

Functional Routing

iOS Presentation Layer

UIViewController

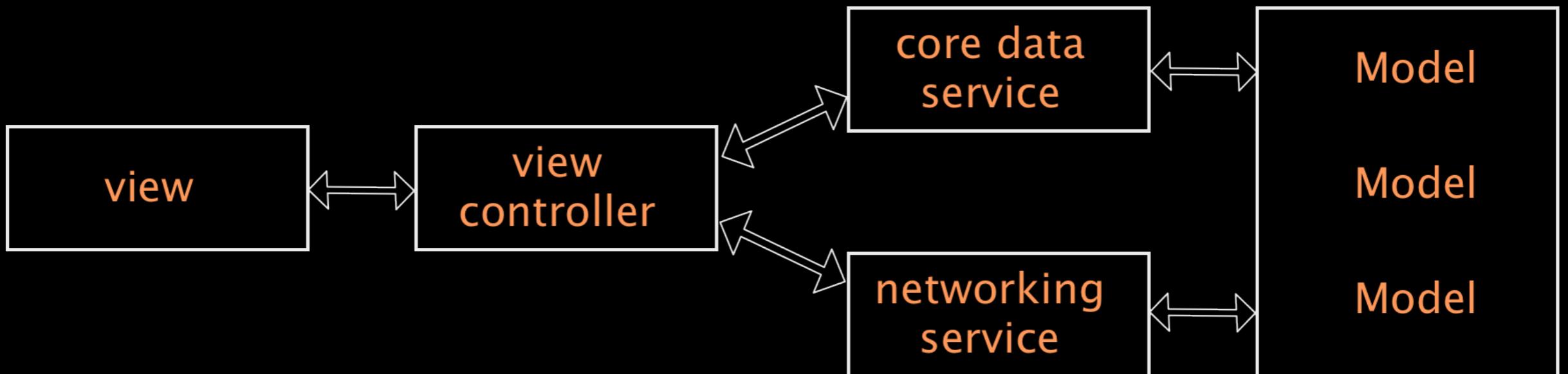
**What to do with
this?**

MVC, MVVM, VIPER, VIP

**How does it look
like?**

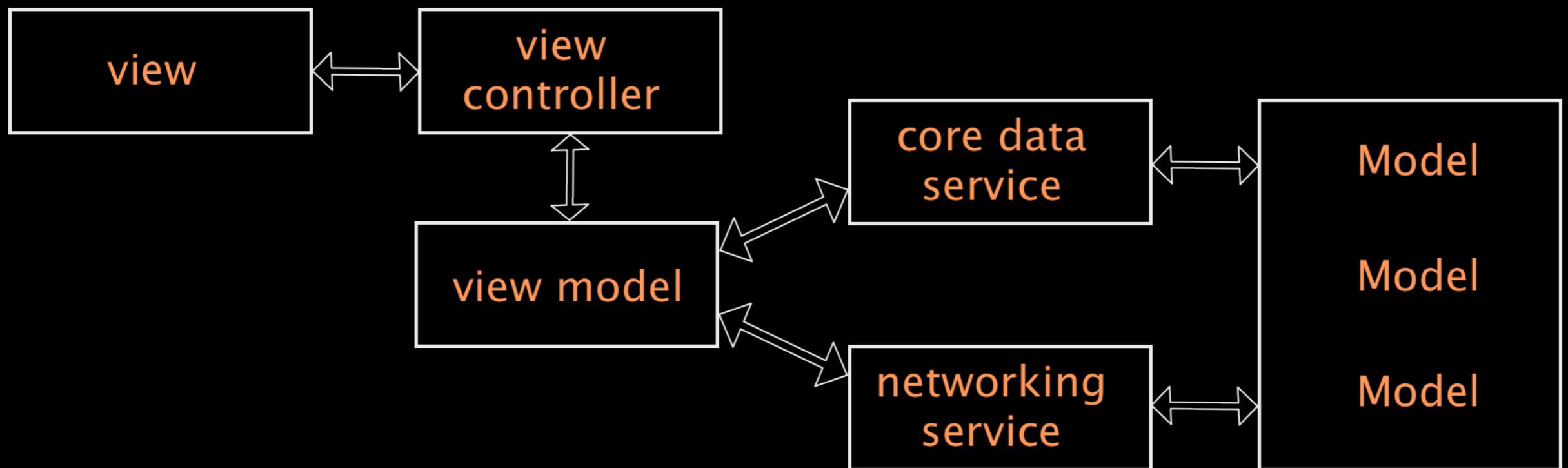
