

Лекция 2. Определение архитектуры

Основные понятия и определения архитектуры ПС

1. Зарождение архитектуры
2. Понятие архитектуры
3. Свойства архитектурных решений и зависимостей

Зарождение архитектуры программных средств

«Есть некоторое дополнение к программированию, и его нужно вытащить на свет. Это **программная архитектура**. Архитектура и проектирование – не одно и то же»

- 1950-е: появления множества языков программирования
- 1960-е: соображения о необходимости структуризации программ
- 1969: вторая конференция Software engineering, первое упоминание о “software architecture”
- 1970-е: концепция модульности и сокрытия информации

Усложнение программ

Появление понятия архитектуры

- До 1980-х: термин «архитектура» в основном применяется к аппаратной части компьютеров
- 1984: основан Software Engineering Institute
- 1990-е: возрождение интереса к теме архитектуры

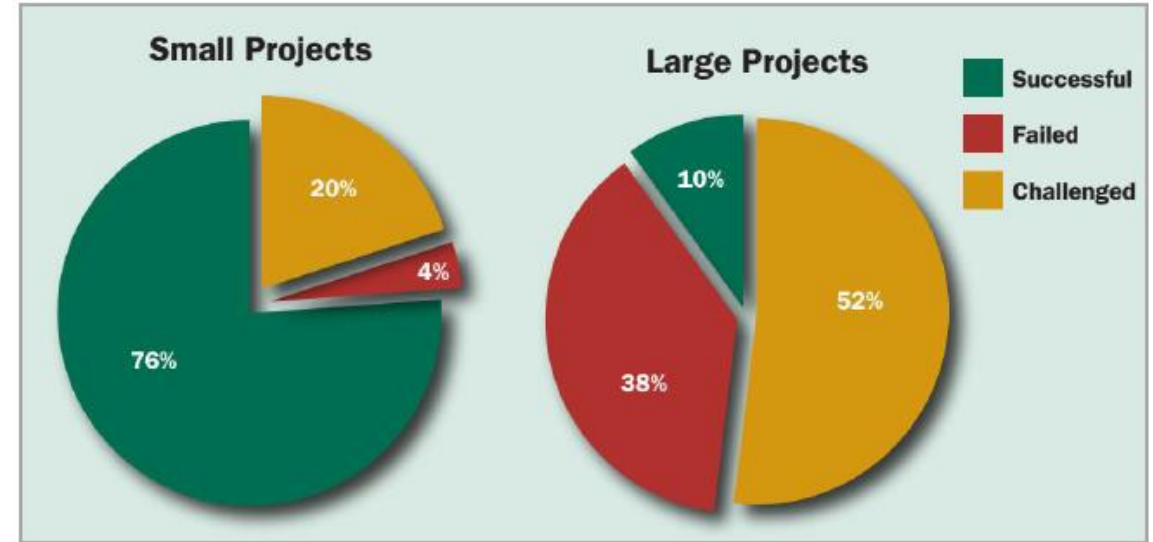
- 1996-2000: создание стандарта IEEE 1471 - описание программной архитектуры
- ISO/IEC/IEEE 42010:2011 “Systems and software engineering – Architecture description”
- 2010: появление дисциплины архитектурного проектирования

Появление стандартов

Зарождение архитектуры программных средств

Трудности:

- Без понимания предмета управление невозможно
- Об архитектуре говорят все, но объяснить не могут



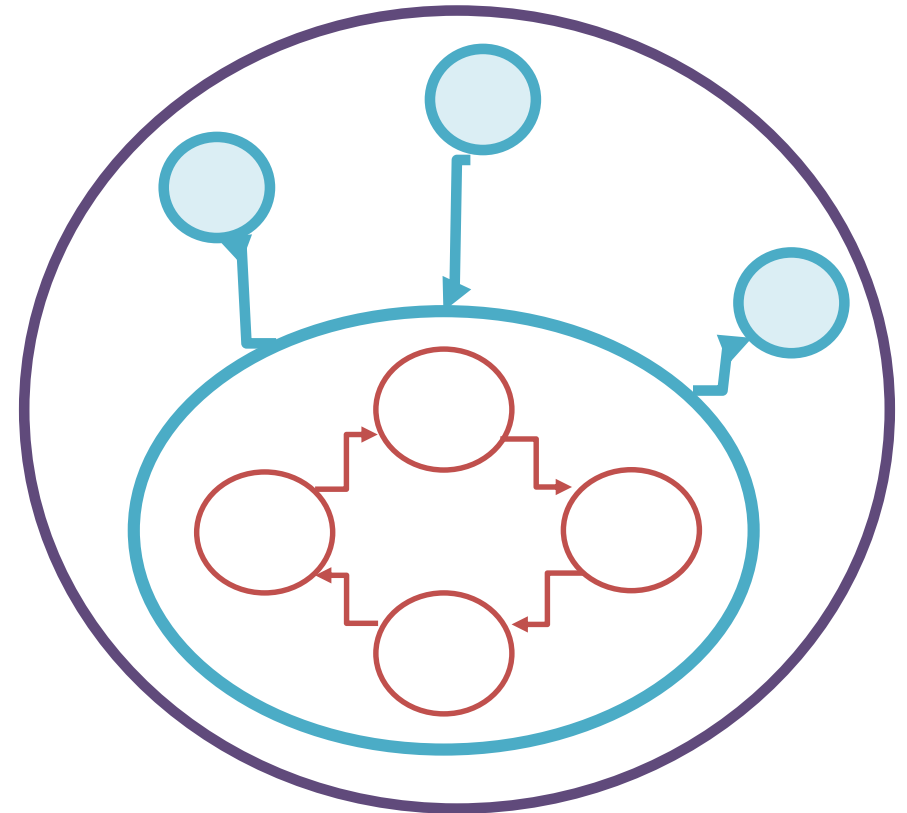
АП* - архитектурное проектирование

The Standish Group Inc. «Мы верим, что нет нужды в больших проектах и большой IT-проект можно разбить на набор маленьких»

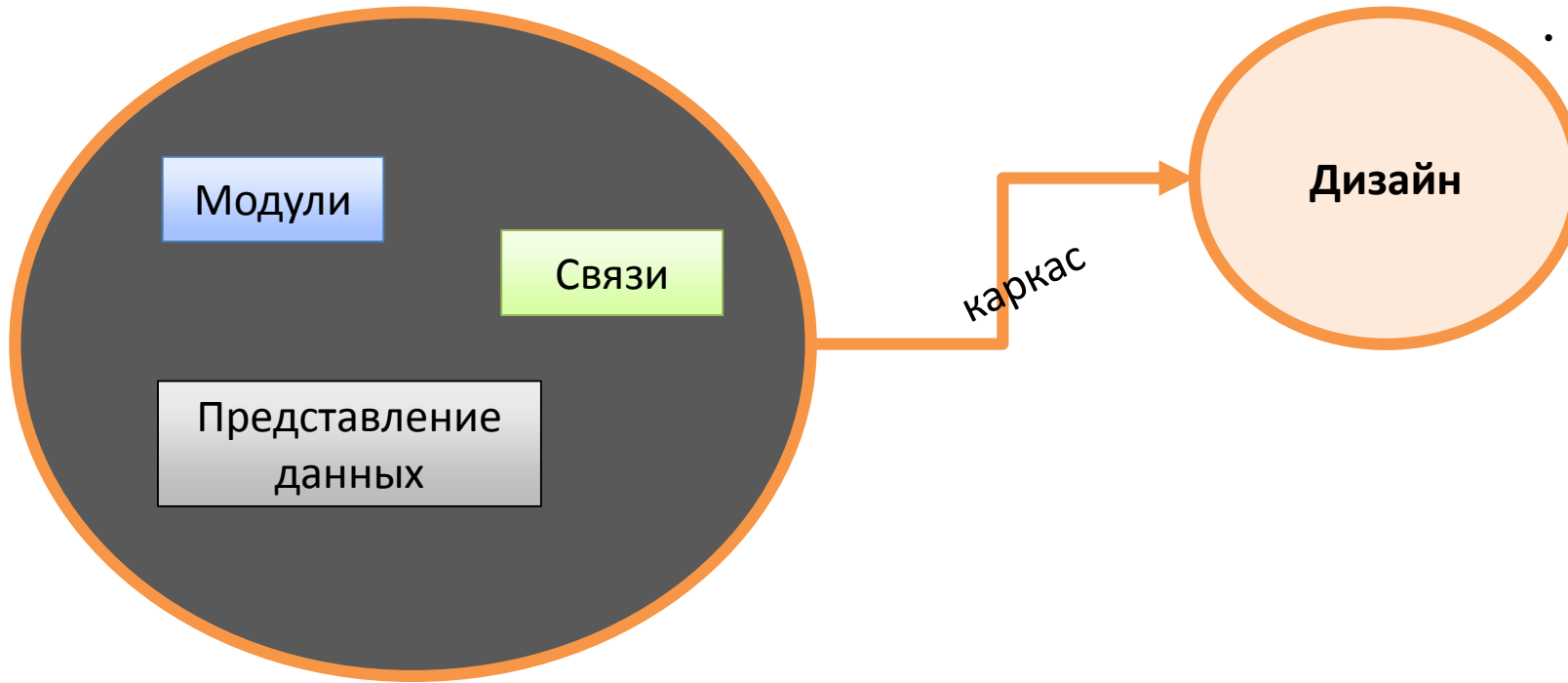
Размер проекта (KSLOC)	Оптимальные затраты на АП*, %	Снижение затрат за счет АП*, %
10	5	18
100	20	38
1000	26	63
10000	33	92

