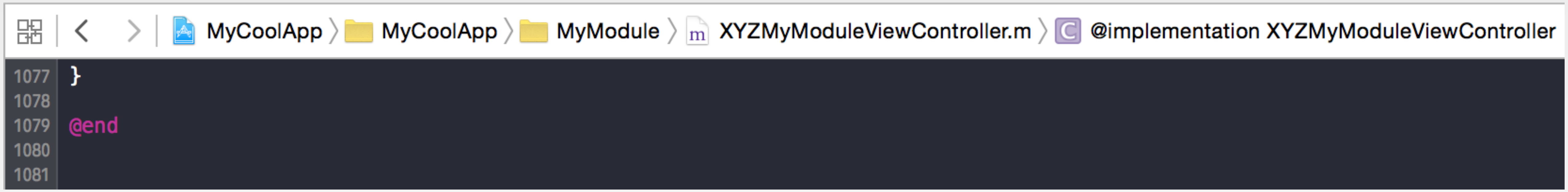
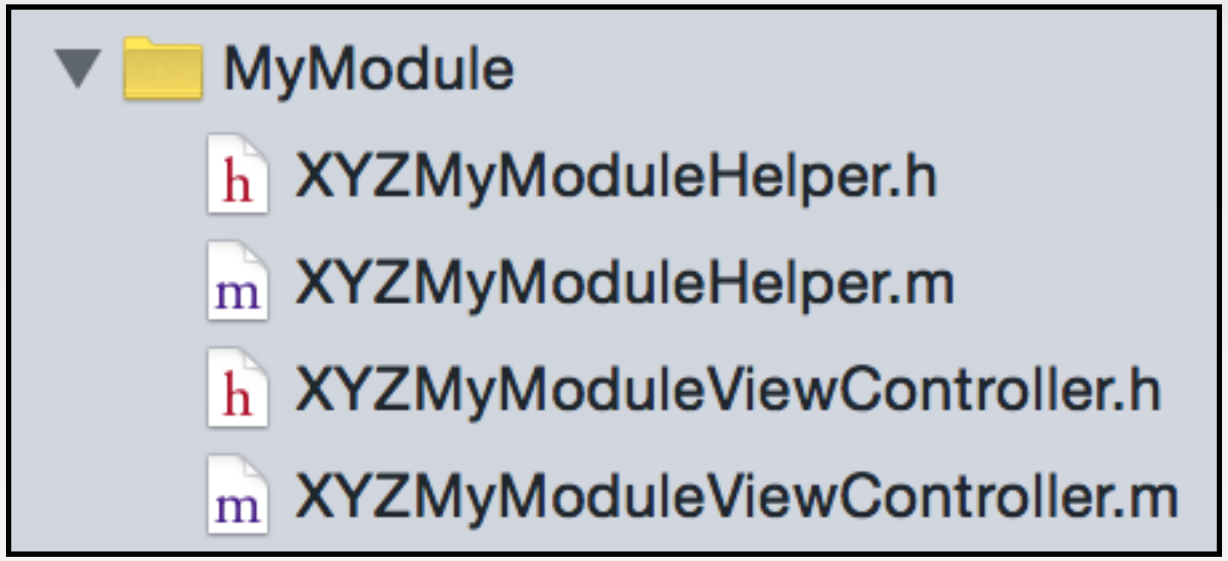
I would like to pay respect to and acknowledge the Traditional Owners of this land we meet on today. I would also like to pay my respects to the Elders past and present.

