

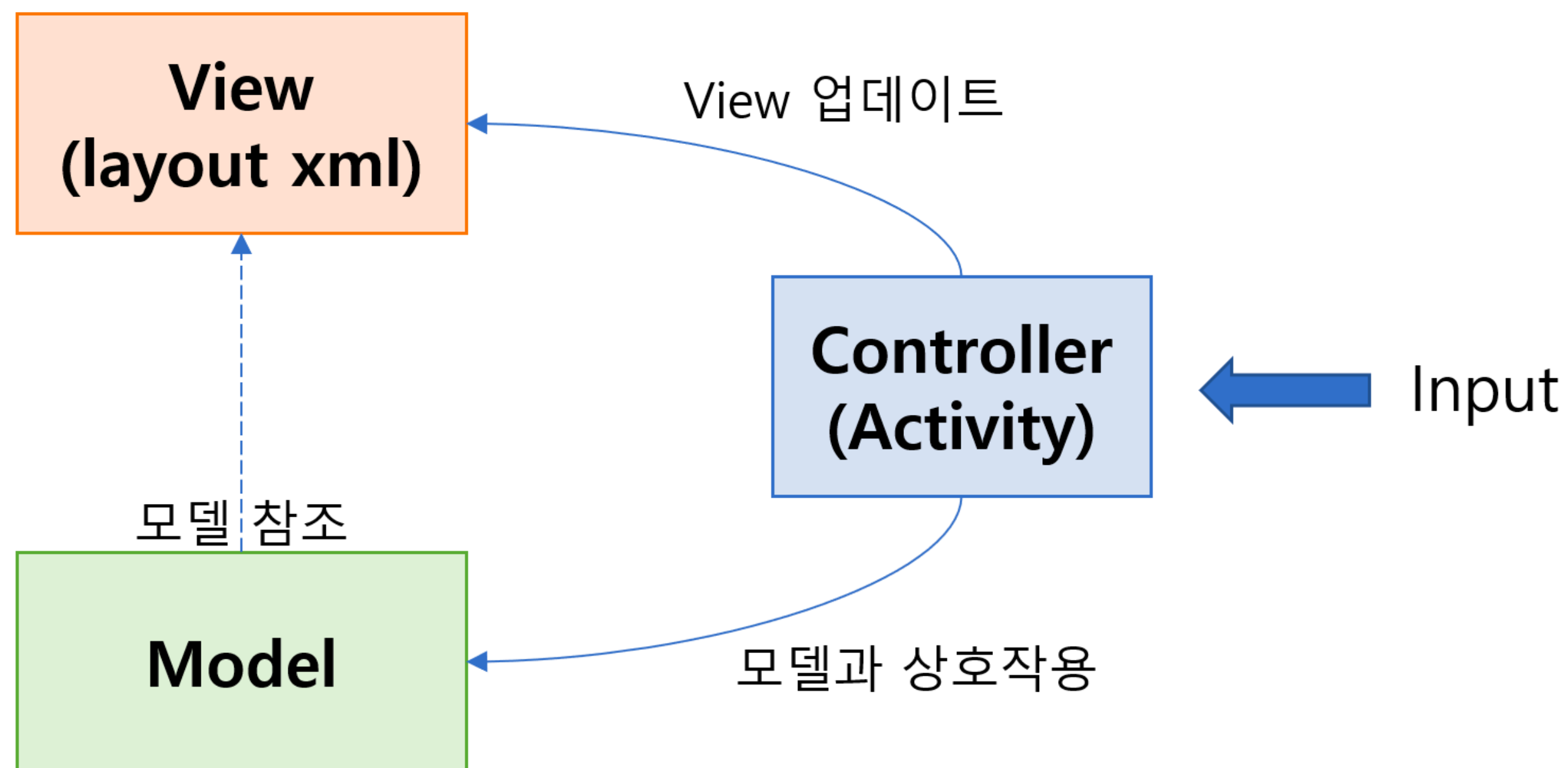
플러터 다트 아키텍처 패턴

2. MVP 패턴

MVP 패턴이란?

