

Яндекс.Практикум

The image features a black background with abstract white and yellow geometric lines. A large white arc is positioned in the upper left, and a smaller white arc is in the lower right. A yellow line segment crosses the lower right area, intersecting the white arcs. The text 'Яндекс.Практикум' is located in the top left, and 'Яндекс Практикум' is in the bottom right. The central text 'Архитектура и организация проектов в Go' is split across two lines.

Архитектура и организация
проектов в Go

Яндекс Практикум

- **Включите камеры**, потому что так гораздо приятнее вести общение и мы сможем запомнить друг друга визуально
- **Не стесняйтесь задавать любые вопросы в чат**, я с радостью на них отвечу
- **Настраивайтесь на позитивный лад и будьте активными** ✨

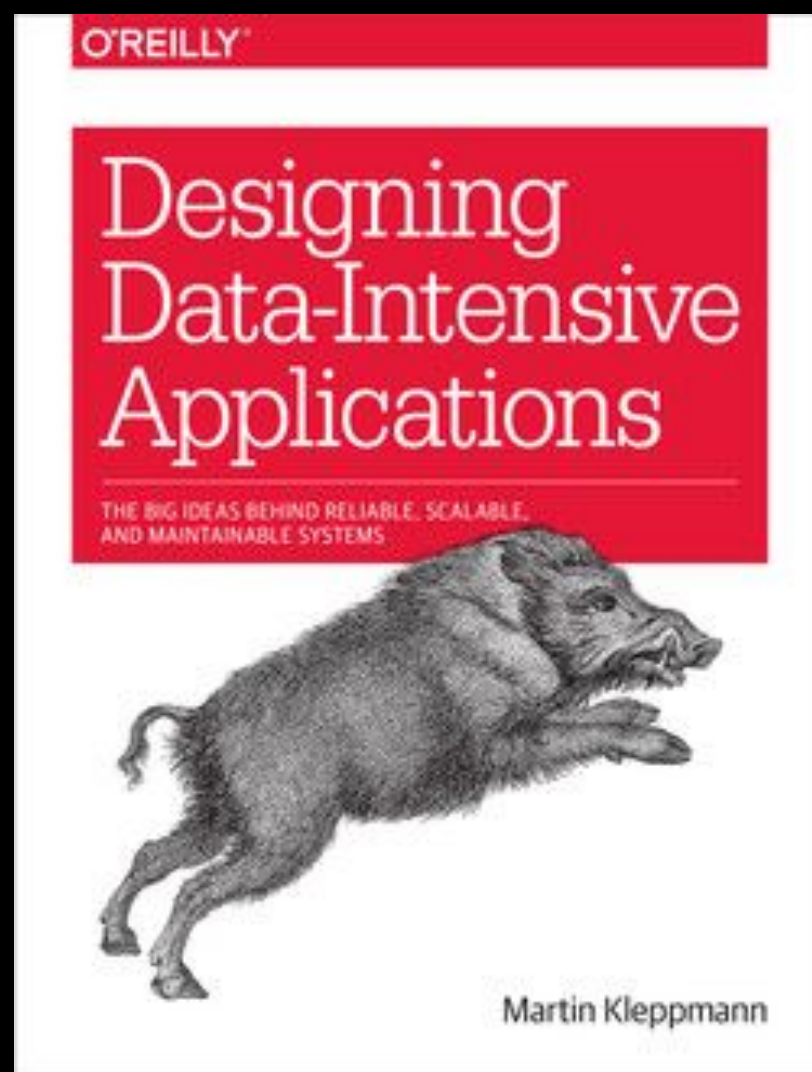
Что нас сегодня ждет

- Рассмотрим популярные подходы к организации архитектуры проектов
- Обсудим варианты структурирования проектов на Go
- Узнаем, что такое циклические зависимости и как их решать
- Разберем ваши вопросы

Что нас сегодня не ждёт

- Мы не будем обсуждать архитектуру систем (System design)
- Мы не будем обсуждать микросервисы
- Мы сконцентрируемся на standalone проектах, не на библиотеках

Что можно почитать об этом:



<https://microservices.io/>

Архитектура

Архитектура – набор договоренностей об организации кода.