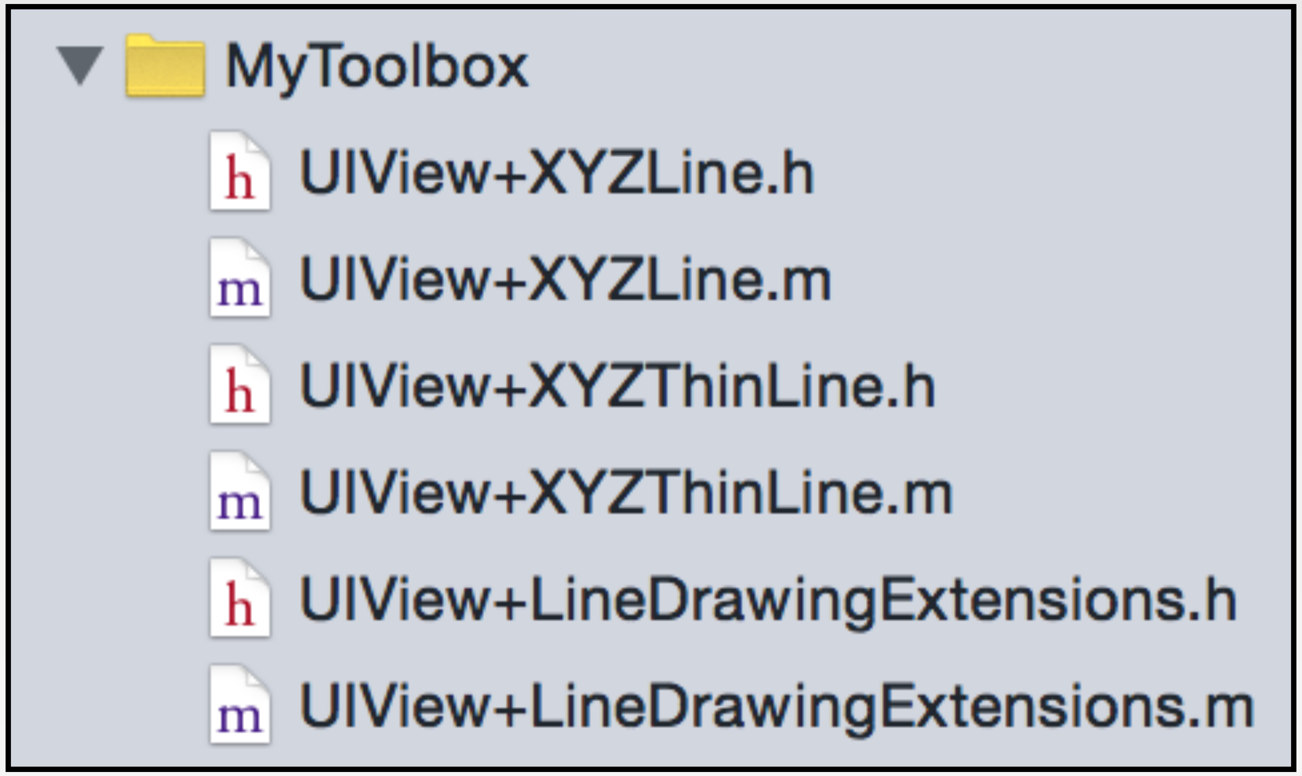

MyCoolApp
MyCoolApp
MyModule
XYZMyModuleViewController.m
@implementation XYZMyModuleViewController

```
}

@end
```



```
▼ MyModule
    XYZMyModuleHelper.h
    XYZMyModuleHelper.m
    XYZMyModuleViewController.h
    XYZMyModuleViewController.m
```



MyToolbox

- UIView+XYZLine.h
- UIView+XYZLine.m
- UIView+XYZThinLine.h
- UIView+XYZThinLine.m
- UIView+LineDrawingExtensions.h
- UIView+LineDrawingExtensions.m

iOS Architecture

Real Life VIPER

iOS Tech Lead

Outware
mobile

iOS Architecture

- What is VIPER
- Real Life use
  - Benefits of a Clean Architecture
  - Problems
  - Easy Improvements
- Learnings