MVC

Model – View – Controller



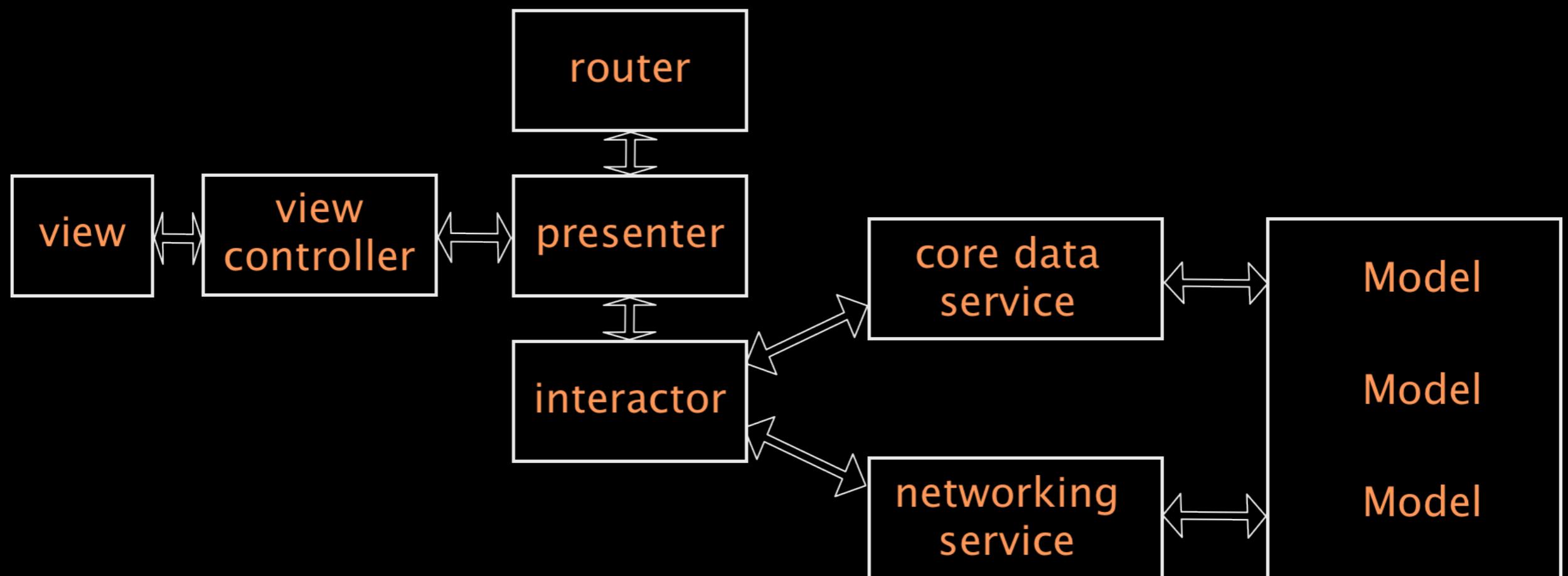
MVVM

Model – View – ViewModel



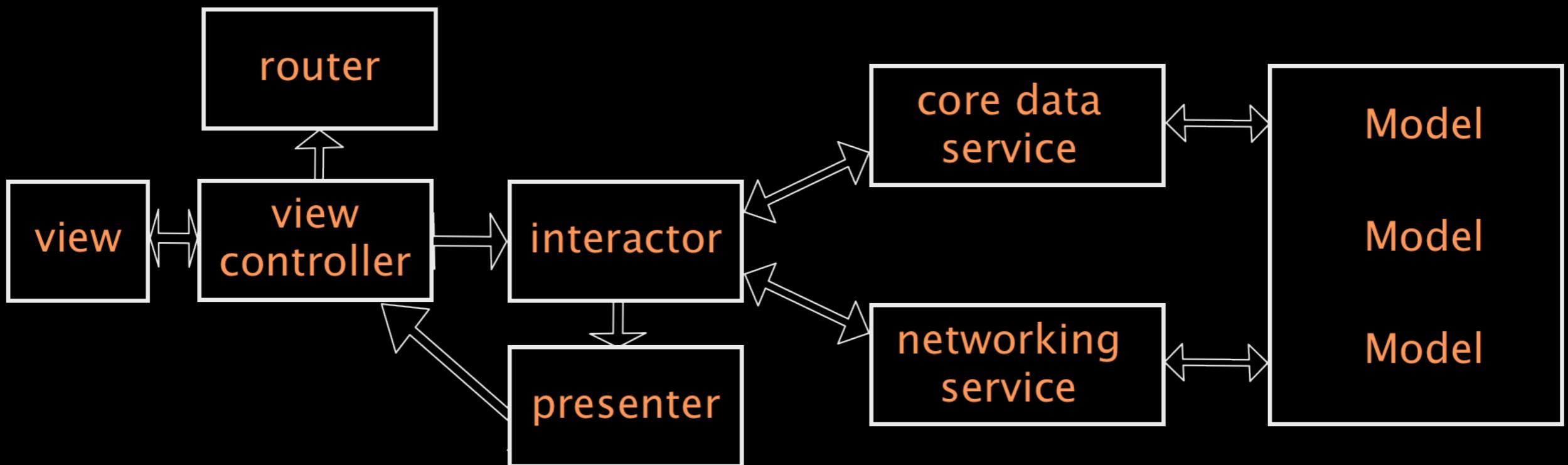
VIPER

View – Presenter – Interactor – Entity – Router



VIP (Clean Swift)

