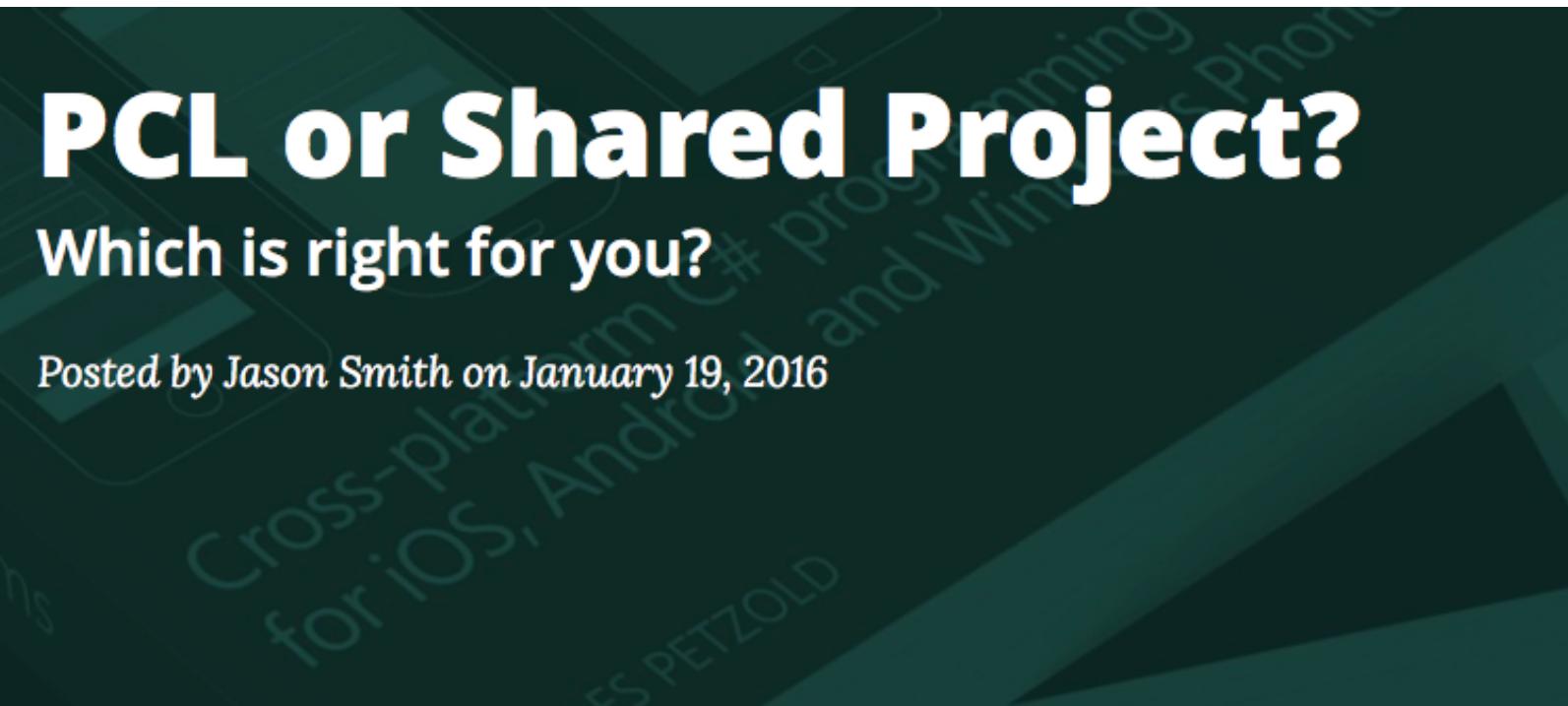
<https://github.com/madvojt/Sunshine-Xamarin>

<https://rink.hockeyapp.net/apps/b632892031c04a5cb7aecc6452a0b1e4>



Common core technique





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 Nugets 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

