Windows Phone 8.1 app development

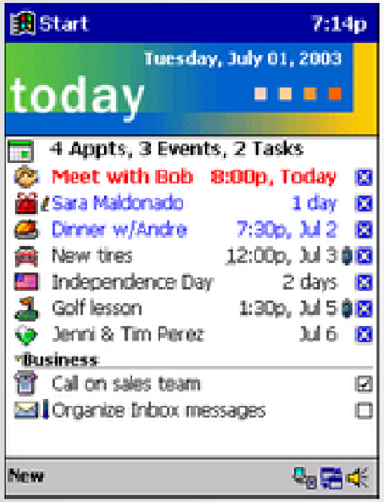
Jan Koběrský, kobersky@avast.com

Roadmap

- little bit of history
- platform overview
- hello WP (controls, templates,...)
- app lifecycle
- contracts and tasks
- best practices (MVVM, binding,...)
- storage
- push notifications, threading, animations

Windows Mobile

2000-2010 (last version 6.5)
Windows CE

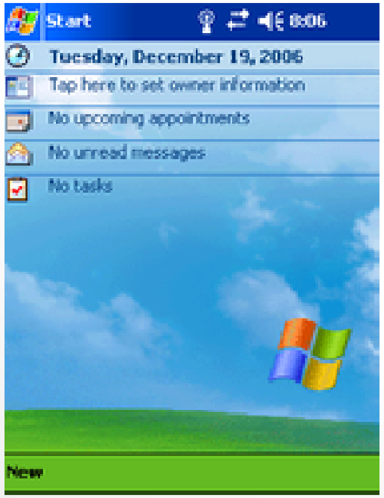


Tuesday, July 01, 2003

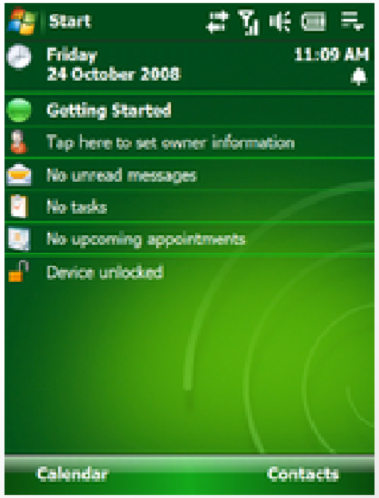
4 Appts, 3 Events, 2 Tasks

Meet with Bob 8:00p, Today ☒
Sara Maldonado 1 day ☒
Dinner w/Andre 7:30p, Jul 2 ☒
New tires 12:00p, Jul 3 ☒
Independence Day 2 days ☒
Golf lesson 1:30p, Jul 5 ☒
Jenni & Tim Perez Jul 6 ☒

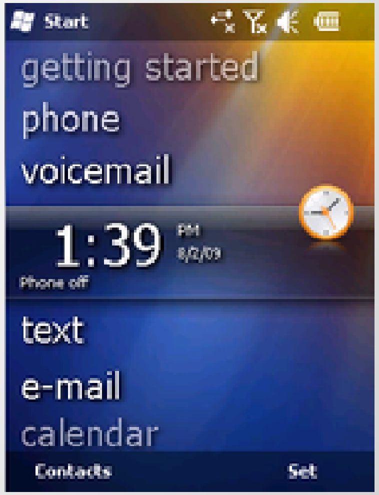
Business
Call on sales team ☑
Organize Inbox messages □



Tuesday, December 19, 2006
Tap here to set owner information
No upcoming appointments
No unread messages
No tasks



Friday
24 October 2008
11:09 AM
Getting Started
Tap here to set owner information
No unread messages
No tasks
No upcoming appointments
Device unlocked



Windows Phone 7.x

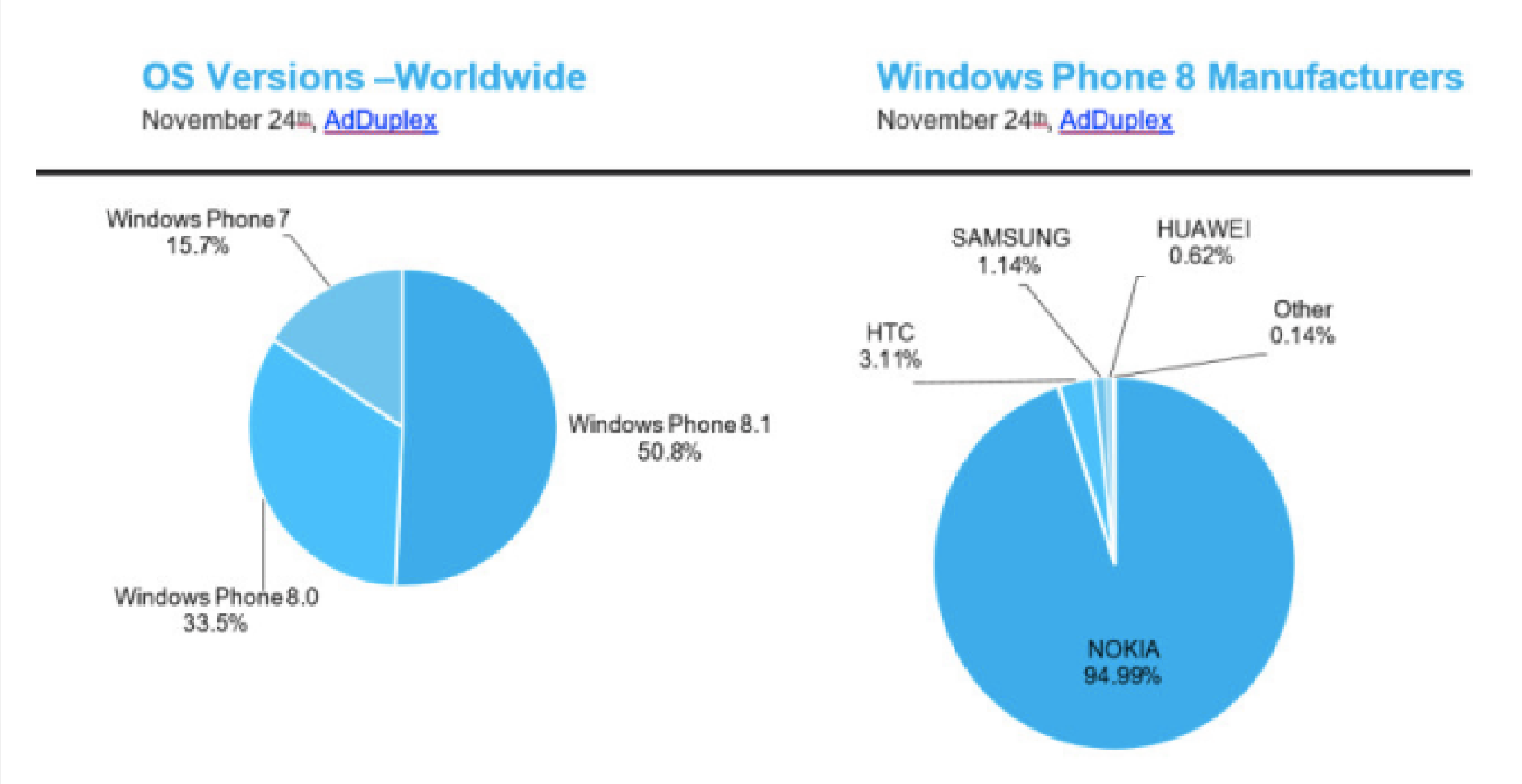
2010-2012 (last version 7.8)
Windows CE

Windows Phone 8 a 8.1

- 2012-today
- Windows NT
- Silverlight vs. WinRT/WinJS
- Code reusability
- in 8.0 30% API convergence; 90% in 8.1
- universal apps

