

Что такое Чистая Архитектура

Для того что бы понять это, лучше всего прочитать книгу [Мартина Роберта «Чистая Архитектура»](#)

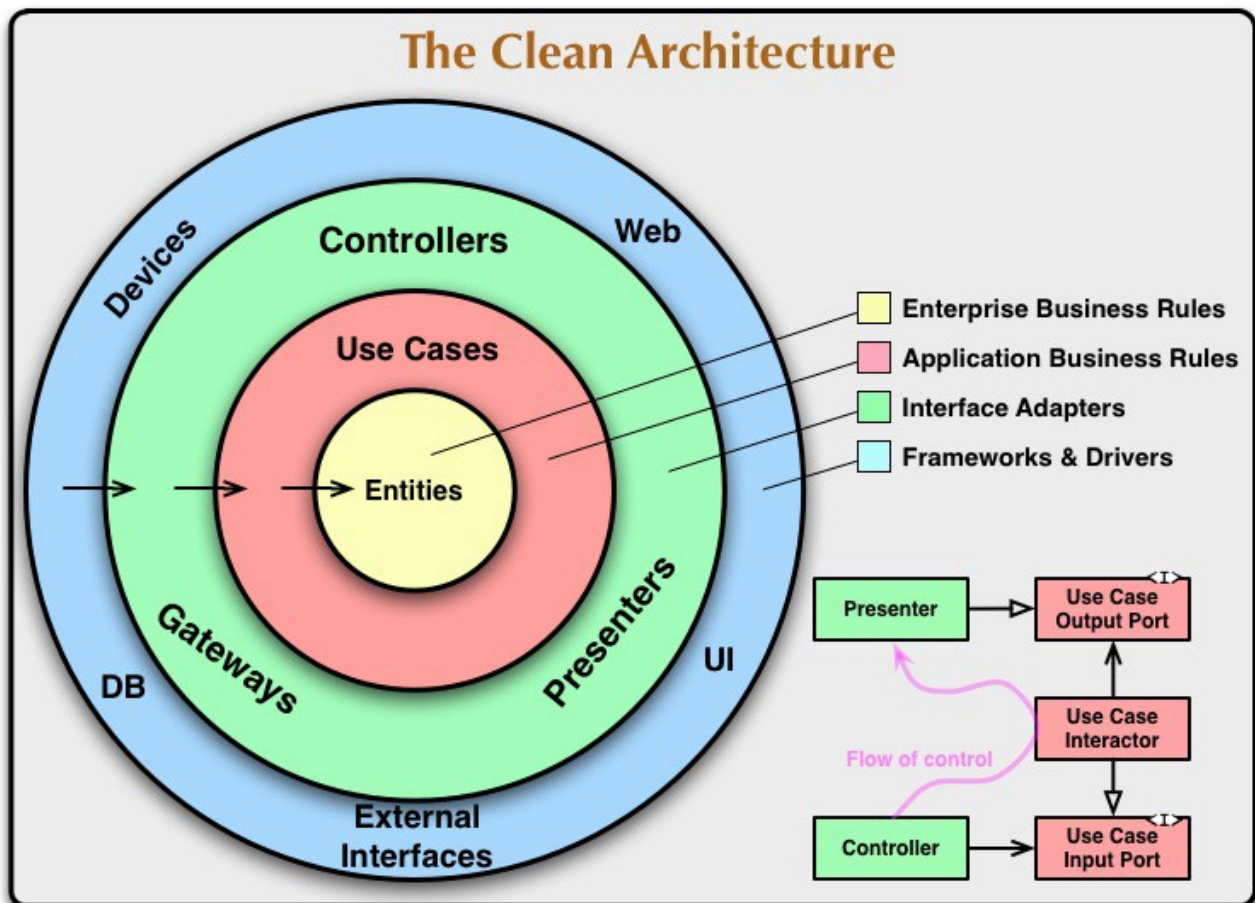
[Ссылка на статью на хабре](#)

Основные идеи заложенные в архитектуру:

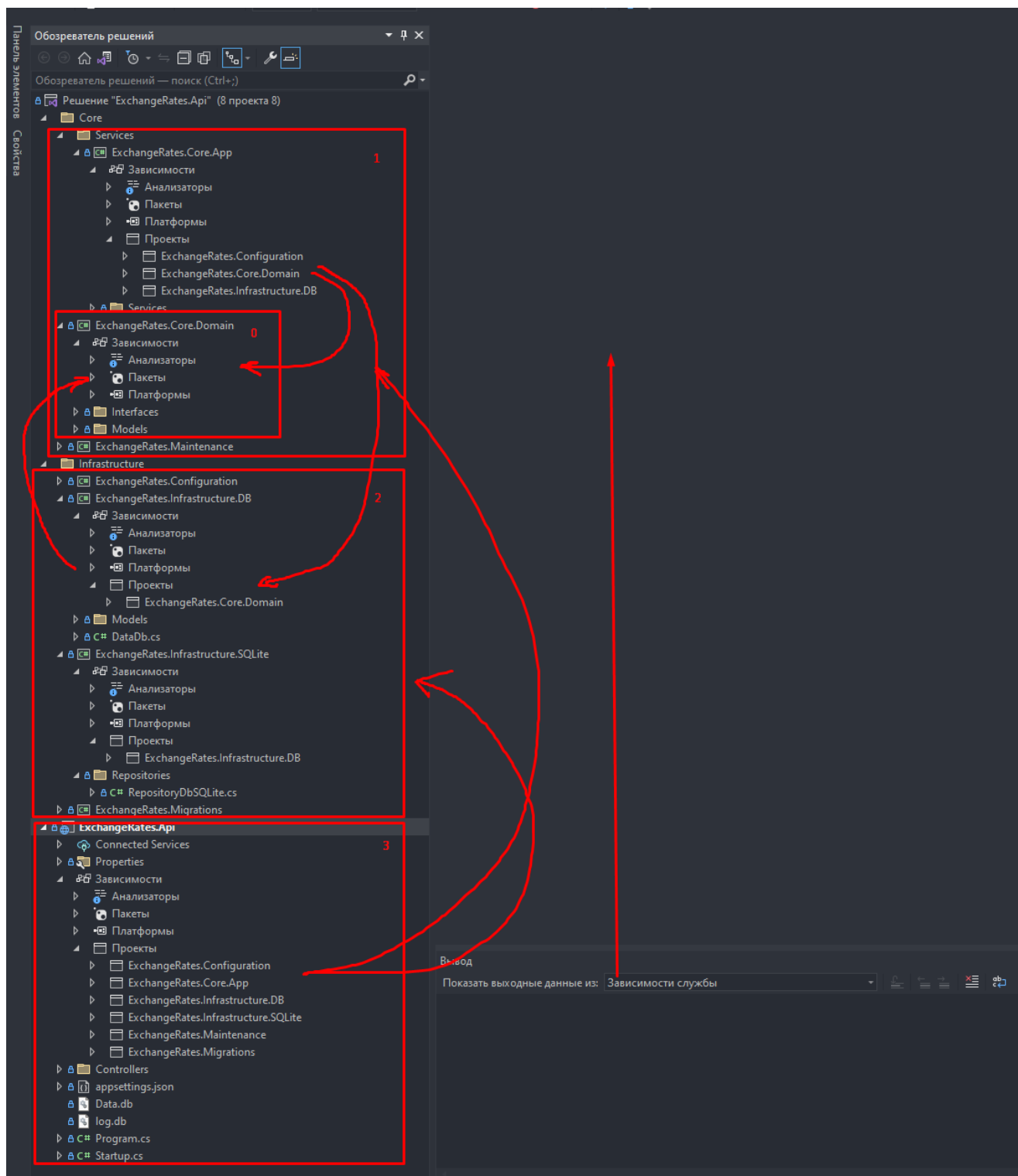
1. **Независимость от фреймворка.** Архитектура не зависит от существования какой-либо библиотеки. Это позволяет использовать фреймворк в качестве инструмента, вместо того, чтобы втискивать свою систему в рамки его ограничений.
2. **Тестируемость.** Бизнес-правила могут быть протестированы без пользовательского интерфейса, базы данных, веб-сервера или любого другого внешнего компонента.
3. **Независимость от UI.** Пользовательский интерфейс можно легко изменить, не изменяя остальную систему. Например, веб-интерфейс может быть заменен на консольный, без изменения бизнес-правил.
4. **Независимость от базы данных.** Вы можете поменять Oracle или SQL Server на MongoDB, BigTable, CouchDB или что-то еще. Ваши бизнес-правила не связаны с базой данных.
5. **Независимость от какого-либо внешнего сервиса.** По факту ваши бизнес правила просто ничего не знают о внешнем мире.