Архитектура может быть простой (и плоской): <https://github.com/appleboy/gorush>

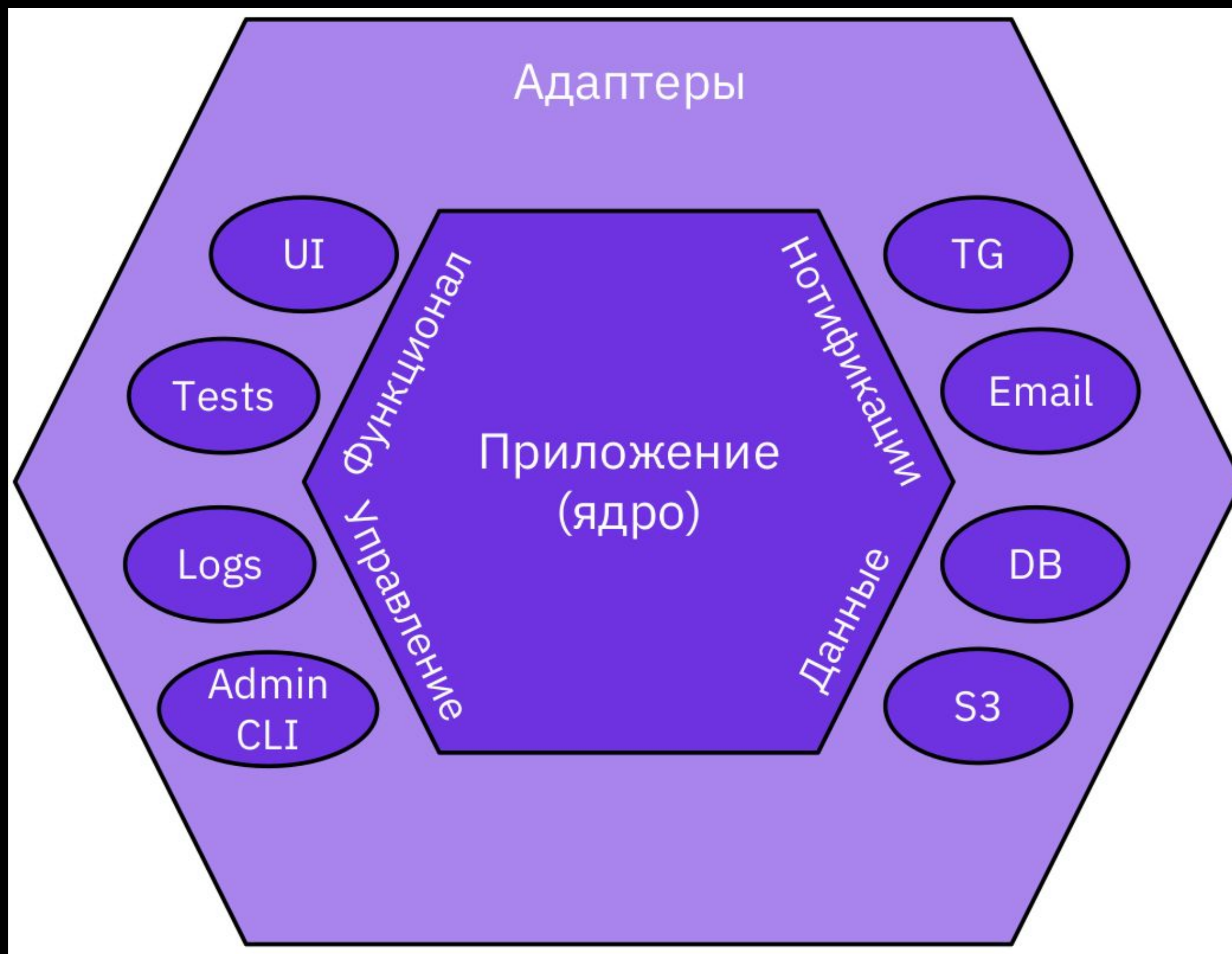
Риск отсутствия архитектурных конвенций – неподдерживаемый код:

- хрупкий
- сложно расширять
- сложно тестировать
- плохо читается

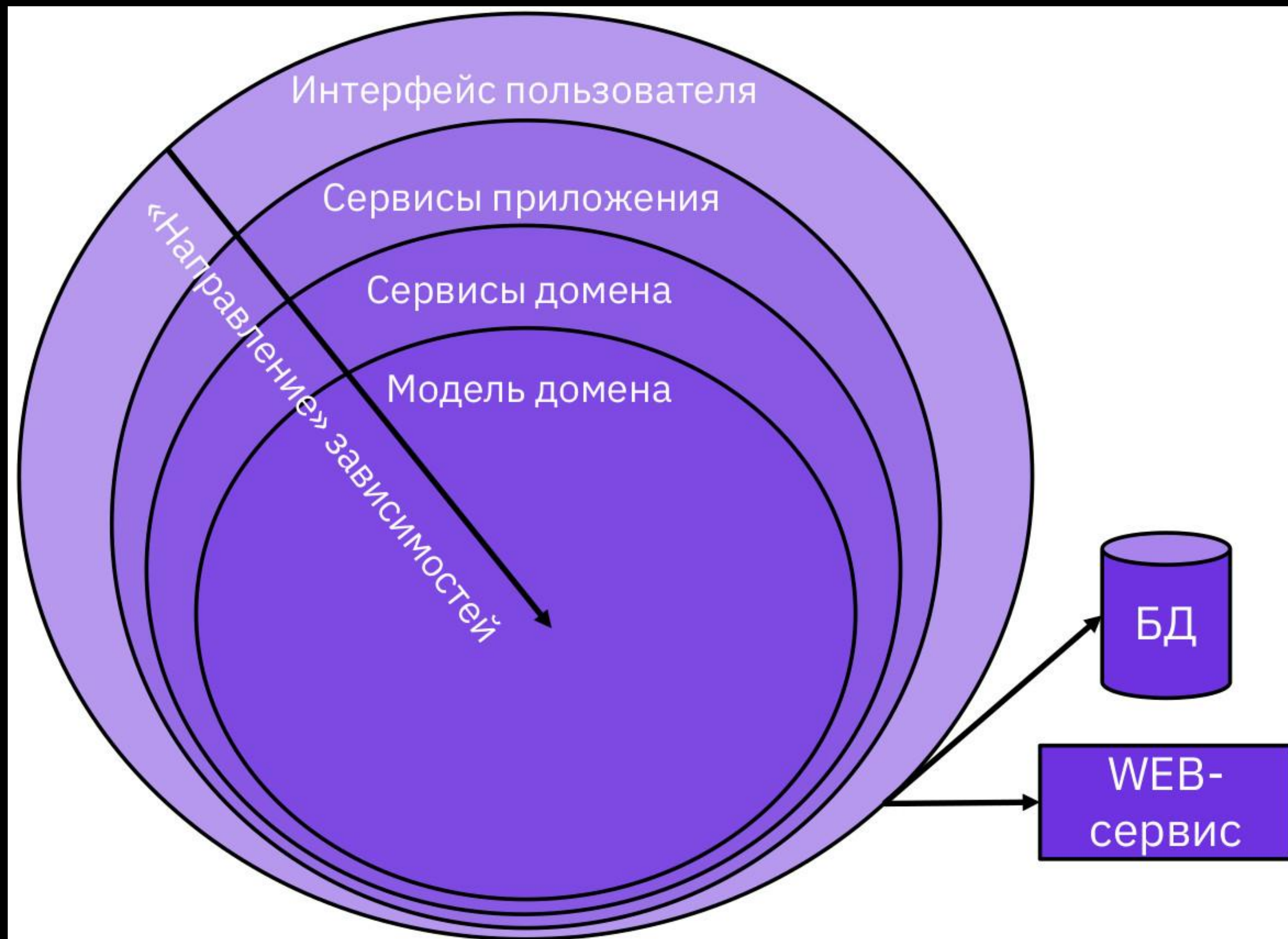
Pasta-theory:

<https://www.techtarget.com/searchsoftwarequality/tip/Fix-spaghetti-code-and-other-pasta-theory-antipatterns>