Clean Architecture

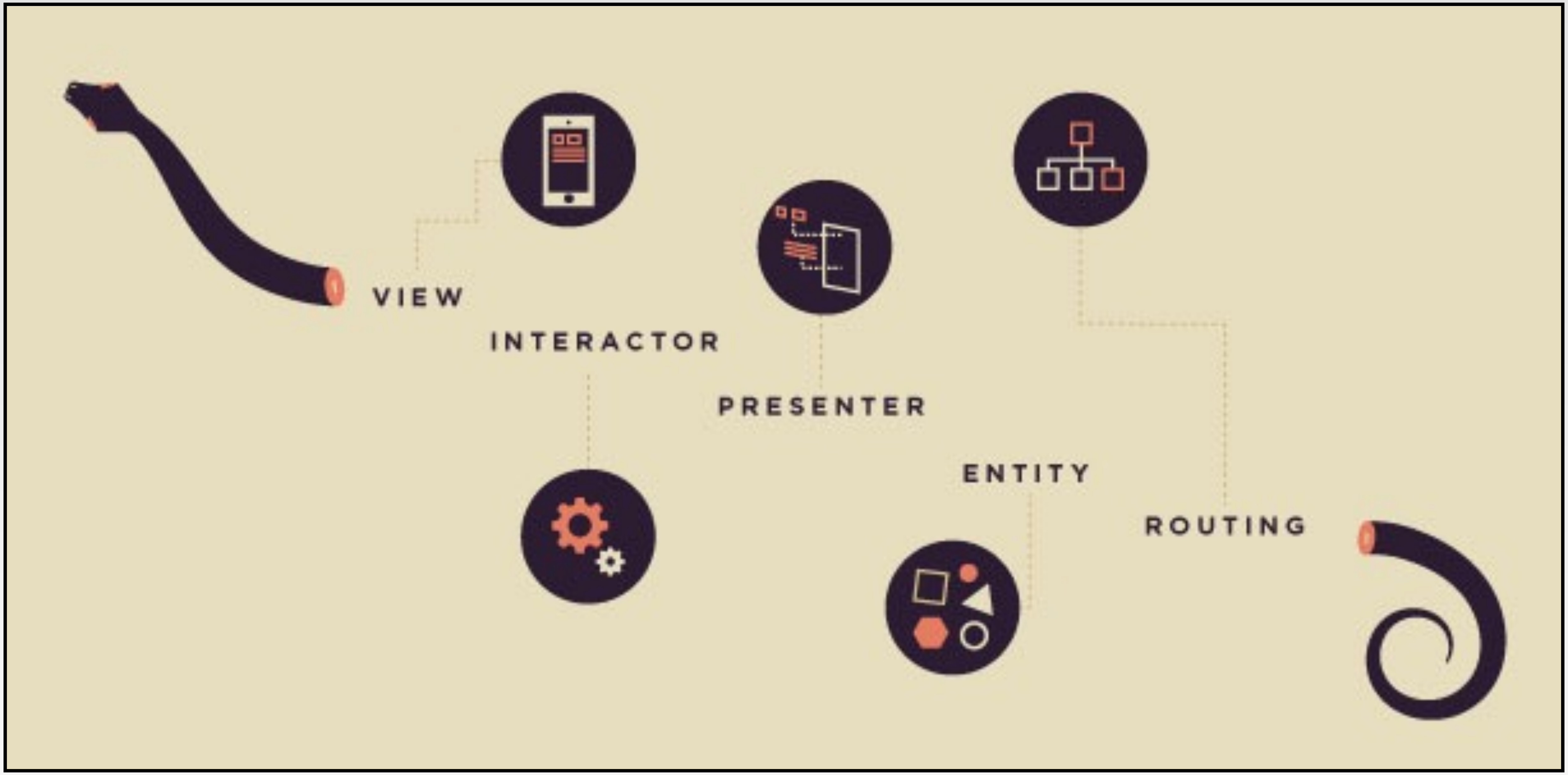
by Robert C Martin
(Uncle Bob)