View – Interactor – Presenter



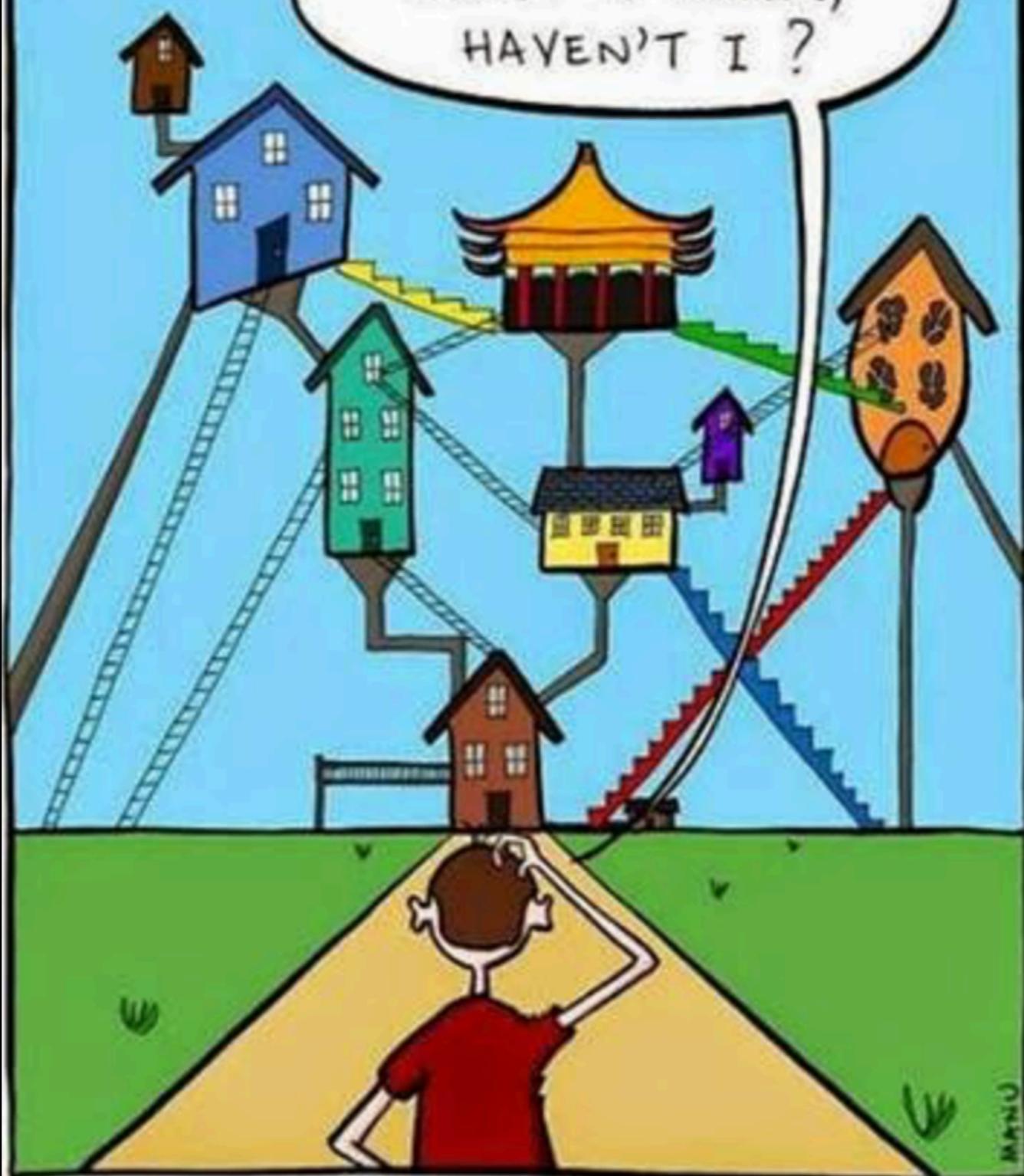
THE LIFE OF A SOFTWARE ENGINEER.

CLEAN SLATE. SOLID
FOUNDATIONS. THIS TIME
I WILL BUILD THINGS THE
RIGHT WAY.

