

# объектно-ориентированное программирование

многослойные архитектуры

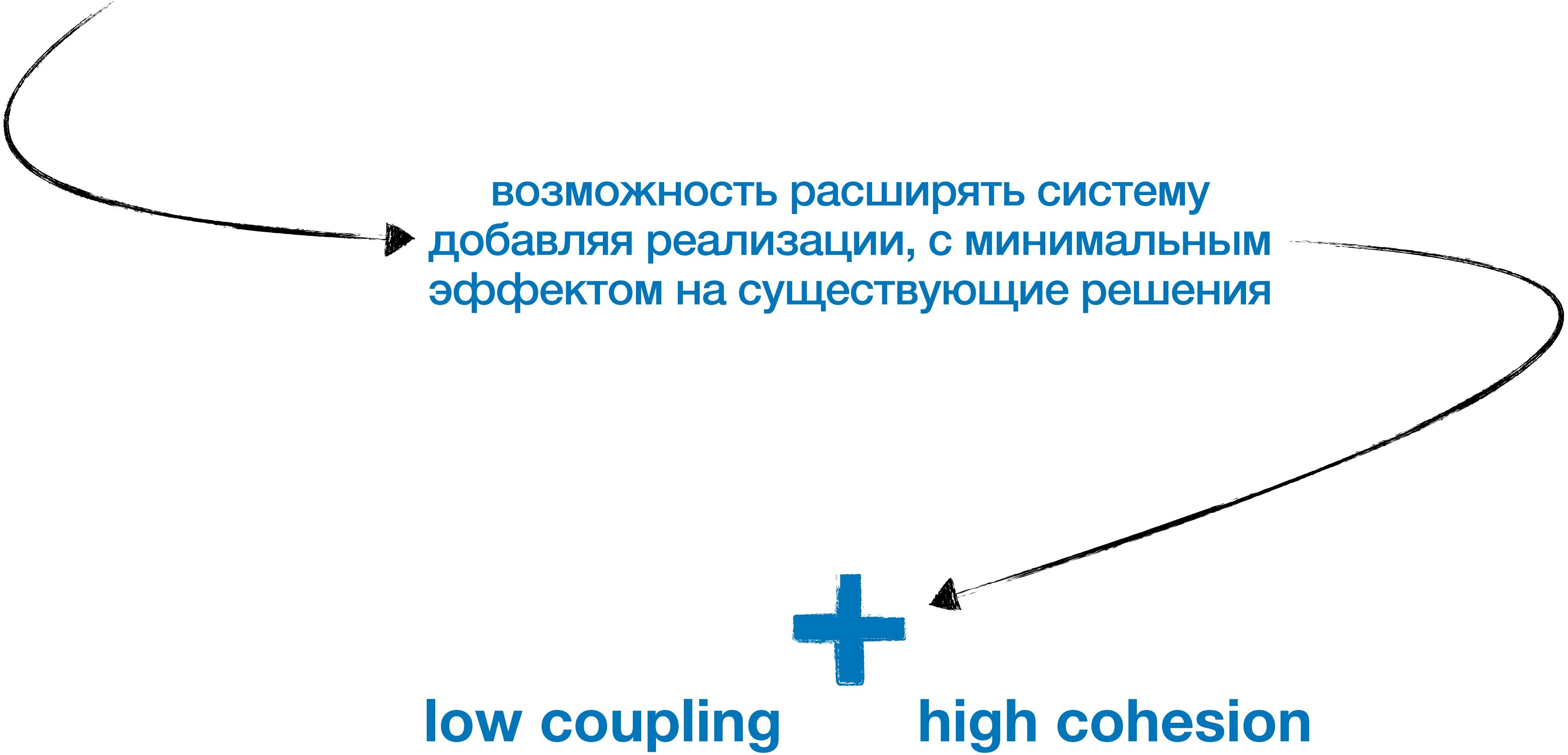
**способ структурирования программных  
компонентов приложения для  
управления сложностью и её приручения**



**архитектура приложения**

# управление и приручения сложности

что это значит?