Понятие и определение архитектуры

- SEI: более 150 определений архитектуры
- Рассмотрим системный подход:
 - Частью чего является
 - Какие функции выполняет
 - Как устроена



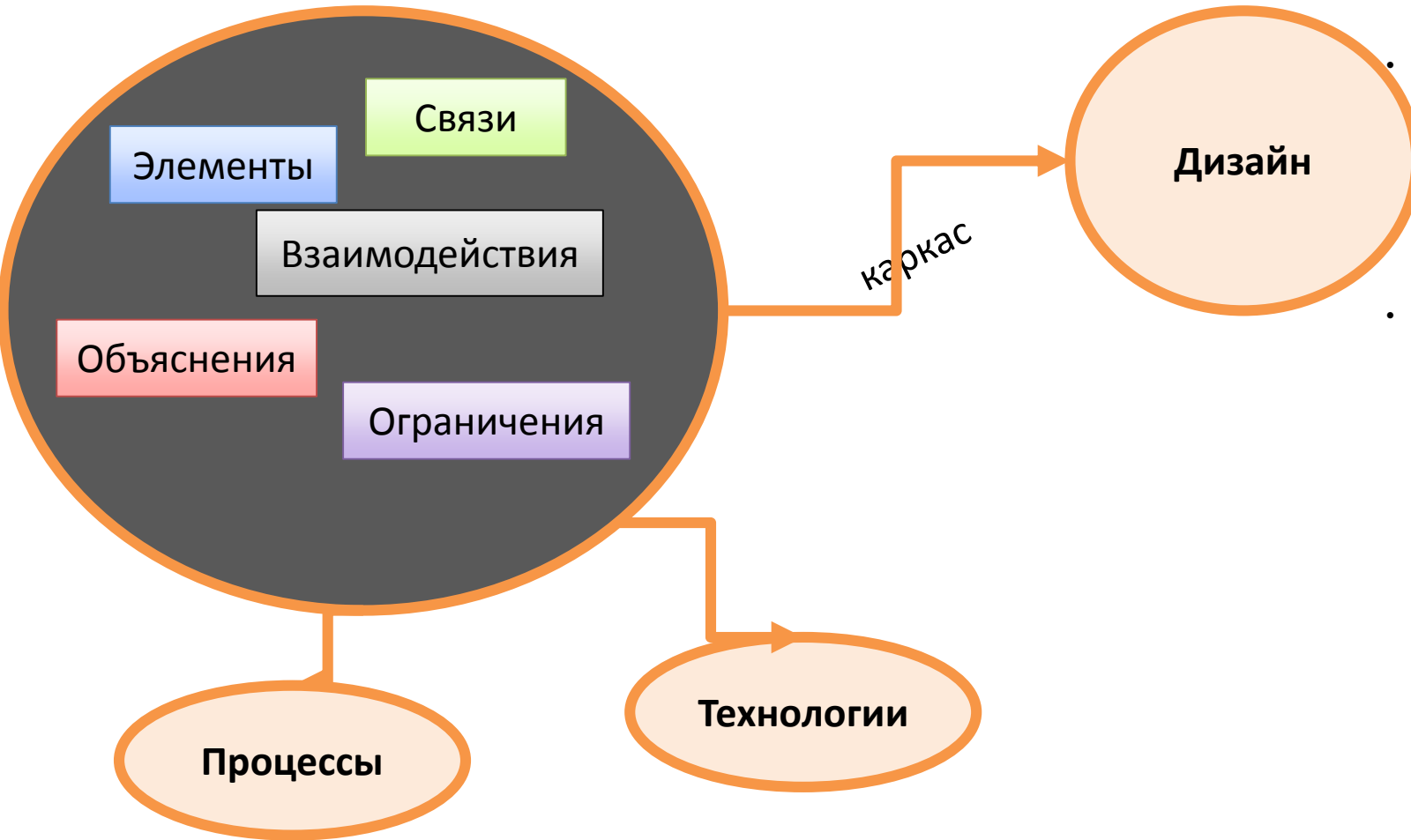
Архитектура



- *«Мы говорим только о таких спецификациях, которые определяют, что программа должна делать ... Но нужно также определять дизайн, форму; и в рамках этого каркаса разработчик должен создавать что-то... понимая, что архитектор имел в виду»*

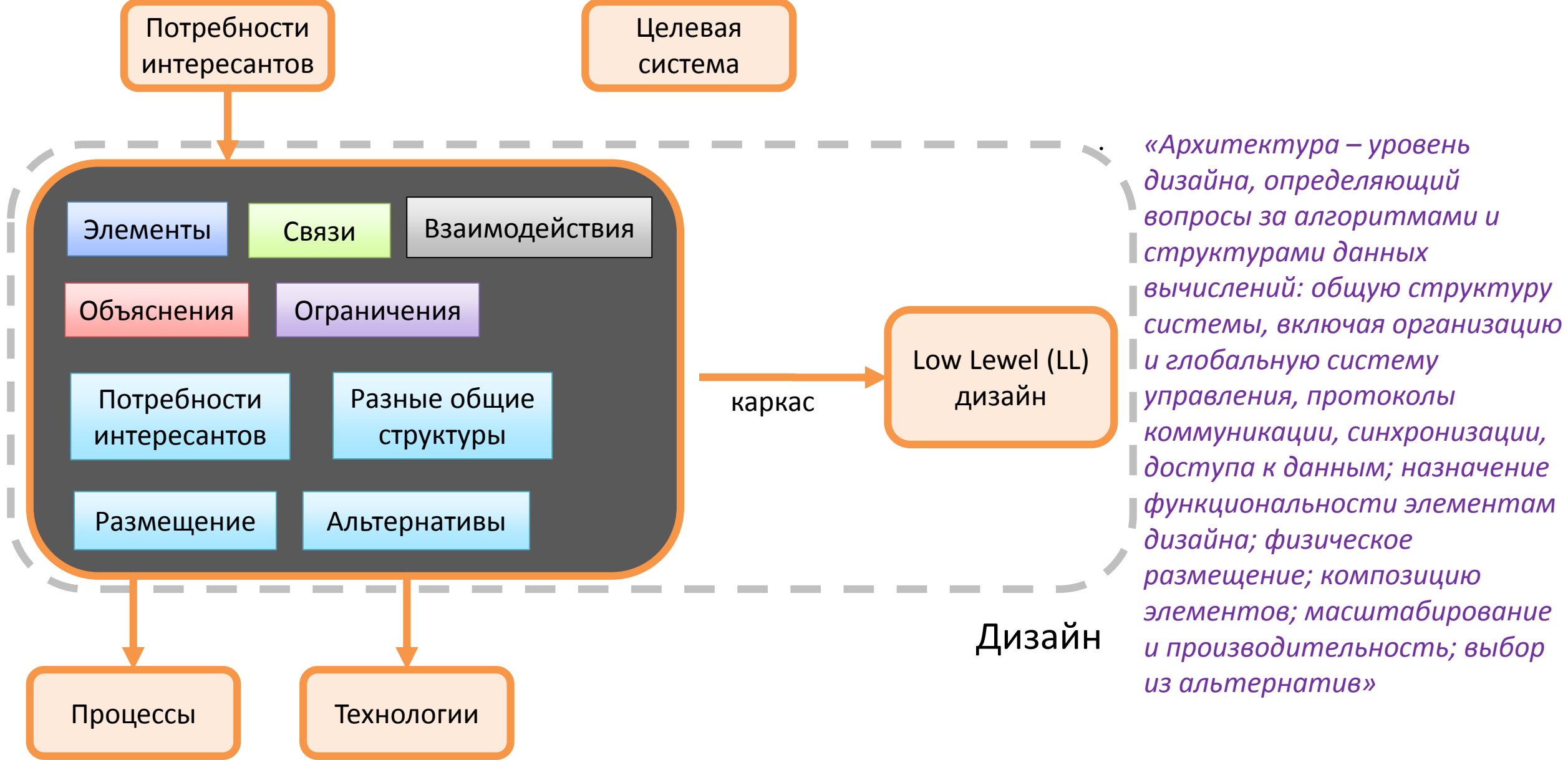
- *«Архитектура включает разделение функций между модулями, средства связи между модулями и представление разделяемых данных»*