Гексагональная архитектура

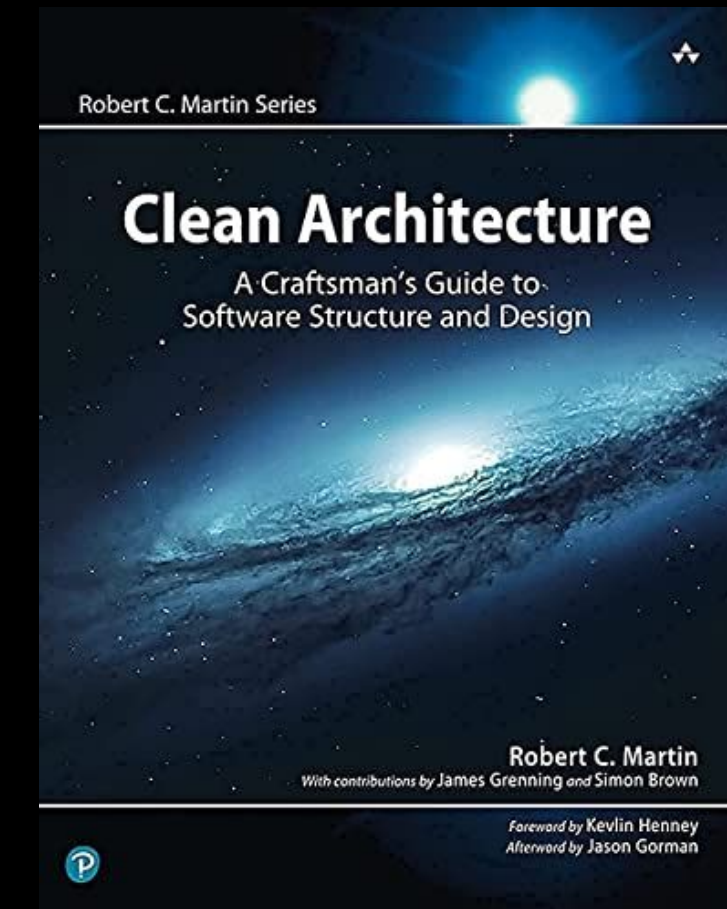
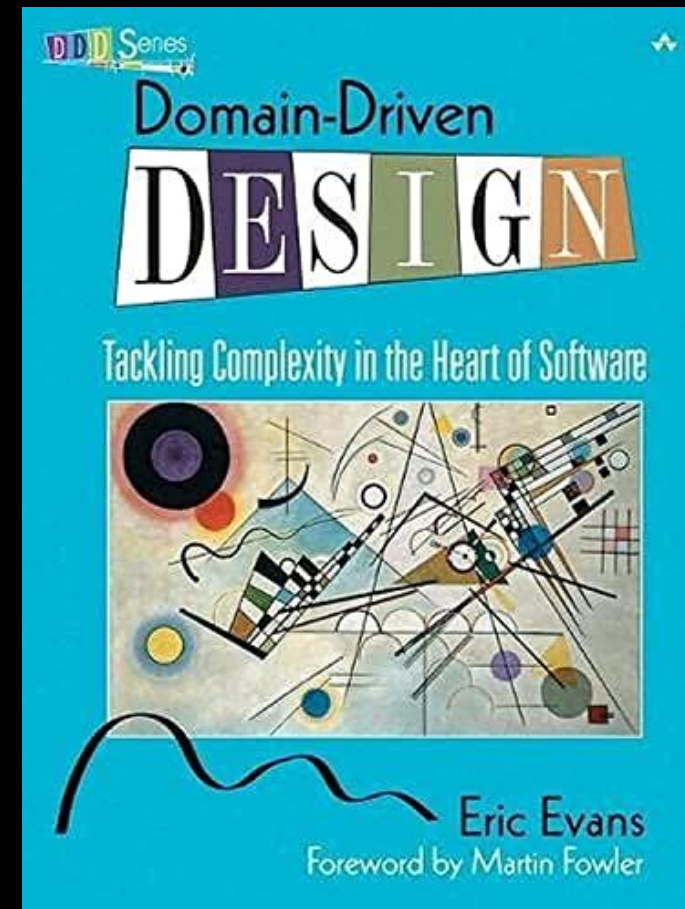