Frank
Prengel: 8.1
Day! :-D

Distribution



OS Versions –Worldwide

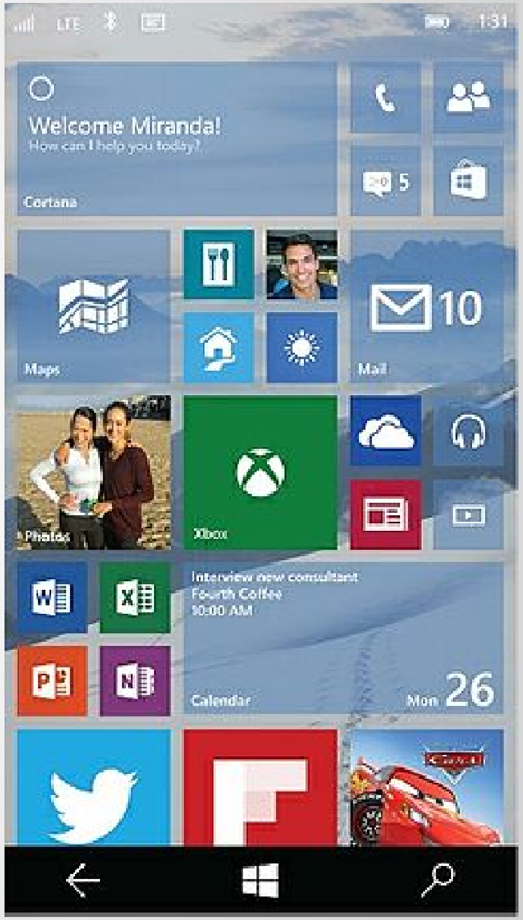
November 24th, AdDuplex

Windows Phone 8 Manufacturers

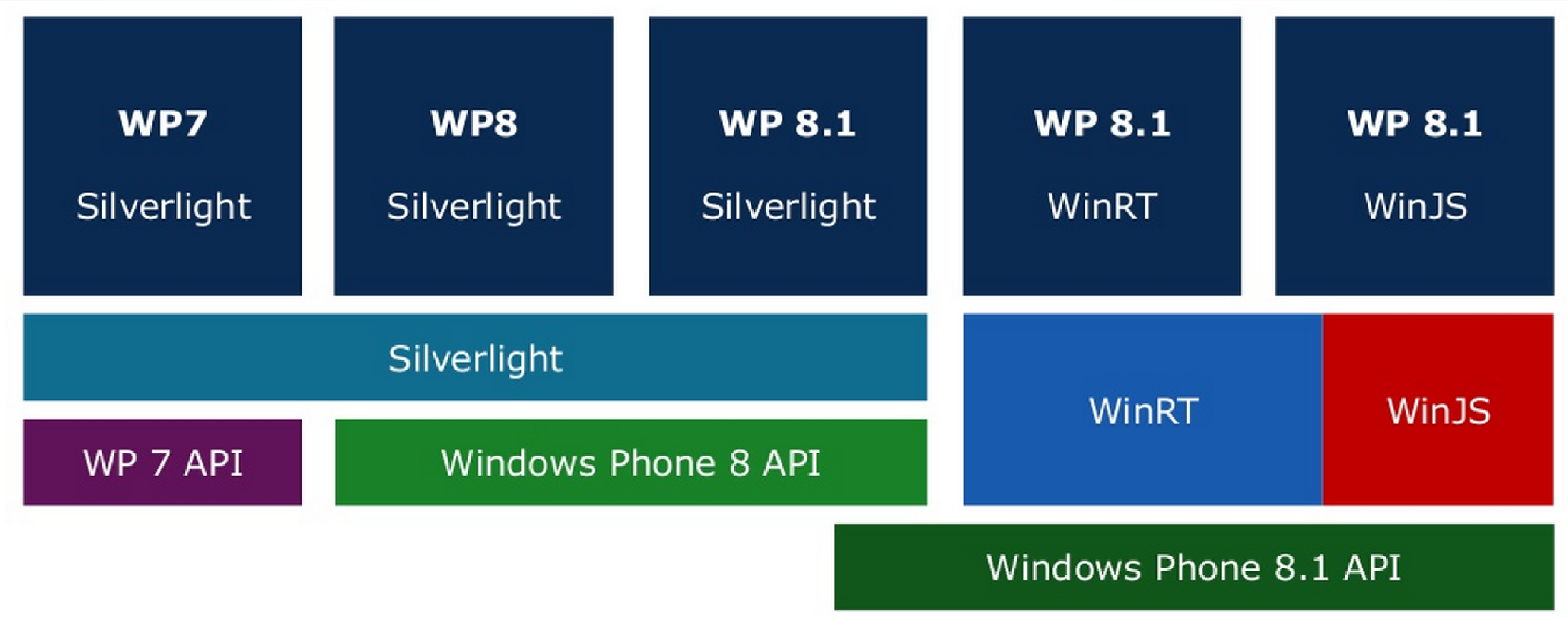
November 24th, AdDuplex

Windows 10 Mobile

- Technical preview available
- Successor of 8.1
- universal apps
- Modern UI “v 2.0”



Platform overview



WP7

Silverlight

WP8

Silverlight

WP 8.1

Silverlight

WP 8.1

WinRT

WP 8.1

WinJS

Silverlight

WinRT

WinJS

WP 7 API

Windows Phone 8 API

Windows Phone 8.1 API

Let’s talk 8.1

What do I need?

- Windows :-)
- Visual Studio
    - free versions (community and express)
- SDK
- hyper-V

Hardware-assisted virtualization.

Second Level Address Translation (SLAT).

Hardware-based Data Execution Prevention (DEP)

Limitations

- analyze and prototype first
- lifecycle limitations
- control limitations
- API limitations

Windows Store

- http://dev.windows.com/
- individual $19
- monetization
  - paid
  - inapp
  - trials
  - ads