Архитектура

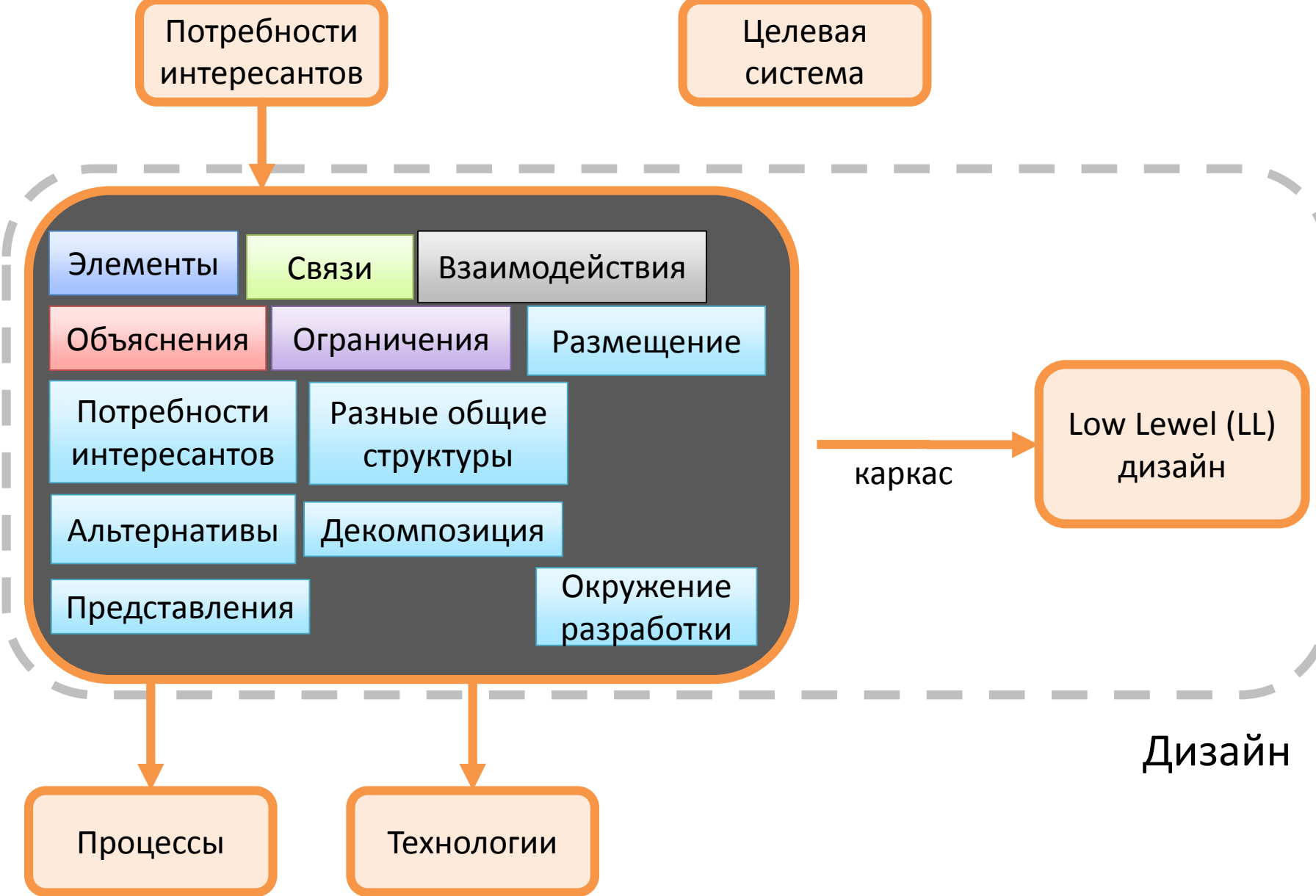


«Архитектура – связующее звено между технологиями и процессами»

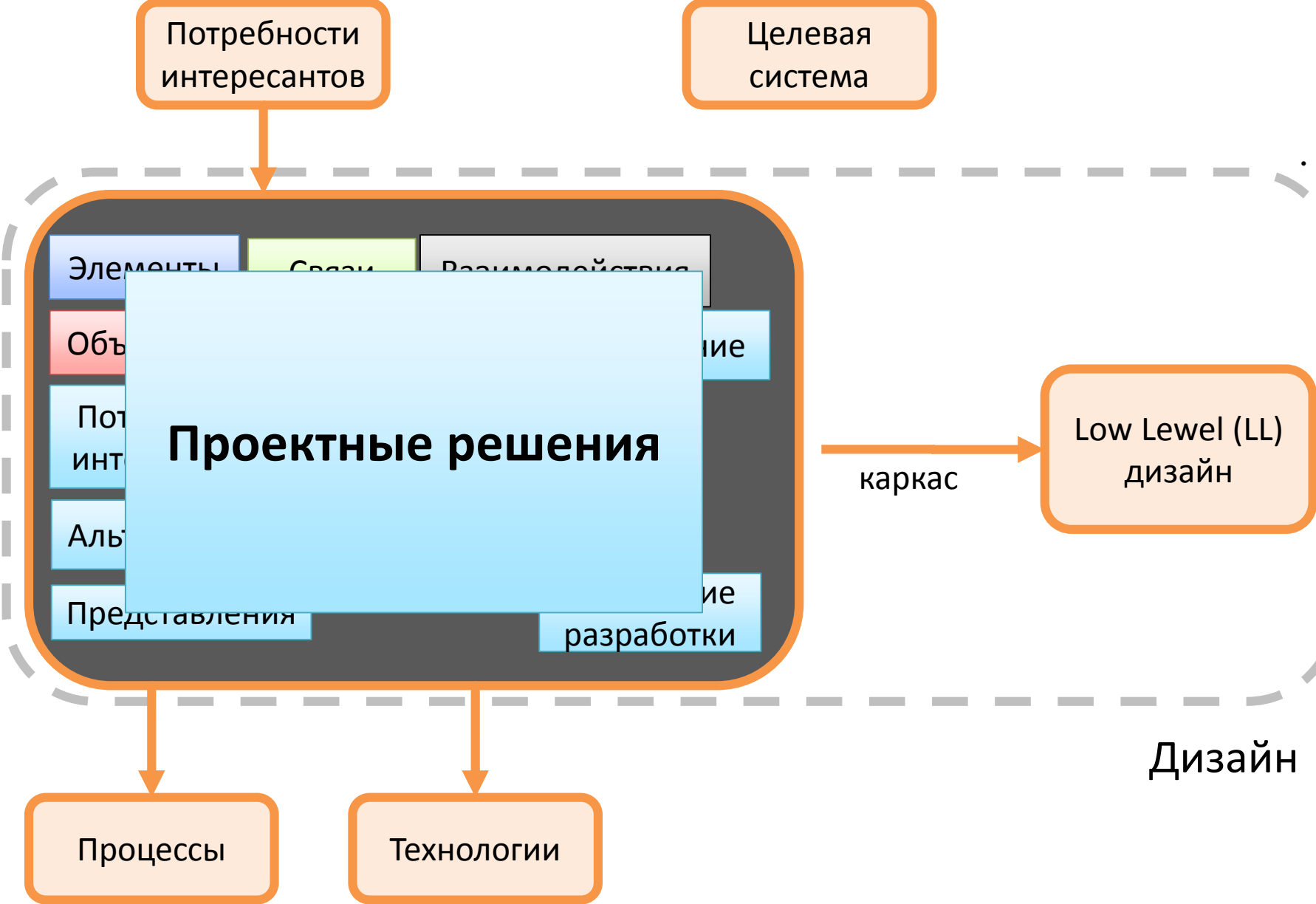
«Архитектура определяет выбор элементов, их взаимодействий и ограничений на эти элементы и взаимодействия, необходимые, чтобы обеспечить каркас, в котором будут удовлетворяться требования и который будет служить базой для дизайна.»



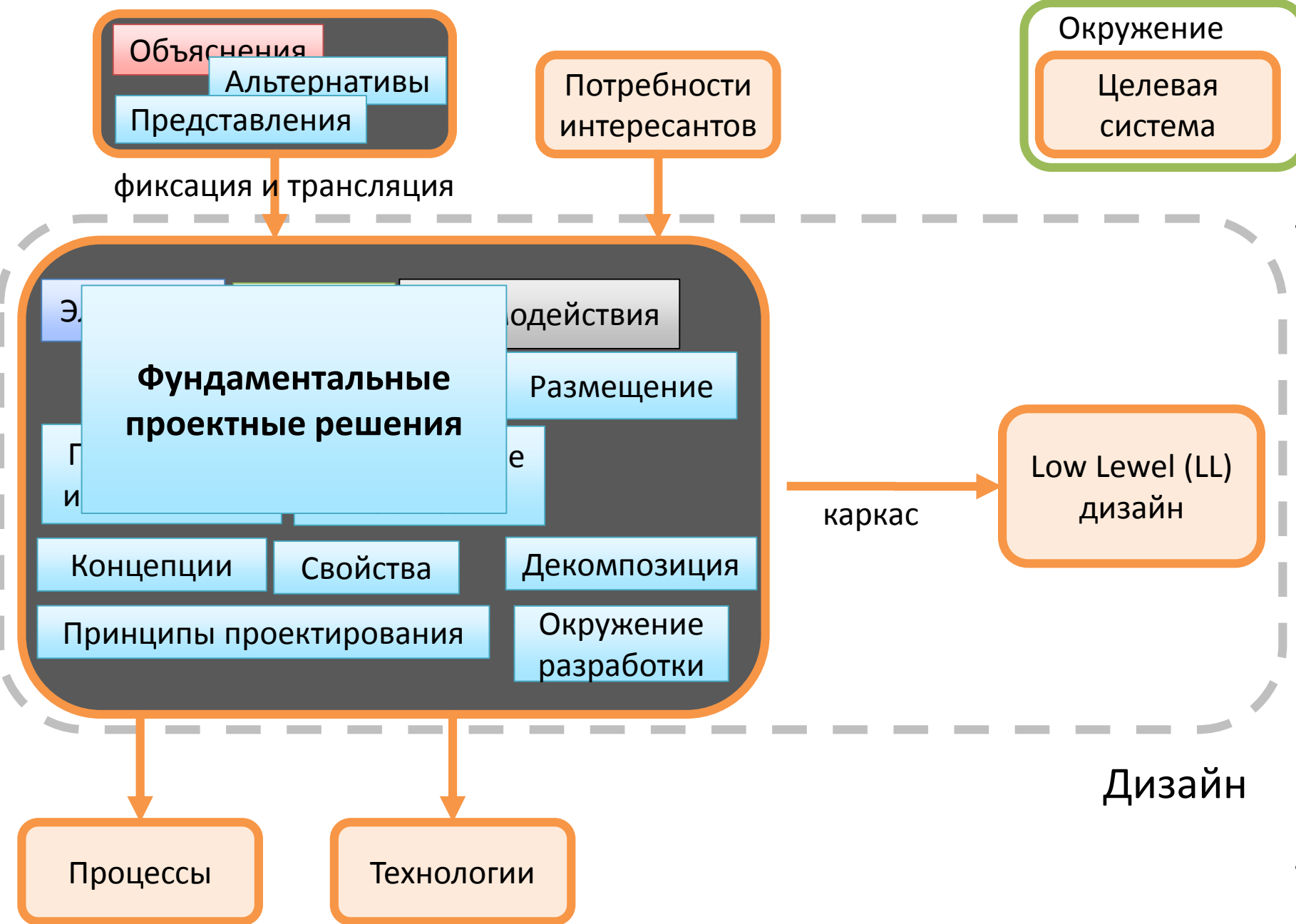
«Архитектура включает: набор компонентов, связей и ограничений; набор утверждений о потребностях заинтересованных сторон; объяснения, показывающие, что компоненты, связи и ограничения определяют систему, которая (если будет реализована) ответит потребностям заинтересованных сторон»



