

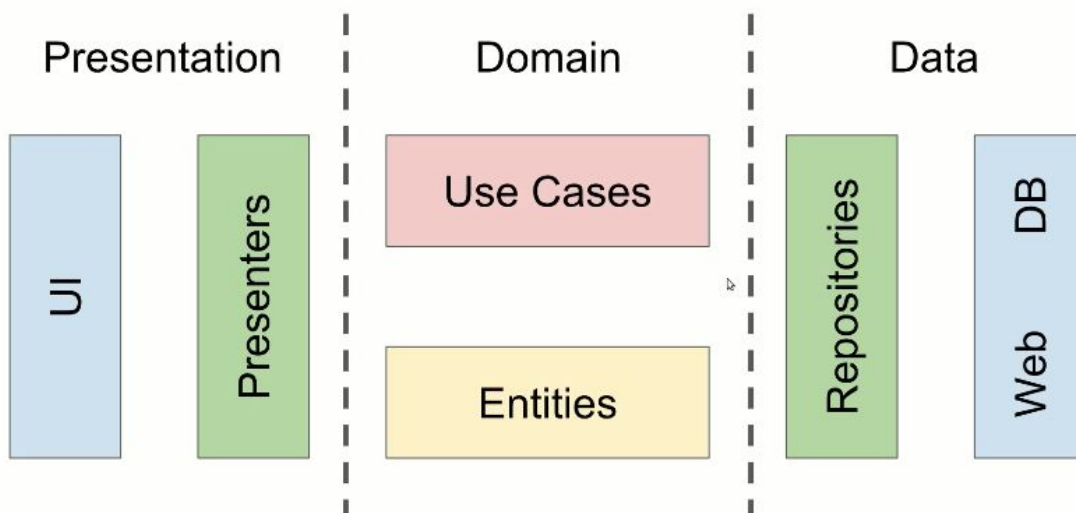
Чистая архитектура:

Позволяет: структурировать код, разделить зависимости и ответственность слоев, обеспечить тестирование.

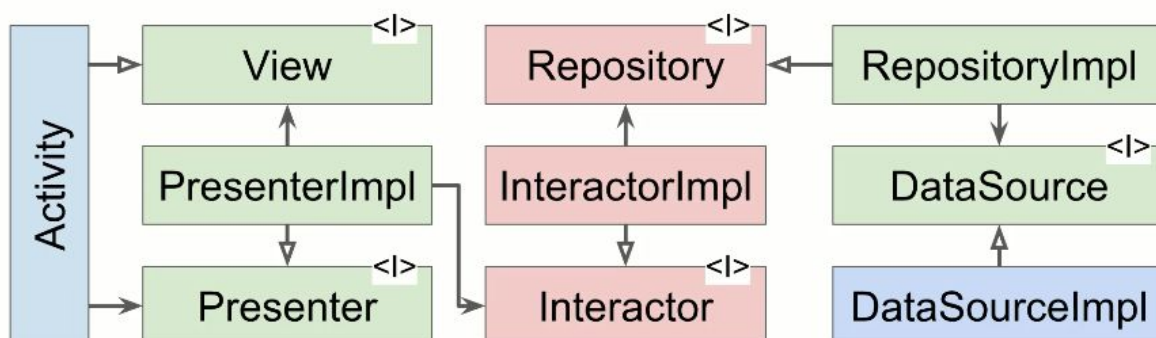
**Внутренний слой (DOMAIN)** - бизнес логика. Все классы и действия над ними, связанные с бизнесом. Например, форматы (классы) заказов, услуг. Формирование заказа, выписка счета, расчет скидок и прочие действия, которые относятся именно к процессу зарабатывания бабла. Внутренний слой ничего не знает про внешние слои. Для взаимодействия с ними он объявляет два интерфейса: InputPort/OutPort. Первый реализует сам и через него получает сообщения. OutputPort реализуется нижним слоем для сообщений от него верхнему слою. То есть внешний слой зависит от внутреннего, а точнее от его абстрактного представления в виде интерфейса InputPort.

Однако внешним слоям разрешено знать структуры классов внутренних слоев.

## Чистая архитектура в Android



Вот как это выглядит:



Вообще в андроиде все фреймворки относятся к периферийным уровням. Тот же самый UI (Activity/Fragments) - это тоже периферия, а значит она не создает своих интерфейсов In/Out, а зависит от интерфейсов презентера (рис выше).

Итак, UI, DB/ORM, WEB/Network/NFC... - это все периферия. Между периферией и уровнем DOMAIN находятся промежуточные компоненты. Со стороны UI - это Presenter'ы или ViewModel, со стороны DB и пр. это Repositories. Presenter'ы и ViewModel играют одинаковую роль - активити/фрагмент сообщает туда обо всех действиях пользователя и получает обратно данные для отображения. От презентера через интерфейс, из VM по подписке. Собственно и все.

Также стоит упомянуть, что полезно отслеживать правильное разделение на уровне пакетов приложения. Например, все презентеры живут в одном пакете и нигде в этом пакете нет import на компоненты из пакета UI. Например на рисунке выше у зеленых компонентов презентера может быть import только на компоненты розового пакета Interactor'a.