

Session 2

Keynote

Madagascar 2021

Coordinator 누구세요?

신윤아

Head of Madagascar Group



Coordinator와 인사하기

Madagascar 2021

Coordinator

안녕하세요. 마다가스카 코딩 클럽

저는 여러분들의 화면전환을 책임지는
Coordinator 입니다.

쉽고 편하게 화면 전환할 수 있도록 제가 도와드릴게요.
저만 믿고 따라와주세요!

C

Madagascar Duna

Coordinator씨 안녕하세요
자기소개를 부탁드립니다 될까요?



Madagascar Duna

제가 Coordinator씨를 기깔나게
설명해드리겠습니다.



Madagascar 2021

Coordinator

Pattern

DI

WHY

Example

Coordinator

Pattern

DI

WHY

Example

Coordinator가 도대체 어떤 패턴인지 함께 알아보시다.

Coordinator

Pattern

DI

WHY

Example

Coordinator가 해결했다는 DI(Dependency Injection), 의존성 주입은 도대체 뭔가요?
의존성 주입이 중요한가요?

Coordinator

Pattern

DI

WHY

Example

Coordinator 패턴도 알겠고 DI도 이제 이해가 다 됐는데
Coordinator를 꼭 써야하는 이유는 모르겠는데요?

Madagascar 2021

Coordinator

Pattern

DI

WHY

Example



자 이제, 실전이다!

Coordinator Pattern

Madagascar 2021

화면 전환을 조금 더 flexible하게 사용할 수 있도록 만들어주는 패턴

하나의 책임만 담당하는 디자인 패턴

DI를 해결한 패턴

VC를 더 쉽게 재사용

계층 관리가 용이한 패턴

전반적인 애플리케이션 구조를 향상시킬 수 있는 놀라운 가능성이 있는 패턴

Dependency Injection

Madagascar 2021

```
class Aclass {  
    var number: Int = 0  
}  
  
class Bclass {  
    var internalVariable = Aclass()  
}  
  
let b = Bclass()  
print(b.internalVariable.number)
```

하나의 클래스는 다른 클래스에 의존(Depend)하게 된다



결합도(coupling)가 높아진다



한 클래스를 수정하였을 때, 다른 클래스도 수정해야 하는 상황 발생


```
class Aclass {  
    var number: Int = 0  
}  
  
class Bclass {  
    var internalVariable = Aclass()  
}
```

의존성 주입을 통해서 독립적으로 만들어주자!

하나의 클래스는 다른 클래스에 의존(Depend)하게 된다



결합도(coupling)가 높아진다



한 클래스를 수정하였을 때, 다른 클래스도 수정해야 하는 상황 발생

Dependency Injection

Madagascar 2021

```
class Aclass {
    var number: Int = 0
}

class Bclass {
    var internalVariable: Aclass

    init(withExternalVariable variable: Aclass) {
        self.internalVariable = variable
    }
}

let b = Bclass(withExternalVariable: Aclass())
print(b.internalVariable.number)
```

외부에서 Aclass를 Bclass에 주입합니다

제어의 주체가 외부에 존재합니다.

Dependency Injection

Madagascar 2021

```
protocol DependencyIndependentInterface: AnyObject {
    var number: Int { get set }
}

class Aclass: DependencyIndependentInterface {
    var number = 1
}

class Bclass {
    var internalVariable: DependencyIndependentInterface

    init(withExternalVariable variable: DependencyIndependentInterface) {
        self.internalVariable = variable
    }
}

let b = Bclass(withExternalVariable: Aclass())
print(b.internalVariable.number)
```

DI는 의존성을 분리시켜서 사용합니다.(의존관계 역전 원칙)

