Системная архитектура

* Architecture *

Основные темы курса

- Дисциплина СА, ее роль и место в структуре знаний ИТ-специалиста
- Понятие архитектуры
  - Необходимость (Organizational Need and Drivers)
  - Заинтересованные стороны (Stakeholders)
  - Риски и препятствия (Barriers)
  - Стратегия [внедрения]
  - Роль [в организационно-штатной структуре] (Positioning)
  - Перспективы (Perspective Taking)
  - Безопасность и управление рисками
- Разработка / построение архитектуры
- Управление и использование

Определения архитектуры

Связь с другими ИТ-дисциплинами

ВВЕДЕНИЕ

Определения – 1/5

- Системная архитектура (Architecture) — имеет множество определений
- ISO 42010 (2011): Архитектура (системы) – фундаментальная организация системы, реализованная в ее компонентах, их взаимосвязях друг с другом и с окружающей средой, и руководящие правила проектирования и развития системы
- Архитектура системы это набор структур, необходимых для рассуждений о системе, каковые структуры состоят из элементов, отношений и свойств этих элементов и отношений
  - Software Architecture: Perspectives on an Emerging Discipline by Mary Shaw, David Garlan (1996)

Определения – 2/5

- Архитектура - это **набор значимых решений** по поводу организации системы программного обеспечения, набор структурных элементов и их интерфейсов, при помощи которых компонуется система, вместе с их поведением, определяемым во взаимодействии между этими элементами, компоновка элементов в постепенно укрупняющиеся подсистемы , а также **стиль архитектуры** который направляет эту организацию – элементы и их интерфейсы, взаимодействия и компоновку
  - Philippe Kruchten, The Rational Unified Process: An Introduction
- Архитектура программы или компьютерной системы - это структура или структуры системы, которые включают элементы программы, видимые извне свойства этих элементов и связи между ними
  - Len Bass, Paul Clements, and Rick Kazman, Software Architecture in Practice

Определения – 3/5

- OMG UML 1.5: [Архитектура - это] структура организации и связанное с ней поведение системы. Архитектуру можно рекурсивно разобрать на части, взаимодействующие посредством интерфейсов, связи, которые соединяют части, и условия сборки частей. Части, которые взаимодействуют через интерфейсы, включают классы, компоненты и подсистемы
- Архитектура программного обеспечения системы или набора систем состоит из всех важных проектных решений по поводу структур программы 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
  - James McGovern, et al., A Practical Guide to Enterprise Architecture

