

# ARCHITECTURES

STRV

# WHAT IS AN ARCHITECTURE

01

# ARCHITECTURE

Fundamental code structure to make code easy to navigate and maintain.

- Order
- Consistency
- Maintainability

# WHY IS IT IMPORTANT?

We actually spend most of our time reading already existing code instead of writing a new one. That's why it's extremely important to have a proper project architecture in place.

## **Key aspects of good architecture:**

- Simplifies navigation and orientation in the project
- Simplifies onboarding of new team members
- Increases scalability, maintainability and reusability
- Simplifies potential refactoring and stands for a safe environment for changes

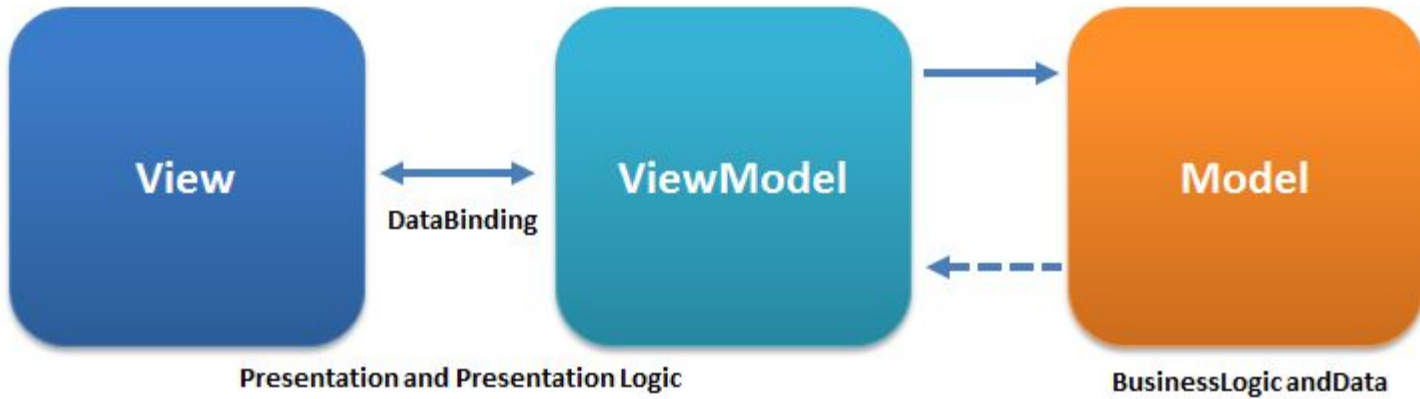
# DESIGN PATTERN

Used to solve more common problems than architecture.

- MVVM - Model View ViewModel
- MVC - Model View Controller
- VIPER - View Interactor Presenter Entity Router
- TCA - The Composable Architecture
- many, Many, MANY more

# MVVM

# 02



# SMART OR DUMB VIEWS

03



# SMART VS DUMB VIEWS

## Smart views

- Easy and fast to write
- Cannot be reused
- Harder to maintain
- They know too much
- They do too much
- Too large

## Dumb views

- Require preparation
- Reusable
- Everything is separated
- Just few lines of code per file
- Super easy to read
- Simple maintenance and extension

6:28



## Rick Sanchez



### Info

Rick Sanchez

Earth (C-137)

Human

Citadel of Ricks

Male

### Episodes

Pilot

S01E01

Characters



Characters



Episodes



Locations



Likes

# @STATEOBJECT

# 04

# @STATEOBJECT

Instance of an object which has observable properties (conforms to ObservableObject).

- Its properties can be observed for updates
- It will notify about upcoming changes
- Instantiated only once per parent structure

**@OBSERVEDOBJECT**

**05**

# @OBSERVEDOBJECT

Instance of an object which has observable properties (conforms to ObservableObject).

- Its properties can be observed for updates
- It will notify about upcoming changes
- Instantiated every time it's parent is recreated (e.g. view is redrawn)

# DEPENDENCY INJECTION

06

# DEPENDENCY INJECTION

Passing a concrete dependency (e.g. view model) from outside of the object to avoid repeated code by extracting initialization

- Reduces repeated code
- Removed knowledge about implementation details where it is not needed
- Helps with interchangeability
- Allows testability



# LIVE CODING

07

# THANK YOU!

Martin Vidovic / [martin.vidovic@strv.com](mailto:martin.vidovic@strv.com)

STRV

# QUESTIONS

STRV

BEER 🍺 ?

STRV