возможность расширять систему  
добавляя реализации, с минимальным  
эффектом на существующие решения

The diagram consists of a central blue plus sign. To its left is the text 'low coupling' and to its right is 'high cohesion'. A curved arrow originates from the 'low coupling' text, loops upwards and to the left, and points towards the definition of low coupling. Another curved arrow originates from the 'high cohesion' text, loops upwards and to the right, and points towards the definition of high cohesion.

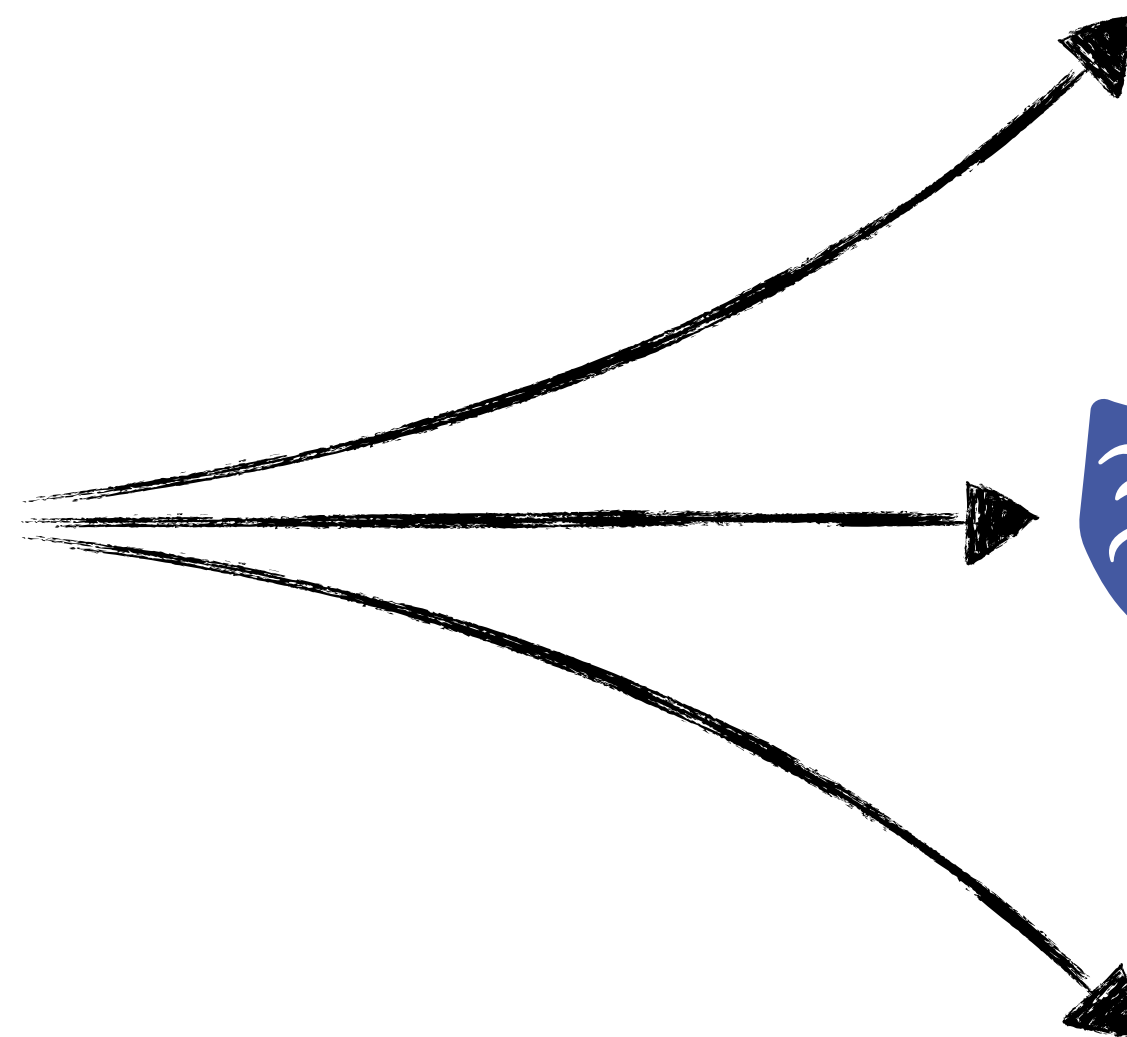
low coupling

+

high cohesion



декомпозиция



?