«Архитектура включает четыре представления:
1. концептуальная а. – элементы и связи
2. модельная а. – функциональная и слоевая декомпозиция
3. а. выполнения – динамическая структура
4. а. кода – организация исходников, бинарников и библиотек в окружении разработки»



«Архитектура состоит из всех важных проектных решений по поводу структур программ и взаимодействий между этими структурами, которые составляют системы. Проектные решения обеспечивают желаемый набор свойств, которые должна поддерживать система, чтобы быть успешной. Проектные решения предоставляют концептуальную основу для разработки системы, ее поддержки и обслуживания»



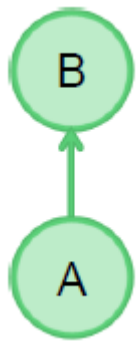
- «Архитектура – фундаментальные концепции и свойства системы в ее окружении, выраженные в ее элементах, отношениях и принципах их проектирования и развития»
- ISO 42010

Стандарт ISO 42010

- Дает приемлемое определение
- Вводит ряд связанных понятий
- Разделяет (разводит) архитектуру и ее описание
- Содержит развернутые рекомендации на описание архитектуры
- Для задач управления недостаточен

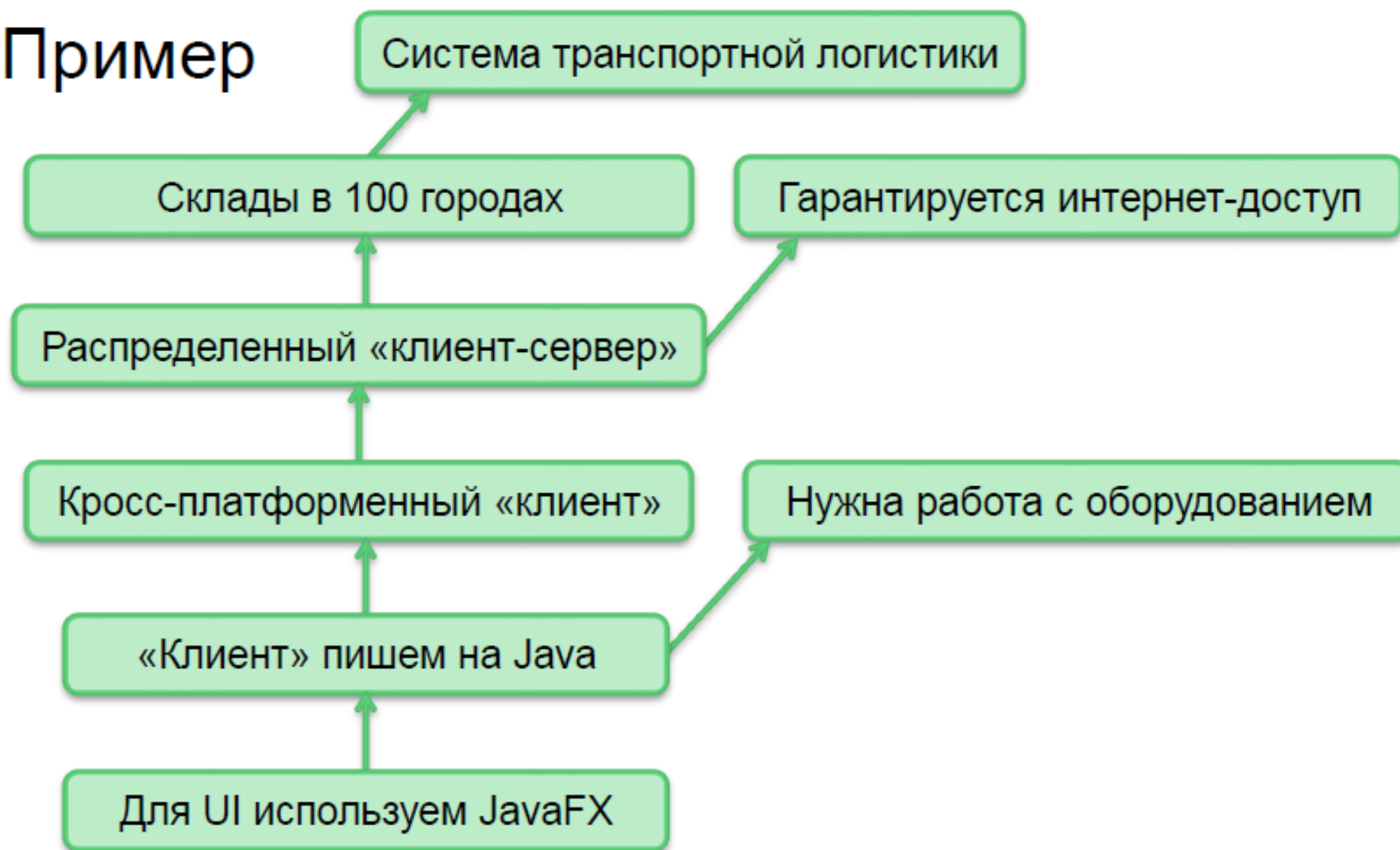
Свойства архитектурных решений

- Элементы, модули, связи, структуры, виды, формы, каркасы, разделения, ограничения, интересанты, потребности, окружение, размещение, управление, интерфейсы, слои, артефакты, технологии, статика и динамика, поведение, свойства, отношения, принципы, стили ...
- Т.е. самые разнообразные **проектные решения, только если они необходимы.**

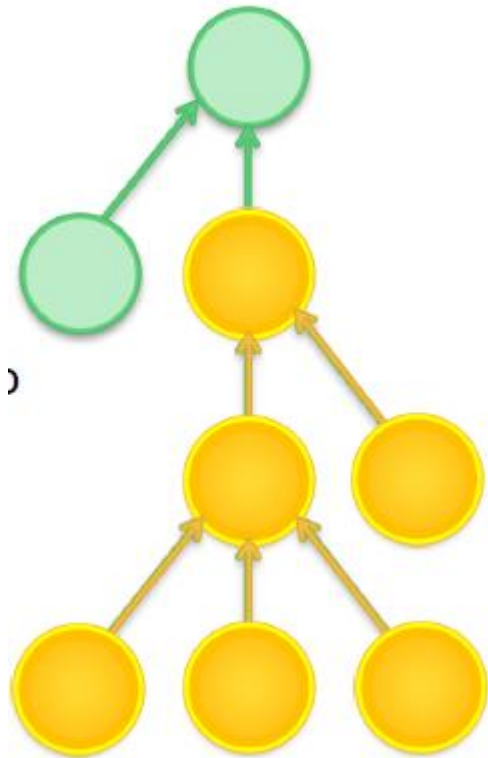


- Решение А зависит от решения В, если А имеет смысл или целесообразно только в том случае, когда принято и актуально решение В

