Windows Phone app development #2

Jan Koběrský
kobersky@avast.com
@koberskyJan

Agenda

- app architecture - MVVM again
- presenting data in UI
- Live? Tiles
- async, await
- HTTP clients
- security
- Windows 10 features

App architecture

- separating data, logic and UI
- maintainable code
- testability

“Always code as if the person who ends up maintaining your code is a violent psychopath who knows where you live.”

Model-View-ViewModel

architectural pattern for software development

MVVM - what it is/isn’t

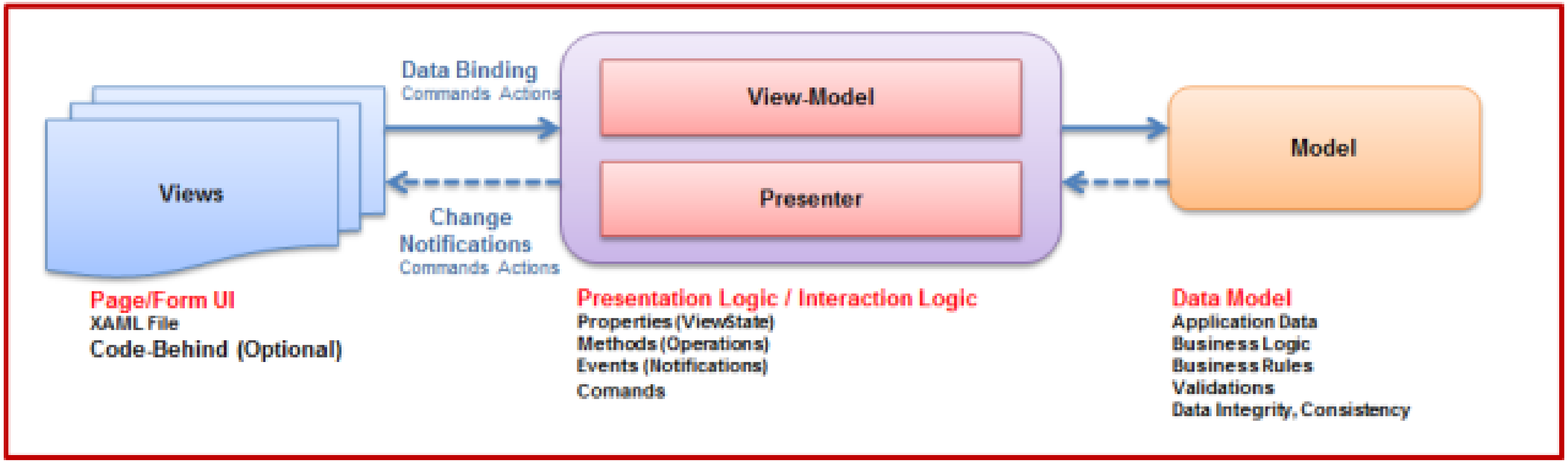
MVVM is

- variation of Fowler’s presentation model
- based on MVC
- developed by Microsoft for WPF and Silverlight
- is using Binding and Commands

MVVM isn’t

- framework
- used only in .NET (Java mostly as MVB)

MVVM - parts


Data Binding
Commands Actions
View-Model
Presenter
Model
Views
Change
Notifications
Commands Actions
Page/Form UI
XAML File
Code-Behind (Optional)
Presentation Logic / Interaction Logic
Properties (ViewState)
Methods (Operations)
Events (Notifications)
Comands
Data Model
Application Data
Business Logic
Business Rules
Validations
Data Integrity, Consistency

