초격차 패키지 Online.

Chapter 5.
Todo 앱
- StateManagement

PART3 | 활용

1. Intro, 이론
2. 상태 관리 없이 앱 만들어보기, ValueNotifier, InheritedWidget이용
3. GetX로 변환해보기
4. Bloc로 변환해보기
5. RiverPod으로 변환해보기
6. RiverPod + Flutter Hook 같이 사용해보기
7. 장단점 총 정리

Todo 앱
- StateManagement

1 Intro, 이론

State Management를 배워야 하는 이유?

=> 앱을 더 쉽게 개발하기 위해서

1. 최초 개발을 빠르게
2. 앱 스펙이 수정되었을때 필요한 부분을 빠르게 찾고 수정하기 위해서

Flutter 이전에도 좋은 개발 방법을 위한 다양한 시도들이 있었다.

=> Android와 iOS의 개발 패턴

Android와 iOS의 개발 패턴

MVC : Model View Controller

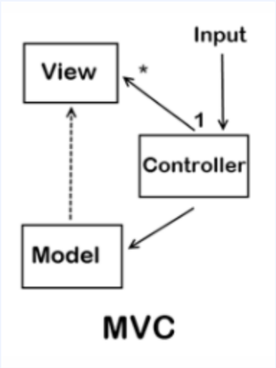
MVP : Model View Presenter

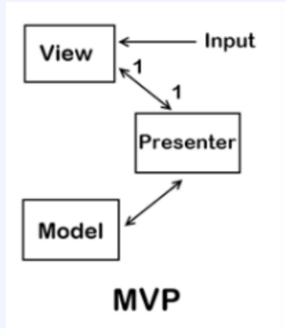
MVI : Model View Intent

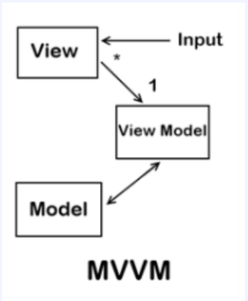
VIPER: View Interactor Presenter Entity Router