Пример



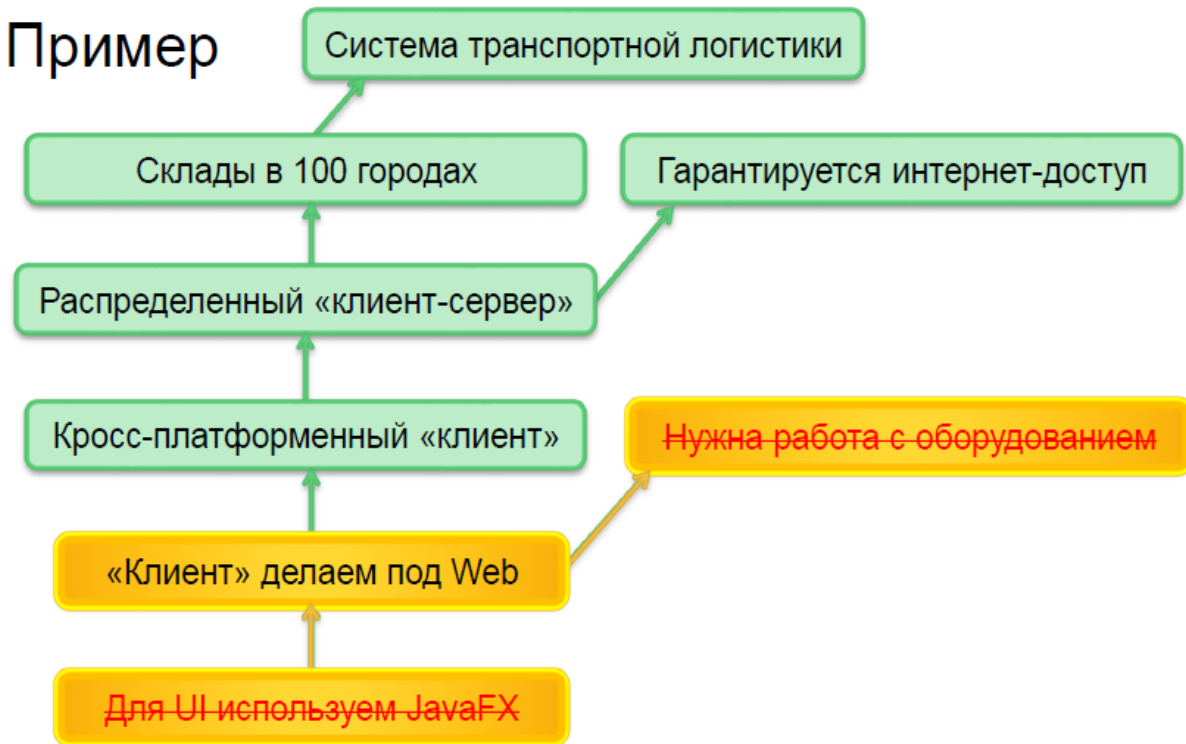
Свойства зависимостей решений



- Зависимости не бывают цикличны, решения образуют направленный граф («дерево решений»)
- Если изменяется некоторое решение, то придется **пересмотреть все решения**, которые прямо или косвенно зависят от него
- Решения ближе к корню дерева менять сложно, решения ближе к листьям менять проще
- Положение в структуре зависимостей определяет уровень решения
- Фундаментальное решение (околокорневые) составляют содержание архитектуры системы

