



**NMCT**

## **Domotica**

Xamarin Cross Platform Mobile App Dev.

# Xamarin



# Xamarin

- Plattformen
  - Apple (Swift/Objective C)
    - iOS
      - iPhone
      - iPad
    - Watch OS
      - Apple Watch
    - tvOS
      - Apple Tv
  - Google (Java)
    - Android
      - Smartphones
    - Android Wear
      - Smartwatches

# Xamarin

- Microsoft (C# UWP)
  - Windows 10
    - Desktop
    - Tablet
    - Phone
    - Xbox
    - IoT Core

# Xamarin

- Conclusie
  - 3 platformen
    - 3 x ontwikkelen
    - 3 x testen
    - 3 x bugfixen
    - 3 x updaten
    - 3 x verschillende competenties bij werknemers (Swift/Java/.NET)
    - 3 x zo duur

# Xamarin

- Oplossing
  - Xamarin (.NET C#)
  - Cordova (HTML/CSS/JS)
  - Appcelerator Titanium (HTML/CSS/JS)
  - ..
- 1 code base => meerdere platformen is doelstelling
  - Niet altijd zo
  - Veel valkuilen
  - Blijft lasting verhaal

# Xamarin

- Wij bekijken Xamarin
  - U mag ook iets anders kiezen

# Xamarin

- Xamarin
  - 3 platformen
    - Xamarin.iOS
    - Xamarin.Android
    - Xamarin Windows
  - 3 x UI ontwikkelen
  - 1 x backend
  - Alles in C# of F#
  - Je moet de platform eigenschappen kennen (ervaring met Swift/Java/C# is zeer handig)
  - Meeste native ervaring





# Xamarin

## Xamarin.Forms

- 1 x UI ontwikkelen
  - XAML
  - Xamarin Forms zal look en feel toepassen van OS
- 1 x backend
- Enkel OS specifieke zaken kan je niet delen
  - Push/Camera/...
- Wij kiezen voor Xamarin.Forms



# Xamarin

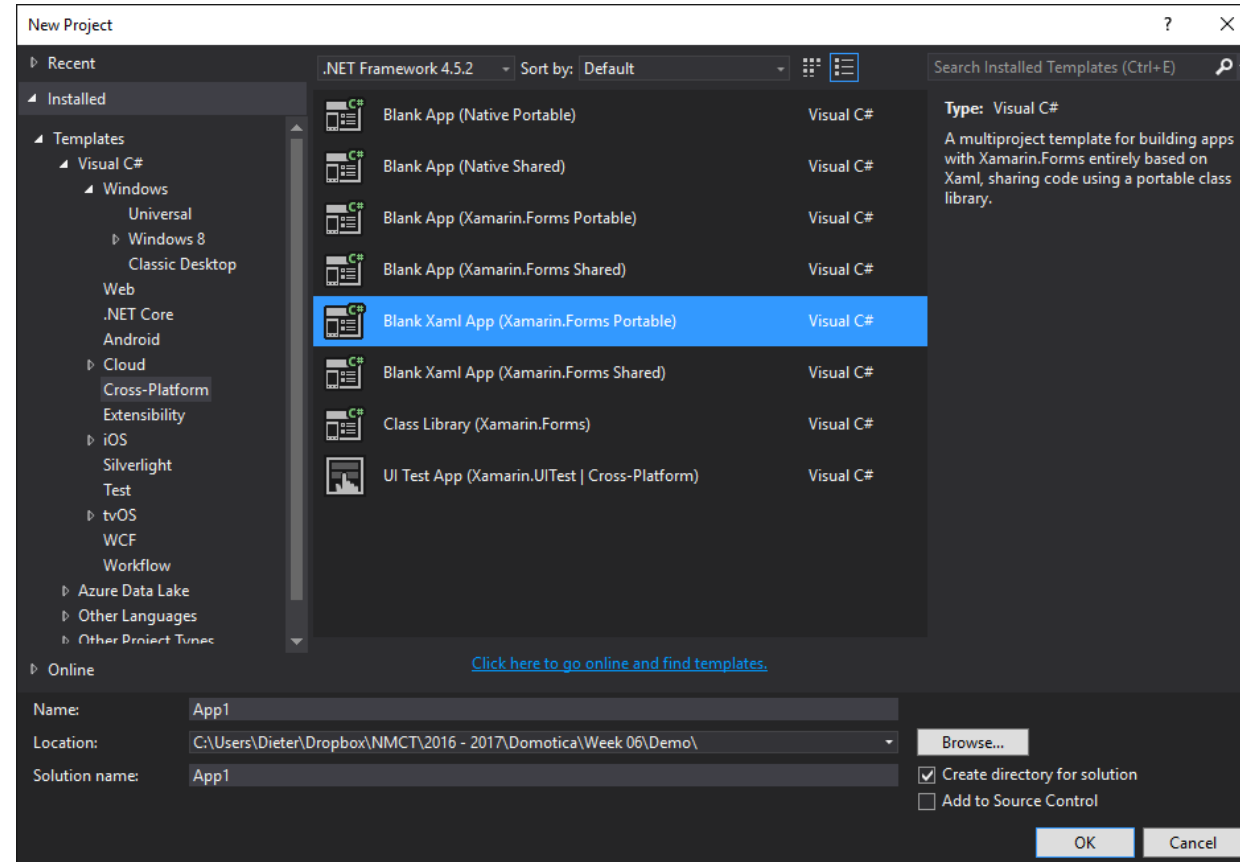
- Wat heb je nodig ?
  - iOS
    - Mac of PC (Deployment enkel via Mac mogelijk, iOS Simulator wel mogelijk op Windows mits een mac in network)
    - Op de Mac moet Xcode staan
    - Xamarin Studio / Visual Studio 2015
  - Android
    - Mac of Pc
    - Android SDK
    - Xamarin Studio/Visual Studio 2015
  - Windows
    - PC
    - Windows 10 SDK
    - Xamarin Studio/Visual Studio 2015