?



?



**MVX**

**model**

**view**

**controller**

# MVX

## model

- содержит бизнес логику
- требует изменений, когда изменяются юзкейсы системы
- требует изменений, когда изменяются бизнес правила системы



# MVX

## view

- реализует представление системы
- GUI, CUI, API
- требует изменений, когда изменяется способ представления

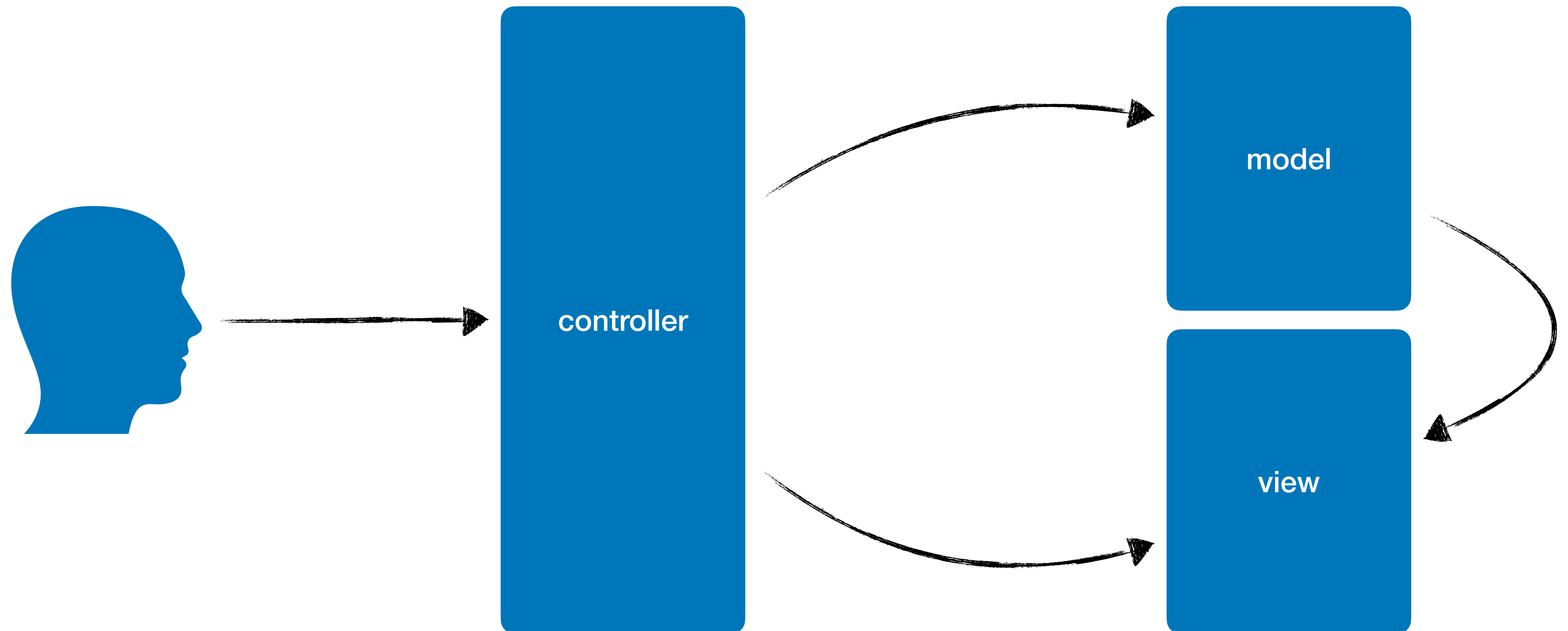
# MVX

## controller

- является посредником между model и view
- отвечает за передачу данных
- требует изменений, когда изменяется механизм взаимодействия между model и view

# MVX

## структура MVC





## **ВЫВОДЫ**

- поддерживает SRP (high cohesion)
- не поддерживает low coupling

трёхслойная архитектура

# трёхслойная архитектура

## presentation

- ответственен за реализацию представления системы
- ответственен за передачу запросов к системе в слой бизнес логики