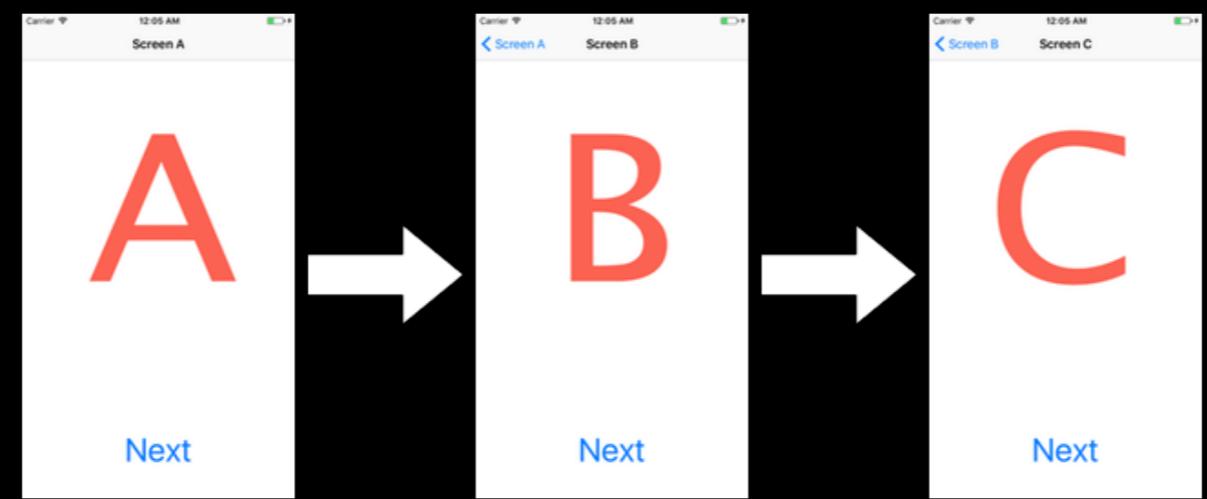


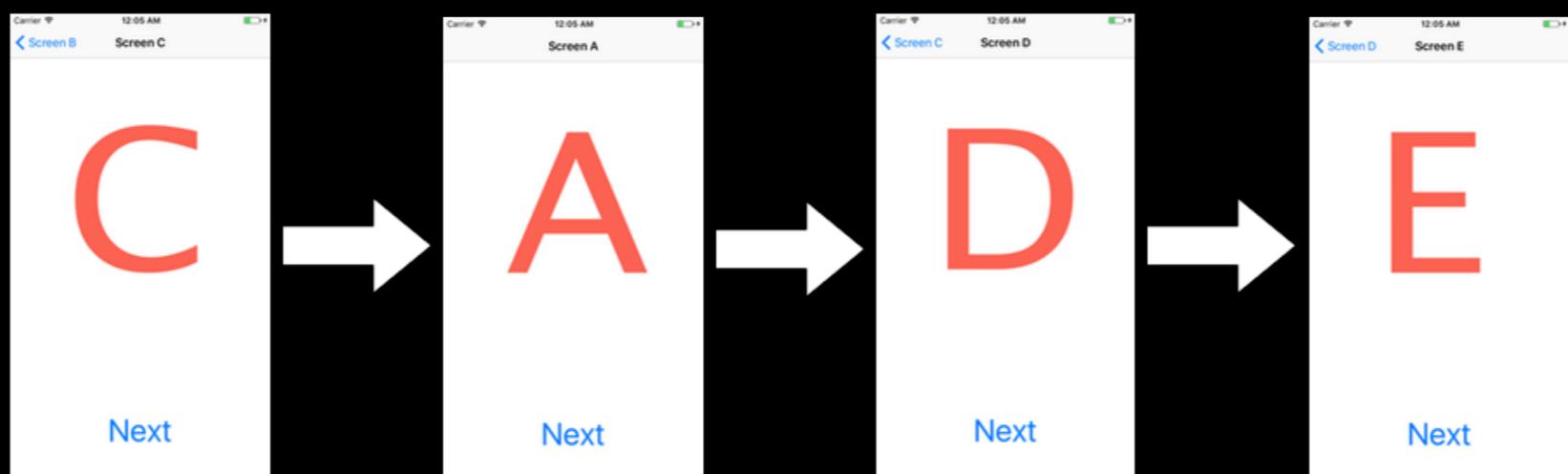