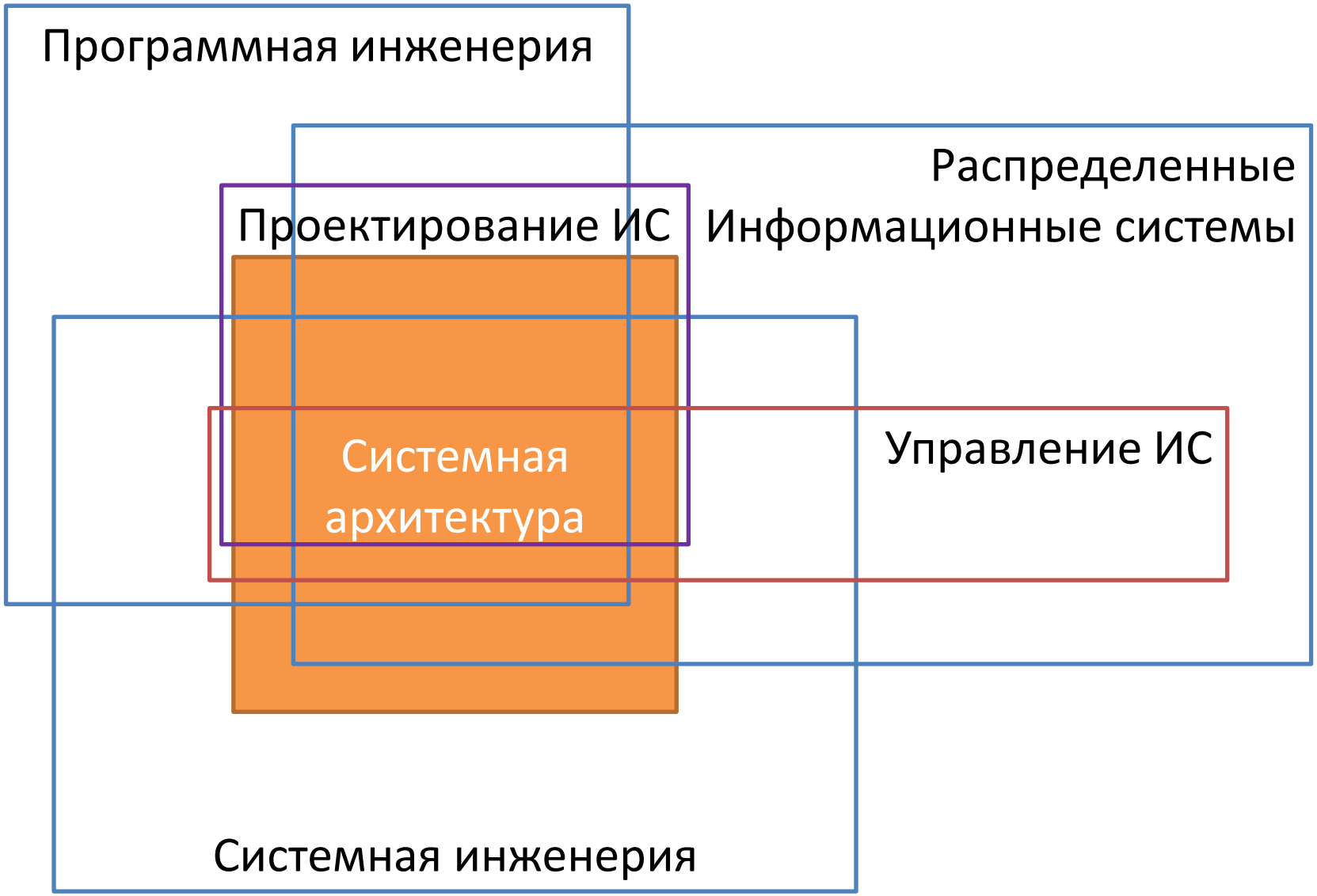
Определения – 4/5

- What Is Your Definition of Software Architecture? Fact Sheet. Carnegy Mellon University December 2010 – Набор из более чем 150 других определений архитектуры, которые предлагались профессионалами системной и программной инженерии
- Who Needs Architect? Ralf Johnson (2003): «Архитектура — это обо всём важном. Что бы это ни было» (Architecture is about the important stuff. Whatever that is)
  - Если представить архитектуру как набор инженерных решений, принимаемых для определения структуры системы, то «важные решения» — это такие, при изменении которых приходится в существенной мере изменять множество других решений, в существенной мере перепроектировать систему
- Системная архитектура – это альфа, пожалуй, важнейшая среди подальф определения системы

Определения – 5/5

- Альфа (ALPHA – Abstract-Level Progress Health Attribute) – неформально это просто «идеальный рабочий продукт», названный отдельным термином («альфой») для уменьшения путаницы с «реальными рабочими продуктами». Аббревиатура для него была подобрана задним числом
  - Альфы — это то, что изменяется в проекте, и изменения чего мы хотим понимать, отслеживать, обеспечивать, направлять, контролировать. То, что альфы (точнее, экземпляры альф) изменяются в ходе стратегирования, инженерной, организационной, операционной деятельностей, отражено в том, что альфы имеют состояния (state), а эти состояния имеют контрольные вопросы (checkpoint) для определения достижения этих состояний. Состояния альф обычно определяются на всём пути от самого появления альфы в проекте до прекращения её существования.

Место системной архитектуры



Материалы

- EABOK
  - SWEBOK, PMBOK
- SEBOK
- http://sewiki.ru – Systems Engineering Thinking Wiki

