

☀수열님의 ReactorKit





ReactorKit 이란?

반응형 단방향 앱을 위한 프레임워크

ReactorKit 이란?

Flux + Reactive Programming

ReactorKit 이란?

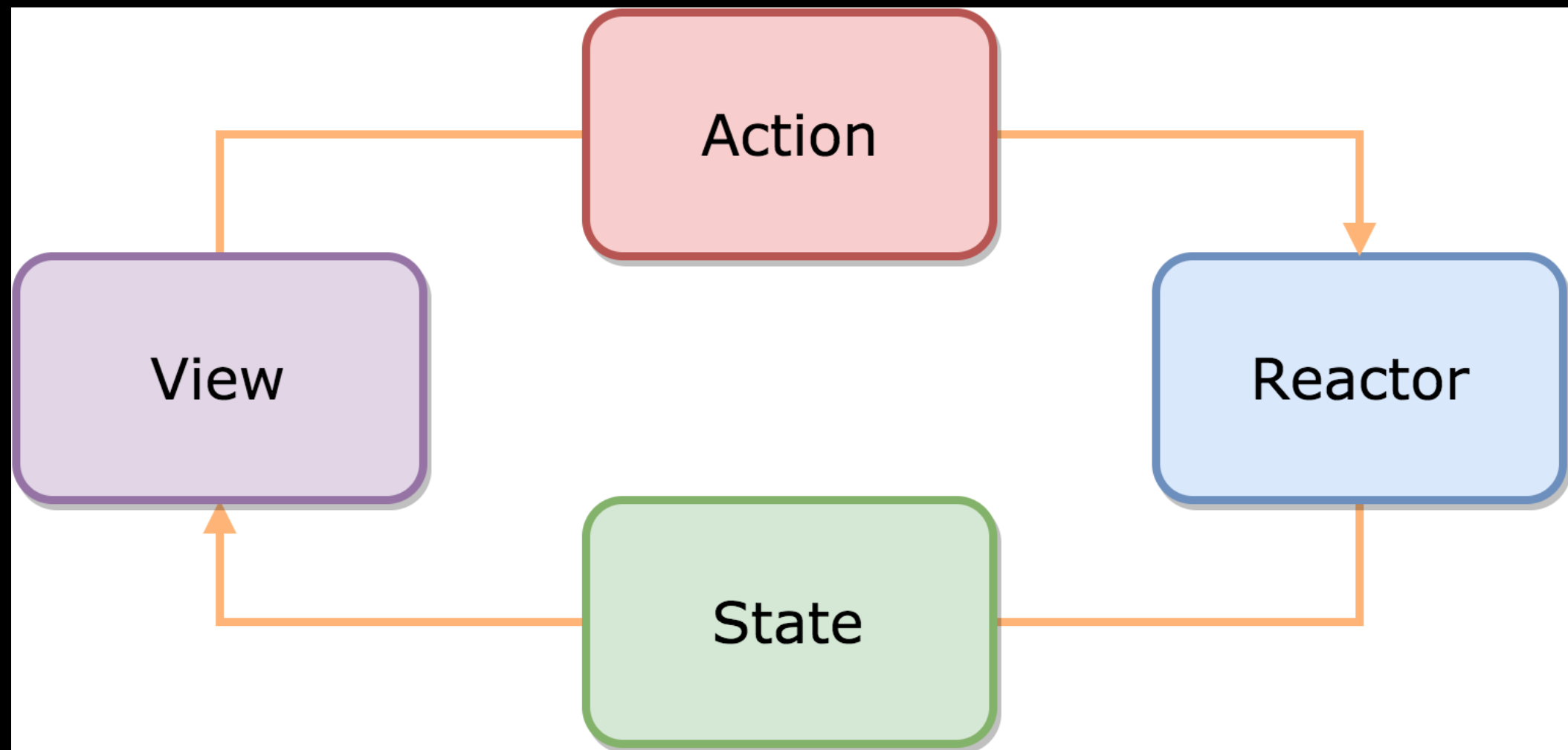
사용자 인터랙션과 뷰 상태가 관찰 가능한 스트림을 통해 단방향으로 전달

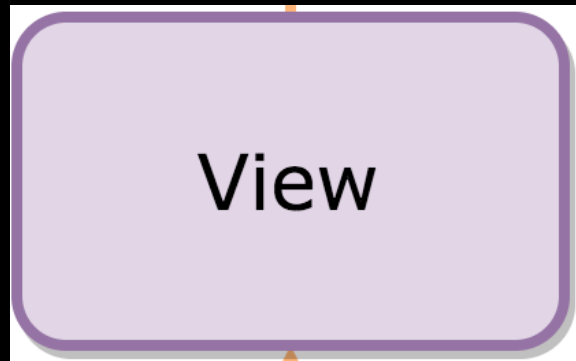
그래서 뭐가 좋은데?