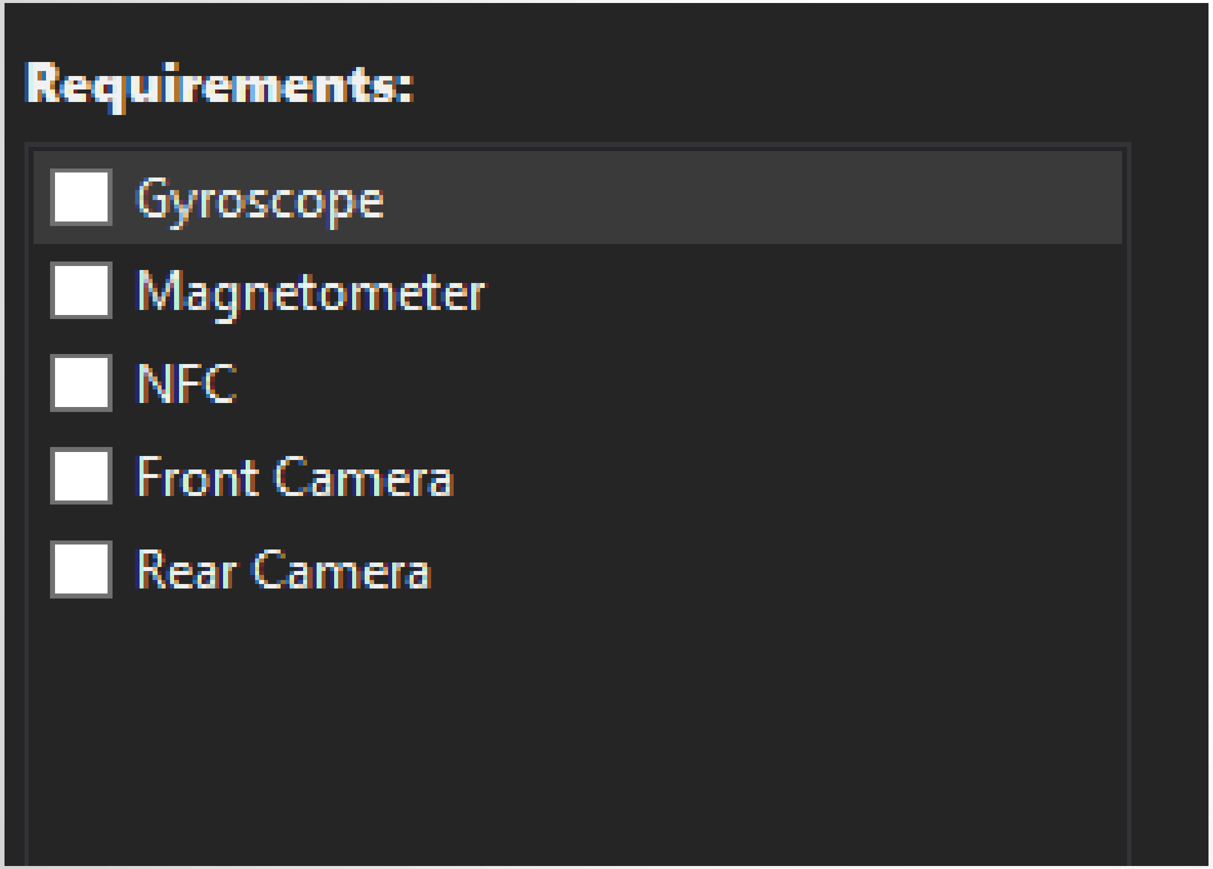
Devices and Emulators

- emulators for:
  - resolutions
  - screen sizes
  - memory limits
  - OS versions
- devices:
  - dev registered devices
  - no limitation

HW specs

- 512MB - 2GB ram
- rear camera
- accelerometr
- directX card
- ...

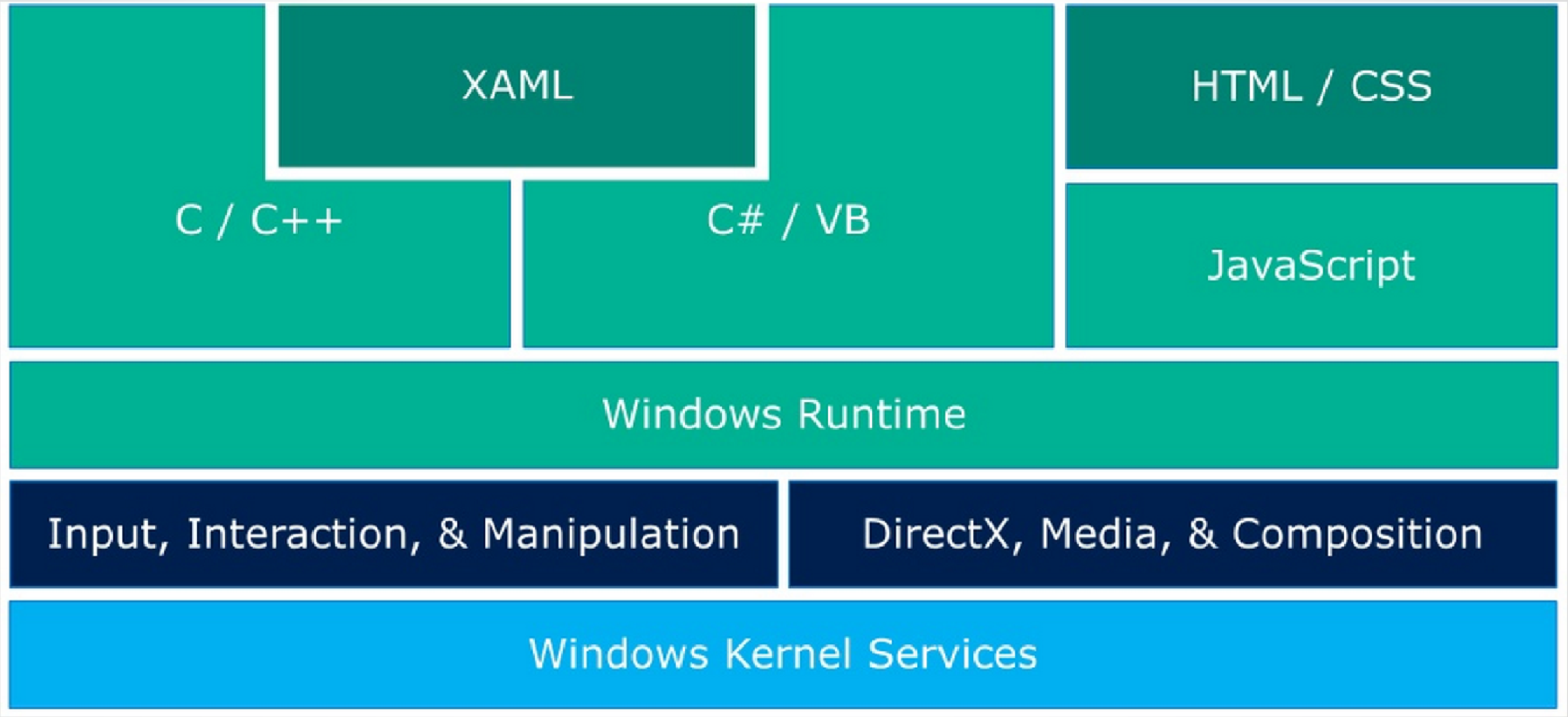
HW specs



Requirements:

- □ Gyroscope
- □ Magnetometer
- □ NFC
- □ Front Camera
- □ Rear Camera

Programming model



XAML

HTML / CSS

C / C++

C# / VB

JavaScript

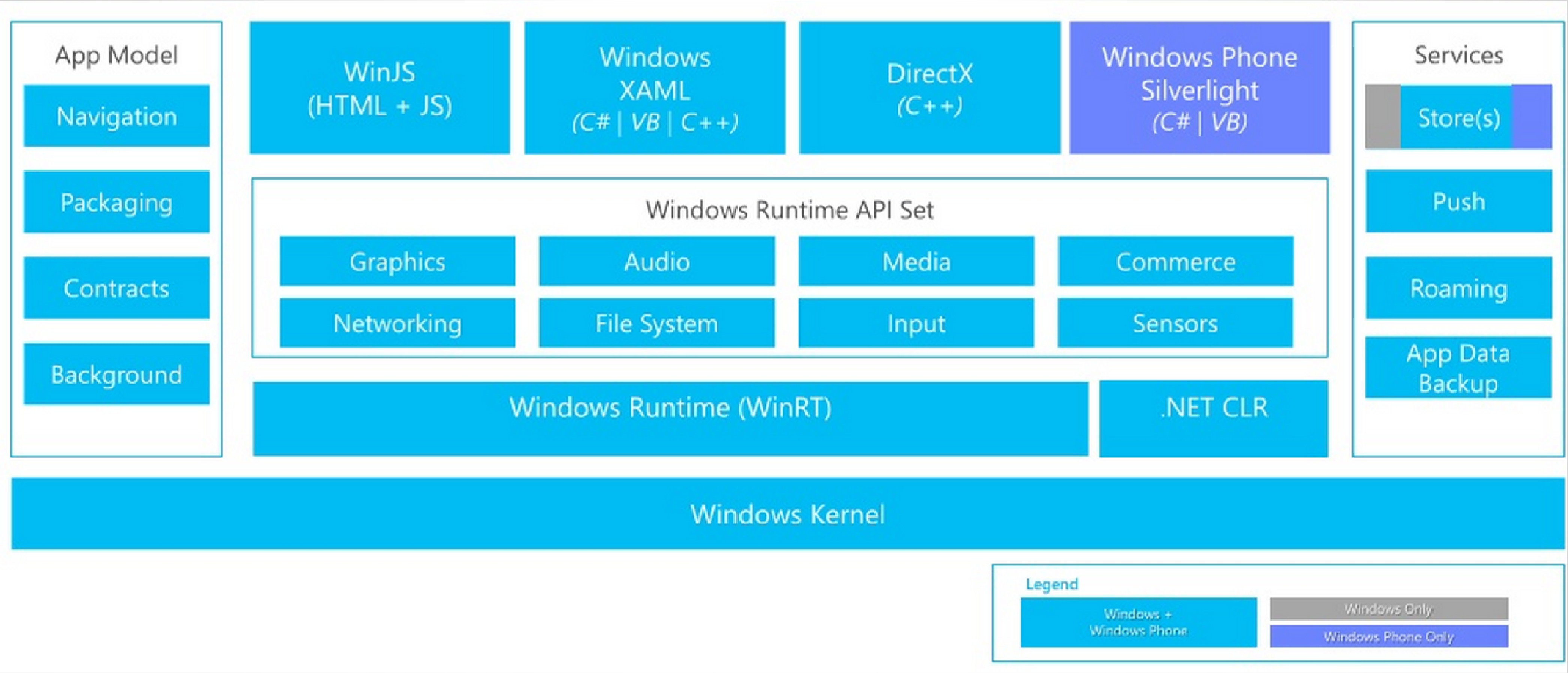
Windows Runtime

Input, Interaction, & Manipulation

DirectX, Media, & Composition

Windows Kernel Services

Programming model



App Model

Navigation

Packaging

Contracts

Background

WinJS (HTML + JS)

Windows XAML *(C# | VB | C++)*

DirectX *(C++)*

Windows Phone Silverlight *(C# | VB)*

Windows Runtime API Set

Graphics

Audio

Media

Commerce

Networking

File System

Input

Sensors

Windows Runtime (WinRT)

.NET CLR

Services

Store(s)

Push

Roaming

App Data Backup

Windows Kernel

Legend

Windows + Windows Phone

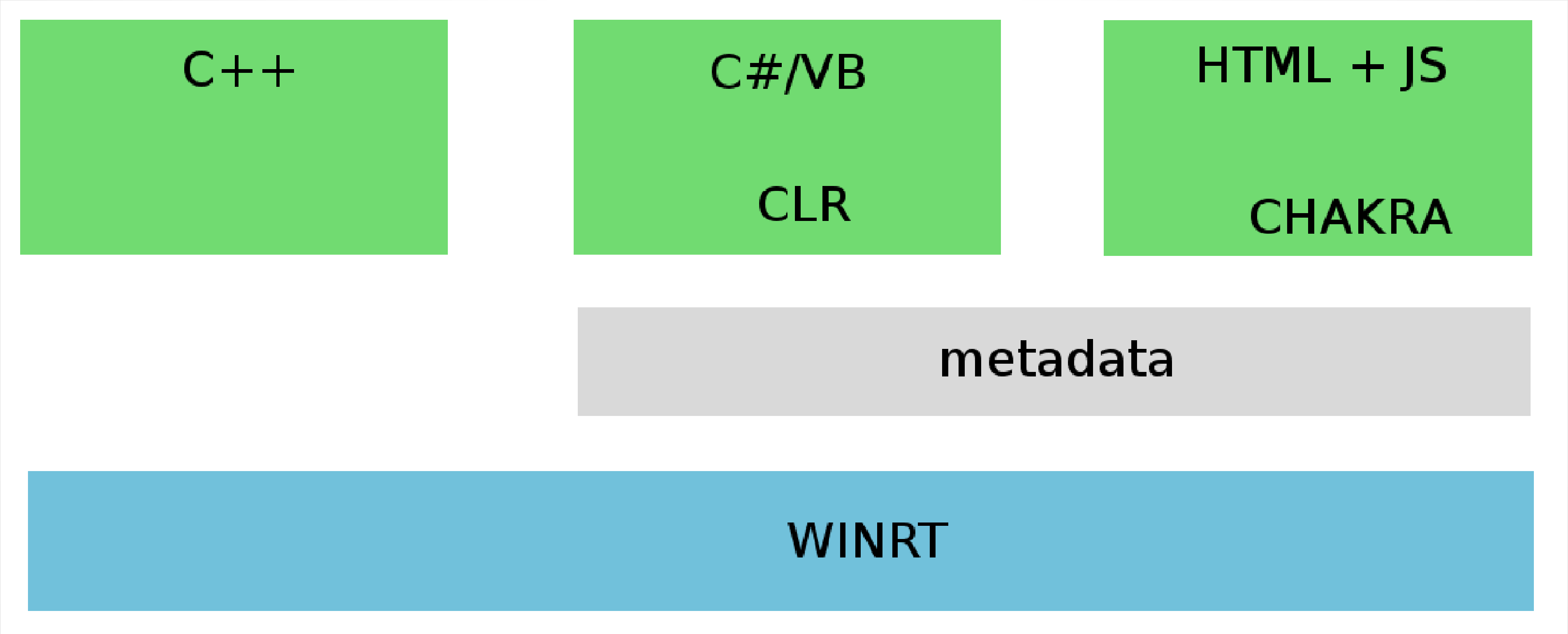
Windows Only

Windows Phone Only

WinRT

- ABI (C++)
- acts as libraries
- described by metadata (winmd)

WinRT



C++

C#/VB

CLR

HTML + JS

CHAKRA

metadata

WINRT

Hello Windows (Phone)

- code sharing (universal)
- assets
- resources
- appxmanifest
- XAML
- nuget
- conditional compilation symbols

Microsoft® Visual Studio®

Page

- partial MyPage : Page
- one screen (view) of an application
- XAML and “code behind”
- basic navigation handling
- provides AppBar

Page

- doesn’t include StatusBar
  - StatusBar.GetForCurrentView()
- rotation
  - DisplayInformation.AutoRotationPreferences

UserControl

- is control with XAML and code behind
- similar to page

Control

- UI without xaml
- providing basic UI events
  - tapped
  - manipulation
  - keyboard handling
- providing basic UI properties
  - IsEnabled
  - Margin
  - ...