Что входит в архитектурное описание

Зачем она создается

Как она строится

На что она влияет

АРХИТЕКТУРА И ВСЕ-ВСЕ-ВСЕ

Структура

Поведение

Значимые элементы

История развития

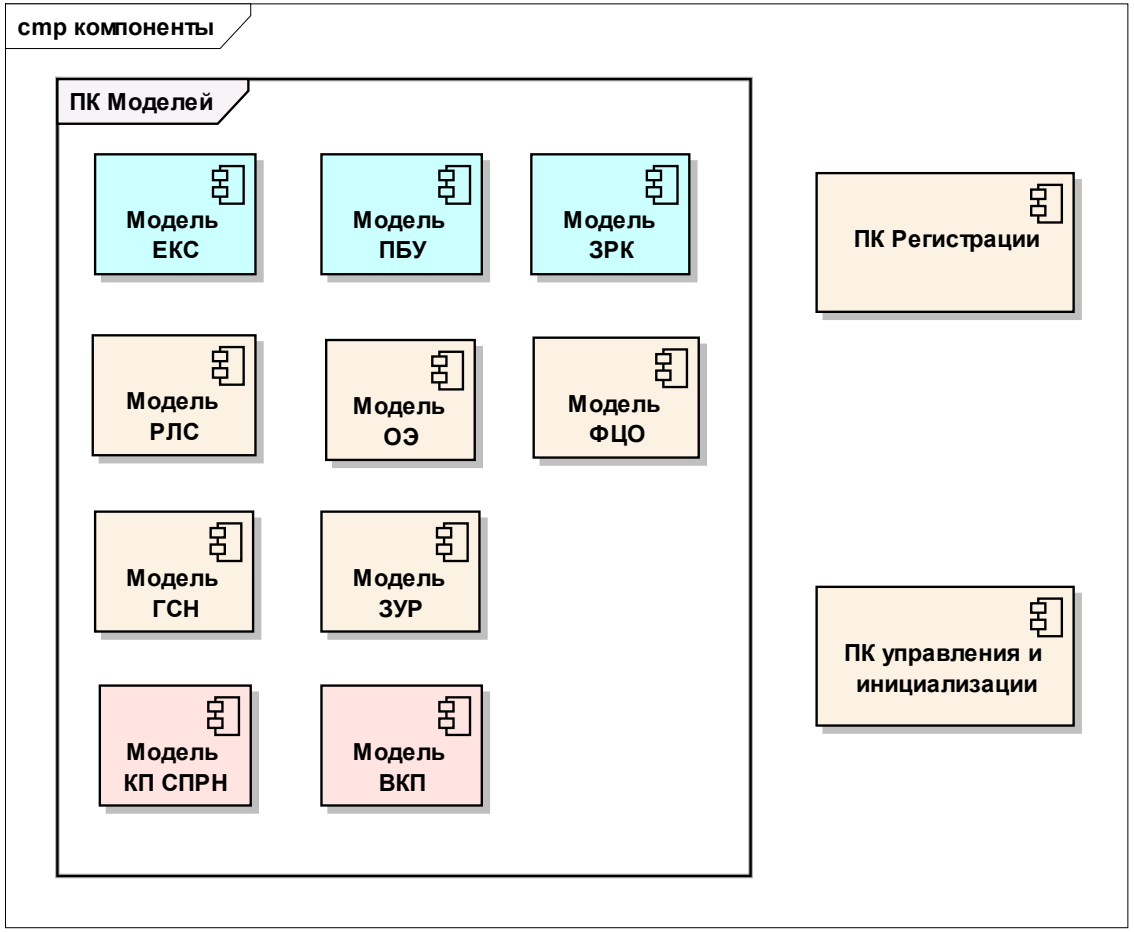
СОСТАВ АРХИТЕКТУРНОГО ОПИСАНИЯ

СТРУКТУРА СИСТЕМЫ

Архитектура и структура

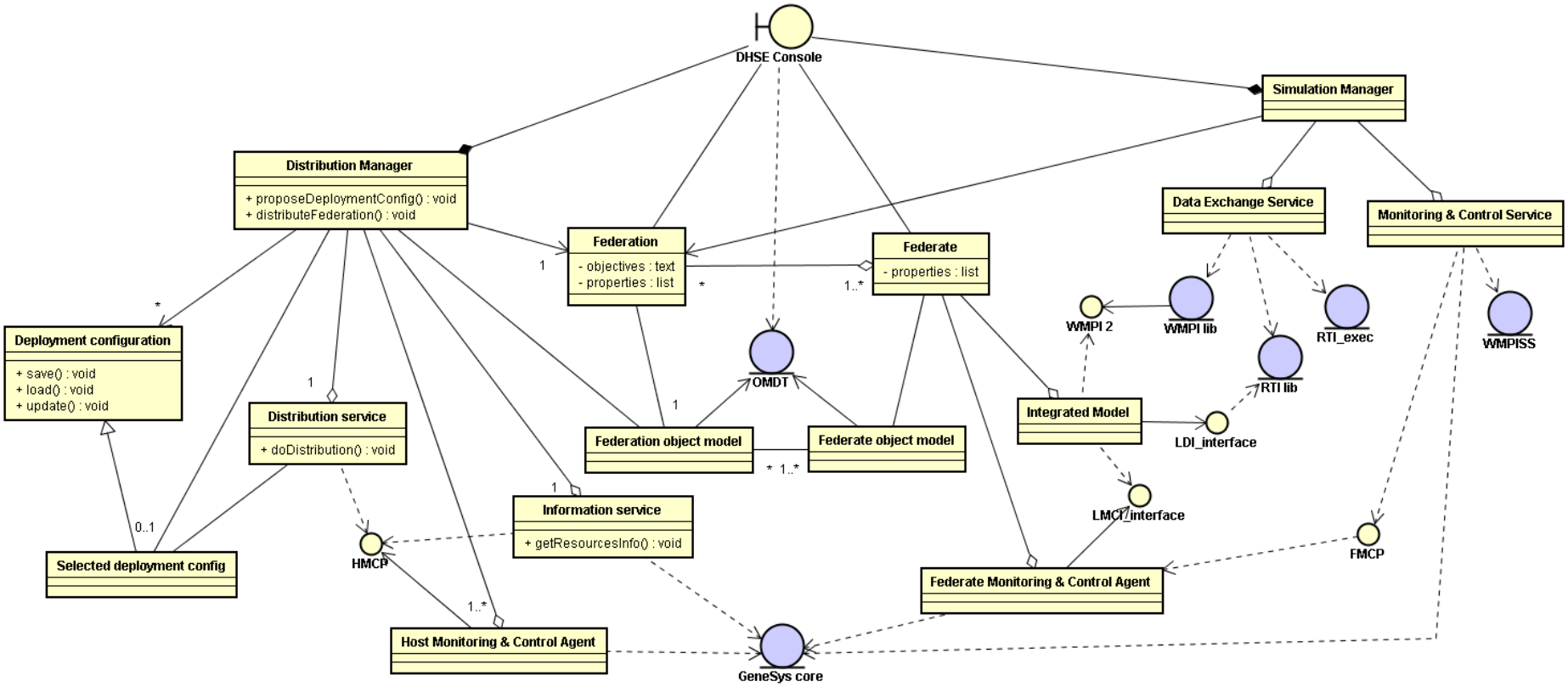
- Описание архитектуры обязательно включает описание структуры, так как надо описать, как минимум, составные части системы
- Структура – это описание системы в виде:
  - Перечня составных частей системы
  - Связей элементов
- Типы связей в структуре:
  - Часть-целое (is-part-of)
  - Классификация (inherits-to или is-a)

Структура – примеры – 1/2



Просто перечень компонентов (иерархический)

Структура – примеры – 2/2



Структура с вхождениями (is-part-of) и наследованиями (is-a)

ПОВЕДЕНИЕ СИСТЕМЫ

Архитектура и поведение – 1/2

- Архитектура определяет обобщенный алгоритм работы системы. Это описание проявляется как описание взаимодействия компонентов системы при выполнении тех или иных задач
- Способы описания взаимодействия / поведения:
  - Блок-схема алгоритма
  - Диаграмма потоков данных

Архитектура и поведение – 2/2

- Способы описания поведения (продолжение):
  - UML: диаграмма последовательности
  - UML: диаграмма деятельности (activity)
  - UML: диаграмма коммуникации (communication)
  - UML: диаграмма состояний
  - UML: диаграмма обзора взаимодействия (interaction overview)
  - IDEF диаграммы