MVVM: Model View ViewModel

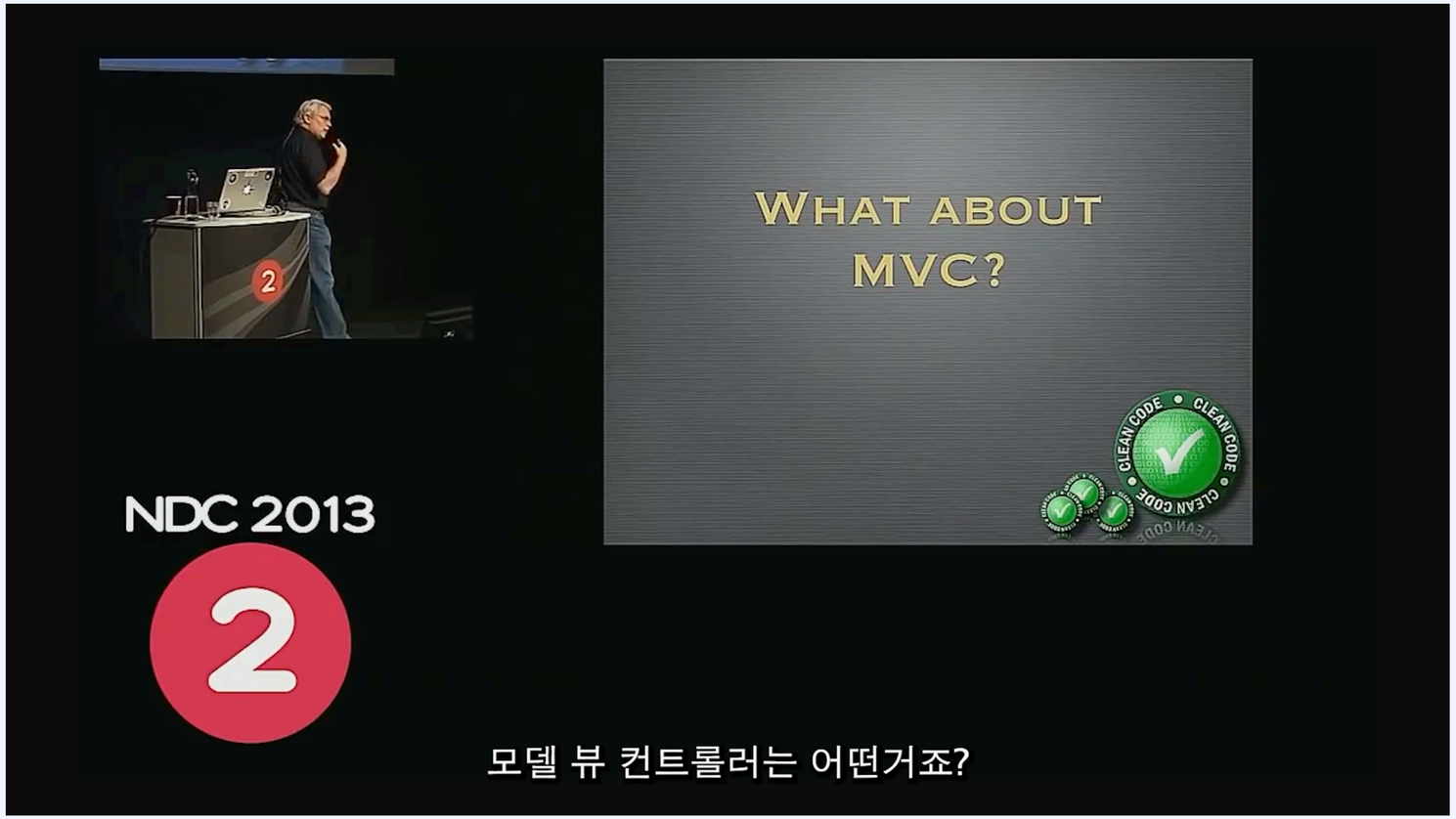
... etc







https://www.thetechedvocate.org/mvc-mvp-mvvm-which-one-to-choose/



출처: https://www.youtube.com/watch?v=Nsjsiz2A9mg

1970년대에 탄생한 MVC의 오해 (Model View Controller):

1. 애초에 제안이 됐을때 부터
   작은 컴포넌트를 위한 설계였음

2. MVVM등에서 말하는 데이터 Observing이 이미 포함되었던 개념

1970년대에 탄생한 MVC의 오해 (Model View Controller):

3. 또 Controller는 키보드와 마우스의 유저 제스쳐 콘트롤을 의미했던 것

그러나 실제 모바일에서는 (모던 웹/앱) 화면+위젯 자체가 콘트롤러

그래서 탄생한 MVP (Model View Presenter):