Basic Controls

- Grid
  - table layout - rows and columns
  - children positions are defined by attached properties
- StackPanel
  - horizontal or vertical
  - children are “stacking” after each other

Basic Controls

- ListView
  - using datatemplates
  - collection binding
  - template selectors
- ItemControls
  - similar to ListView
- ContentControl

Templates and styles

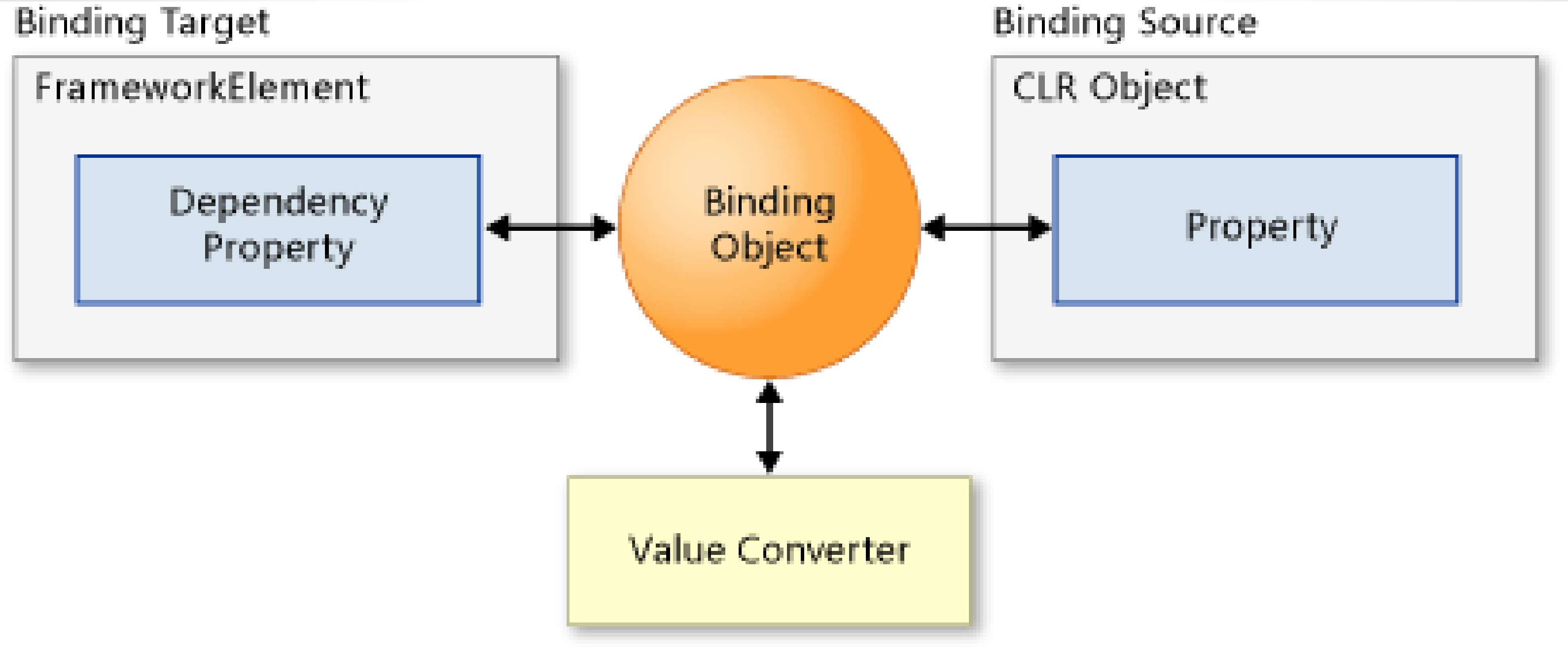
- foe easy UI reusability
- imagine styles as CSS with inheritance
- templates are chunks of design
- conditional templates (selectors)
- extract default templates and DIY
- Blend - tool for creating UI design

DependencyProperty

- properties using of DepedencyObject for storing values
- using backing field for storing values
- binding mechanism build in

Binding

- propagating data in UI



Attached properties

- easy way to extend xaml controls with custom properties
- most commost is Grid.Row, Grid.Column, ...

INotifyPropertyChanged

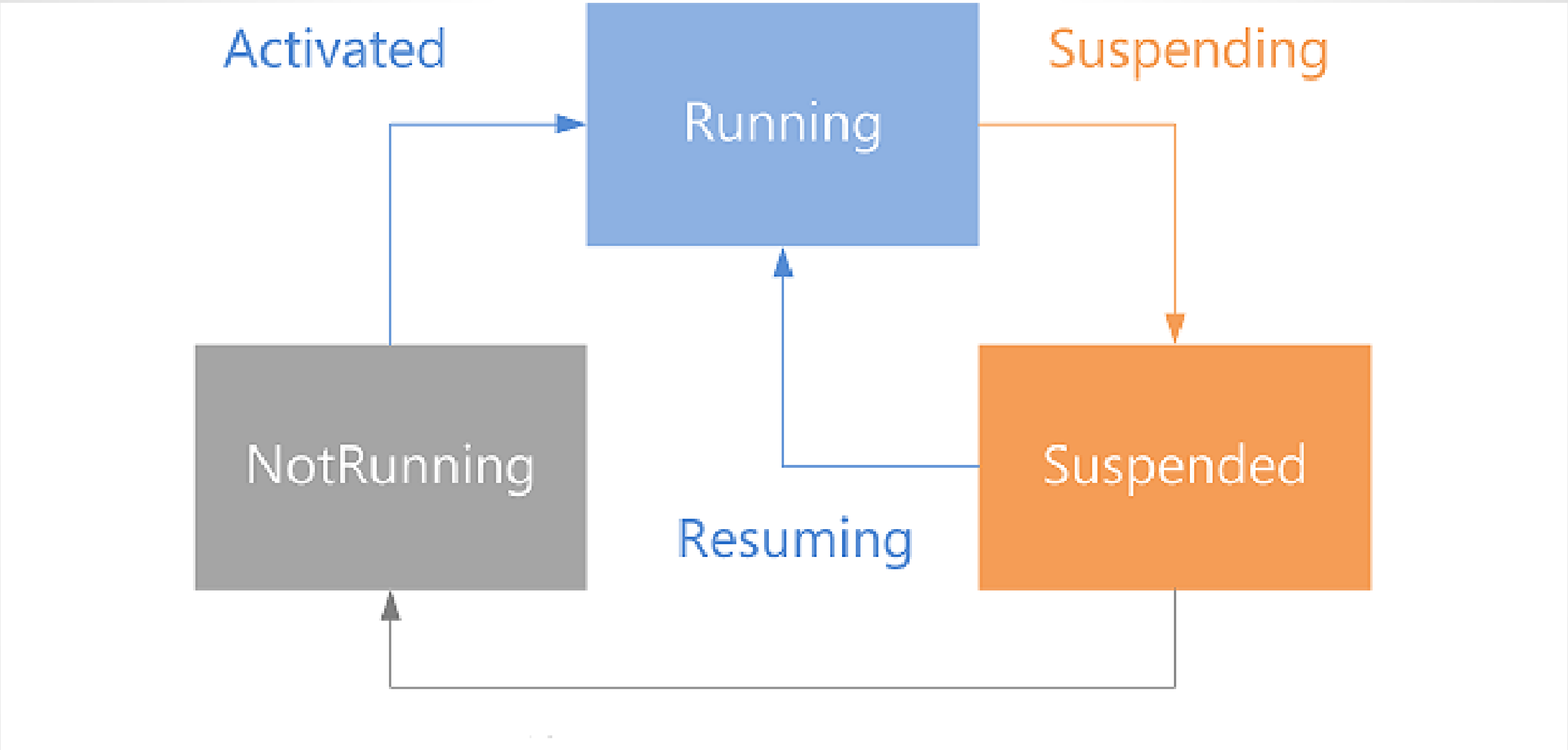
- Notyfiying to UI when property changed
- Make you life easy with fody

