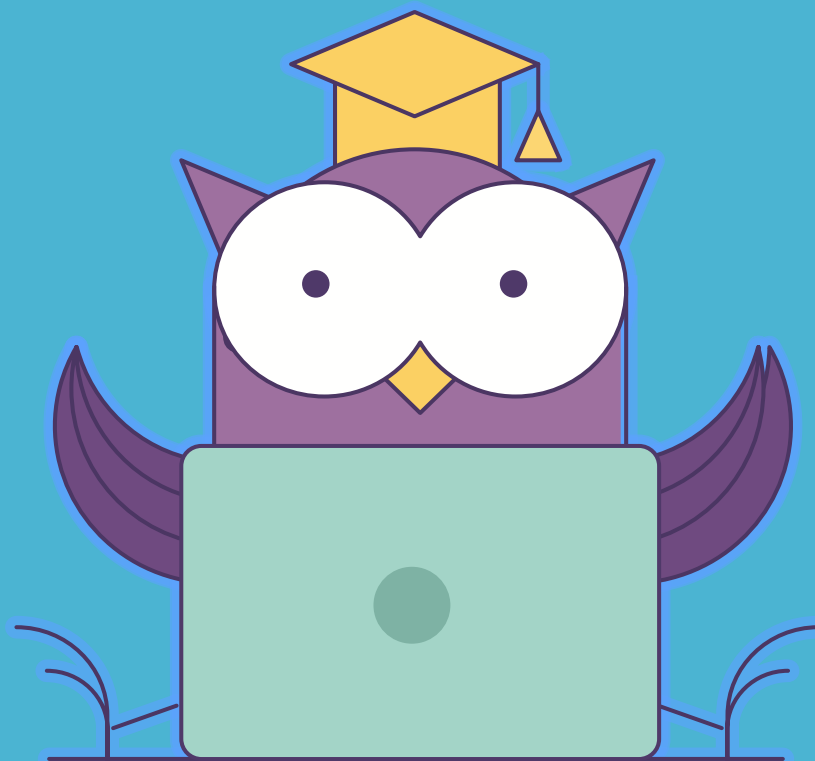




ОНЛАЙН-ОБРАЗОВАНИЕ

Как меня слышно и видно?



> Напишите в чат

+ если все хорошо

- если есть проблемы со звуком или с видео

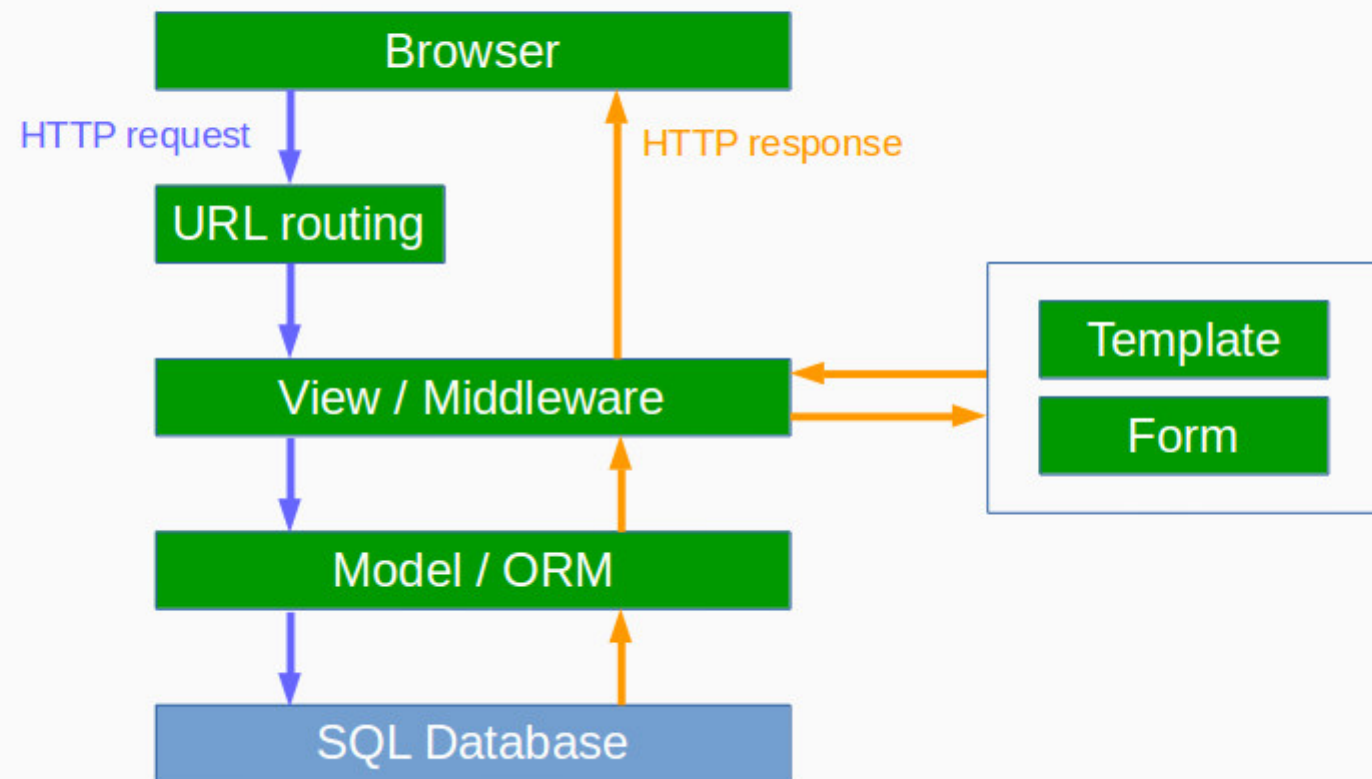
!проверить запись!