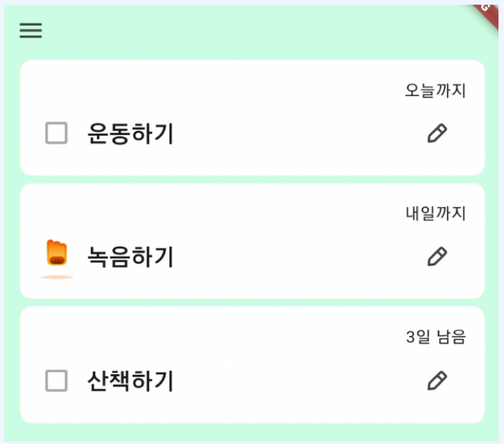
Android에서는 View가 XML

iOS에서는 View가 View Controller & Story Board

코드와 분리되어있는 이 View Layer를 통제할 Presenter가 필요했음

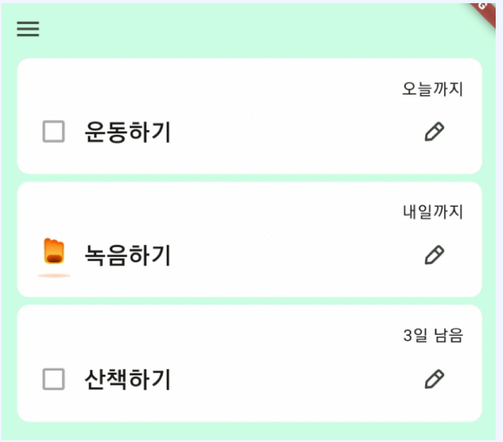
하지만 불편했던 MVP

presenter.showLoading()



presenter.showList(listItems)

때마침 우리를 (잠시) 구원해준 MVVM



viewModel.state = State.Loading

viewModel.list = listItems

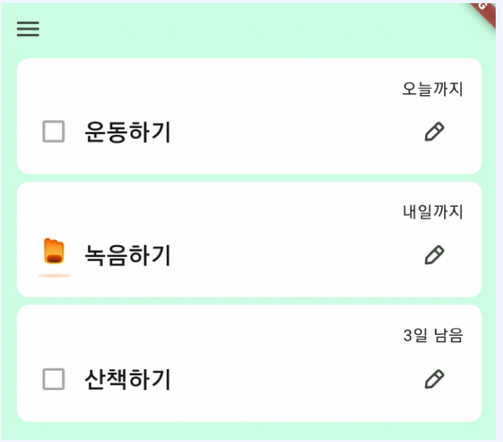
그래도 불편한 구석

Android에서는 View가 XML

iOS에서는 View가 View Controller & Story Board

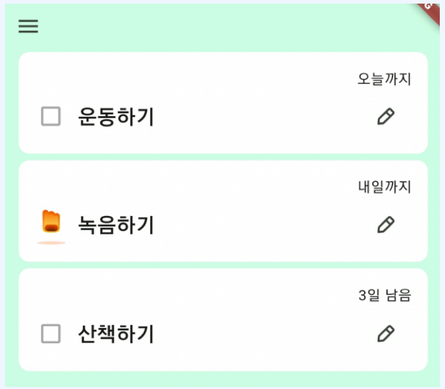
최종 BOSS - 선언형 UI (Declarative UI)

ToListView(list: null)



ToListView(list: todoList)

이제는 View의 패턴 구조 논의할 필요가 없어짐 대신에



ToListView(list: todoList)