Scaling and resolution detection

Navigation

- via Frame by default
  - e.NavigationMode
- page overrides
  - OnNavigatedTo
  - OnNavigatedFrom
  - OnNavigatingFrom
- NavigationCacheMode
- NavigationParameter - naming convention

App life cycle



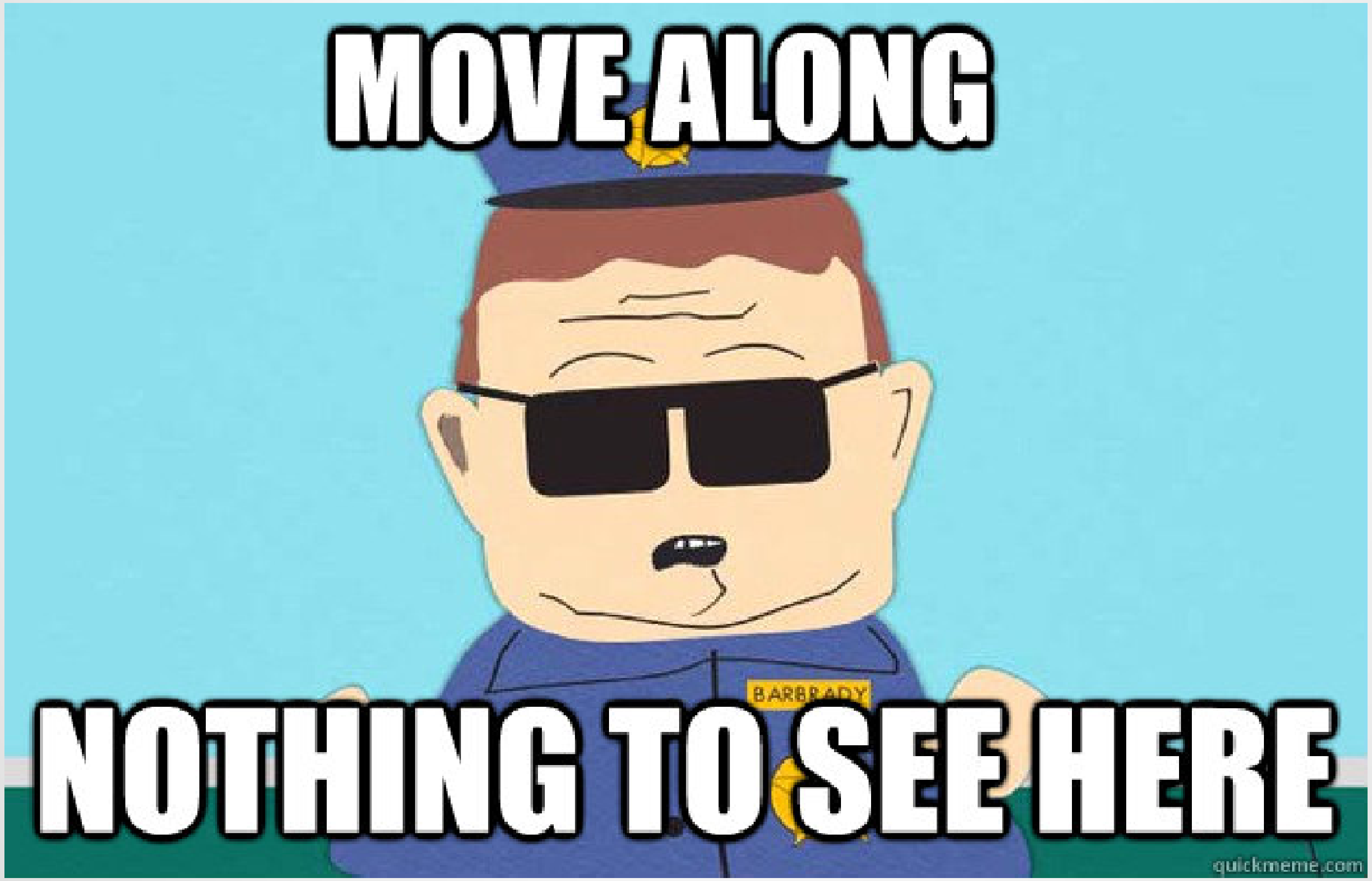
Activated

- user or system
- 16 different starts (Launch, Search, File, FilePicker, LockScreenCall…)
- execution states
- arguments
- different tiles

Suspending

- state saving - system vs. developer
- know your deadline
- we don’t know if app will be terminated

Resuming



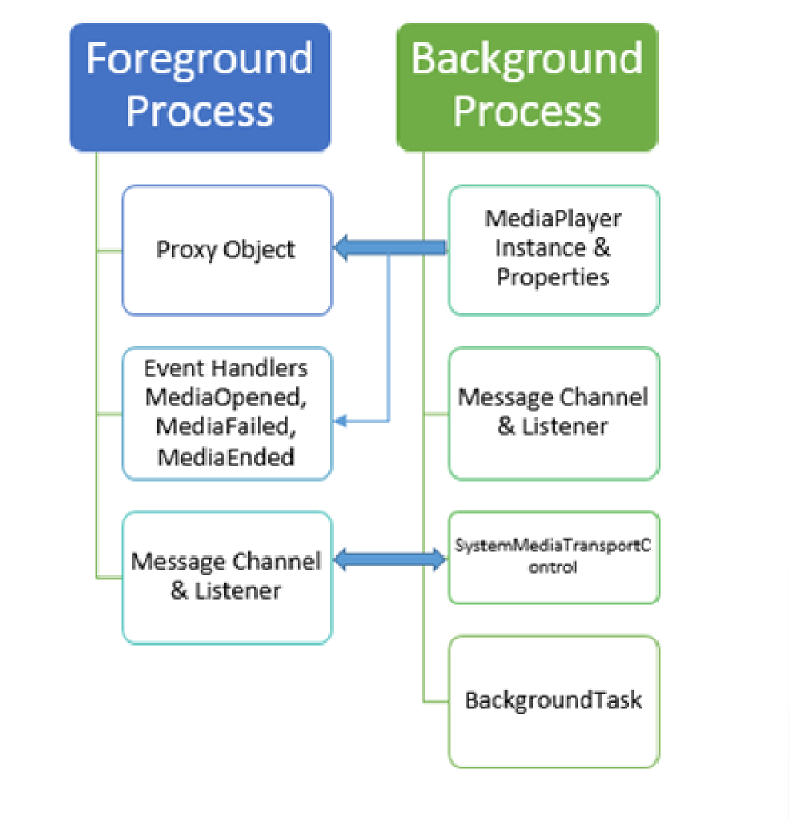
Contracts

- file picker
- share
- open uri
- file association

Tasks and triggers

- simulating multitasking
  - **SmsReceived**
  - **UserPresent/UserAway** (lock)
  - **NetworkStateChange**
  - **InternetAvailable**
  - **ServicingComplete**
  - **TimeZoneChange**
- limited - refresh time/bandwith
- maintenance trigger