# трёхслойная архитектура

## business logic

- ответственен за реализацию бизнес логики
- для хранения данных использует слой доступа к данным

# трёхслойная архитектура

## **data access**

- ответственен за реализацию персистентности данных



**ТИПЫ НЕ ИМЕЮТ ПОВЕДЕНИЙ, СЛУЖАТ  
ЛИШЬ ДЛЯ ХРАНЕНИЯ ДАННЫХ  
ВСЯ ЛОГИКА НАХОДИТСЯ В СЕРВИСАХ**



**анемичная модель данных**

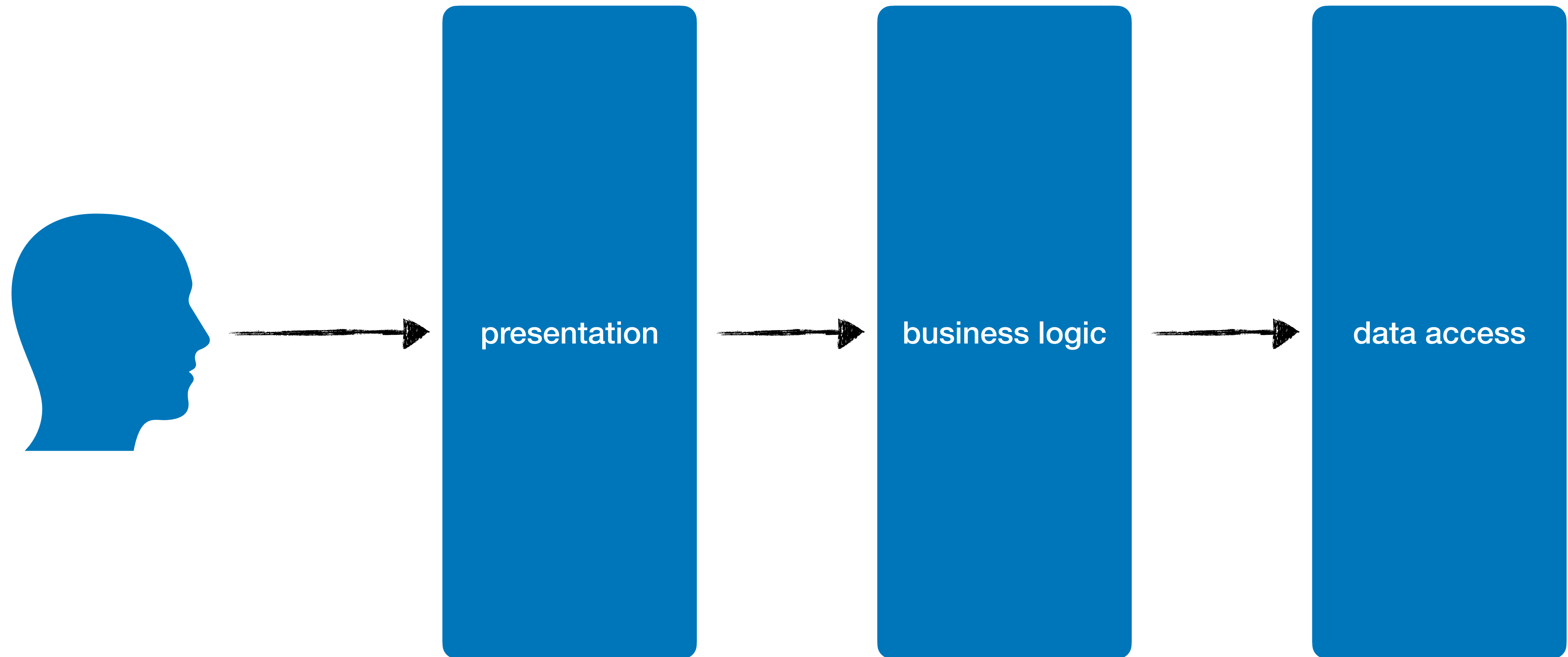
**ТИПЫ ОПИСЫВАЮТ ПОЛНОЦЕННЫЕ  
объекты  
логика находится в типах  
описывающих домен**