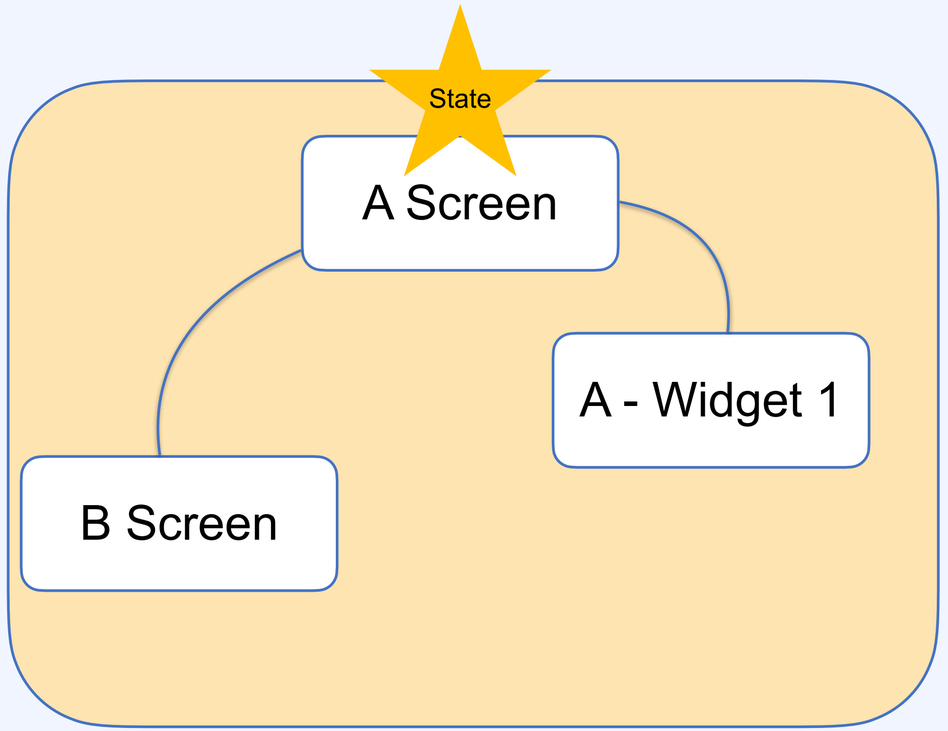
Data & State를 어떻게 관리할지? <= 중요해짐

State Management 구조의 종류

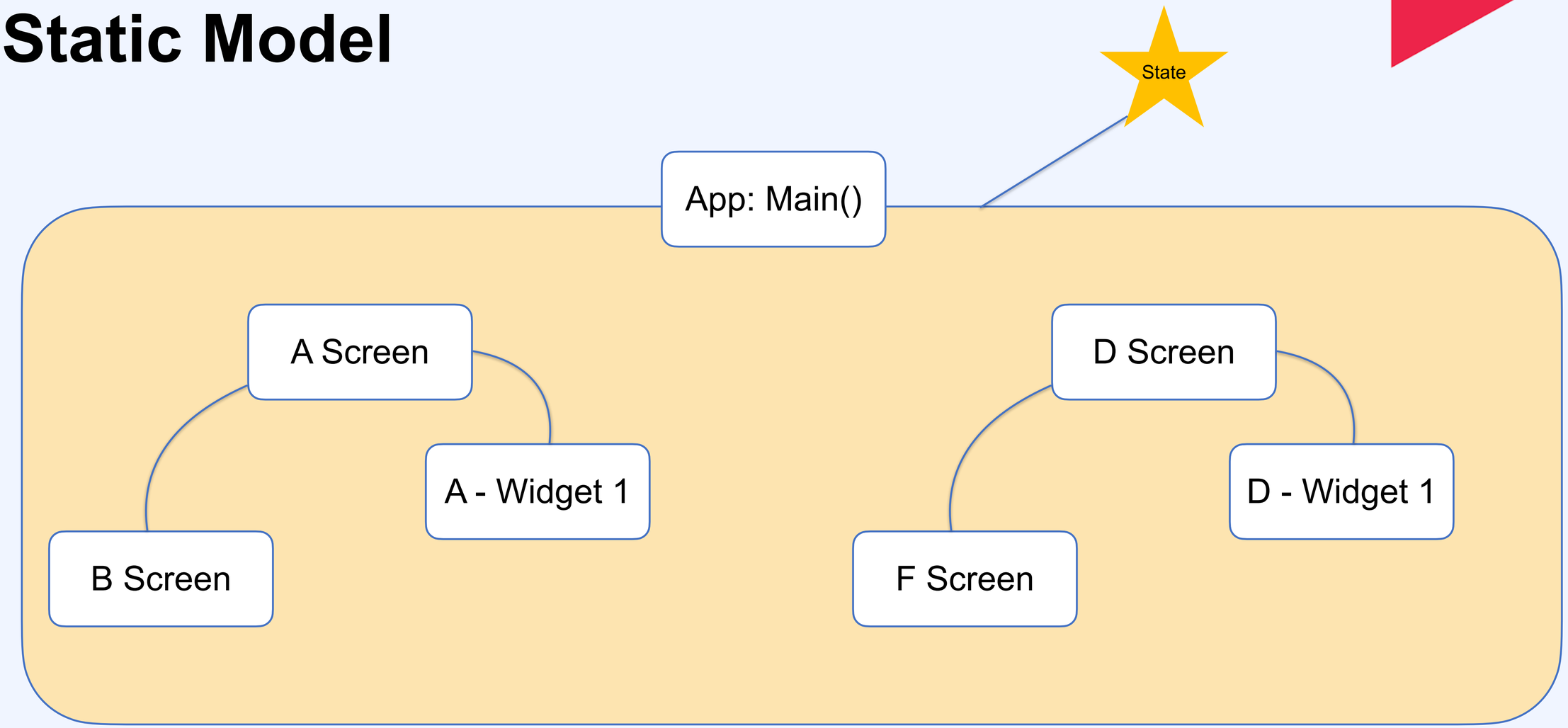
Scoped Model