뷰와 비즈니스 로직을 분리할 수 있게 되면서 모듈간 결합도가 낮아지고 테스트하기 쉬워짐

비동기 코드를 일관되게 작성할 수 있게 됨

데이터 흐름

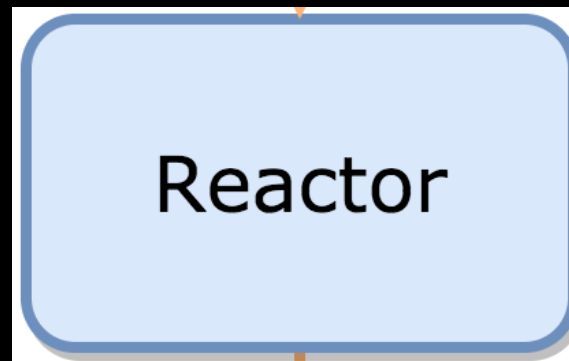




상태를 표현

사용자 인터랙션을 추상화하여 리액터에 전달

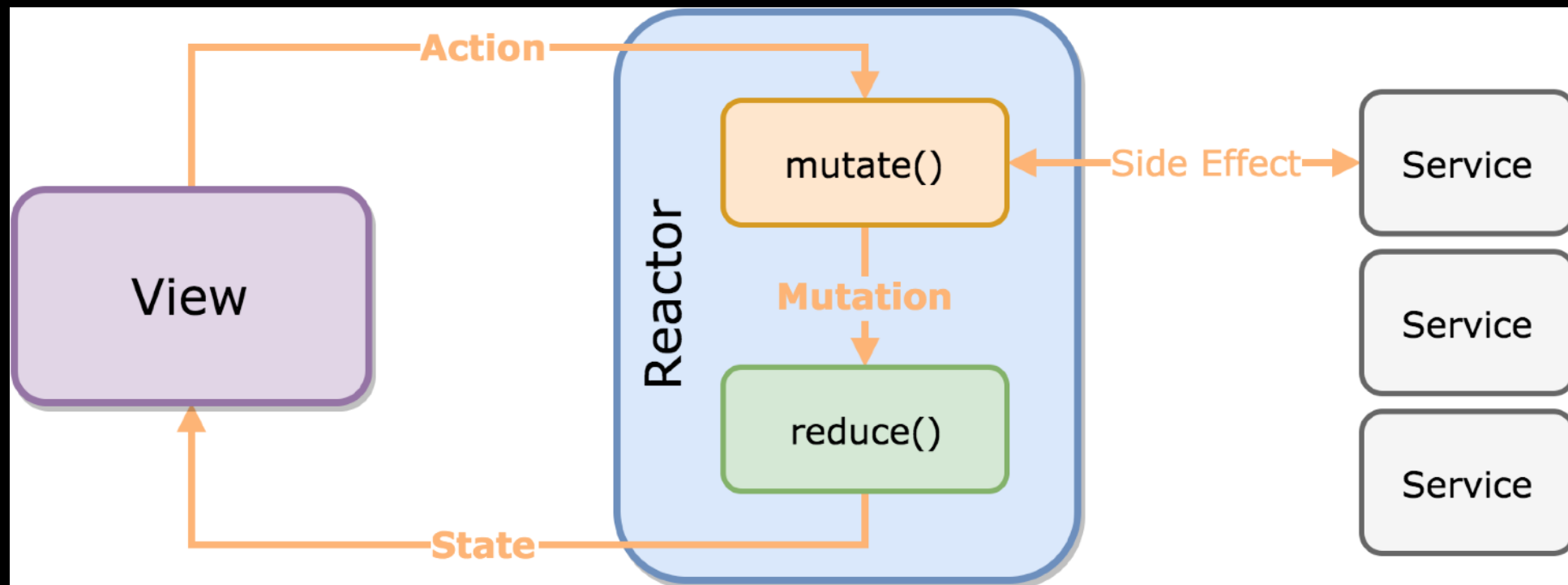
리액터에서 전달받은 상태를 각각의 뷰 컴포넌트에 바인딩



뷰의 상태 관리

뷰에서 액션을 전달받으면 비즈니스 로직을 수행한 뒤 상태를 변경하여 다시 뷰에 전달

리액터는 UI 레이어에서 독립적이기 때문에 비교적 테스트가 용이



적용은 어떻게 해?

적용은 어떻게 해?

```
import ReactorKit
import RxSwift

class UserViewController: UIViewController, View {
    var disposeBag = DisposeBag()

    func bind(reactor: UserViewReactor) {
    }
}
```

적용은 어떻게 해?

```
func bind(reactor: UserViewReactor) {  
    // Action  
    self.followButton.rx.tap  
        .map { Reactor.Action.follow }  
        .bind(to: reactor.action)  
        .disposed(by: self.disposeBag)  
  
    // State  
    reactor.state.map { $0.isFollowing }  
        .distinctUntilChanged()  
        .bind(to: self.followButton.rx.isSelected)  
        .disposed(by: self.disposeBag)  
}
```

적용은 어떻게 해?

```
import ReactorKit
import RxSwift

final class UserViewReactor: Reactor {
    enum Action {
        case follow
    }

    enum Mutation {
        case setFollowing(Bool)
    }

    enum State {
        var isFollowing: Bool
    }

    let initialState: State = State(isFollowing: false)
}
```

출처

<https://www.slideshare.net/devxoul/reactorkit/1>

<https://medium.com/styleshare/reactorkit-시작하기-c7b52fbb131a>

<https://eunjin3786.tistory.com/100>