Interface를 통해 의존관계가 독립되는 상황 발생

Dependency Injection

Madagascar 2021

클래스나 구조체의 책임이 더욱 명확

Unit Test 용이

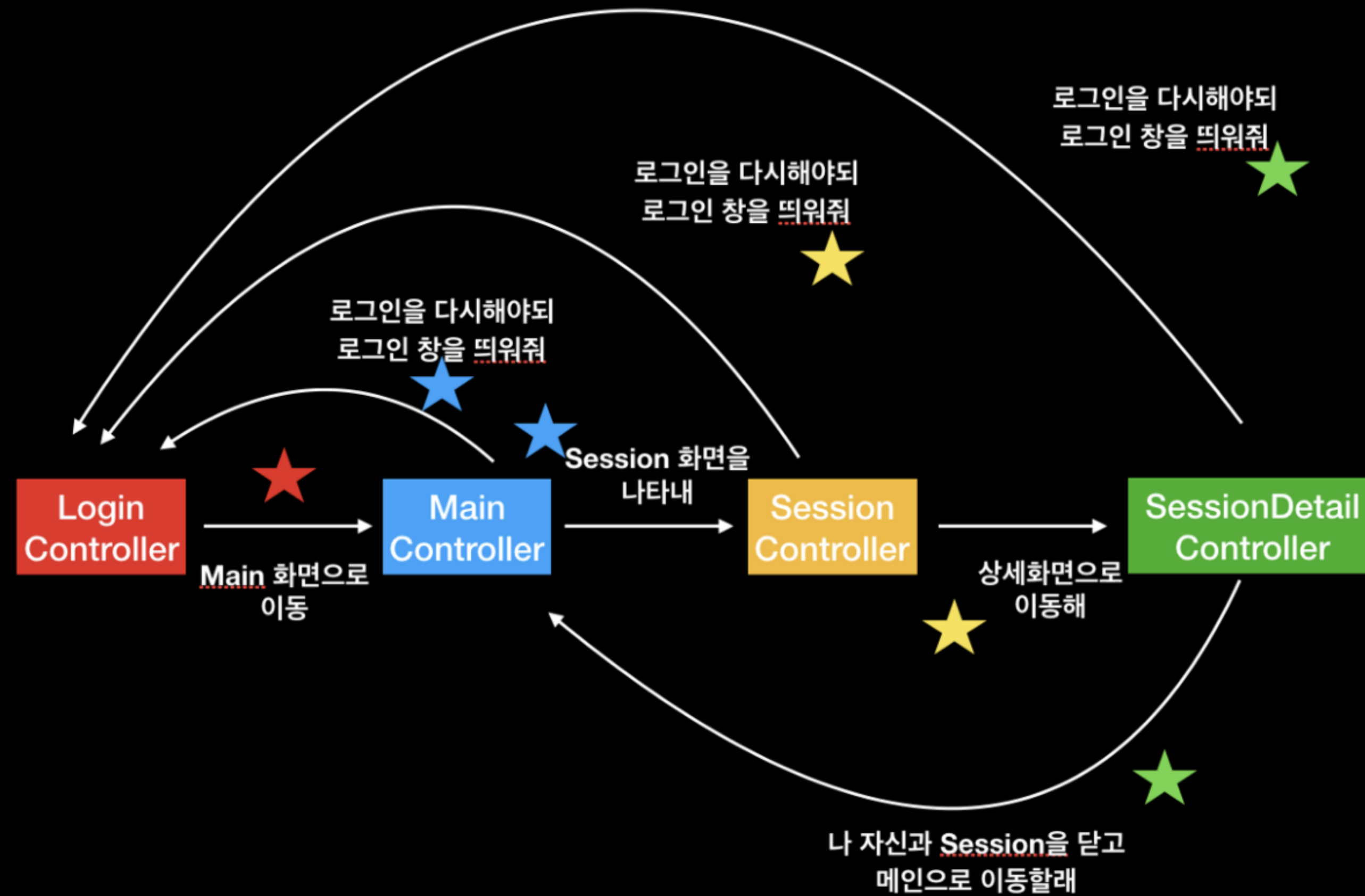
코드의 재활용성을 높임

결합도를 낮추면서 유연한 코드 작성

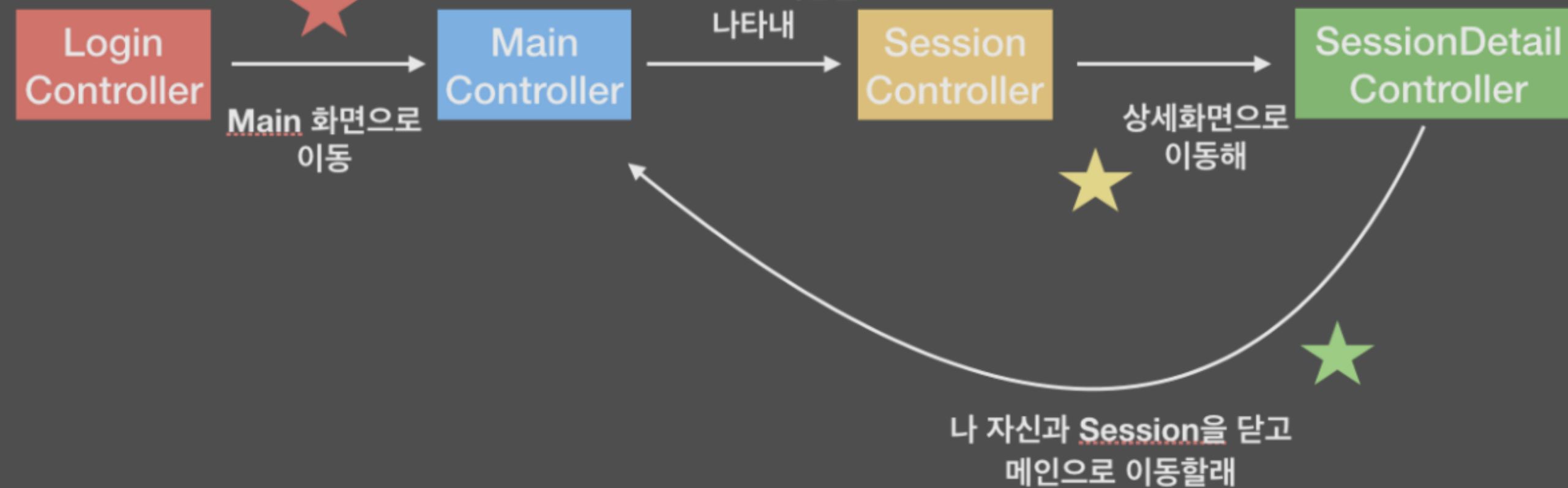
의존성(종속성)을 줄이거나 없앴

Why

Madagascar 2021

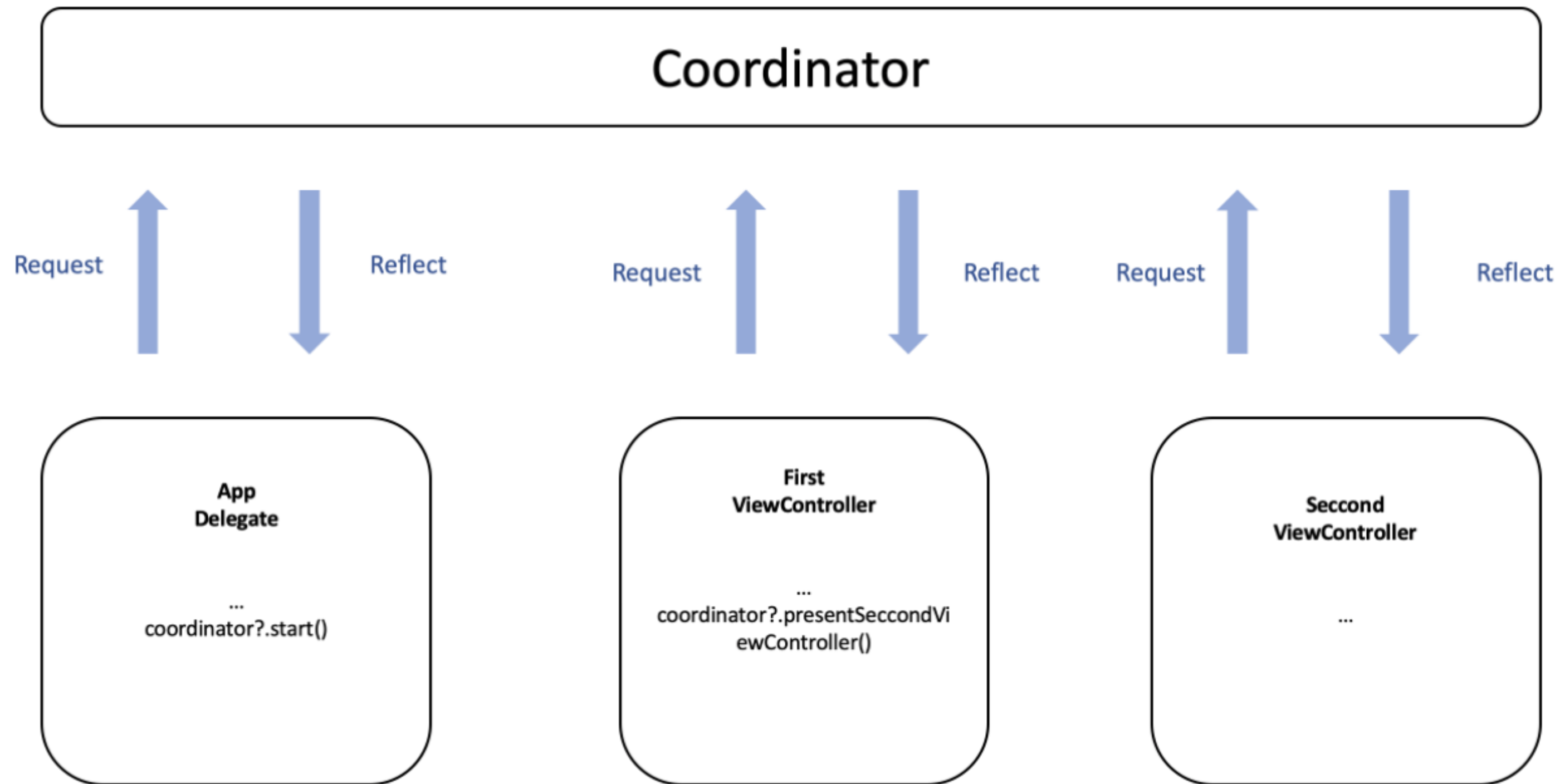


서로 종속적인 ViewController의 모습



Why

Madagascar 2021



Why

Madagascar 2021



Example

쉽게 적용해보는 Coordinator



심화 Coordinator

```
protocol CheckUserCoordinator: AnyObject {
    func presentTo(_ viewController: UIViewController)
}

extension CheckUserCoordinator {
    func presentTo(_ viewController: UIViewController) {
        let vc = CheckVC()
        viewController.present(vc, animated: true, completion: nil)
    }
}
```

delegate protocol로 전환하는 Coordinator를 생성

» CheckUserCoordinator

extension를 통해서 필요한 작업들을 구현합니다.

» presentTo 함수를 통해서 CheckVC를 Present하는 코드 구현

Example

Madagascar 2021

```