**богатая модель данных**

# трёхслойная архитектура

выпрямление зависимостей



# трёхслойная архитектура

фокус декомпозиции

**MVC**

view

controller

model

**TLA**

presentation

business logic

data access

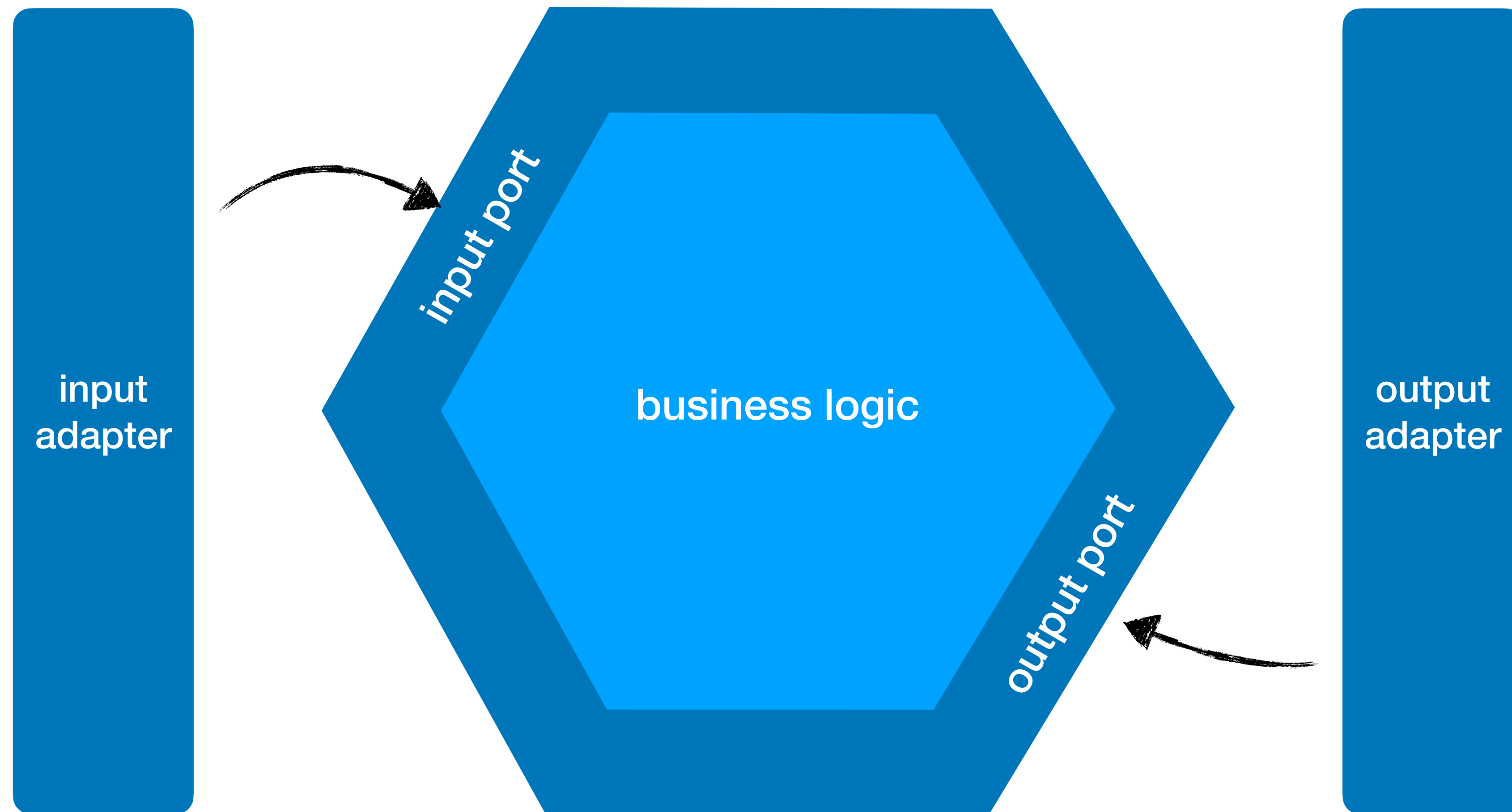
гексагональная архитектура

# гексагональная архитектура

- развитие тулинга
  - веб-фреймворки
  - бд-фреймворки
  - другие фреймворки и библиотеки
- нужно абстрагировать бизнес логику от инфраструктуры