Луковичная архитектура



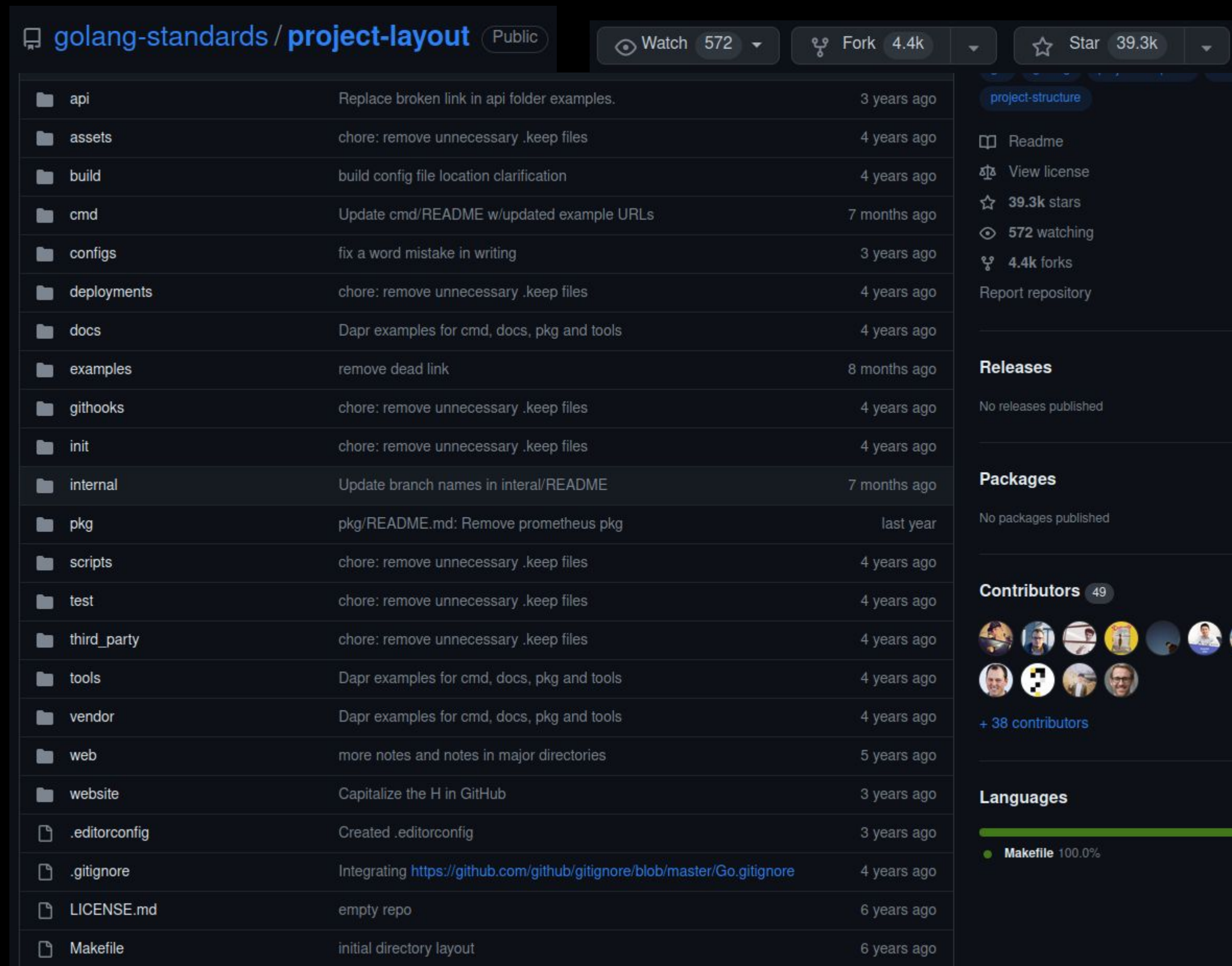
А как же DDD? Clean architecture? ...?

- DDD описывает организацию моделей и сущностей приложения, т.е. дополняет подходы к организации архитектуры. Рассчитан на большие приложения
- Clean architecture – onion architecture + SOLID + Component principles + ...
- Существующих теоретических подходов очень много, а их практических имплементаций еще больше



Standard Go Project Layout

- “Это базовый макет организации проектов, разработанных на Golang”
- “... данный макет будет явным перебором”
- многие части стали де-факто стандартом



Дополнение

Если мы еще не обсудили, то самое время поговорить о:

- директории *internal*
- правиле “Accept interfaces, return structures”
- циклических зависимостях
- Dependency injection

Спасибо за внимание!

Вопросы?