Issue 13

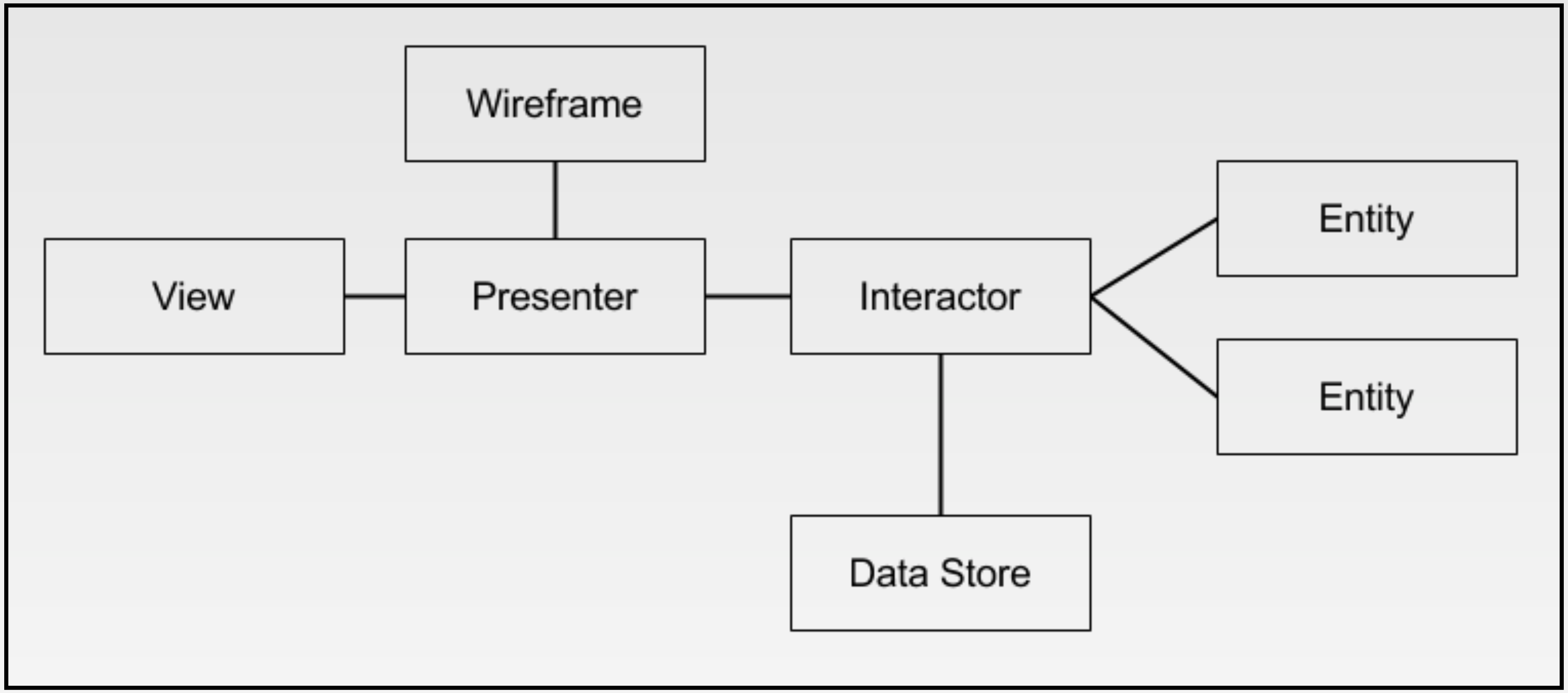
Architecture

June 2014

Cover art by Sarah Lincoln
@sarahlincoln



Artist unknown
Sourced from objc.io article



Artist unknown
Sourced from objc.io article

Appeal of VIPER

- Use of PONSOs (Plain Old `NSObjects`)
- Dependency Inversion Principle
- Router