Свойства зависимостей решений

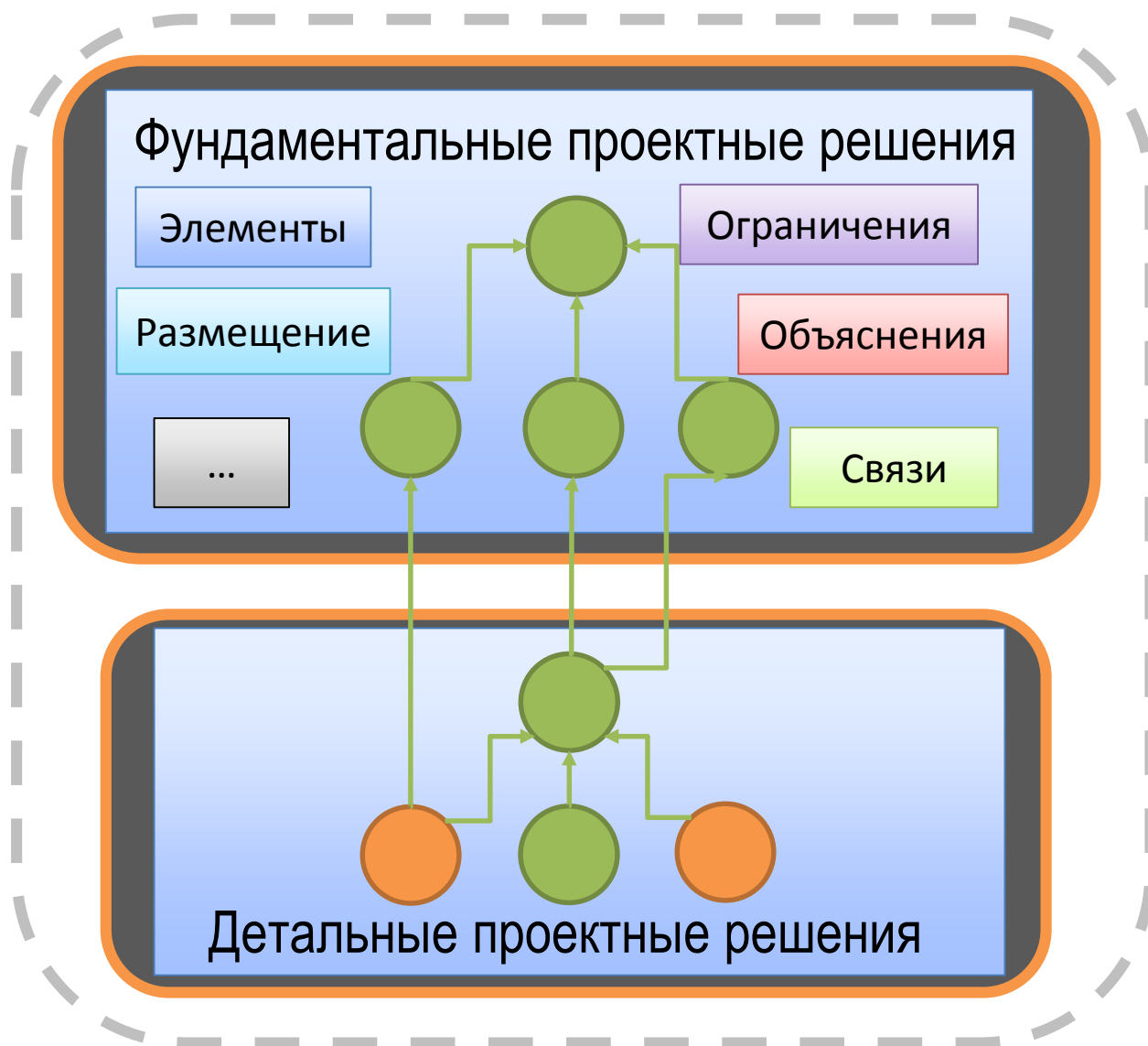
Пример



Пример



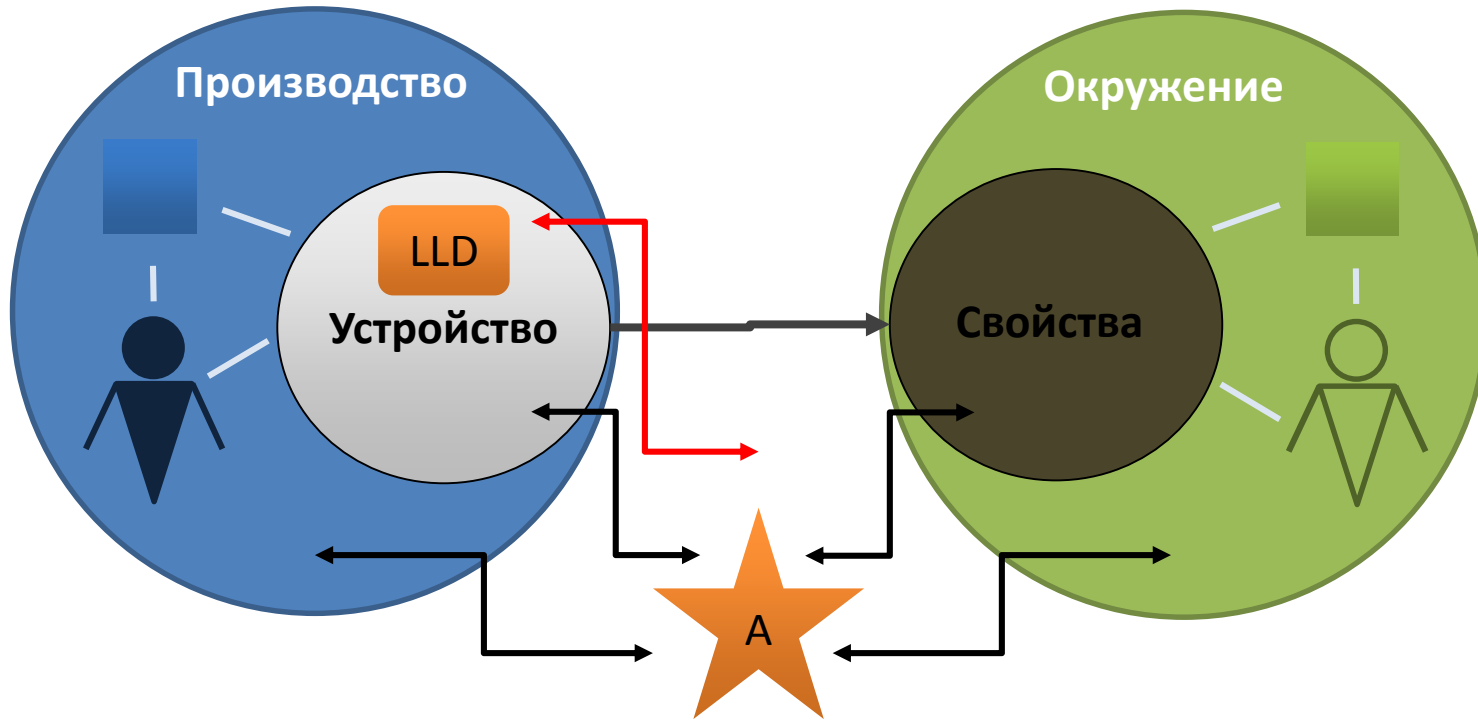
Дизайн



Архитектура

LL-дизайн

Области проектных решений

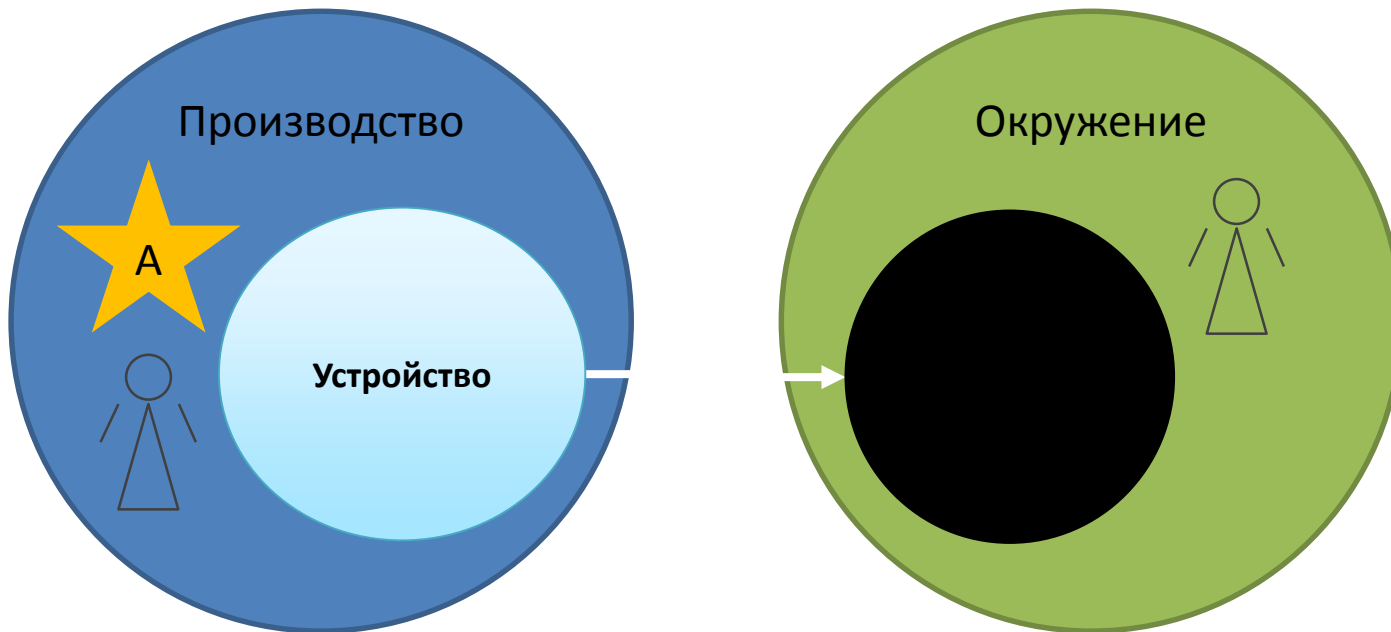


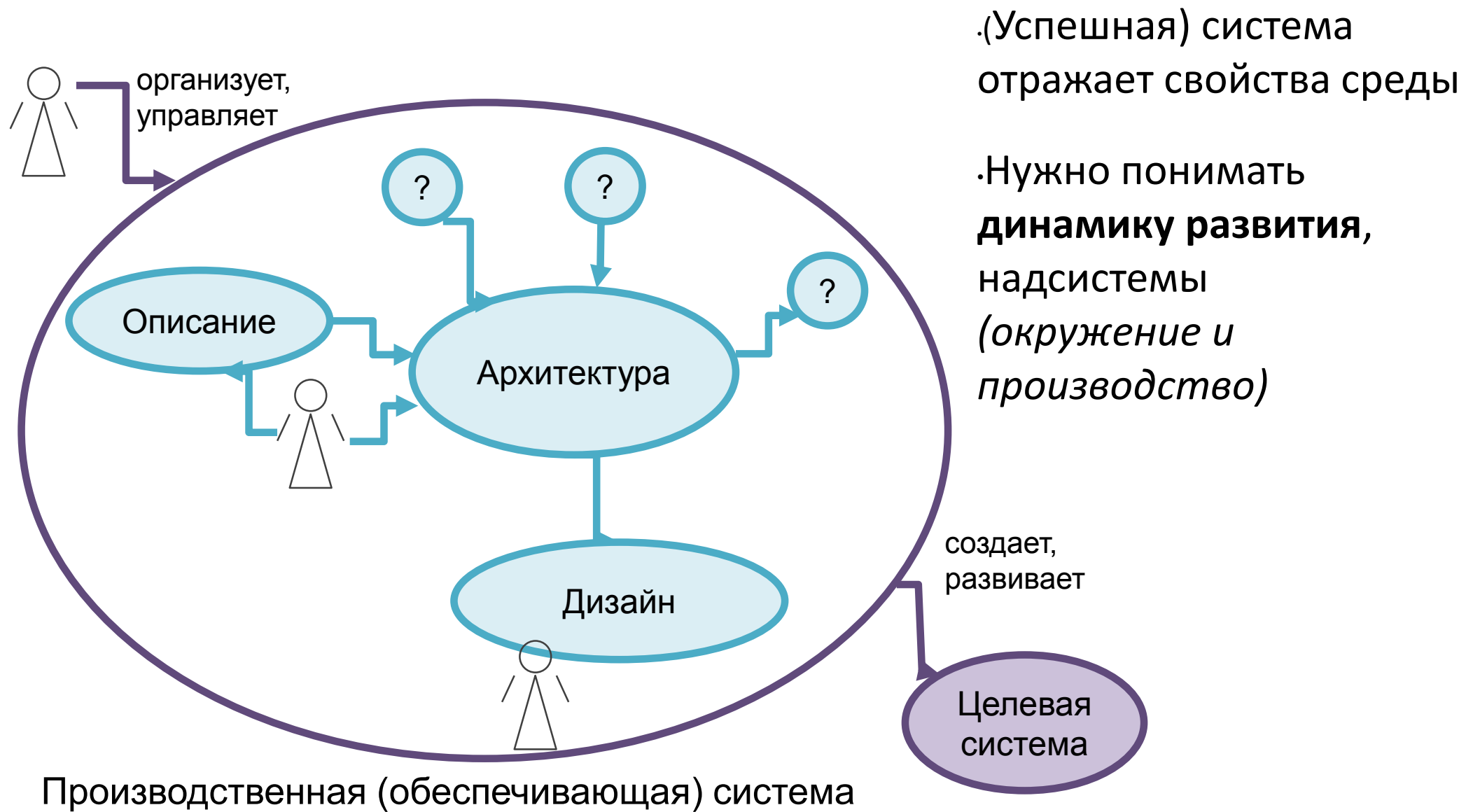
- Свойства целевой системы (черный ящик)
- Устройство системы (прозрачный ящик)
- Окружение целевой системы (надсистема)
- Производство (обеспечивающая система)

- В архитектуру могут входить решения не только об устройстве, если они существенно влияют на устройство

Надсистема архитектуры

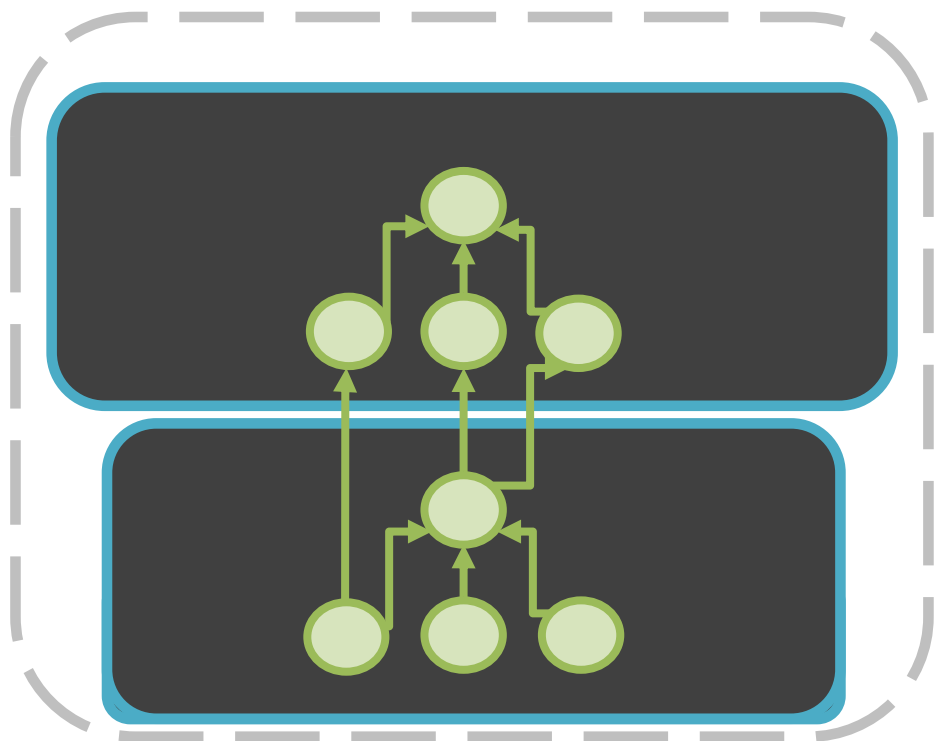
- Архитектура не «внутри» целевой системы
- Архитектура – внутри производственной системы





