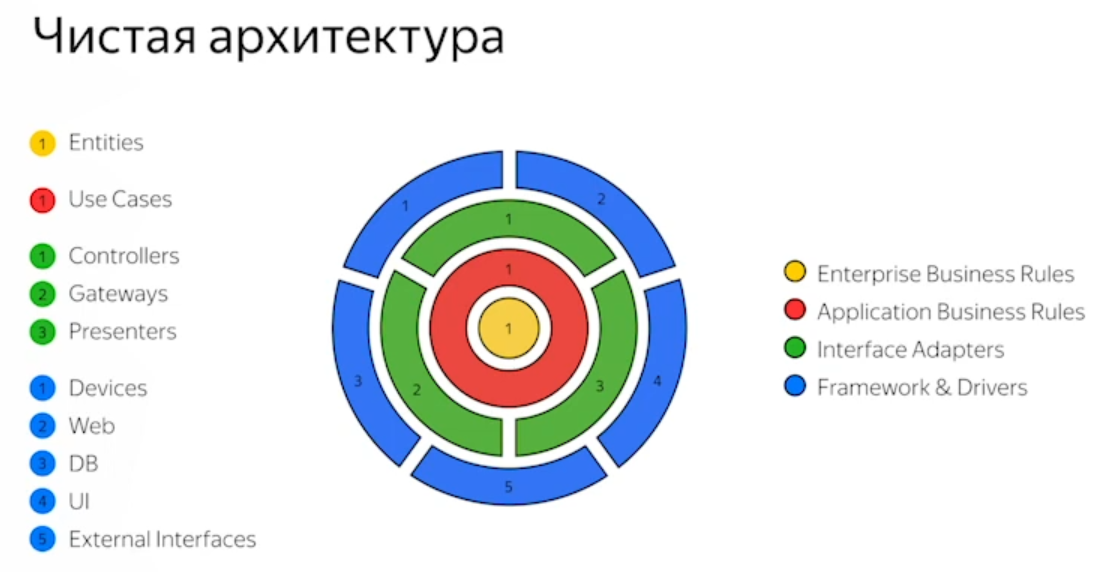
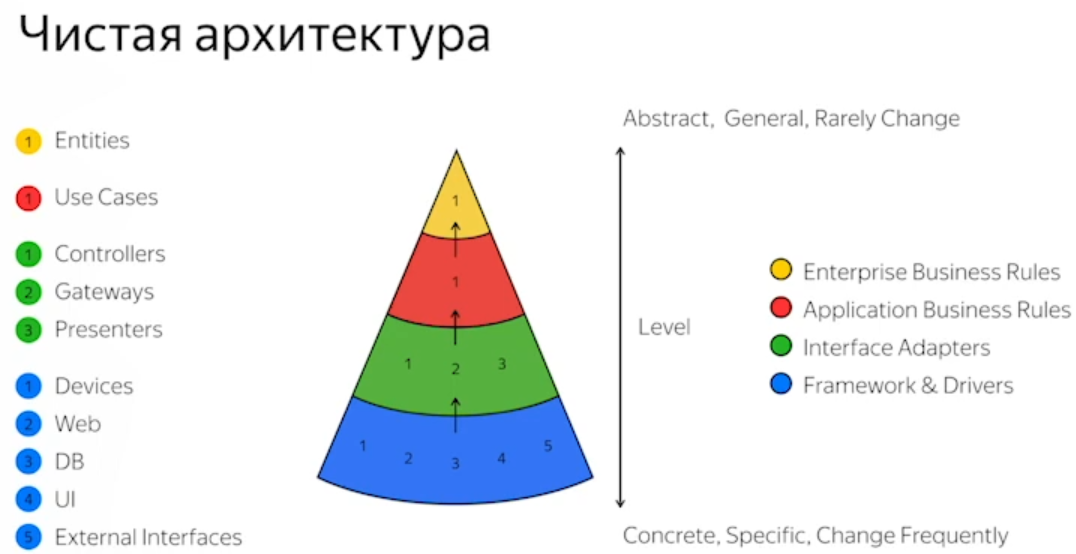
Архитектура – набор методов и шаблонов, которые помогают разработчикам создавать структурированные приложения.

Другими словами, архитектура это не про работоспособность приложения, а про лёгкость его дальнейшего сопровождения/развития и лёгкости последующего рефакторинга.



Чистая архитектура нужна для повышения стабильности кода при его изменениях. Внутренние слои не зависят от изменений внешних, они о них даже ничего не знают.



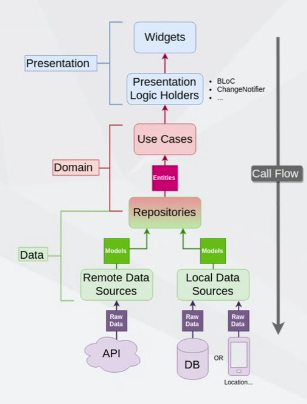
Стрелки – направление зависимостей. Внутренние слои более абстрактны и редко изменяются. Внешние же слои более специфичны и зависят от библиотек ввода-вывода. Части внешних слоёв могут быть заменены на другие и это будет безболезненно для внутренних слоёв. Это упрощает развитие приложения и упрощает его тестируемость.

Данная архитектура делает развитие приложения надёжным и предсказуемым.

ВСТРЕЧА

Не бывает идеальной архитектуры.

Мы разделяем наше приложение на 3 основных слоя:



* Presentation – грубо говоря, наш UI.
* Domain – бизнес-логика всего приложения. В нём находится абстрактный репозиторий. Модели (entity) в domain ничего не знают о Data.
* Data – реализацию работы с данными. Находятся классы, которые реализуют интерфейсы работы с Domain. Mapper – класс, который преобразует модели уровня Data в модели уровня Domain.

UI знает о Domain, Domain знает о Data.

В Domain называется просто класс, например, Car, в Data – CarEntity.

Репозиторий – сущность, которая обращается к данным.



Создание пакета.

В пакете coreui лежат виджеты, которые мы используем более, чем в одном месте.

Все основные зависимости проекта находятся в core, в других пакетах находятся только зависимости на другие пакеты.