class LoginVC: UIViewController {
    private let signInButton = UIButton().then {
        $0.addTarget(self, action: #selector(didTappedLoginButton), for: .touchUpInside)
    }

    weak var coordinator: CheckUserCoordinator?

    @objc
    private func didTappedLoginButton(_ sender: UIButton) {
        coordinator?.presentTo(self)
    }
}
```

```
class SignUpVC: UIViewController {
    private let signUpButton = UIButton().then {
        $0.addTarget(self, action: #selector(didTappedSignupButton), for: .touchUpInside)
    }

    weak var coordinator: CheckUserCoordinator?

    @objc
    private func didTappedSignupButton(_ sender: UIButton) {
        coordinator?.presentTo(self)
    }
}
```

```
class LoginVC: UIViewController {  
    private let ...  
    $0.addT ... (chUpInside)  
}  
  
weak var coordinator: CheckUserCoordinator?
```

같은 형식으로 다른 뷰에 사용 가능

coordinator로 화면 전환을 요청하기만 하면 된다

```
class Si ...  
priv ... (side)  
}
```

ViewController에서 수정할 필요가 없다

```
weak var ...  
  
@objc ...  
private ...  
    coordinator?.presentTo(self)  
}  
}
```

ViewController는 view에 집중 가능

Example

쉽게 적용해보는 Coordinator

심화 Coordinator



Example

Madagascar 2021

```
enum TransitionStyle {
    case root
    case push
    case modal
}

enum TransitionError: Error {
    case navigationControllerMissing
    case cannotPop
    case unknown
}
```

TransitionModel.swift

Transition이 되는 Style들을 구분

Transition이 되면서 나타나는 Error들을 구분

Example

Madagascar 2021

SceneCoordinatorType.swift

```
import RxSwift

protocol SceneCoordinatorType {
    /// 새로운 Scene를 표시
    @discardableResult
    func transition(to scene: Scene, from storyboard: String, using style:
        TransitionStyle, animated: Bool) -> Completable

    /// 현재 씬 닫기
    @discardableResult
    func close(animated: Bool) -> Completable
}
```