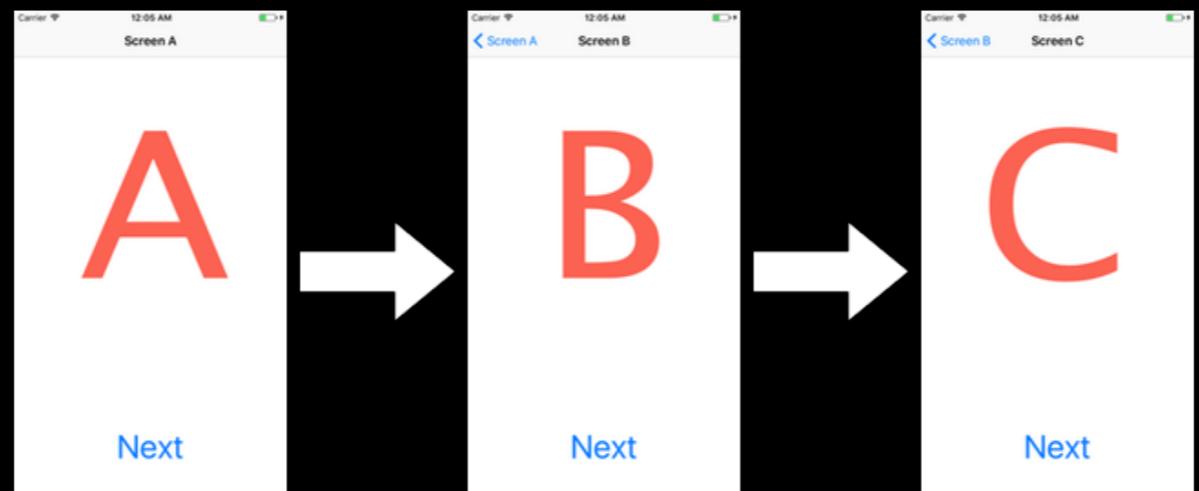
MUCH LATER...