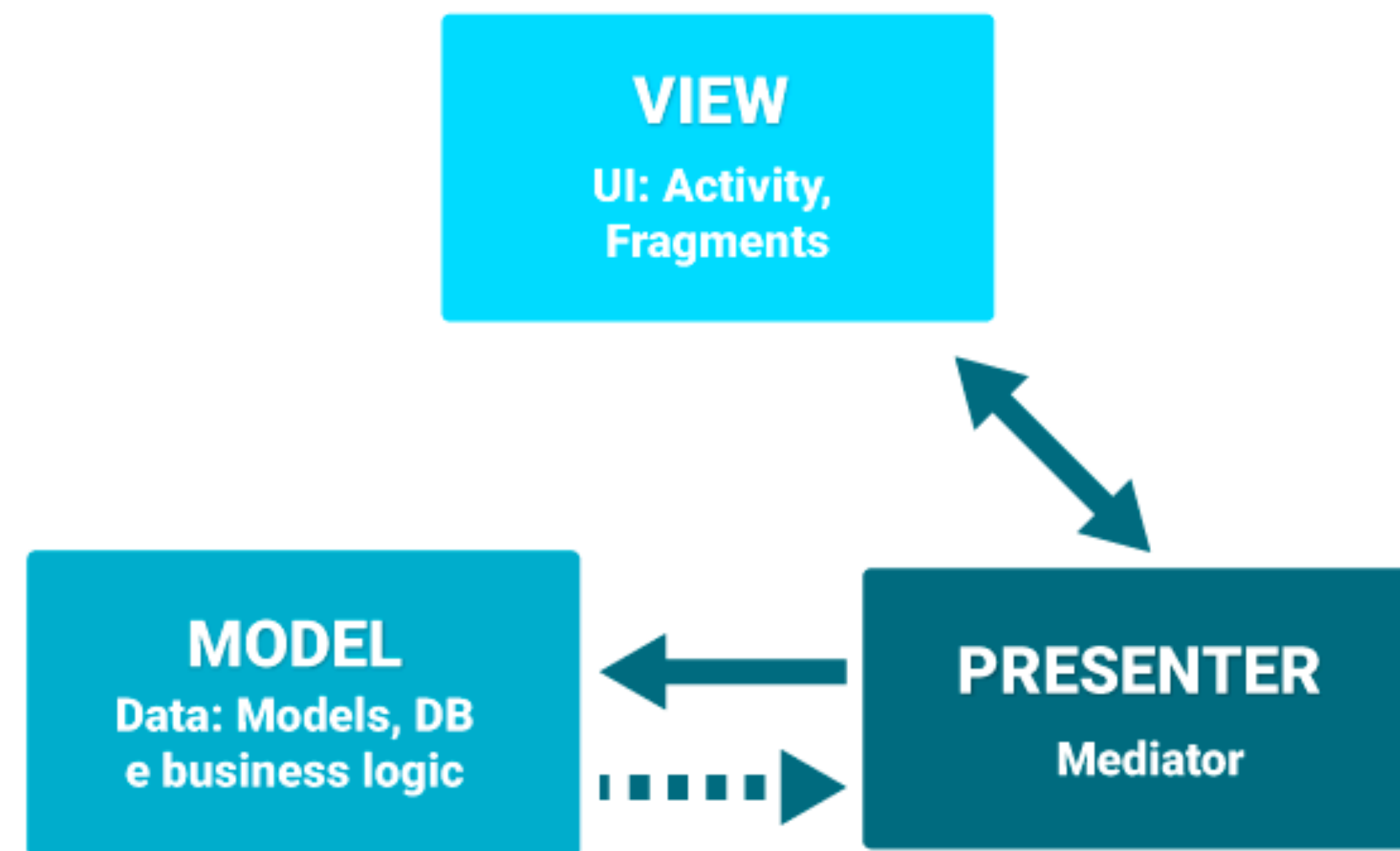
- MVC(Model-View-Controller) 패턴의 문제점을 해결하기 위해 나온 패턴으로 Model-View-Presenter로 이루어짐
- Model은 데이터와 비즈니스 로직을 담당하며
- View는 사용자에게 보여지는 UI
- Presenter는 Model과 View의 메신저 같은 역할을 함
- 실제 요소들의 역할을 볼 때는 MVC와 큰 차이가 없어보임