# Xamarin

- Testen
  - iOS
    - Microsoft iOS Emulator (Windows , maar Mac in network)
    - Apple iOS (OSX)
    - Toestel (hiervoor Apple Dev. Account nodig \$\$)
  - Android
    - Microsoft Android Emulator (Windows)
    - Google Android Emulator (Windows en OSX)
    - Toestel (geniet voorkeur)
  - Windows
    - Microsoft Windows Phone Simulator (Windows)
    - Toestel (geniet voorkeur)

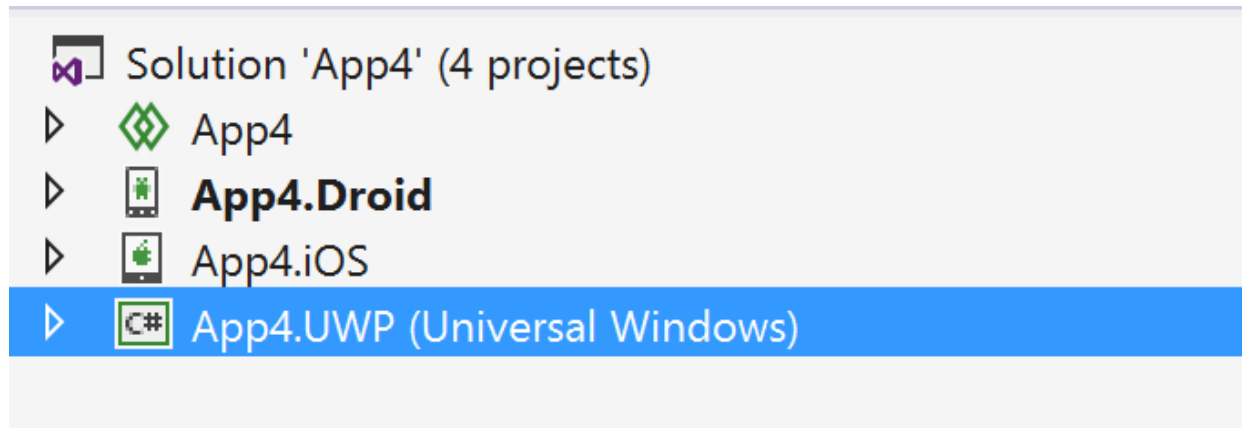
# Xamarin

- Nieuw project

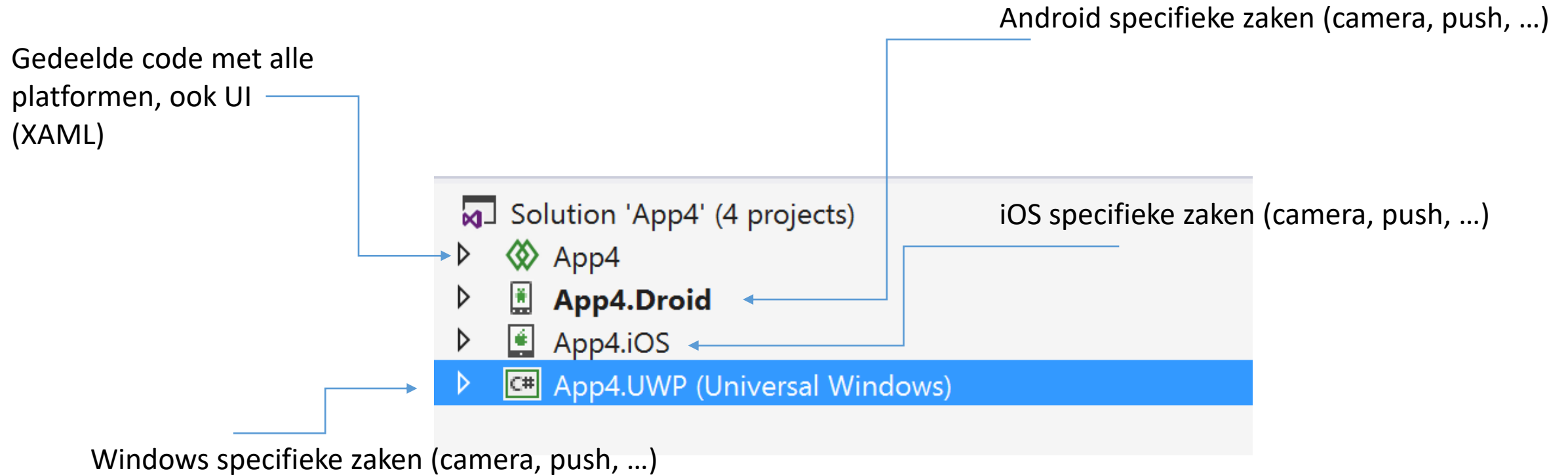


# Xamarin

- Hangt af van de geïnstalleerde SDK's welke projecte je ziet



# Xamarin



# Xamarin

- Leercuue
  - Wat moet je kennen ?
    - C# ✓
    - .NET ✓
    - XAML ✓
    - Platform specifieke zaken
      - Camera aanspreken
      - Push
      - HTTP Client ✓
      - Bluetooth
      - ...
  - Documentatie
    - Quickstarts
    - Veel code examples
    - Mooie community
  - Learn by doing

# Xamarin

- Quickstart 1 (Samen)
  - <https://developer.xamarin.com/guides/xamarin-forms/getting-started/hello-xamarin-forms/quickstart/>



# Xamarin

- Basis is dependency injection
  - Interface definiëren (IDial in dit voorbeeld) in het “shared project”
  - Per platform implementatie van IDial

```
[assembly: Dependency(typeof(PhoneDialer))]  
namespace Phoneword.Droid {  
    1 reference  
    public class PhoneDialer : IDialer {  
        0 references  
        public bool Dial(string number) {  
            var context = Forms.Context;  
            if (context == null)  
                return false;  
  
            var intent = new Intent(Intent.ActionCall);  
            intent.SetData(Uri.Parse("tel:" + number));  
  
            if (IsIntentAvailable(context, intent)) {  
                context.StartActivity(intent);  
                return true;  
            }  
  
            return false;  
        }  
    }  
}
```