Поведение – примеры – 1/4

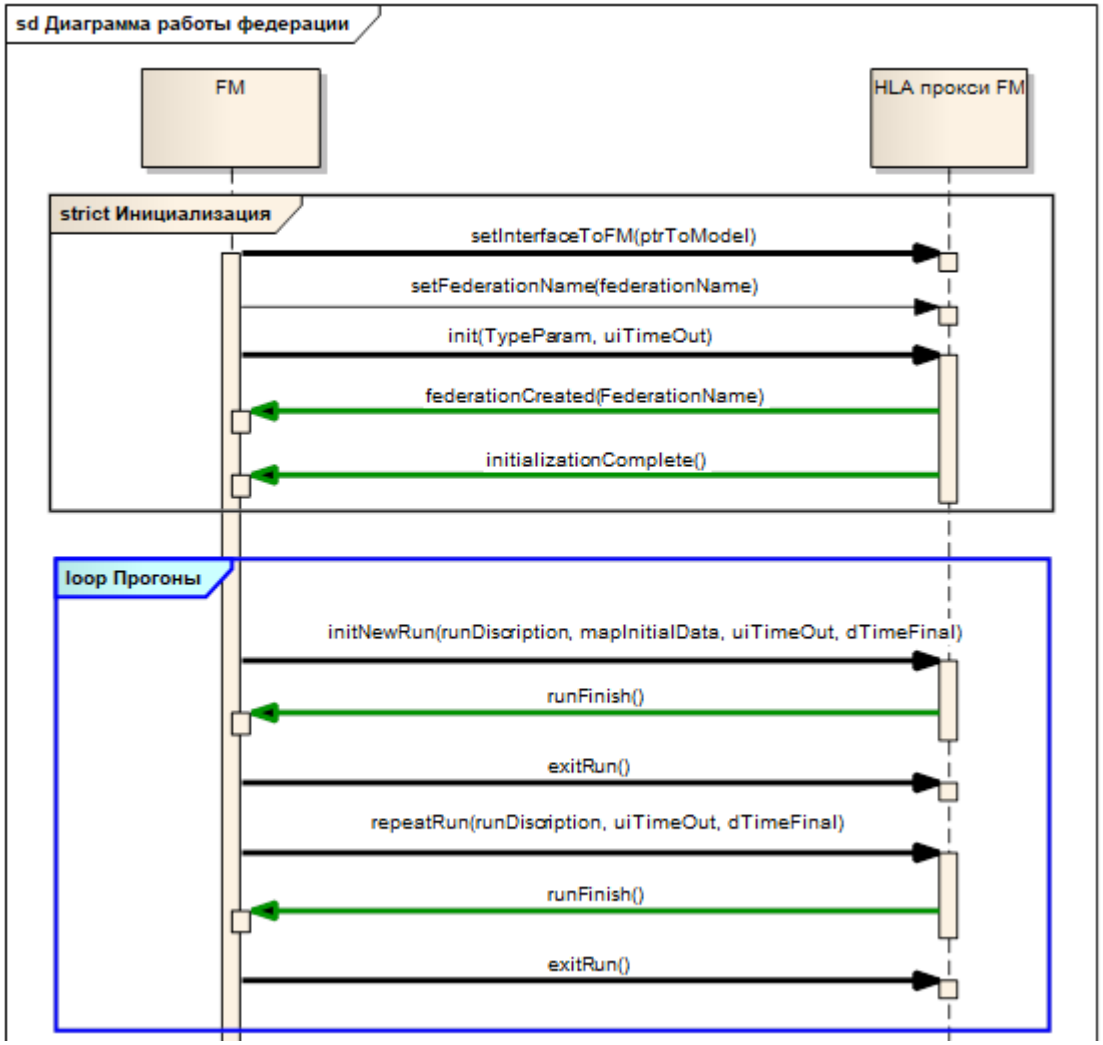


Диаграмма последовательности

Поведение – примеры – 2/4

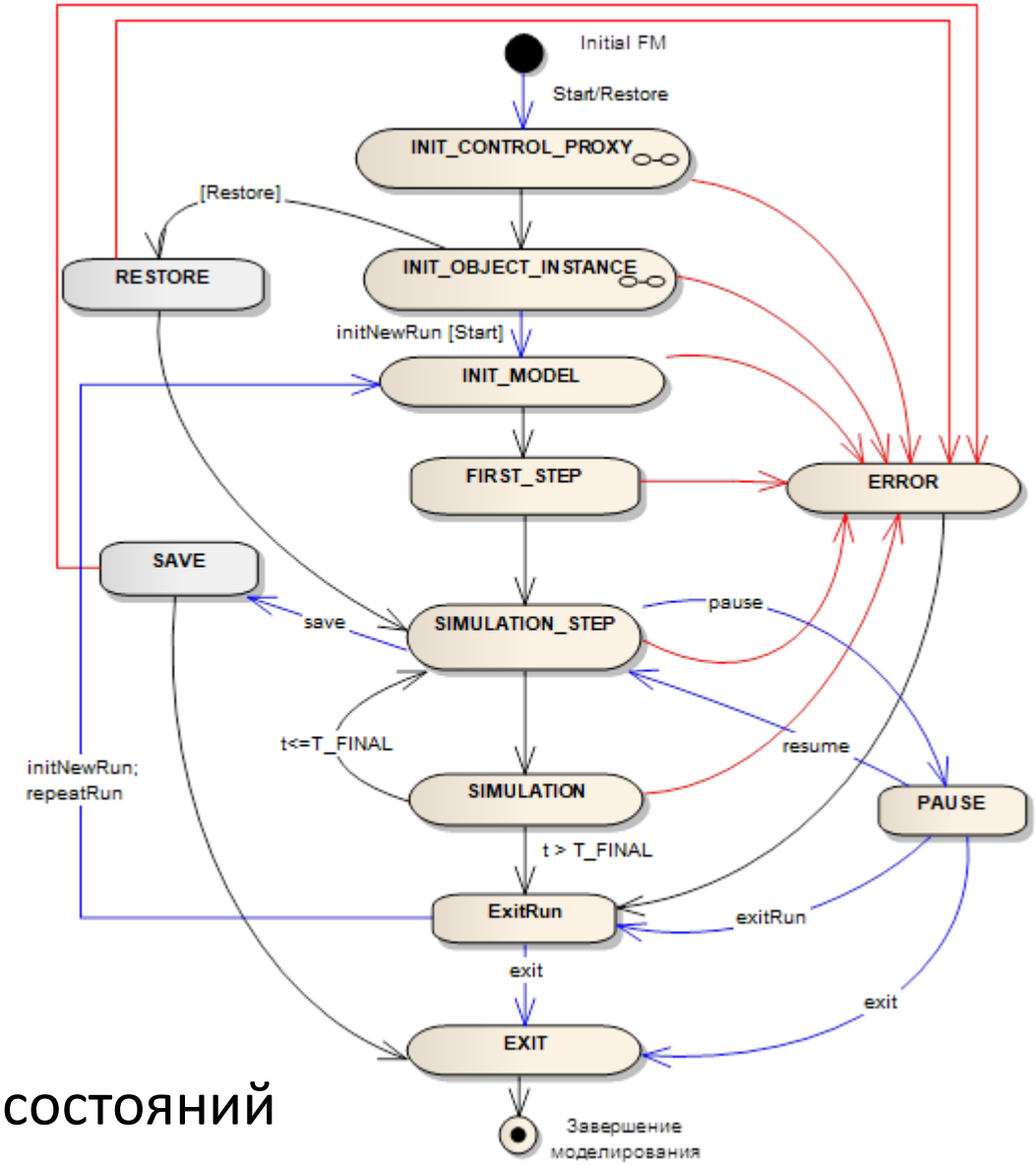


Диаграмма состояний

Поведение – примеры – 3/4