Real Life Use

In a (reasonably) big enterprise project

10+ developers

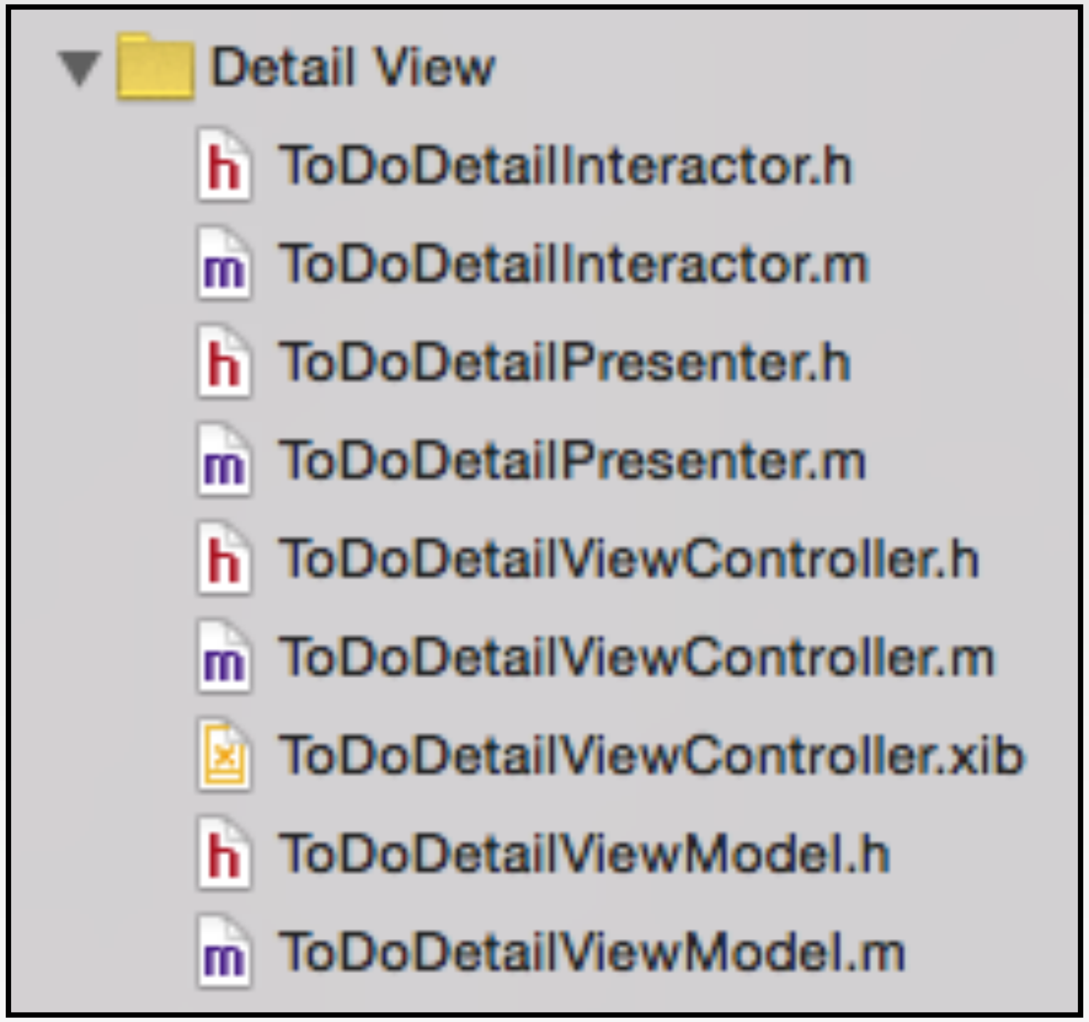
8+ months

Benefits of a Clean Architecture

- Clear cut blocks
- Smaller files
- Better code coverage
- Consistency

Clear cut blocks

- Makes it obvious what goes where
- Presentation logic / Business logic



Detail View
- ToDoDetailInteractor.h
- ToDoDetailInteractor.m
- ToDoDetailPresenter.h
- ToDoDetailPresenter.m
- ToDoDetailViewController.h
- ToDoDetailViewController.m
- ToDoDetailViewController.xib
- ToDoDetailViewModel.h
- ToDoDetailViewModel.m