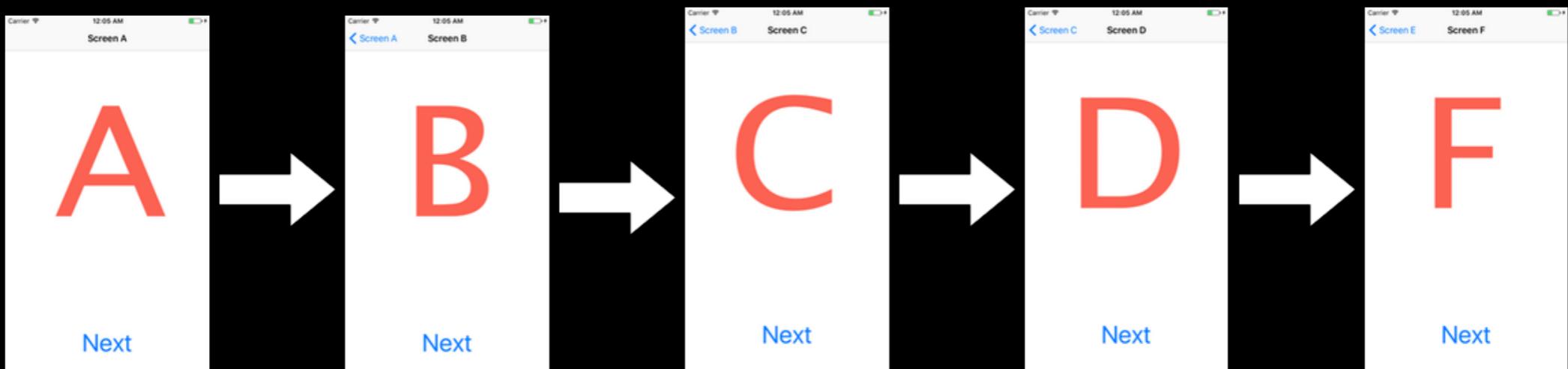
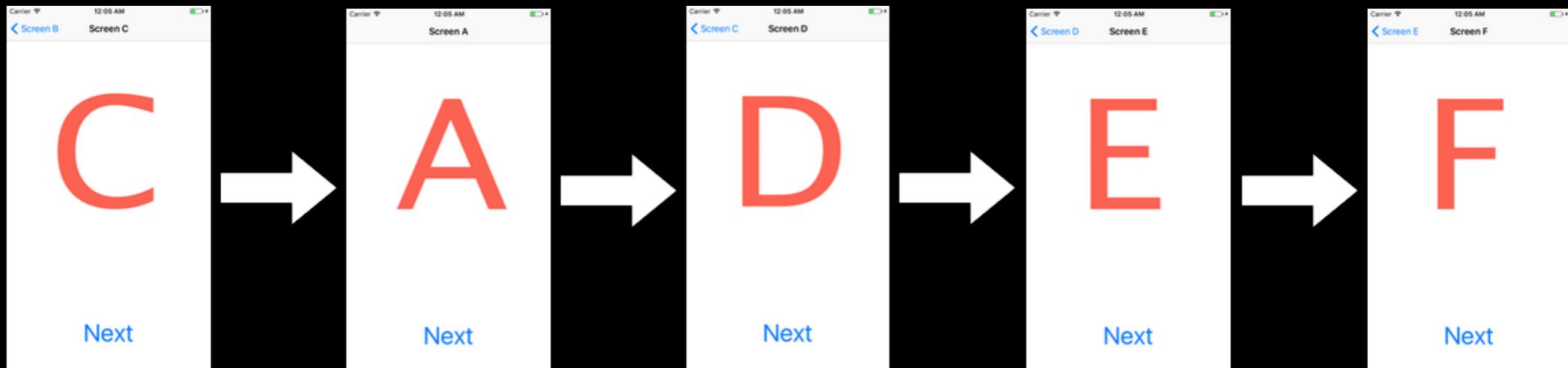