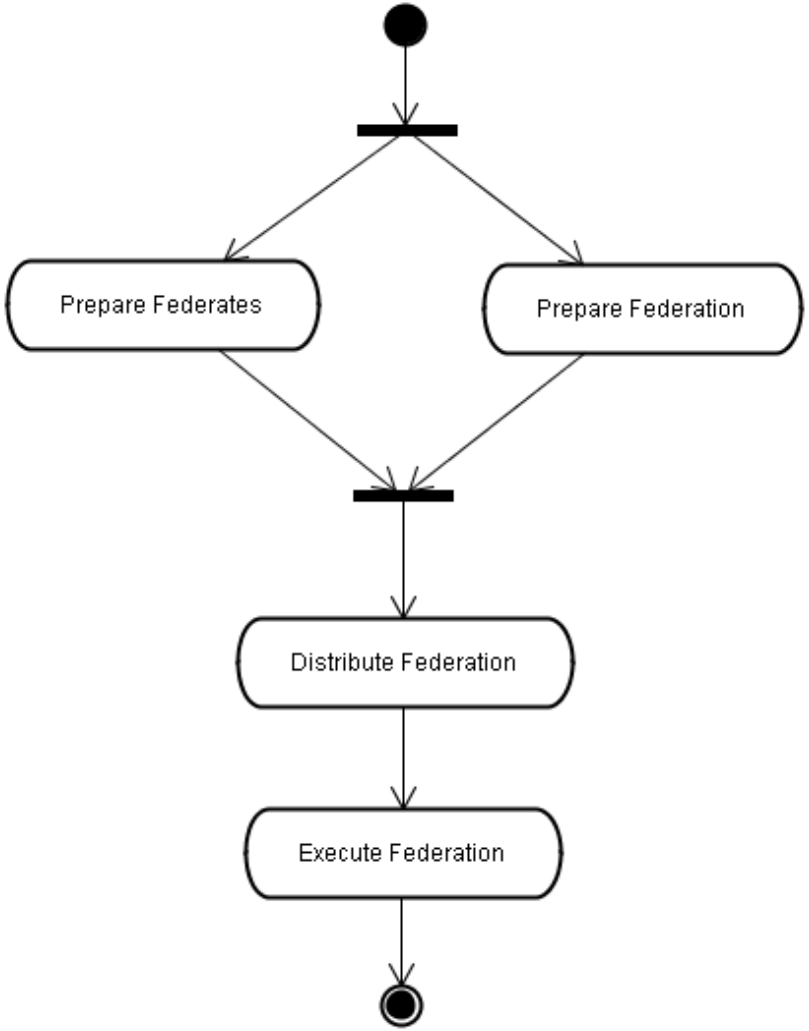


Диаграмма деятельности

Поведение – примеры – 4/4

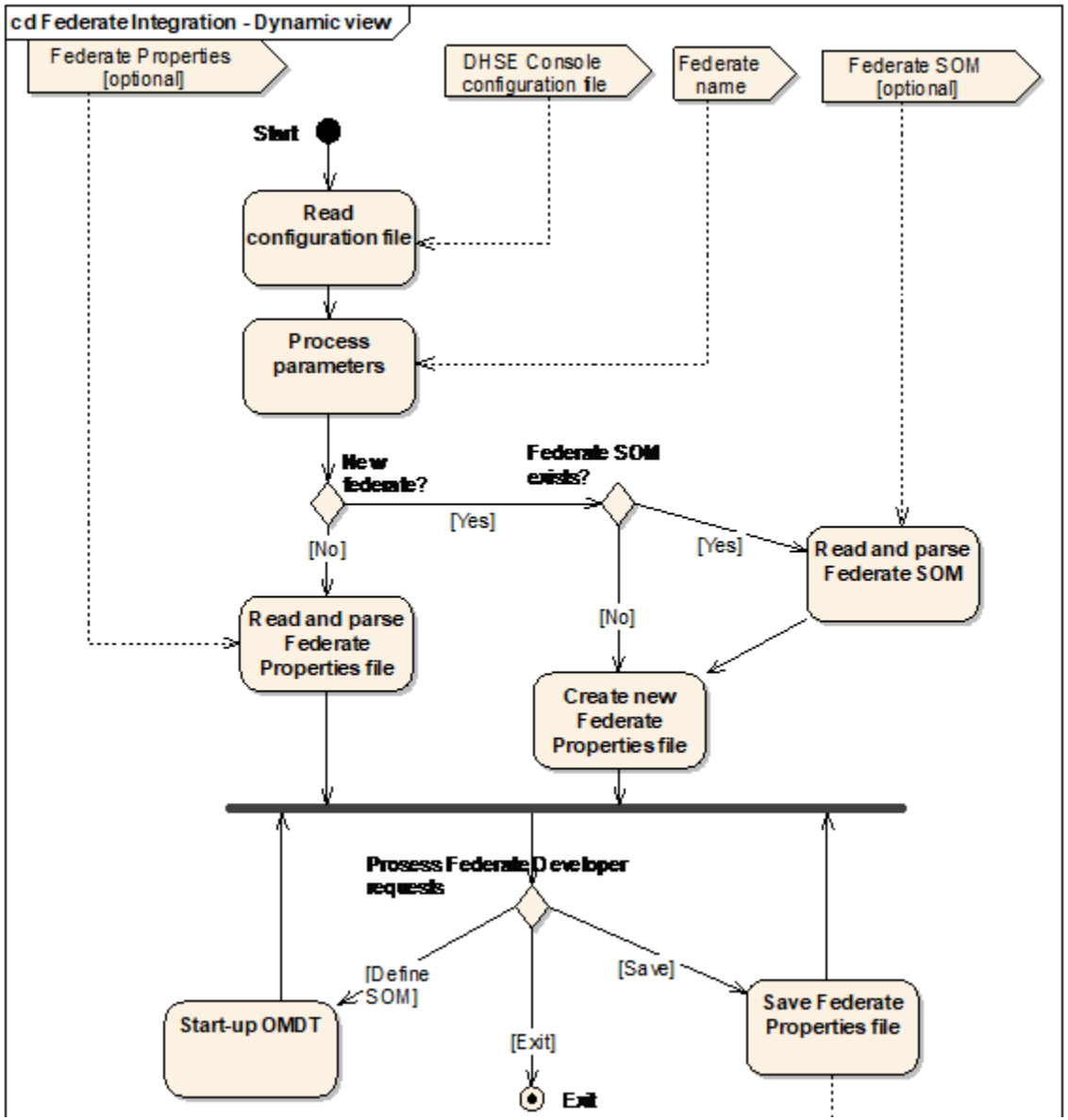


Диаграмма коммуникаций

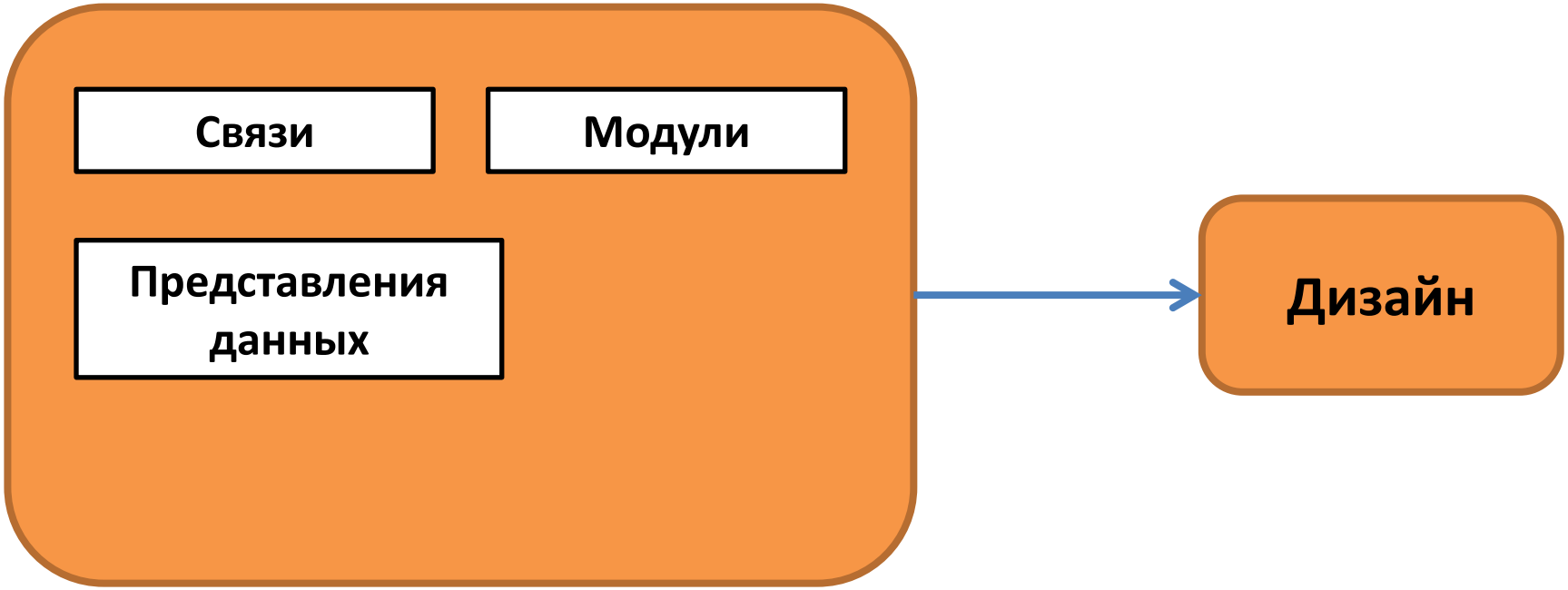