# гексагональная архитектура

## структура



# гексагональная архитектура

## ВЫВОДЫ

- подразумевает абстракцию приложения над инфраструктурой (output port)
- подразумевает абстракцию представления над приложением (input port)
- позволяет сделать бизнес логику независимой от вспомогательных реализаций
- не имеет ничего общего с шестиугольниками



**бизнес логика**

# бизнес логика

## составляющие

### бизнес правила

набор данных и операций над этими данными, отражающий основные принципы работы доменной области

в ОО мире бизнес правила представляются типами

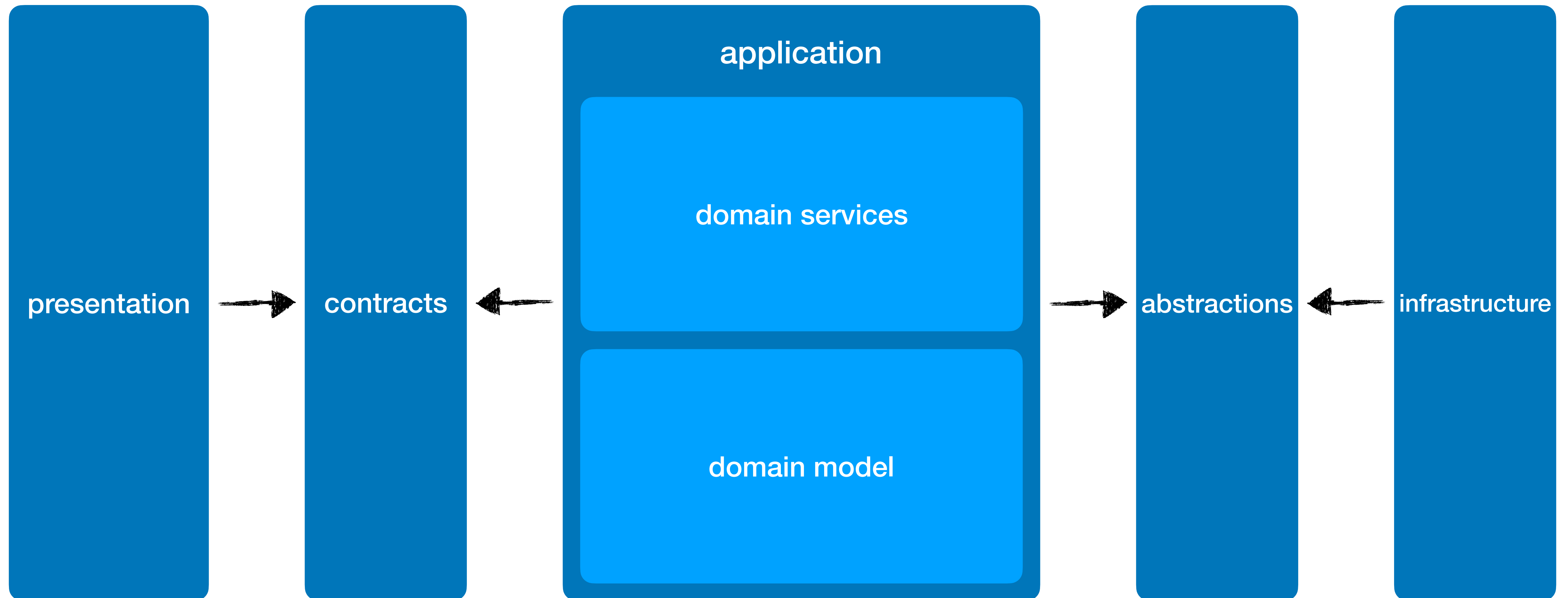
### юзкейсы

какие-либо операции, поддерживаемые приложением

луковая архитектура

# луковая архитектура

## структура



# луковая архитектура

## ВЫВОДЫ

- основывается на гексагональной
- больше декомпозирует бизнес логику
  - доменные модели - определения бизнес правил
  - доменные сервисы - определения юзкейсов
- выделяет слой приложения - связующее звено между инфраструктурными абстракциями и доменом

Всё!