Static Model

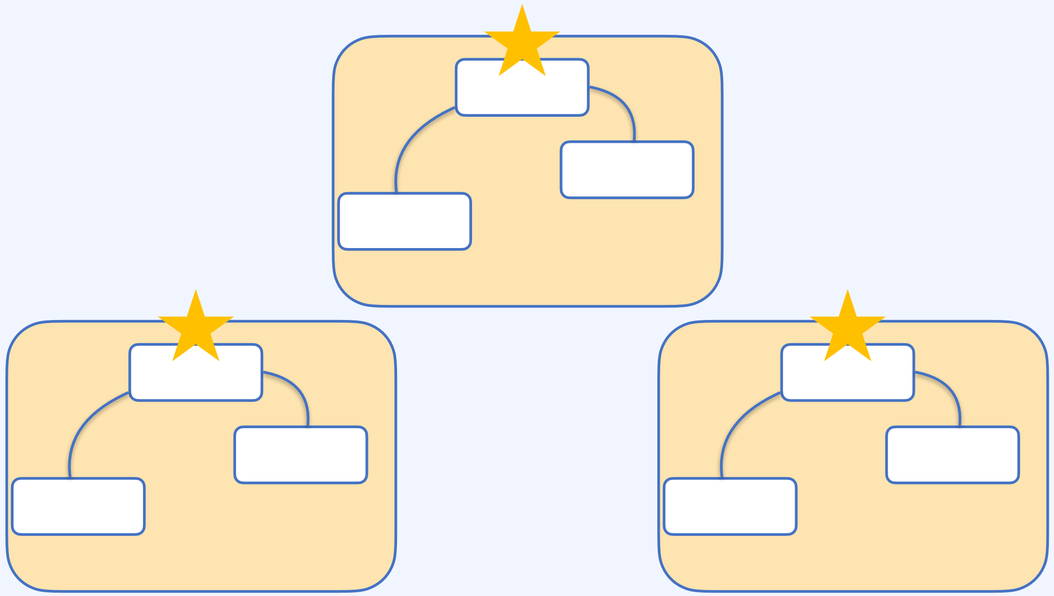
Scoped Model



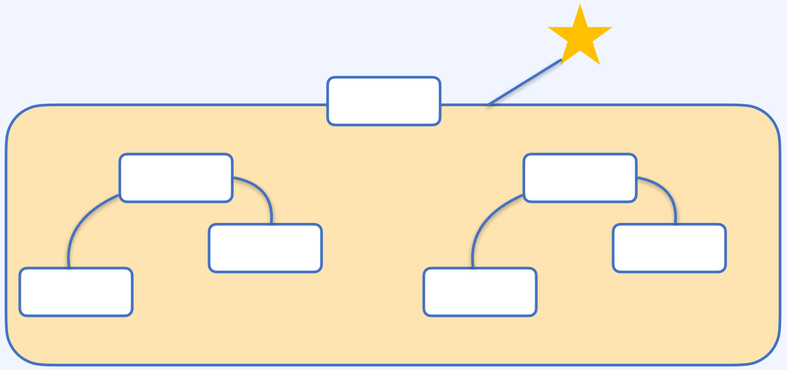
Static Model

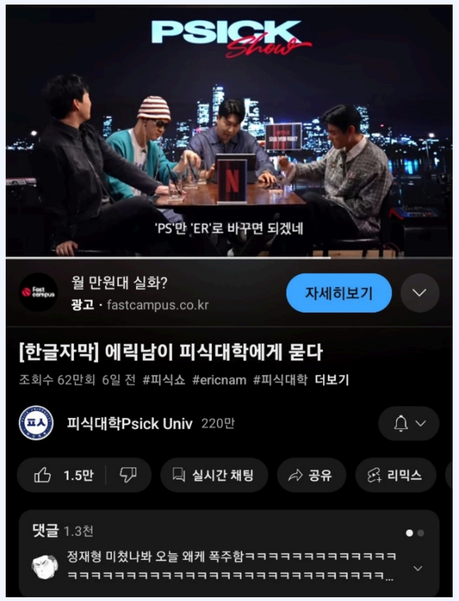