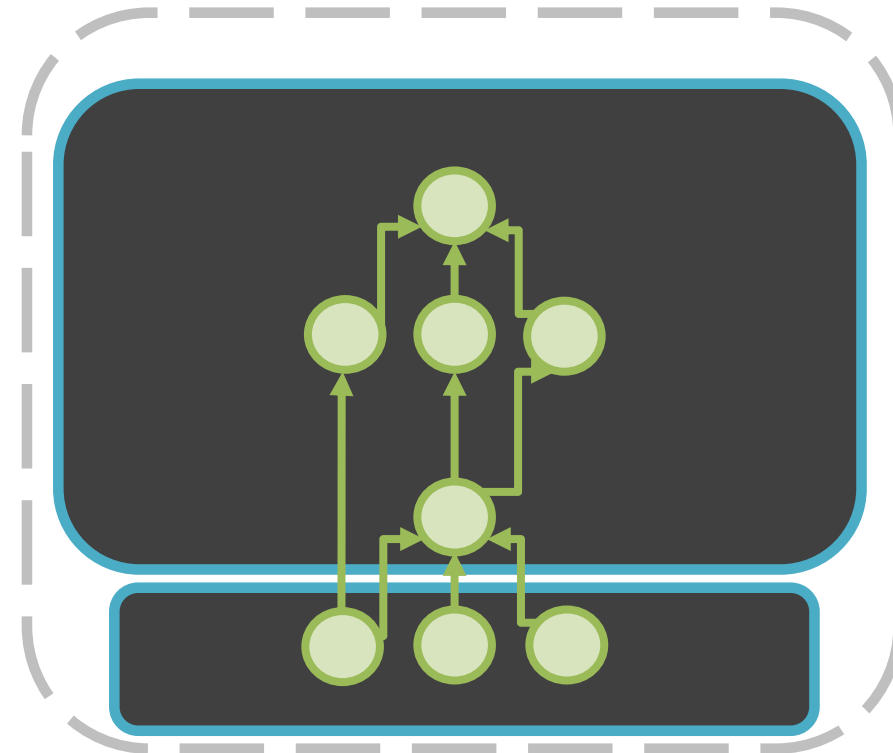
Области проектных решений

- Граница «архитектура-дизайн» определяется «мерой существенности» («фундаментальности») проектных решений
- Т.е. граница **подвижна**, архитектура и дизайн – **относительны**



Архитек
тура

LL-
дизайн



Архитек
тура

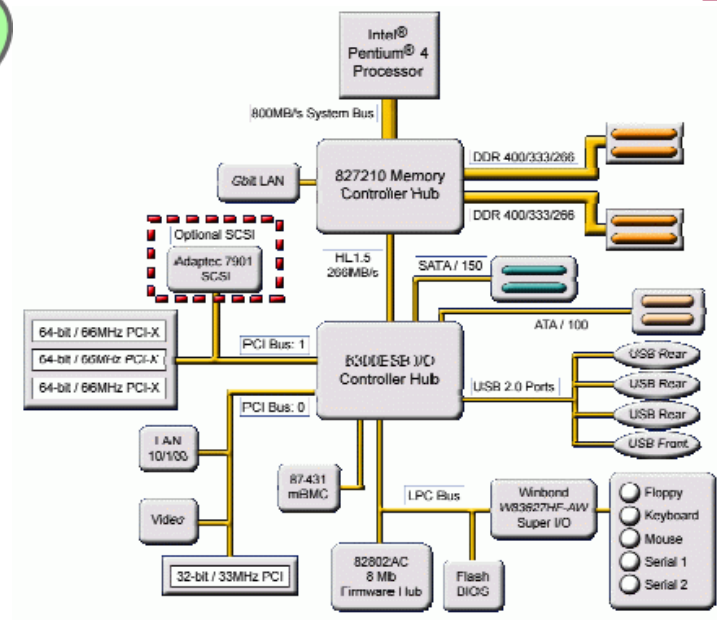
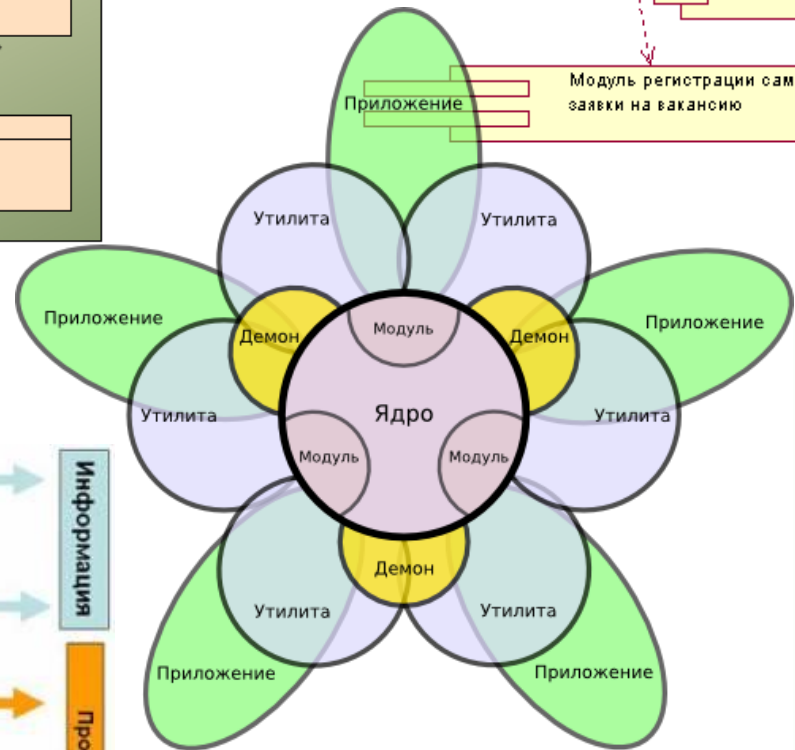
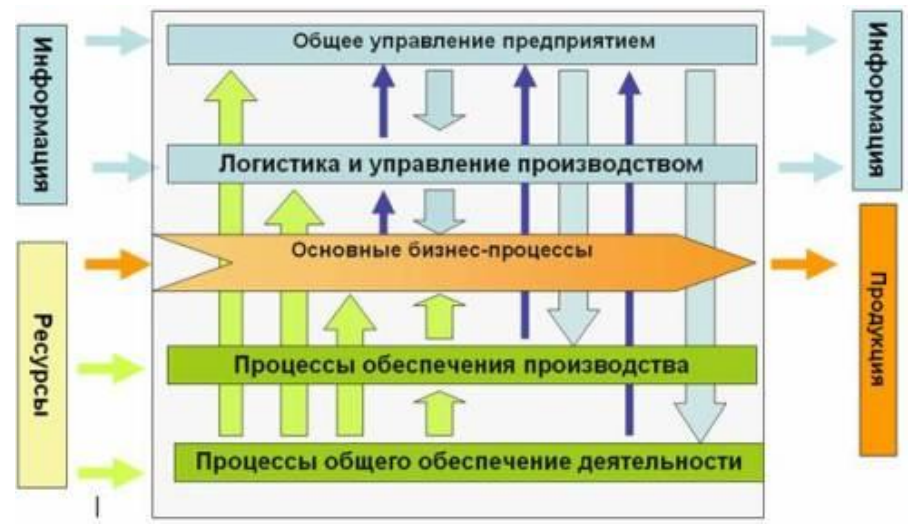
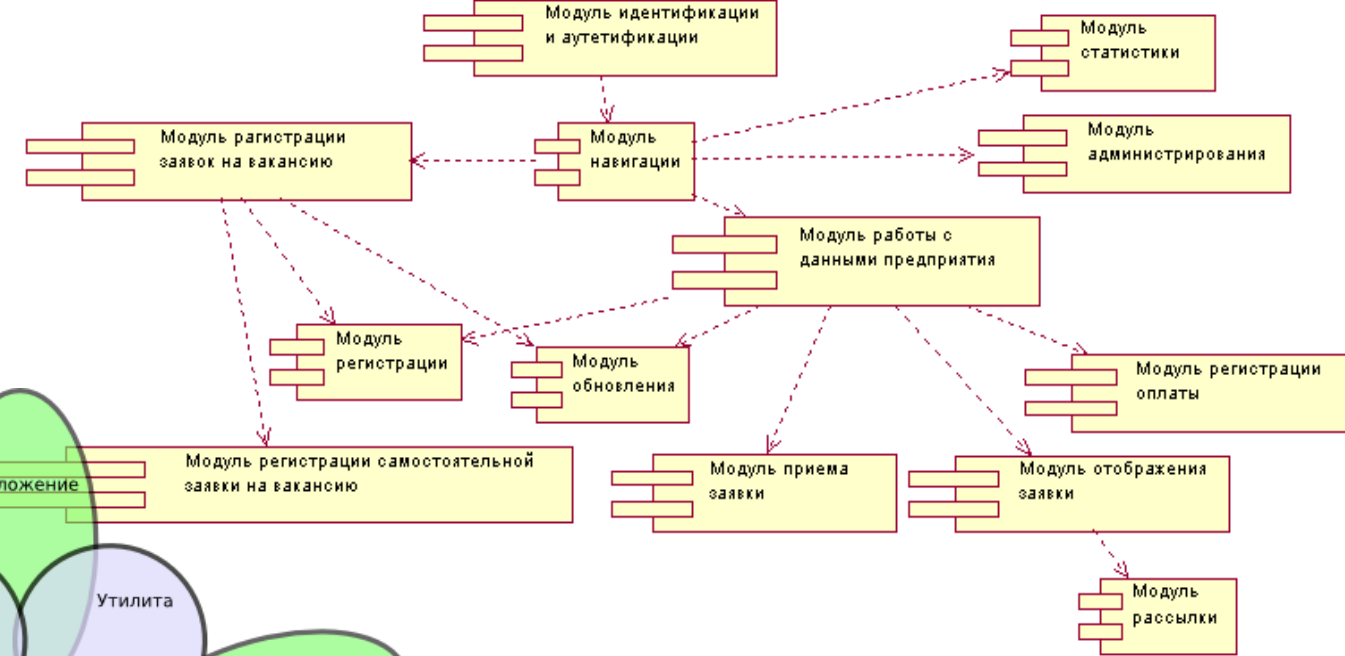
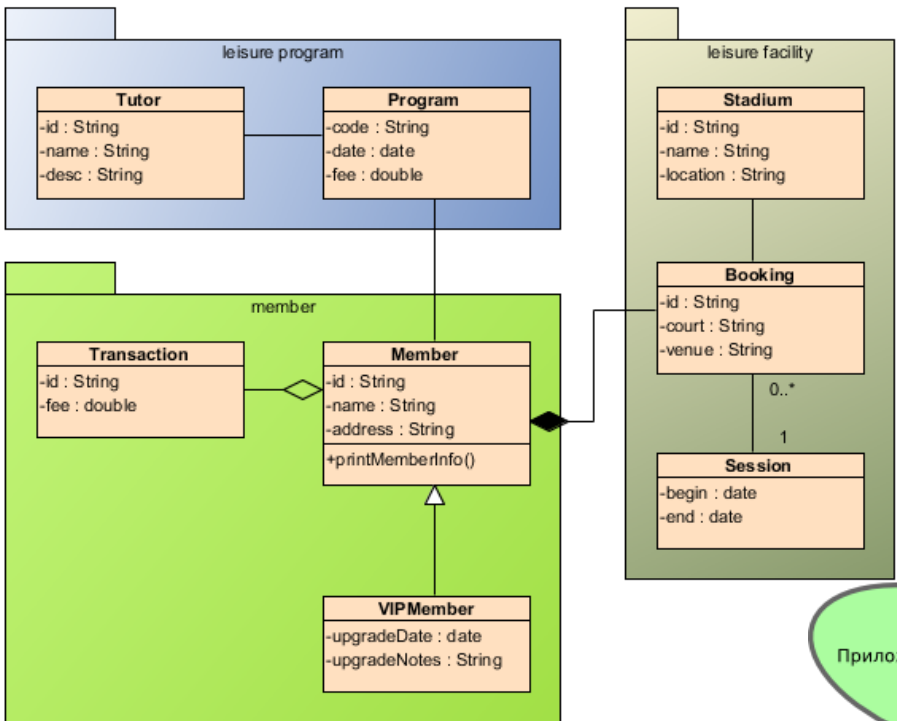
LL-
дизайн