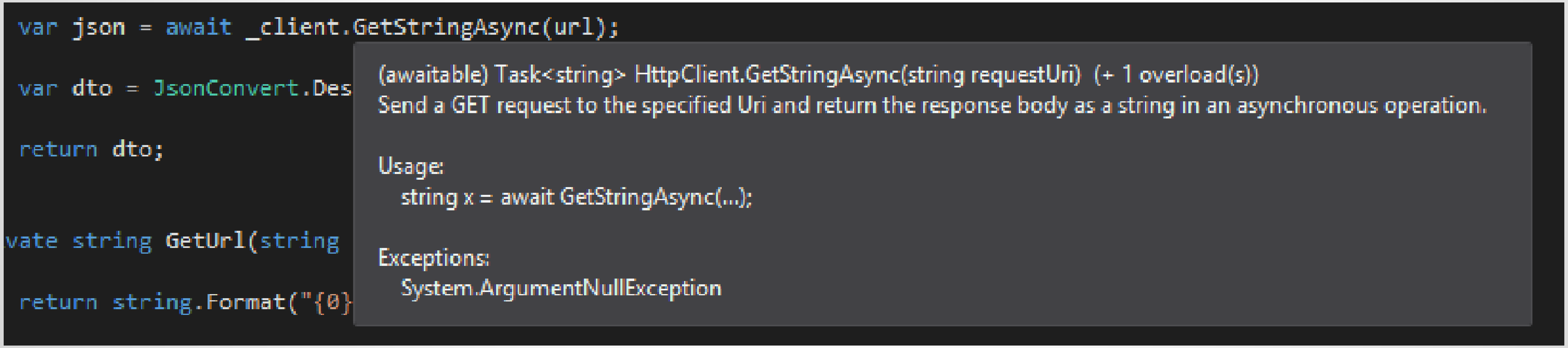
Background Audio


Foreground Process
Background Process
Proxy Object
Event Handlers MediaOpened, MediaFailed, MediaEnded
Message Channel & Listener
MediaPlayer Instance & Properties
Message Channel & Listener
SystemMediaTransportControl
BackgroundTask

Best practices

Async and await

- removing “callback hell”
- async: Task, Task<T>, void



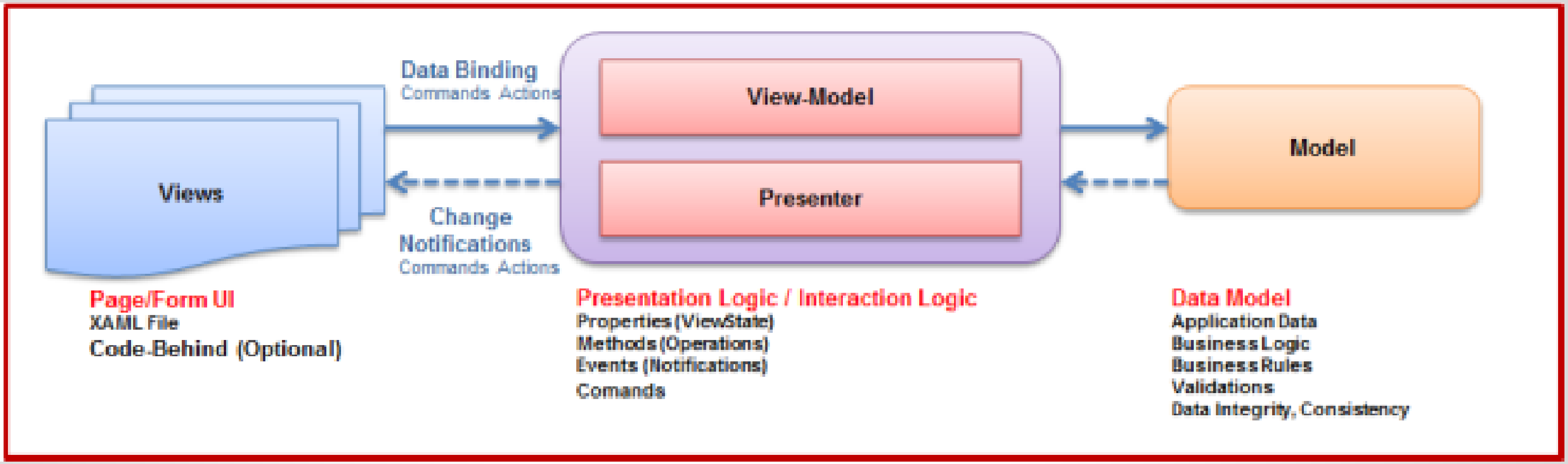
```
 var json = await _client.GetStringAsync(url);

 var dto = JsonConvert.Des

 return dto;


vate string GetUrl(string

 return string.Format("{0}
```

(awaitable) Task<string> HttpClient.GetStringAsync(string requestUri) (+ 1 overload(s))
Send a GET request to the specified Uri and return the response body as a string in an asynchronous operation.

Usage:
string x = await GetStringAsync(...);

Exceptions:
System.ArgumentNullException

BP: MVVM (Model-View-ViewModel)

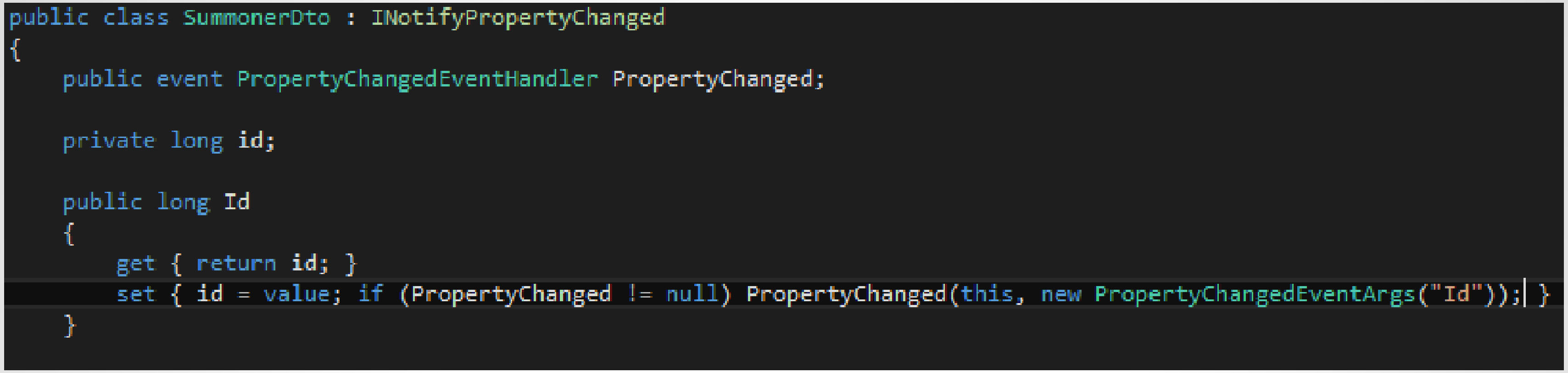


Binding everywhere!