참조: 깃뚜님의 블로그
<https://gitsu.tistory.com/38>



참조: 봄석님의 블로그
<https://beomseok95.tistory.com/212>

Model View Presenter



MVC 패턴과의 차이점(장점)과 단점

- 각 요소들이 직접적으로 참조가 되기 때문에 필연적으로 종속성이 높아지는 MVC와 달리 MVP는 각 요소들에 대한 인터페이스를 선언, 활용하여 참조하기 때문에 보다 의존성 낮아짐
- 유저 인풋을 컨트롤러가 아닌 뷰에서 받기 때문에 컨트롤러/프리젠퍼 부담이 MVC보다 감소
- 뷰와 프리젠퍼가 일대일 관계이기 때문에 프로젝트 규모가 높아질수록 파일 수가 함께 높아짐
- 기능 추가 시 인터페이스까지 손 보아야하기 때문에 작업량이 많아짐 - 의존성이 낮아진다는 장점이 있는데 해당 단점을 보면 모순되는 것처럼 보이나 낮아지는 요소 간의 의존성이고 인터페이스와 구현 객체의 사이는 의존성이 있는 게 당연

플러터와 MVP 패턴

- 유저 입력을 컨트롤러에서 담당하던 MVC와 달리 뷰에서 담당한다고 보기 때문에 플러터에서도 적용하기 쉬워짐 (실제로 액티비티에서 하는 일은 동일하지만 입력을 받는 것까지 뷰의 역할이라고 정의를 달리 했을 뿐)
- 다만 하나의 클래스에서 UI 요소 할당, 입력 처리를 한번에 할 수 있어 각각 코드 구획을 나누어 정리가 가능한 안드로이드 개발과 달리 개별 위젯들마다 UI, 이벤트 리스너를 들고 있는 플러터는 가독성이 조금 떨어질 수 있음
- 다트는 인터페이스 키워드를 따로 제공하지 않으며 모든 클래스가 인터페이스로 활용가능하다는 컨셉을 기본으로 함 - 자바, 코틀린은 인터페이스 중첩이 되지만 플러터는 클래스이기 때문에 클래스 중첩이 안 되어 MVP 기본 구조인 Contract 인터페이스가 View 인터페이스, Presenter 인터페이스를 포함하는 구조가 불가능

플러터와 MVP 패턴

- 또 뷰(페이지) 자체를 프리젠티어에 주입할 수 있어 프리젠티어에서 뷰를 업데이트할 수 있으나 뷰의 각 속성들을 다 접근할 수 있는 안드로이드와 달리 플러터에서는 프리젠티어에서 뷰에 대한 세부적인 처리가 어려움(페이지도 위젯으로 이루어져 있는데 해당 위젯들에 접근할 방법이 없기 때문)
- 그러나 아쉬움이 조금 남는다 정도이고 MVC처럼 태생적으로 불가능한 것은 아님

코드 공유 시간

인터페이스 구조는

- 1. Contract - [View, Presenter] 2. Presenter - View 등이 있으나 해당 코드는 View, Presenter 따로 선언 후 관리
- 모델 또한 인터페이스로 추상화할 수 있지만 이번 코드에서는 적용하지 않음. 또한 이번 코드의 모델은 모델이 단순히 저장소의 역할만 한다고 가정(CRUD는 프리젠테터)
- 인터넷 참조 코드의 경우 데이터 바인딩이나 DI 패키지, 어댑터 패턴 등 여러 가지 기술이 쓰였지만 해당 코드는 순수 코드로만 구현

결론

- 플러터에서도 그럭저럭 구현할 수 있지만 상태관리를 적용한다면 딱히 효용성이 느껴지지 않음
- MVC가 플러터에 적용하기 어려워서 관심을 받지 못했다면 MVP는 굳이 이걸 해야하나라는 이유 때문에 관심을 받지 못했을 것 같음(MVC는 직관적으로 UI와 비즈니스 로직을 나누었다는 게 느껴지지만 MVP는 목적이 MVC의 높은 결합도를 낮추겠다는 의도로 각 레이어를 추상화했기 때문에 직관적이지는 않음)
- 그러나 아키텍처 패턴의 발달 과정을 보며 이전 개발자들이 추구하고자 했던 것과 극복하고자 했던 문제를 이해하면 플러터에 필요한 구조가 어떤 것일지 더 잘 알 수 있을 것 이기에 의의가 있다고 생각 (mp3 - pmp - 스마트폰에서 pmp의 포지션과 비슷하다는 느낌)