Registratie in Dependency Container

```
[assembly: Dependency(typeof(PhoneDialer))]  
namespace Phoneword.iOS {  
    1 reference  
    public class PhoneDialer : IDialer {  
        0 references  
        public void Demo(string fromOs) {  
            UIAlertView _error = new UIAlertView("bericht", fromOs, null, "Ok", null);  
            _error.Show();  
        }  
    }  
}
```

# Xamarin

- Aanroepen methode in shared project
  - DependencyService.Get<... Interface ...>

```
0 references
async void OnCall(object sender, EventArgs e) {
    if (await this.DisplayAlert(
        "Dial a Number",
        "Would you like to call " + translatedNumber + "?",
        "Yes",
        "No")) {
        var dialer = DependencyService.Get<IDialer>();
        if (dialer != null)
            dialer.Dial(translatedNumber);
    }
}
```

# Xamarin

- <https://developer.xamarin.com/guides/xamarin-forms/user-interface/navigation/>



ContentPage



MasterDetailPage



NavigationPage



TabbedPage



CarouselPage

# Xamarin

- Controls
  - The Usual Suspects
    - Button
    - Textbox
    - Listview
    - ...
  - Extra controls
    - Maps
    - Webview
    - TableView
    - ...

# Xamarin

- XAML
  - Kennen we van WPF & Windows UWP
  - Databinding aanwezig, zelfde manier van werken
  - Styling aanwezig, ongeveer zelfde manier van werken

# Xamarin

- Webservices
  - Gebruik HTTP Client
  - Gebruik JSON.NET

# Xamarin

- MVVM
  - Aan te raden manier voor apps die lang onderhoudbaar moeten zijn
  - Valt buiten de scope van dit vak
  - Indien iteresse
    - [https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data\\_bindings\\_to\\_mvvm/](https://developer.xamarin.com/guides/xamarin-forms/xaml/xaml-basics/data_bindings_to_mvvm/)
    - MVVM Light
    - MVVM Cross
  - Geen verplichting voor project !!

- <https://developer.xamarin.com/guides/xamarin-forms/xaml/>

The screenshot shows the Xamarin developer website. The top navigation bar includes links for Products, Customers, Pricing, Developers (selected), Support, and Resources. A secondary navigation bar contains links for Welcome, Guides, Recipes, APIs, Samples, Forums, Components, and Videos, along with a search bar. On the left sidebar, the 'Xamarin.Forms' section is expanded, and the 'XAML' link is highlighted. The main content area displays the title 'eXtensible Application Markup Language (XAML)' and an introductory paragraph. Below this, there are links to 'XAML Basics', 'XAML Compilation', 'XAML Previewer', 'XAML Namespaces', and 'Passing Arguments'. A right-hand sidebar offers a 'PDF for offline use' download link and feedback buttons ('I love it!', 'I have a problem.').

Xamarin

Products Customers Pricing Developers Support Resources

Welcome Guides Recipes APIs Samples Forums Components Videos Search...

Cross-Platform

Android

iOS

Mac

Test Cloud

Insights

Xamarin.Forms

Getting Started

Controls Reference

**XAML**

XAML Basics

XAML Compilation

XAML Previewer

XAML Namespaces

Passing Arguments

Bindable Properties

Attached Properties

Resource Dictionaries

User Interface

DataPages

Themes

Platform Features

Working with...

Behaviors

Custom Renderers

DependencyService

Efficiency

MessagingCenter

Xamarin.Forms

## eXtensible Application Markup Language (XAML)

XAML is a declarative markup language that can be used to define user interfaces. The user interface is defined in an XML file using the XAML syntax, while runtime behavior is defined in a separate code-behind file.

[XAML Basics](#)

XAML allows developers to define user interfaces in Xamarin.Forms applications using markup rather than code. XAML is never required in a Xamarin.Forms program but it is toolable, and is often more visually coherent and more succinct than equivalent code. XAML is particularly well suited for use with the popular Model-View-ViewModel (MVVM) application architecture: XAML defines the View that is linked to ViewModel code through XAML-based data bindings.