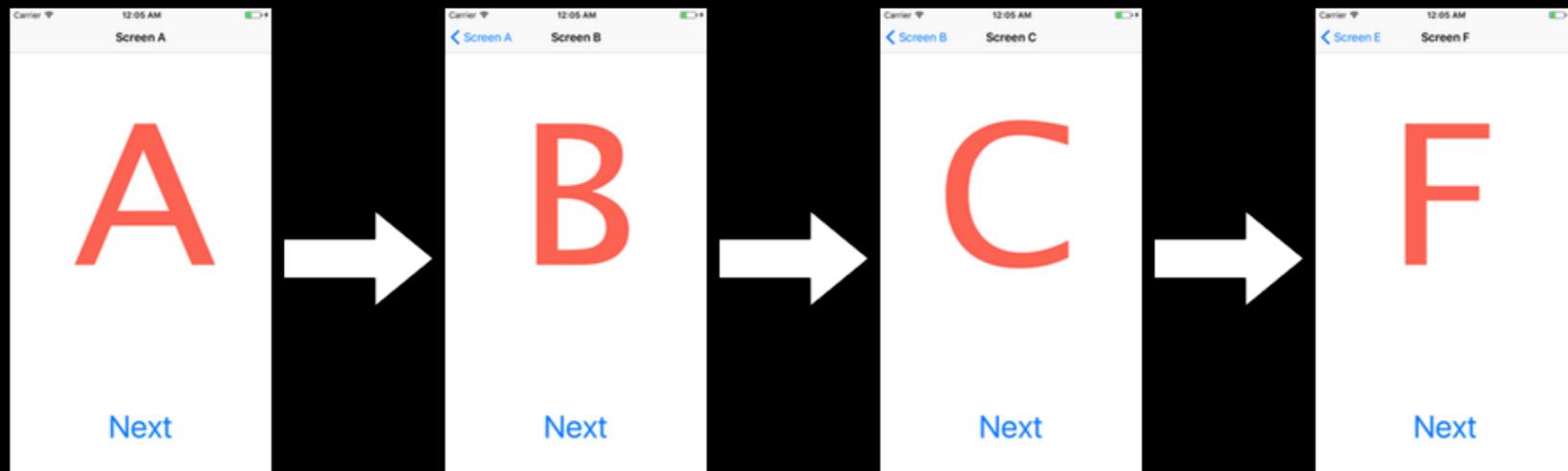
OH MY. I'VE
DONE IT AGAIN,
HAVEN'T I ?