Xamarin Core

Android

iOS

Windows

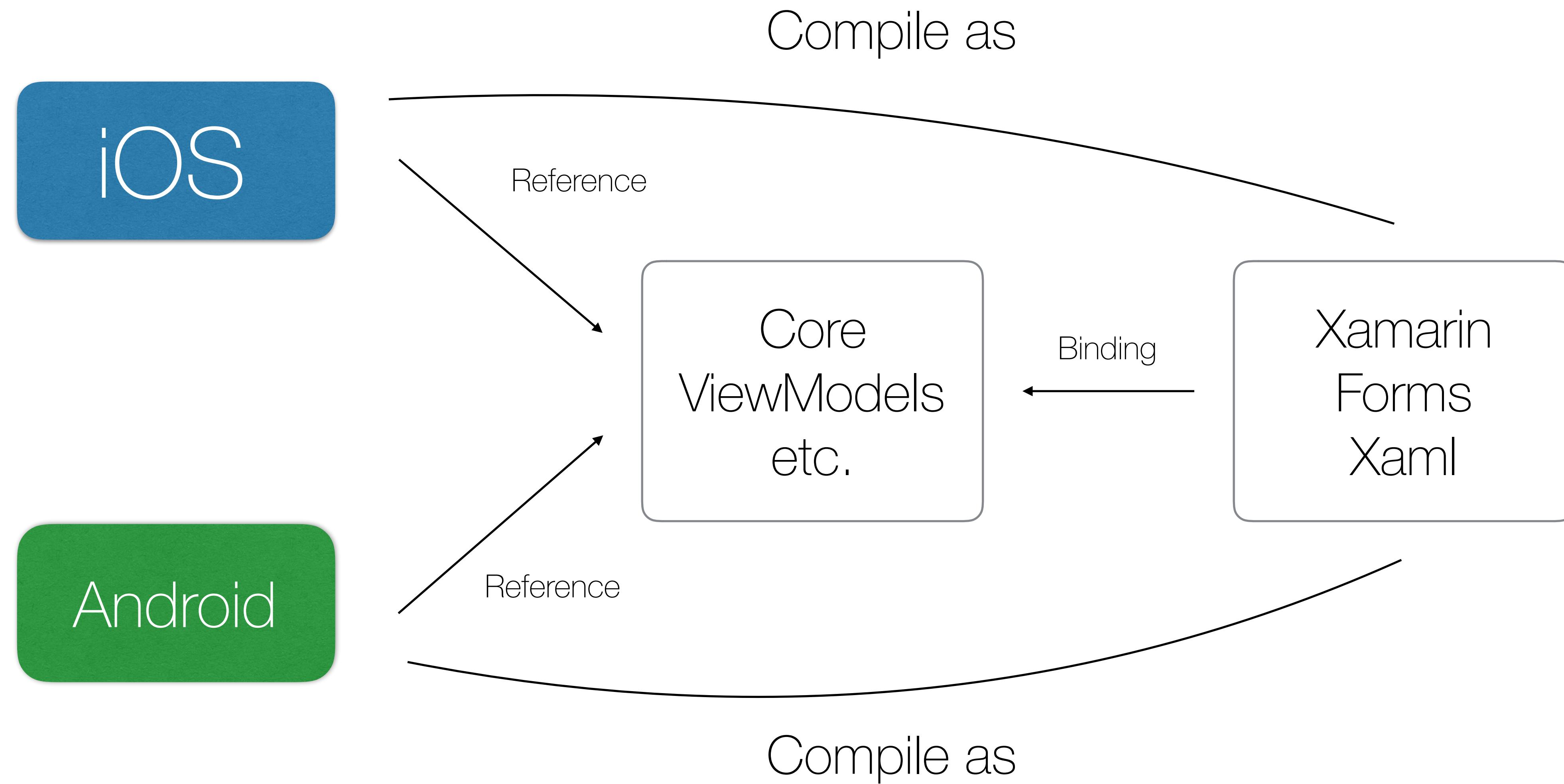
Core

Xamarin Forms

All Platforms View

ViewModel

Core





Prague DevOps Meetup

<https://madrvojt.wordpress.com>



<https://www.eman.cz/blog/>

twitter : @madrvojt

email : madrvojtech@outlook.com



Thanks for watching

www.eman.cz