Scene를 transition하는 타입과 close하는 타입으로 나눠서 함수로 생성

Example

Madagascar 2021

```
class SceneCoordinator: SceneCoordinatorType {

    private let bag = DisposeBag()
    private var window: UIWindow
    private var currentVC: UIViewController

    required init(window: UIWindow) {
        self.window = window
        currentVC = window.rootViewController!
    }

    @discardableResult
    func transition(to scene: Scene, from storyboard: String, using style:
        TransitionStyle, animated: Bool) -> Completable {
        let subject = PublishSubject<Void>()
        let target = scene.instantiate(from: storyboard)

        switch style {
        case .root:
            currentVC = target.sceneViewController
            window.rootViewController = target
            subject.onCompleted()
        case .push:
            print(currentVC)
            guard let nav = currentVC.navigationController else {
                subject.onError(TransitionError.navigationControllerMissing)
                break
            }

            nav.interactivePopGestureRecognizer?.isEnabled = true
            nav.interactivePopGestureRecognizer?.delegate = nil

            nav.rx.willShow
                .subscribe(onNext: { [unowned self] evt in
                    self.currentVC = evt.viewController.sceneViewController
                })
                .disposed(by: bag)
        }
    }
}
```

SceneCoordinator.swift

```
        nav.pushViewController(target, animated: animated)
        currentVC = target.sceneViewController

        subject.onCompleted()
    case .modal:
        currentVC.present(target, animated: animated) {
            subject.onCompleted()
        }
        currentVC = target.sceneViewController
    }

    return subject.ignoreElements().asCompletable()
}

@discardableResult
func close(animated: Bool) -> Completable {
    return Completable.create { [unowned self] completable in
        /// modal 형식으로 되어 있다면 dismiss
        if let presentingVC = self.currentVC.presentingViewController {
            self.currentVC.dismiss(animated: animated) {
                self.currentVC = presentingVC.sceneViewController
                completable(.completed)
            }
        }
        /// navigationController 형식이라면 pop
    } else if let nav = self.currentVC.navigationController {
        guard nav.popViewController(animated: animated) != nil else {
            completable(.error(TransitionError.cannotPop))
            return Disposables.create()
        }
        self.currentVC = nav.viewControllers.last!
        completable(.completed)
    } else {
        completable(.error(TransitionError.unknown))
    }

    return Disposables.create()
}
}
```

Example

Madagascar 2021

Scene.swift

```
/// 나중에 ViewModel를 case안에 넣어줘야 한다.
enum Scene {
    case splash(SplashViewModel)
}

/// 스토리 보드에 있는 씬을 생성하고 연관값이 저장된 뷰 모델을 바인딩해서 리턴하는 메소드를 구현
extension Scene {
    func instantiate(from storyboard: String = "Main") -> UIViewController {
        let storyboard = UIStoryboard(name: storyboard, bundle: nil)

        switch self {
        case .splash(let viewModel):
            guard var splashVC = storyboard.instantiateViewController(identifier:
                Const.ViewController.Identifier.splash) as? SplashVC else {
                fatalError()
            }

            splashVC.bind(viewModel: viewModel)

            return splashVC
        }
    }
}
```

Example

Madagascar 2021

SplashViewModel.swift

```
lazy var loginAction: Action<String, Void> = {
    return Action { _ in
        /// scene 생성 + sceneCoordinator에서 transition method 호출
        let loginViewModel = LoginPresentViewModel(sceneCoordinator:
            self.sceneCoordinator)
        let loginScene = Scene.login(loginViewModel)
        return self.sceneCoordinator.transition(to: loginScene,
            from: "Login",
            using: .modal,
            animated: true).asObservable().map { _
                in }
    }
}()
```

단순하게 sceneCoordinator transition코드를 return함으로써 화면 전환 완료

Example

Madagascar 2021

AppDelegate.swift

```
@main
class AppDelegate: UIResponder, UIApplicationDelegate {
    var window: UIWindow?

    func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        if let window = window {
            let coordinator = SceneCoordinator(window: window)
            let splashViewModel = SplashViewModel(sceneCoordinator: coordinator)
            let splashScene = Scene.splash(splashViewModel)
            coordinator.transition(to: splashScene,
                                from: "Splash",
                                using: .root,
                                animated: false)
        }

        return true
    }
}
```

Coordinator는 **AppDelegate**부터 사용해주어야 합니다.

Root가 되는 뷰부터 Coordinator를 사용해야 그 다음 뷰들도 Coordinator가 적용됩니다.

Madagascar 2021

Co-ordinator

프로젝트 팀원 여러분께 감사드리고 싶어서 써보아야겠네요

잠깐!!!!!!

Madagascar 2021

Coordinator

프로젝트에서 꼭 써봐야겠네요

Thanks

Coordinator❤️
Coordinator❤️
Coordinator❤️
Coordinator❤️