But why?

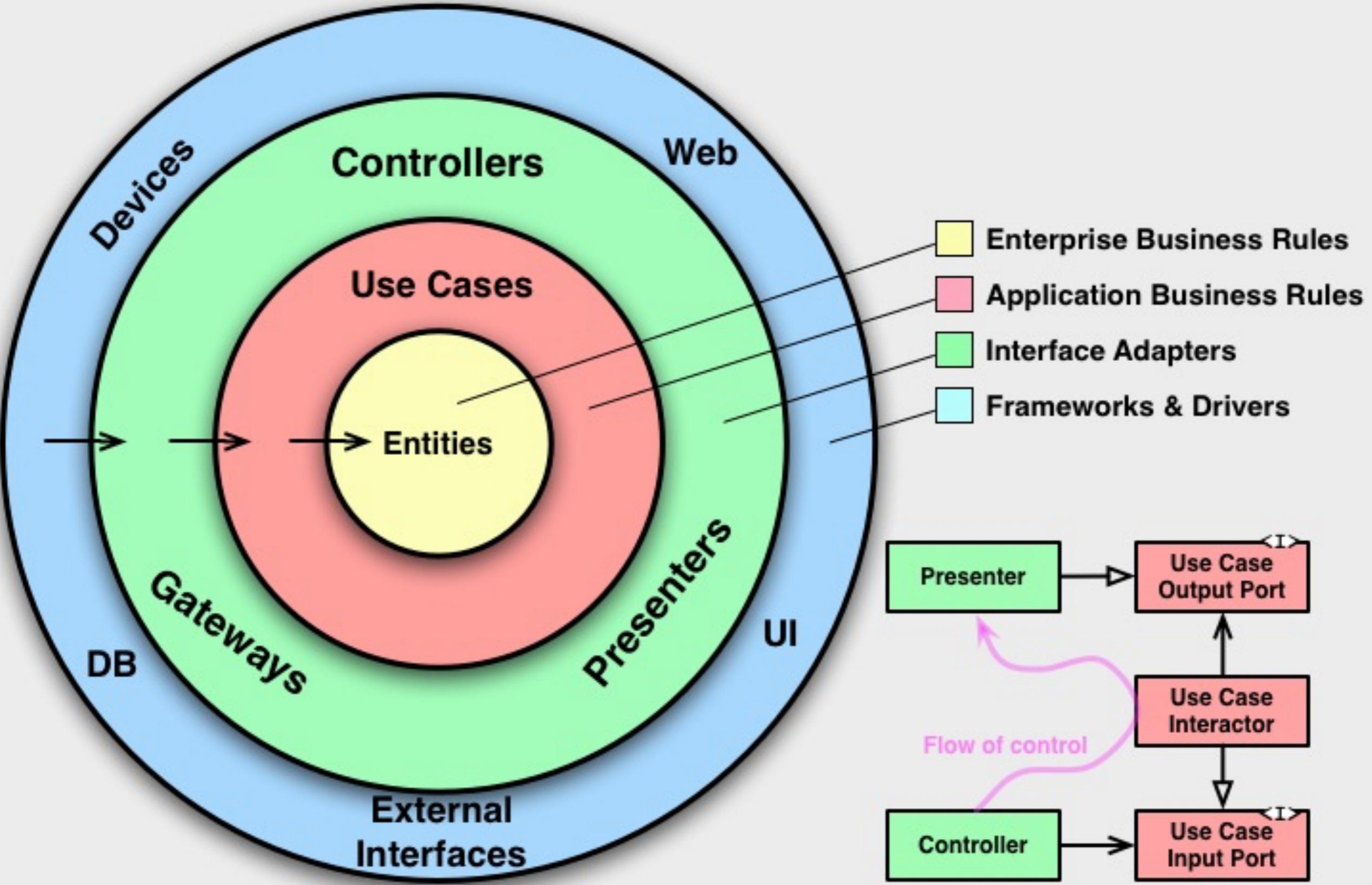






**why
Router?**

The Clean Architecture



*Let's rethink what
the interface is*

Interface

```
func interface(configuration: Configuration,  
completion: (Result) -> ())
```

Interface

```
func interface(configuration: Configuration,  
completion: (Result) -> ())  
  
struct ViewController<Result> {  
    let run: (@escaping (Result) -> ()) -> UIViewController  
}
```

Navigation

```
struct NavigationController<Result> {
    let build: (_ f: @escaping (Result, UINavigationController) -> ())
-> UINavigationController
}
```

Let's do some magic

**What have we
learned?**

Resources

- #24 Architecture Wars – A New Hope, swifting.io
- Functional View Controllers, Chris Eidhof
- The Clean Architecture, 8thlight.com

Questions?

Thank you

-  @DmitriiCelpan
-  lexorus
-  /  celpand