Smaller files

- Shorter files
- Straight to the point
- No file bigger than 450 LOC
- Average 60 LOC
- Caveat

Better code coverage

- Better tests
- Tests are less hacky
- Barrier to entry is lower
- Tests used as documentation

Consistency

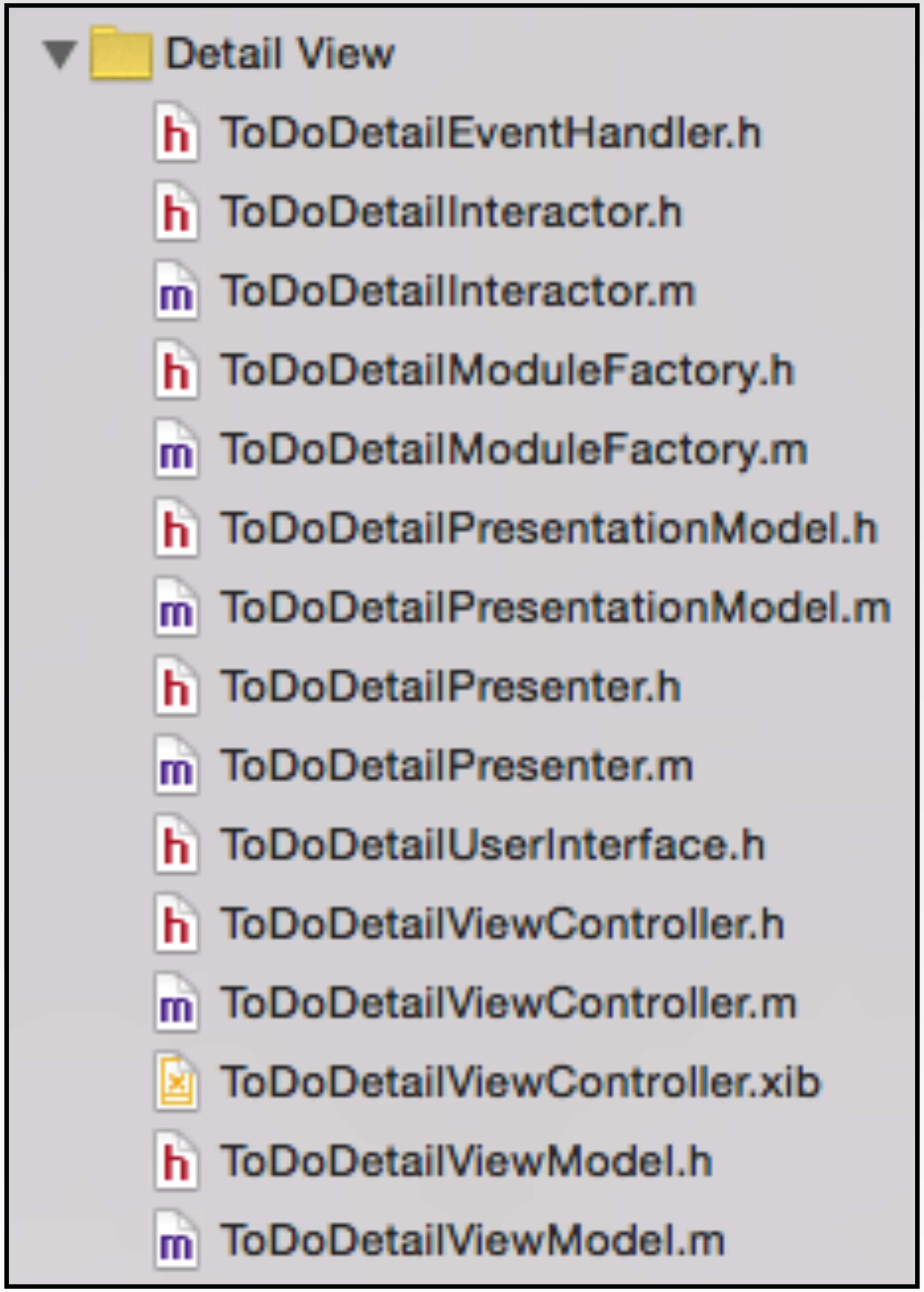
- Uniformity of codebase
- All modules look the same
- Any developer can maintain any module