Scoped Model

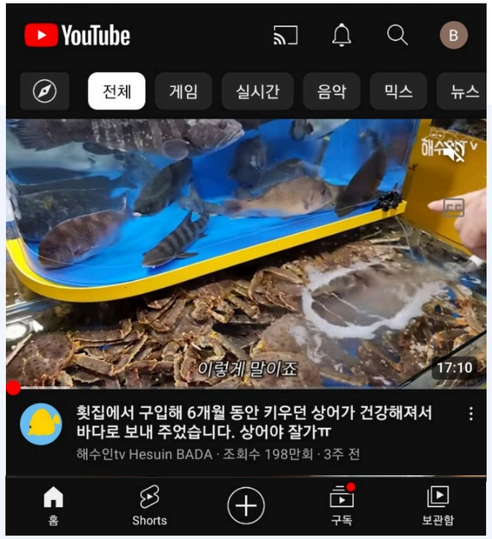


Static Model

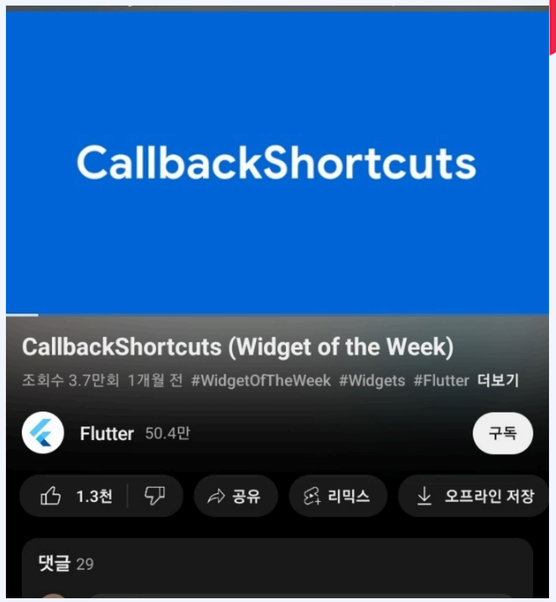




VideoScreen: id = "abcd456"