[XAML Compilation](#)

XAML can be optionally compiled directly into intermediate language (IL) with the XAML compiler (XAMLC). This article describes how to use XAMLC, and its benefits.

[XAML Previewer](#)

The [XAML Previewer](#) announced at Xamarin Evolve 2016 is available for testing in the Alpha channel.

[XAML Namespaces](#)

XAML uses the `xmlns` XML attribute for namespace declarations. This article introduces the XAML namespace syntax, and demonstrates how to declare a XAML namespace in order to access a type.

[Passing Arguments](#)

XAML can be used to pass arguments to non-default constructors or to factory methods. This article demonstrates using

PDF for offline use:  
[Download PDF](#)

Let us know how you feel about this.

last updated: 15 days ago



# Xamarin

- Opdracht
  - Maak een Tabed Navigation Page Application
  - Op Tab 1: opvragen van alle temperatuur registraties
  - Op Tab 2: temperatuur doorsturen
  - We maken gebruik van volgende webservice uit de vorige lessen:
    - <http://domotica-nmct.azurewebsites.net/api/pi>
    - Zie klasse TemperatuurData voor ontvangen en versturen van data
    - PiName is uw eigen unieke naam














```
5 references
public class TemperatuurData {
    1 reference
    public decimal Value { get; set; }
    1 reference
    public string Sensor { get; set; }
    1 reference
    public string PiName { get; set; }
}
```

# Xamarin

- Wat met cross platform hardware features ?
  - Foto/video
  - Push
  - Bluetooth
  - ...
- Je kan zelf implementeren in platform project en via DI aanspreken in het shared project
- Meestal wel een plugin beschikbaar
  - Vb xam.plugins....
  - Bekijk zelf een <https://github.com/jamesmontemagno/MediaPlugin>
  - Probeer een foto te nemen en deze weer te geven in de applicatie
  - Plugins zoveel mogelijk toevoegen via
    - Nuget
    - Components

# Xamarin

- Tips
  - Herstarten Visual Studio kan veel helpen
  - Clean & Build
  - Bij Android
    - Check goed welke API je target (versie van Android)
  - Als je functionaliteit nodig hebt zoek ALTIJD
    - Is er een nuget package?
    - Is er een component?
  - Voor je project check zeker xam.plugins....

	<b>Xam.Plugin.Connectivity</b> by James Montemagno, 198K downloads Get network connectivity information such as network type, speeds, and if connection is available.	v2.2.12 
	<b>Xam.Plugin.ExternalMaps</b> by James Montemagno, 74,3K downloads Open external maps to navigate to a specific geolocation or address.	v3.0.0
	<b>Xam.Plugin.Geolocator</b> by James Montemagno, 55,5K downloads Easily access geolocation across Xamarin.iOS, Xamarin.Android and Windows. View full project page for README	v3.0.4
	 <b>Xam.Plugin.Media</b> by James Montemagno, 54,7K downloads Take or pick photos and videos from a cross platform API.	 v2.6.0
	<b>Xam.Plugin.EmbeddedResource</b> by Joseph Hill, 29,6K downloads Mobile applications often need to bundle files with the app, such as a SQLite database or static HTML or images, that will later be accessed from the file system.	v1.0.1
	<b>Xam.Plugin.DeviceInfo</b> by James Montemagno, 27,7K downloads Get device information such as Model, OS, and other properties.	v2.0.2
	<b>Xam.Plugin.Version</b> by Mark Trinder, 24,4K downloads Easily get your apps version	v1.0.0
	<b>Xam.Plugin.PushNotification</b> by rdelrosario, 9,59K downloads Cross Platform Push Notifications Support	v1.2.4
	<b>Xam.Plugin.LocalNotifications</b> by Savelij Bondini, 5,27K downloads Create and parameterize local notifications, schedule them or send immediately, cancel if required	v1.0.0
	<b>Xam.Plugin.DeviceMotion</b> by rdelrosario, dimitrijevic, 4,55K downloads Get vector value for the different motion sensors in the device.	v1.1.2