ЗНАЧИМЫЕ ЭЛЕМЕНТЫ СИСТЕМЫ

Значимые элементы – 1/3

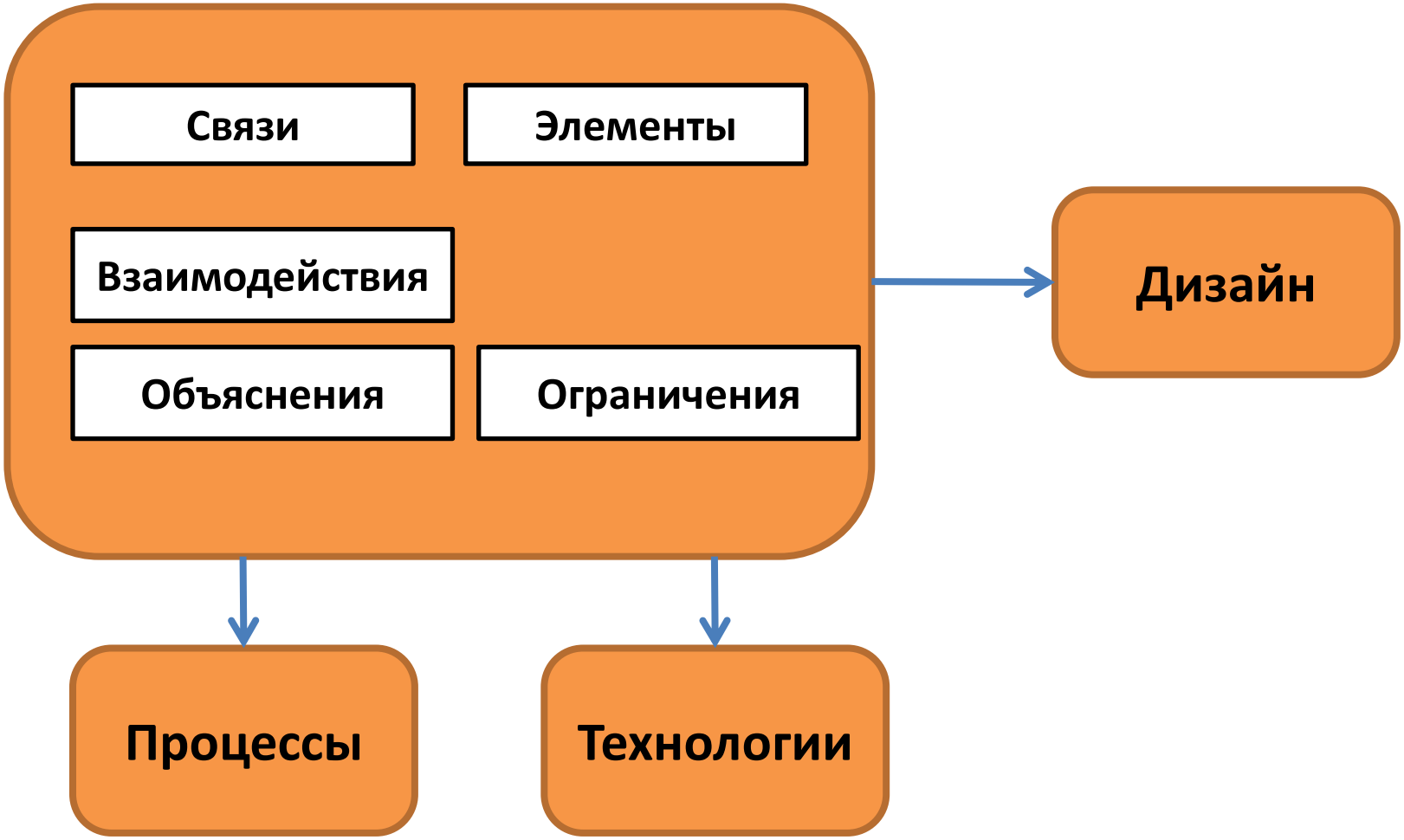
- Значимые элементы имеют продолжительное и устойчивое действие:
  - главные структурные элементы
  - связанные с основным поведением
  - определяющие значимые свойства (например, надежность и масштабируемость
- Обобщенно, значимость – это стоимость создания и стоимость изменения элемента в смысле затрат ресурсов (не только $)