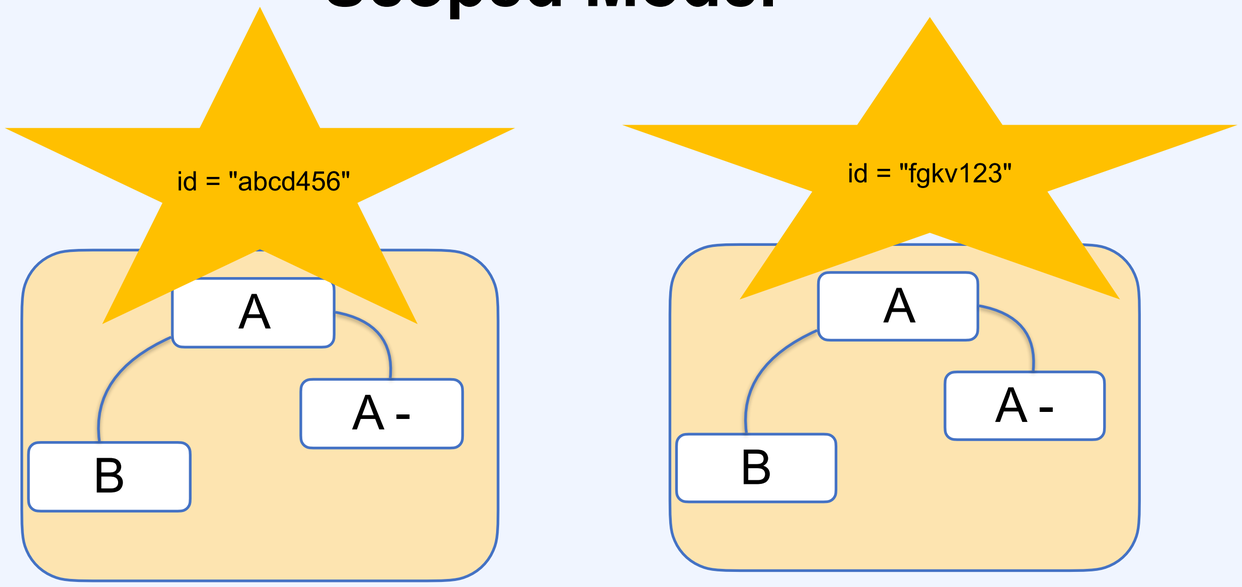
Home Screen



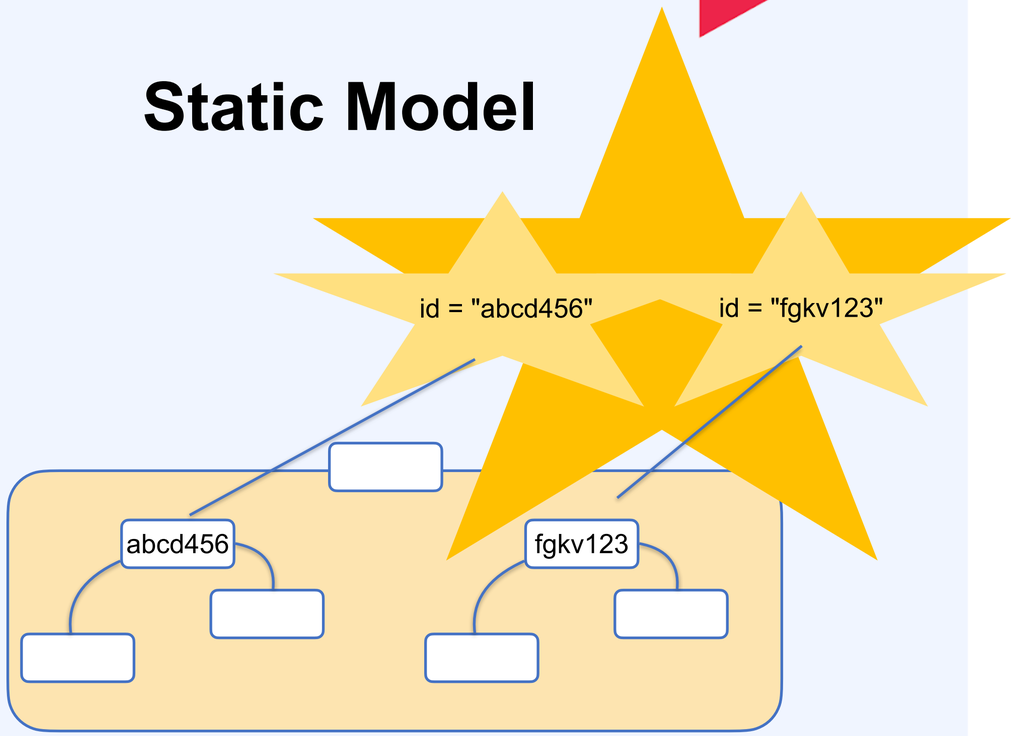
VideoScreen: id = "fgkv123"

