







# Серверная часть

## Сущности

### User → Пользователь

 Название поля	 Тип	 Комментарий
<u>Идентификатор</u>	Целое число	
<u>Фамилия</u>	Строка	
<u>Имя</u>	Строка	
<u>Дата создания</u>	Дата	
<u>Дата последнего изменения</u>	Дата	
<u>Статус</u>	Перечисление	Возможные значения → Активен, Отключен, Заблокирован

### Task → Задача

 Название поля	 Тип	 Комментарий
<u>Идентификатор</u>	Целое число	
<u>Название</u>	Строка	
<u>Описание</u>	Строка	
<u>Дата создания</u>	Дата	
<u>Дата последнего изменения</u>	Дата	
<u>Статус</u>	Перечисление	Возможные значения → Не начата, В процессе, Выполнен, Отменен, Отклонен
<u>Постановщик</u>	Пользователь	
<u>Исполнитель</u>	Пользователь	

## База данных

Для работы с базой данных использован подход Code First и инициализатор со свойством обновления БД при запуске приложения (при каждом запуске приложения происходит авто-генерация данных)

## API

### Пользователи

- получение списка (с пагинацией);
- получение одного пользователя;
- редактирование данных (кроме статуса);
- поставить задачу исполнителю.

### Задачи

- получение списка задач пользователя-создателя (с пагинацией);
- получение списка задач пользователя-исполнителя (с пагинацией);
- получение одной задачи;
- редактирование данных (кроме статуса и создателя);
- изменение статуса задачи;
- смена постановщика.

## Используемые технологические и проектные решения

Web API, CQRS, MediatR, AutoMapper, FluentValidation, PostgreSQL, Entity Framework, DI, Swagger, Docker, Blazor (библиотека UI компонентов Blazorise), чистая архитектура (рисунок 1).

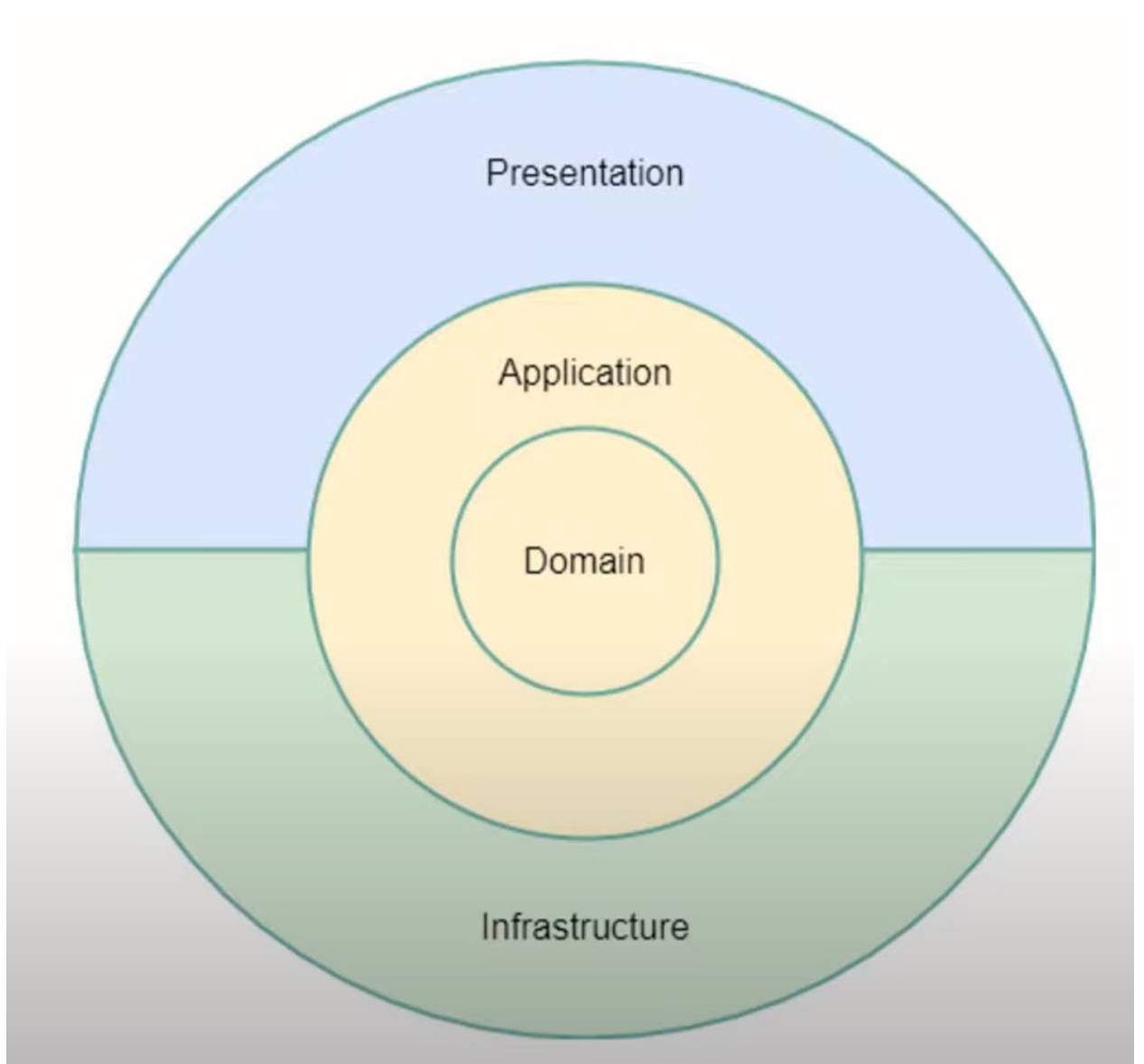


Рисунок 1 – Архитектура приложения