Достигается такая гибкость за счет разделения приложения на **слои** Service, Repository, Model. получил следующие слои:

- **View** — выводит данные клиенту, фактически визуализирует состояние логики клиенту.
- **Controller** — отвечает за взаимодействие с пользователем посредством IO (ввод-вывод).
- **Service** — отвечает за бизнес логику и ее переиспользование между компонентами.
- **Repository** — отвечает за получение данных из внешних источников, такие как база данных, api, локальное хранилище и пр.
- **Models** — отвечает за перенос данных между слоями и системами, а также за логику обработки этих данных.

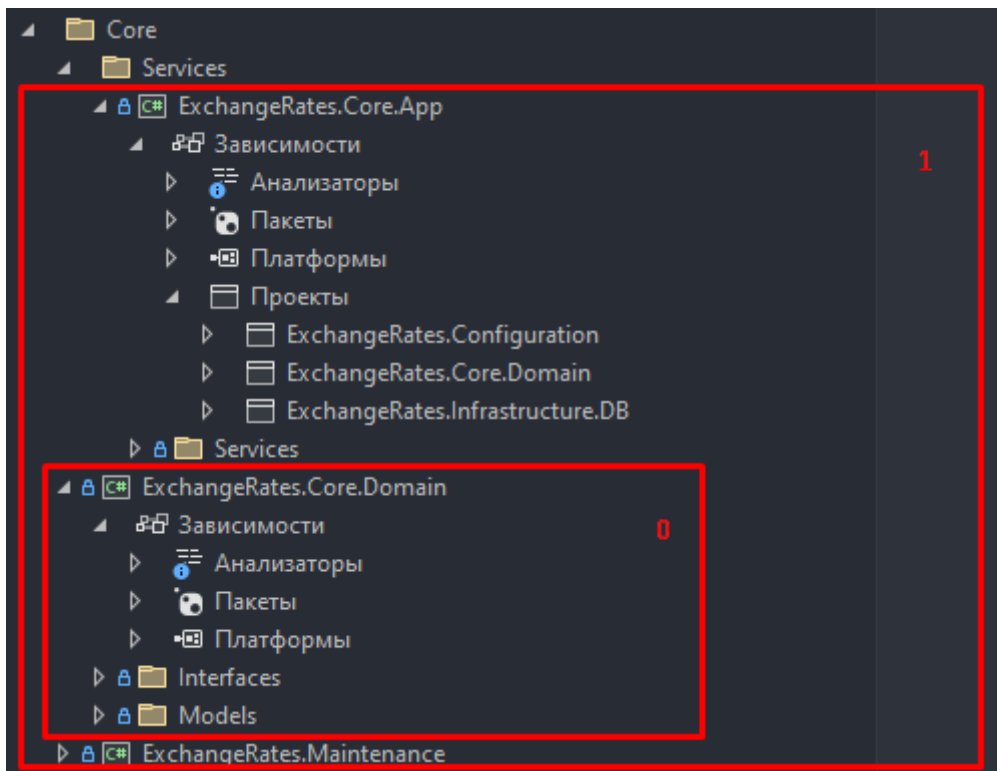


Если говорить простым языком. Приложение делится на слои, каждый слой имеет связь только с определенным слоем в этом виде архитектуры.
Флоу идет внутрь приложения....от краев внутрь...



если посмотреть на скрин выше
мы имеем "луковую" структуру приложения, а точнее луково-слоистую.

квадраты:

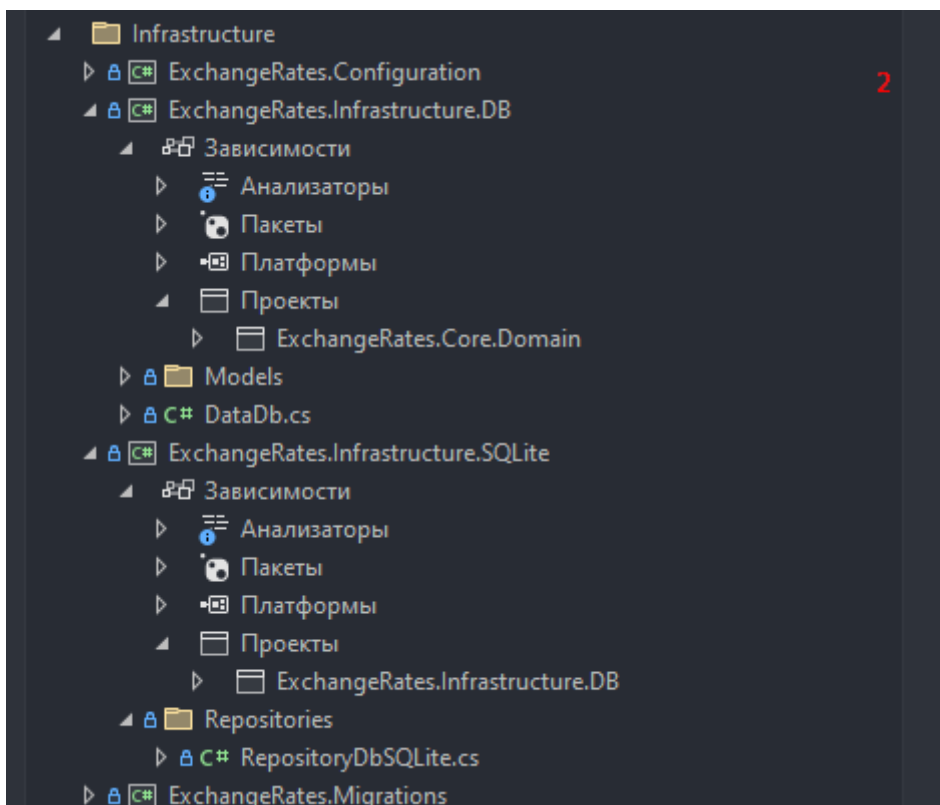


0 - это так называемые домейн модели или Entity или Enterprise логика. Этот проект не имеет связей. В своей структуре имеет модели и интерфейсы которые используются приложением.

1 - Это так называемое ядро приложения, в своей структуре может содержать Core.App(бизнес логика Core.BLL) и Core.Domain

Core.App слой содержит в себе бизнес логику проекта, различные сервисы и обработчики.

НЕ СОДЕРЖИТ интерфейсы. Интерфейсы(контракты) содержатся в слое Domain

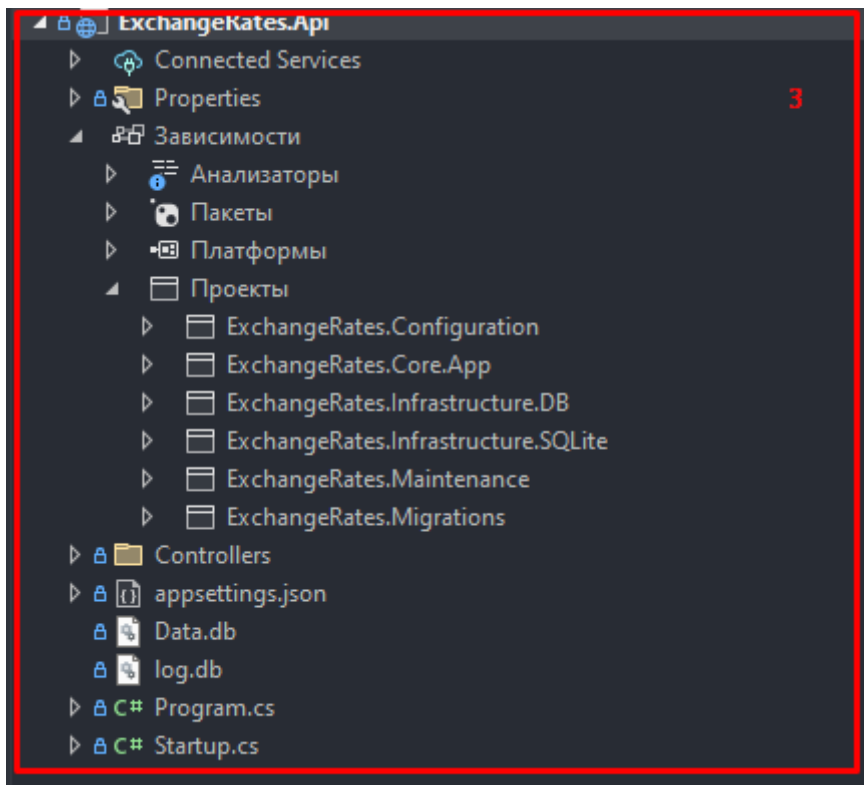


2 - Это инфраструктурный слой, там содержатся слои конфигурации приложения, миграции и подключения к базам данных в частности здесь использован Entity Framework.

Имеет зависимости только на Domain слой.

Разбит на несколько проектов

- проект с миграциями
- проект где настраивается DbContext + содержатся модели DB слоя
- Проект с репозиториями



3 - Это презентер(UI интерфейс). В нашем случае им выступает web API. Она имеет в связях зависимости практически от всех проектов приложения.

[#архитектура](#) [#луковая](#) [#чистая](#) [#онион](#)