Problems

- Verbosity
- Developer On-Boarding
- Developer Engagement

Verbosity

- #1 complaint among developers
- Module have ~10 files
- Xcode isn’t good at jumping through protocols



- Detail View
  - ToDoDetailEventHandler.h
  - ToDoDetailInteractor.h
  - ToDoDetailInteractor.m
  - ToDoDetailModuleFactory.h
  - ToDoDetailModuleFactory.m
  - ToDoDetailPresentationModel.h
  - ToDoDetailPresentationModel.m
  - ToDoDetailPresenter.h
  - ToDoDetailPresenter.m
  - ToDoDetailUserInterface.h
  - ToDoDetailViewController.h
  - ToDoDetailViewController.m
  - ToDoDetailViewController.xib
  - ToDoDetailViewModel.h
  - ToDoDetailViewModel.m

Developer On-Boarding

- 3 weeks to not feel lost anymore
- 1.5 months to feel confident

Developer Engagement

- Share the architecture with your team
- Not everyone is excited about VIPER, etc.

Easy Improvements

- Naming Convention
- Dogma

Naming Convention

- Chose a sensible naming convention early
- Stick to it
- Enforce it
  - During code review?

Dogma

- Not everything is a nail
- Custom UI components aren’t

What we learned with VIPER

V-I-P-E-R

View, Interactor, Presenter, Entity, Router

V-C-VM-M-R

View, Controller, ViewModel, Model, Router

VIPER®

VIPER™

MVC

Massive View Controller

Design Patterns

should be prescriptive

Help make small choices upfront

so you can spend more time on the code that matters

Facts

- Best test coverage ever
    - 80+%
    - 5000+ test cases
    - ViewControllers all smaller than 450 lines
- Pretty good for a big project
    - 10+ developers
    - 8+ months

Forced us to think about architectural design

more than before

Epilogue

Epilogue

- Processes
- Tooling
  - Xcode templates
  - Scripting
  - Anything that reduces boilerplate

Acknowledgements

- Jeff Gilbert and Conrad Stoll (Mutual Mobile)
  - “Inventors” of VIPER
- Robert C Martin *aka* Uncle Bob
- Objc.io
  - Sara Lincoln (Issue #13 artwork)

Links

Founding article: https://www.objc.io/issues/13-architecture/viper/

Source: https://mutualmobile.github.io/blog/2013/12/04/viper-introduction/

MVVM: https://www.objc.io/issues/13-architecture/mvvm/

Outware Mobile’s VIPER demo project: https://github.com/outware-mobile/om-viper

Clean Architecture by Robert C Martin

https://blog.8thlight.com/uncle-bob/2012/08/13/the-clean-architecture.html

https://vimeo.com/43612849

UIKonf 2015 - Brian Gesiak: iOS API Design: Swift Patterns - “Parameter Objects”

https://www.youtube.com/watch?v=yu6KND7dJBA

Justin Spahr-Summers - Enemy of the State

https://www.youtube.com/watch?v=7AqXBuJOJkY

WWDC 2014 - Session 229: Advanced iOS Application Architecture and Patterns

Theses slides: https://github.com/jeanetienne/viper-slides

Jet

@jeanetienne

Outware
mobile

Thank you!