MVVM - parts

Model

- object representation of your data or **service(s) providing data**

View

- UI

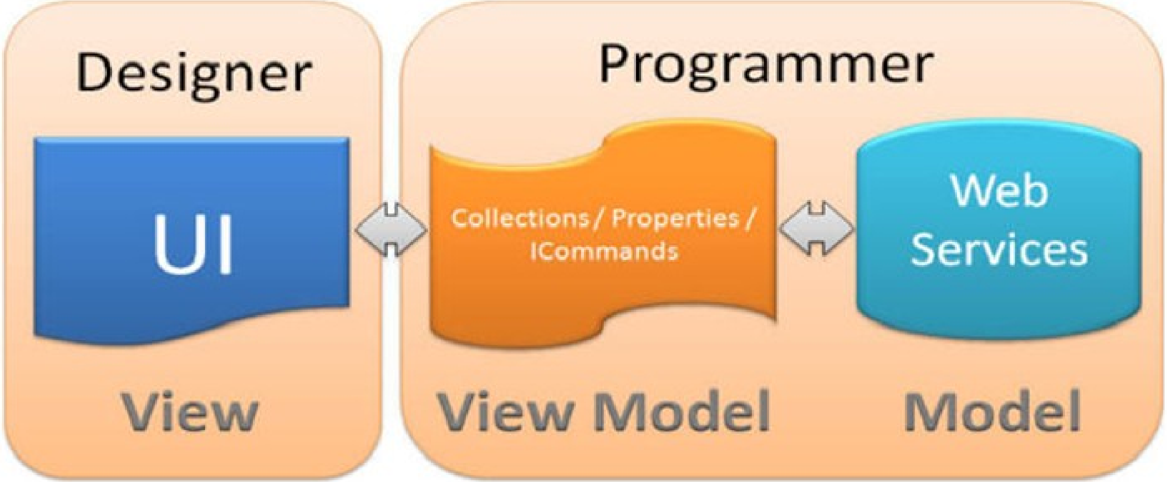
ViewModel

- business logic between data and ui
- bound (provides data) to the View

MVVM - parts

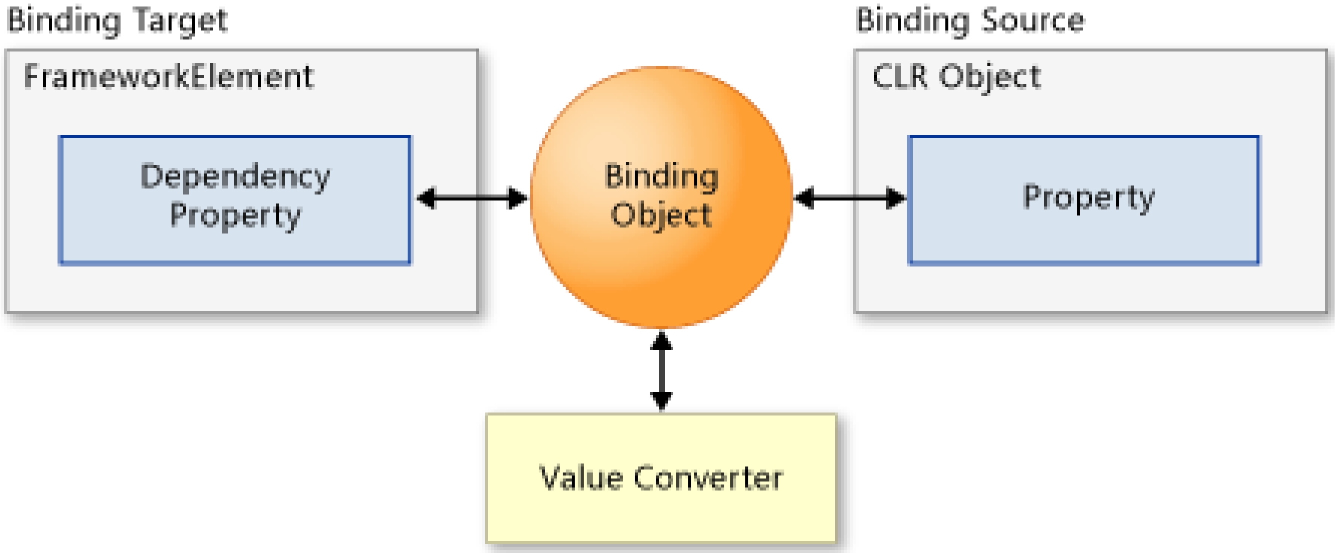
Binder

- XAML



Binding

- propagating data in UI



MVVM - implementation

ViewModel is basically class, which is exposing properties and implementing INotifyPropertyChanged

Easy to use frameworks - Caliburn Micro, MVVM Light.

MVVM - Caliburn micro

- base classes for VM’s
- binding VM’s to UI is name based
- ViewModel first navigation
- Page lifecycle handling
- IoC
- messaging
- much simpler commanding

MVVM - Commands

- handling UI events in ViewModel
- not every event
- Attached Messages in Caliburn

MVVM - messaging

- type based messaging (<T>)
- we don’t need references
- subscribing/publishing

MVVM live demo

Presenting data in UI

POCO, partial classes, converters, template selectors, behaviors, attached properties

Presenting data in UI

- POCO
- modify TO/Contract data/structure only if necessary
- create own data structures and convert them to transport objects

Presenting data in UI

Partial classes

- awesome for generated classes

“Wrapper” classes

- using original data as one of properties
- deriving from TO object

Presenting data in UI

Converters

- can convert data both ways
  - from VM to UI
  - from UI to VM
- parameterizable
- reusable
- binder part in MVVM schema

Presenting data in UI

Behaviors

- code attached to control
- multiple behaviors to one control
- reusable

Presenting data in UI

Attached properties

- used mostly in XAML
- global property that is settable on any object

Presenting data in UI

Template selectors

- can show different UI based on data
- awesome for inheritance
- can be replaced by converters to some extend

Live? Tiles

Live? Tiles

- icons with custom information
- multiple tiles for one application
- can be transparent
- WP8 - 3 types
- usage of XAML templates in 8.1
- over 70 templates
- flip tiles

async & await

- based on promises
- provide task cancellation
- thread marshalling
- configure await
- run mutliple and wait for all/one
- Task, Task<TResult>, void

HTTP clients

- WebClient available only in 8.1 Silverlight
- HttpClient
  - System.Net.Http
  - Windows.Web.Http
- fully asynchronous
- similar, but not same

Think performance

- reflection
- Do heavy work in threads
- Throttling
- Use what is framework giving you

WP security

- storage encryption only available for enterprise users
- use DPAPI for encrypting your data
- OpenSSL is portable to WP
- obfuscate your code
- decompiler/reflectors
- ATIV

Windows 10

Windows 10

- UAP
- one binary for all
- contracts as sets of API
- not developing vs system version but UAP version
- min and max version

Windows 10 - migrating code

- 10 API is superset of 8.1 XAML
- old silverlight projects needs to be reimplemented
- all the apps from 7.X and 8.X will be running

Windows 10

- conditional compiling #WINDOWS_PHONE removed
- user runtime checking in ApiInformation
    - IsTypePresent
    - IsApiContractPresent

Windows 10 - bits

- you can now launch specific app and it can return result!
- publisher folder
- app services
- better notification interaction