BP: CaliburnMicro - XAML made easy

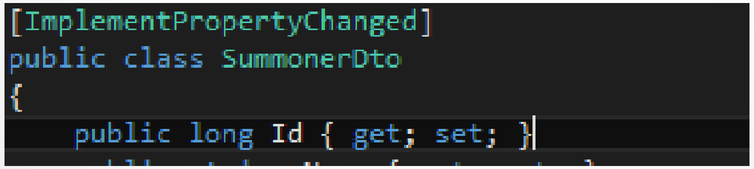
- Dependency Injection
- Navigation made simple
- Broadcasting messages
- Simple commands from UI to VM
- Naming conventions

https://github.com/Caliburn-Micro



```
public class SummonerDto : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    private long id;

    public long Id
    {
        get { return id; }
        set { id = value; if (PropertyChanged != null) PropertyChanged(this, new PropertyChangedEventArgs("Id")); }
    }
```

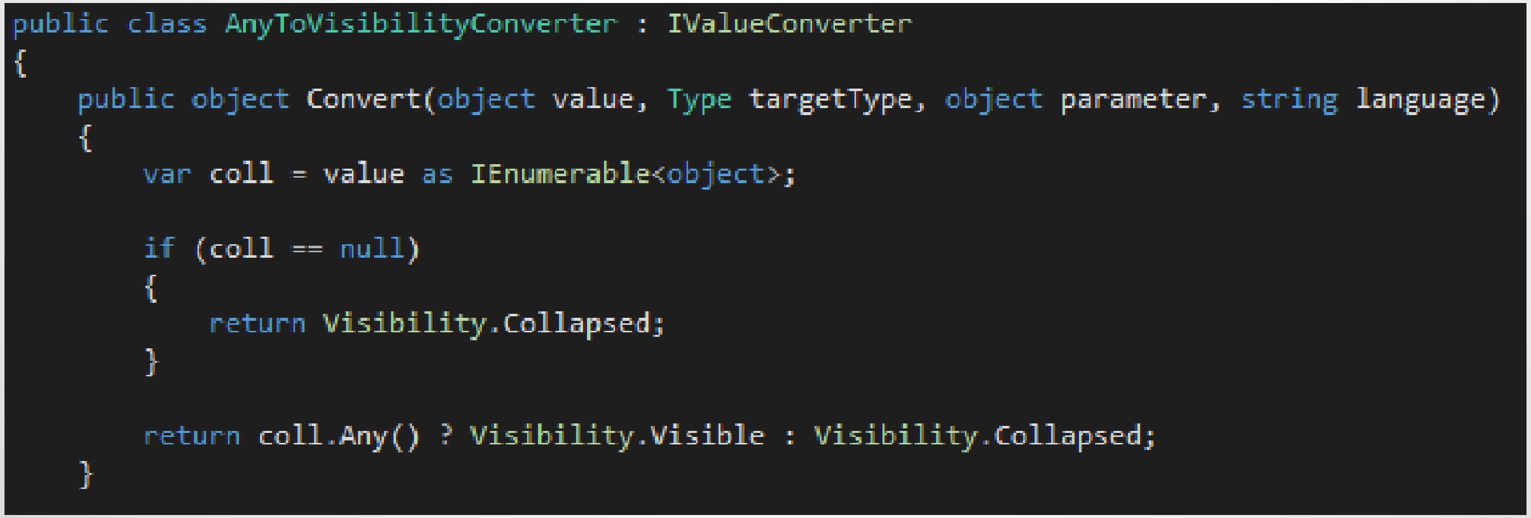
VS



```
[ImplementPropertyChanged]
public class SummonerDto
{
    public long Id { get; set; }
```

BP: Converters

- for presentation
- result isn’t propagated to VM (if you don’t want to)



```
public class AnyToVisibilityConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, string language)
    {
        var coll = value as IEnumerable<object>;

        if (coll == null)
        {
            return Visibility.Collapsed;
        }

        return coll.Any() ? Visibility.Visible : Visibility.Collapsed;
    }
```

BP: Vector graphics

- autoscaling
- for icons and basic grphics
- canvas
- Adobe Illustrator nad Fireworks, XDraw, ...

BP: i18n and other resources

- applicable via Uid
- can target every property of control
- kind of tricky in design time
- never hardcode strings

Runtime components

- written in C++
- code reusing
- wrappers for using well known library

Storage

- static class ApplicationData
  - your app only
  - LocalSettings
  - RoamingSettings
  - LocalFolder
  - RoamingFolder
  - TemporaryFolder
- KnownFolders

Storage - settings

- IPropertySet : IDictionary<string, object>

Storage - files

- files and folders (StorageFile, StorageFolder)
- fully asynchronous
  - CreateFileASync
  - CreateFolderAsync
  - GetFileAsync
  - GetFolderAasync

Storage - files

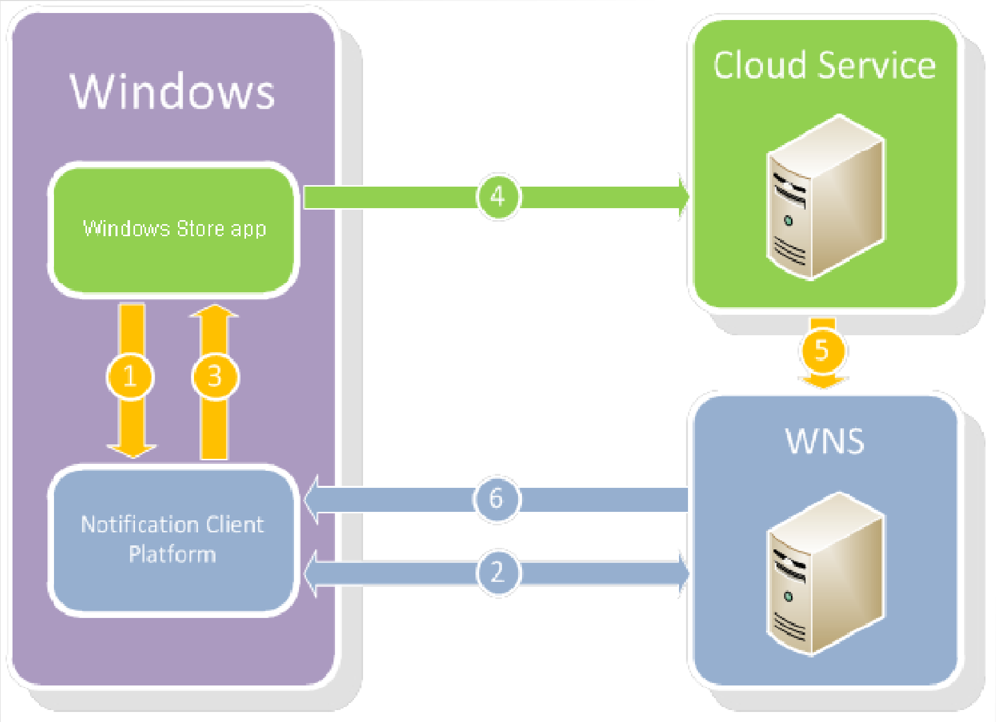
- queries
  - enumerating folders/files
- getting tuhmbnails with GetThumbnailAsync
- basic properties (Modified, Created, Size)

Storage - KnownFolders

- CameraRoll
- MusicLibrary
- PicturesLibrary
- RemovableStorage
- VideosLibrary

Push notifications

Windows Push Notification Services



- channel represented by URI
- channel should be refreshed as much as possible (uri can change)
- lasts 30 days
- channel is not guaranteed
- channel per app
- notification is handled by system (deeplinks)

- toast and tiles

Threading

- do not block UI!
- using ThreadPool or ThreadPoolTimer
- report back to UI with CoreDispatcher

Animations

- cool StoryBoard bro
- can be created in XAML and code
- easings
- keyframes
- Double, colors, objects

Questions?

Jan Kobĕrský
kobersky@avast.com