~~Гибкая~~ архитектура

- Не бывает
- Гибкой (легко изменяемой) может быть только структура
 - *«Мы очень легко можем сделать любой аспект ПО гибким. И это не сложно. Но мы не можем сделать ПО гибким во всех аспектах. Это породит сложность, а сложность трудноизменяема» Ральф Джонс*

Модели в архитектуре

- Основная форма представления и описания проектных решений
- Вся современная инженерия моделиеориентирована
- Не все модели одинаковы полезны
 - Детализация
 - Вид модели
 - Уровень абстракции
- Некоторые решения плохо выражаются классическими box-and-line моделями (ограничения, принципы)



Цели архитектуры

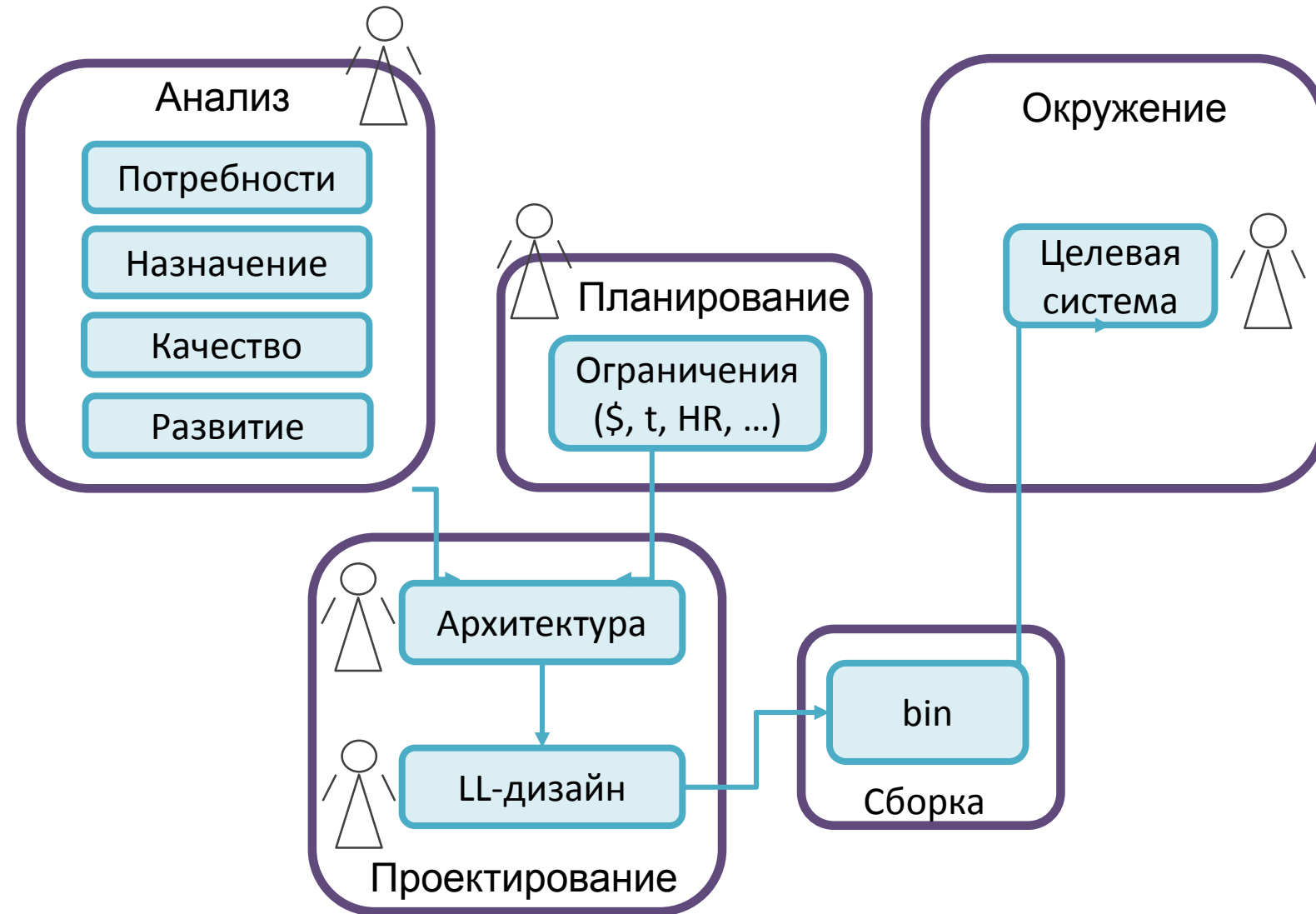
- Назначение системы **Не требования!**
- Атрибуты качества
- Ограничения (включая финансы и время)
- Свойства окружения

Архитектура - стратегия проектирования

Функция архитектуры - **направлять** детальное проектирование

Анализ, архитектура и дизайн

связаны **логически** (как зависимые решения), но **не хронологически!** сложнее, много итераций



- Архитектор или команда?
- **НЕ ИМЕЕТ ЗНАЧЕНИЯ!**
- До тех пор, пока ответственность не лежит ни на ком и архитектуры **нет** как механизма управления в производстве

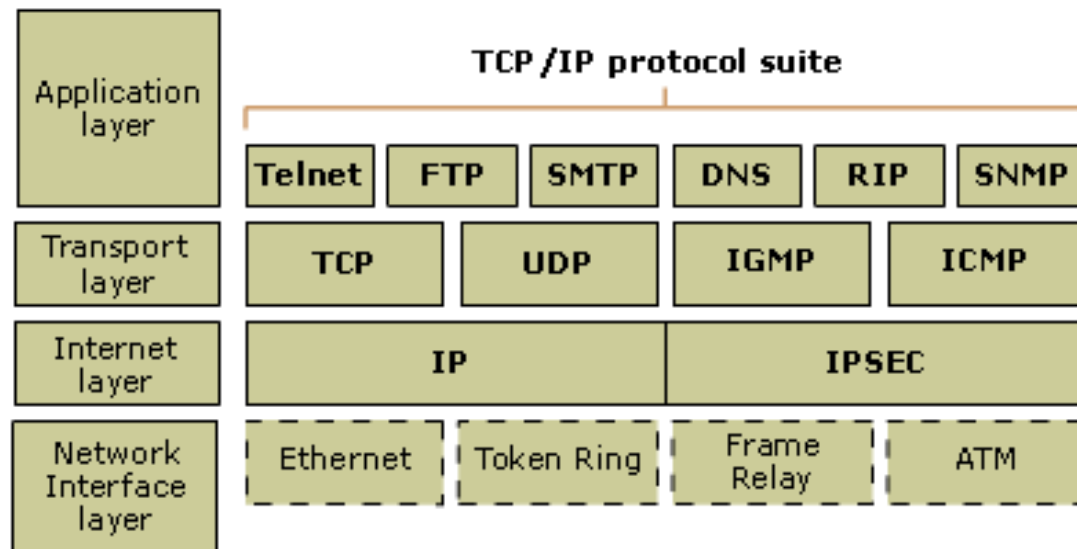
Выводы

- Про архитектуру можно разговаривать предметно!
 - Называть конкретные решения
 - Показывать, что они фундаментальны для устройства
 - Обосновывать этим, что решение входит в архитектуру
 - Находить подходящие модели и тексты для наилучшего выражения решений

Интернет

- Из чего состоит архитектура интернета?
- Элементы, соединения и топология практически произвольны
- Ключевые решения собраны в **стеке протоколов TCP/IP**
- Структура гибка (произвольна), архитектура – нет!

TCP /IP model



Процессор

- Фредерик Брукс:
- *«Архитектура процессора – это его система команд»*

