
Charles'
Android

MVVM

Start from the bottom

The need for architecture

- Lower maintenance costs
- Improve performance
- Improve security
- Improve stability
- Make unit test easier

principles of App Architecture

SOLID

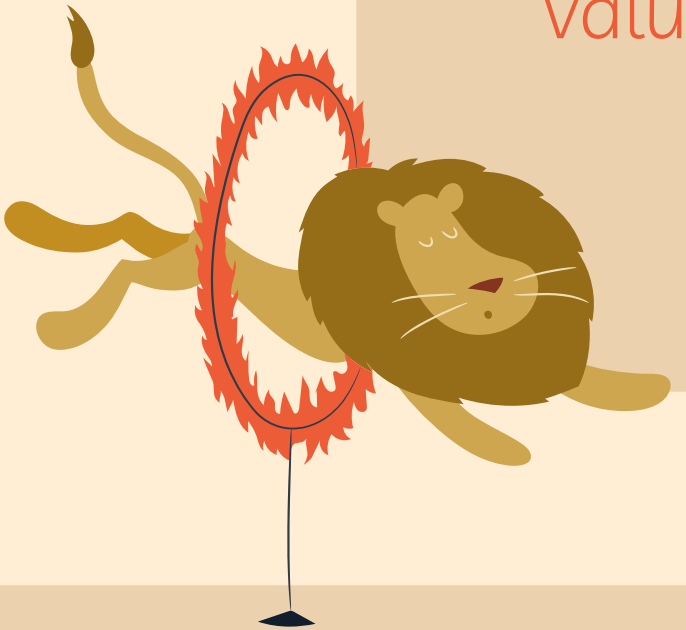
- SRP
- OCP
- LSP
- ISP
- DIP

Clean Architecture

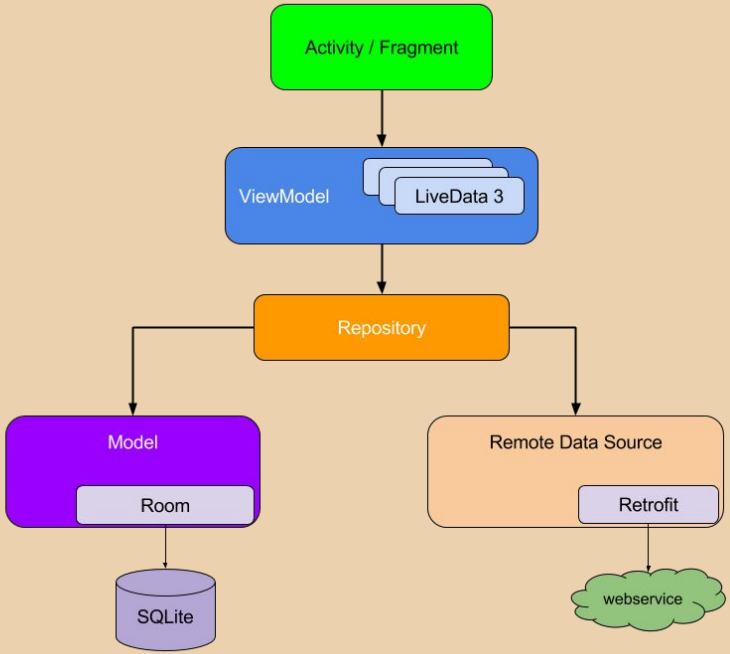
- Layering
- Separation
- Indepence

"Good design adds value faster than it adds cost."

- Thomas C. Gale



Recommended App Architecture



What do you choose?



MVC

MVP

MVVM

MVC

Model

Provide data,
Everything excepting
Controller and View

Controller

Manipulate Events and Data

View

Display contents

MVC

MVP

Model

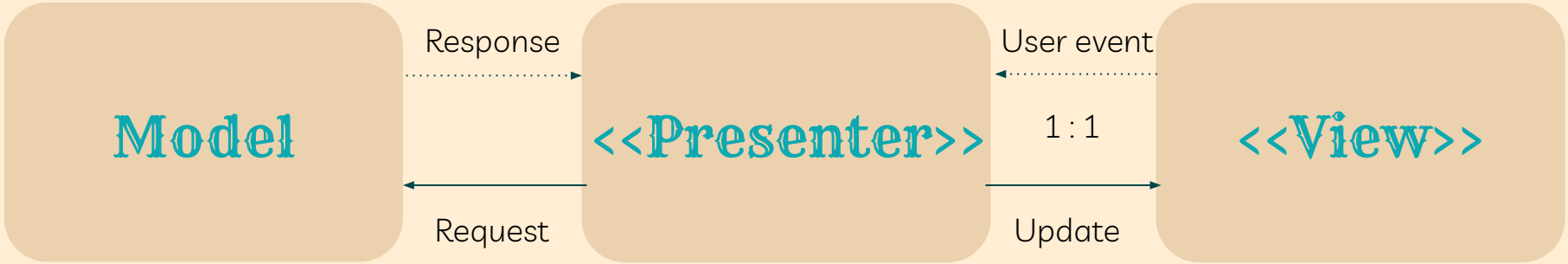
Presenter

Manipulate Events and Data
(a.k.a Supervising Controller or
Mediator)

View

Display contents
Passively

MVP


Model
Response
Request
<<Presenter>>
User event
1 : 1
Update
<<View>>

MVVM

Model

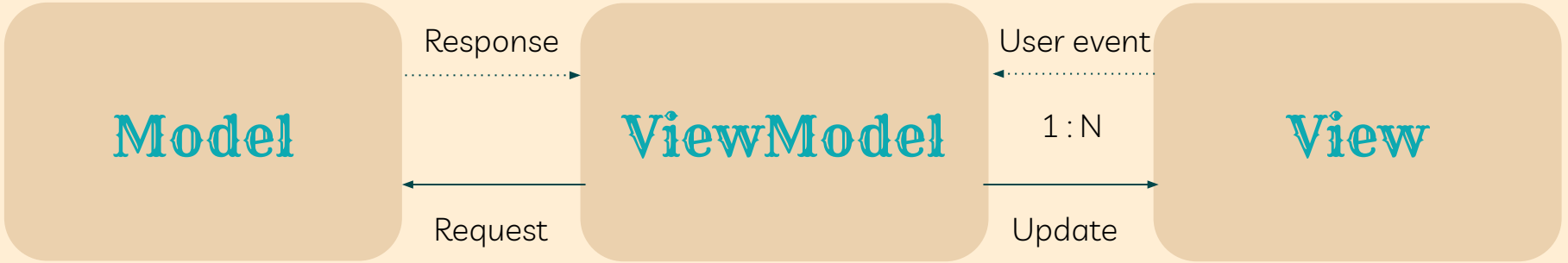
ViewModel

Manipulate Events and Data
Independent from View completely

View

Display contents

MVVM


Model
Response
Request
ViewModel
User event
1 : N
Update
View

Trait of Architecture pattern

	Maintenance	Testable	Reusable
MVC	X	X	X
MVP	△	O	X
MVVM	O	O	O

Which class can be for ViewModel?

BaseObservable

- Included in data binding package.
- Can contain Context
- Used notifyChange() or notifyPropertyChanged() to update
- Updates views automatically using ObservableField

ViewModel

- Lifecycleaware
- **Prohibits to refer context but only App's context**
- Can be a problem when using Dagger2
- Has own it's scope
- Used with LiveData usually
- Provides saving UI States method
- Includes support for Kotlin coroutines

THANKS

SCAN ME