Scoped Model? Or **Static Model?**

Scoped Model

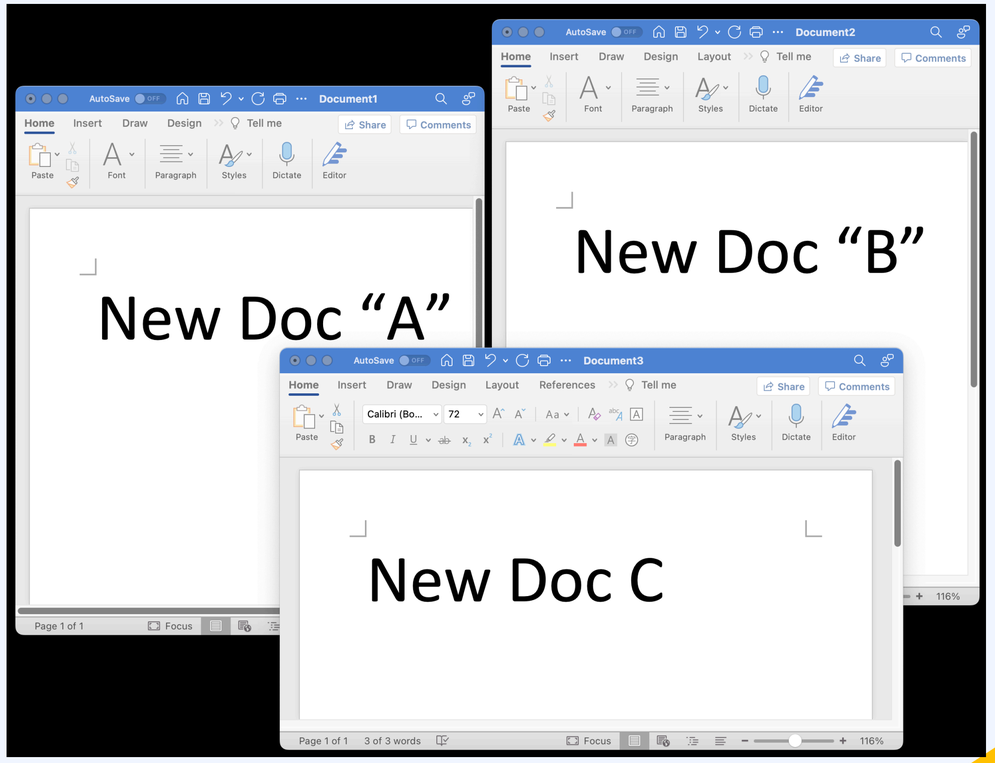


Static Model



1. Provider나 Scope를 상위에 지정
2. Key(Provider, Bloc), Family (Riverpod)

Tag(GetX)



Scoped Model? Or Static Model?

Scoped Model? Or Static Model?

스펙에 따라서 결정하셔야 합니다.

Scoped Model

1. 해당 Scope 화면이 사라지면 자동으로 메모리 해제가 된다.
2. Child에서 데이터 참조를 할때 id(구별자)없이도 상위 데이터 참조 가능

Static Model

1. 메모리 관리를 개발자가 직접 관리해야한다.
2. Child에서 데이터 참조를 할때 id(구별자)가 항상 필요하다.
3. Scope 모델보다 구현이 쉽다.
4. 모든 코드에서 원하는 State에 접근하고 수정이 가능하다.

개발을 더 좋게 하기 위해서
+ 테스트가 가능한 환경을 위해서

- 개발 패턴
- State Management

Todo 앱 만들기 실습