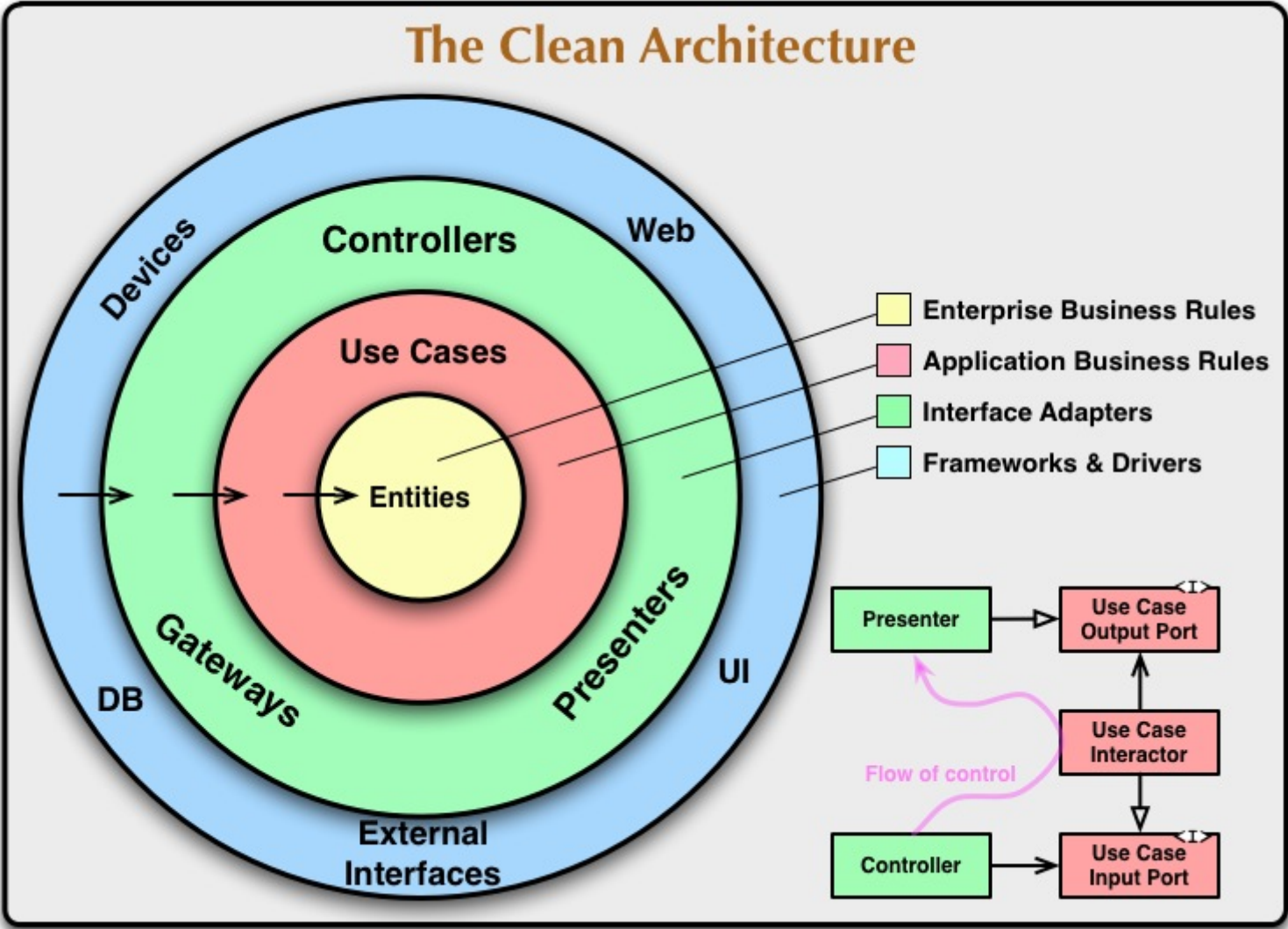
Clean Architecture

Дмитрий Смаль



Django's architecture





Ключевое правило: внутренние слои НЕ зависят от внешних

<https://github.com/OtusTeam/Go/tree/master/cleancalendar>

```
├── Makefile
├── api
│   └── api.proto
├── cmd
│   ├── grpc_client.go
│   ├── grpc_server.go
│   └── root.go
├── internal
│   ├── domain
│   │   ├── errors
│   │   │   └── error.go
│   │   ├── interfaces
│   │   │   └── event_storage.go
│   │   ├── models
│   │   │   └── event.go
│   │   └── services
│   │       └── event.go
│   ├── grpc
│   │   └── api
│   │       ├── api.pb.go
│   │       └── server.go
│   └── maindb
│       └── maindb.go
├── main.go
└── sql
```


Devives, DB, UI..

- Postgres / драйвер pgx
- библиотека GRPC
- библиотека Cobra

Gateways, Presenters, Controllers

- `cleancalendar/cmd/*.go`
- `internal/grpc/*.go`
- `internal/maindb/*.go`

Entities, Usecases

- `internal/domain/models/*.go`
- `internal/domain/services/*.go`
- `internal/domain/interfaces/*.go`

- Обращение к `domain` слою через интерфейсы
- Dependency Injection <https://habr.com/ru/company/funcorp/blog/372199/>

Плюсы

- Чистая модель и бизнес-логика
- Возможность простого Unit-тестирования

Минусы

- Зачастую выглядит как overkill
- Есть шанс получить Anemic Model

Микросервисы ?

- Реализация бизнес-логики (валидация)
- Тестирование бизнес-логики
- Альтернативный интерфейс (REST/http)

Заполните пожалуйста опрос

<https://otus.ru/polls/5138/>



Спасибо за внимание!