Что входит в архитектуру – 1/4

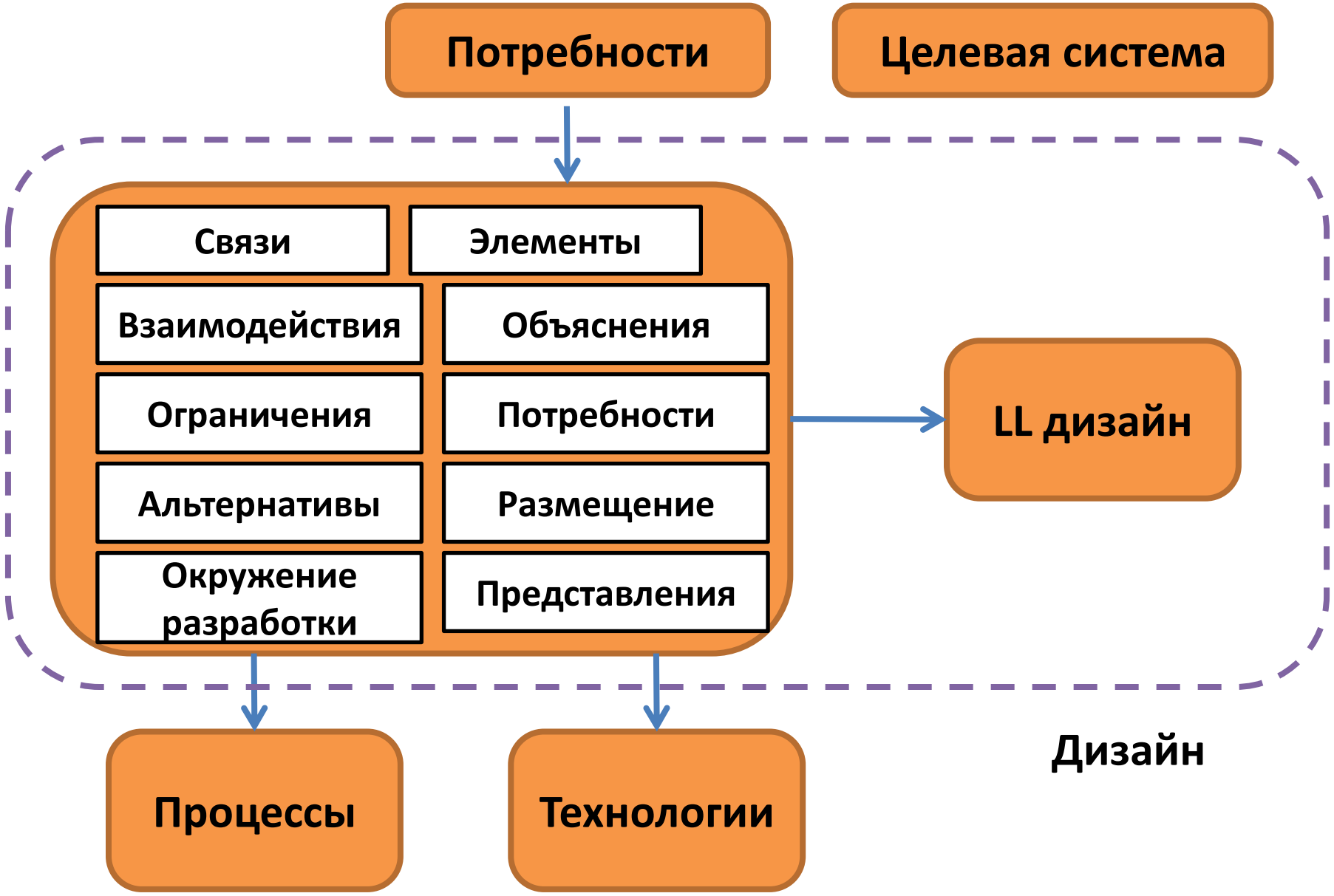
До 1990-х



Что входит в архитектуру – 2/4



Что входит в архитектуру – 3/4



Что входит в архитектуру – 4/4

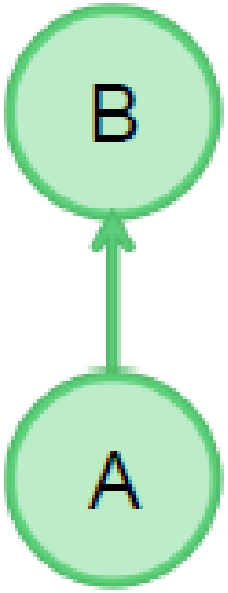
- Элементы, модули, связи, структуры, виды, формы, каркасы, разделения, ограничения, интересанты, потребности, окружение, размещение, управление, интерфейсы, слои, артефакты, технологии, статика и динамика, поведение, свойства, отношения, принципы, стили…
- → То есть самые разнообразные **проектные решения**

Значимые элементы – 2/3

- Архитектура определяет несколько верхних уровней иерархического разбиения. То есть, разработчик архитектуры определяет конкретную перспективу системы, значимую для ~~него~~ заинтересованных сторон, и управляет сложностью описания
- Набор значимых элементов не является статичным и может измениться при:
  - Уточнении требований
  - Идентификации рисков
  - Реализации системы

Зависимость проектных решений – 1/7

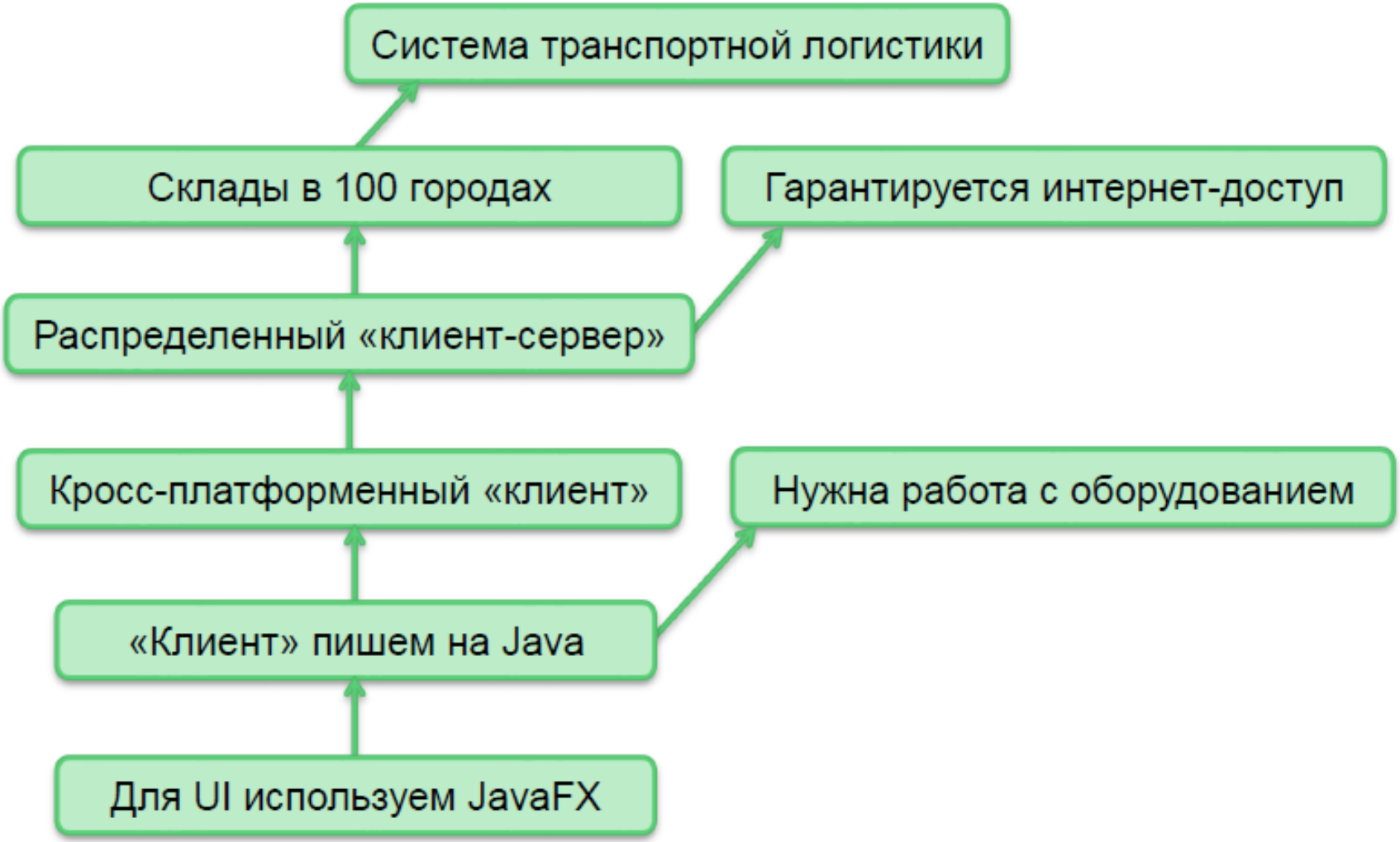
- Решение А зависит от решения В, если А имеет смысл или целесообразно только в том случае, когда принято и актуально решение В



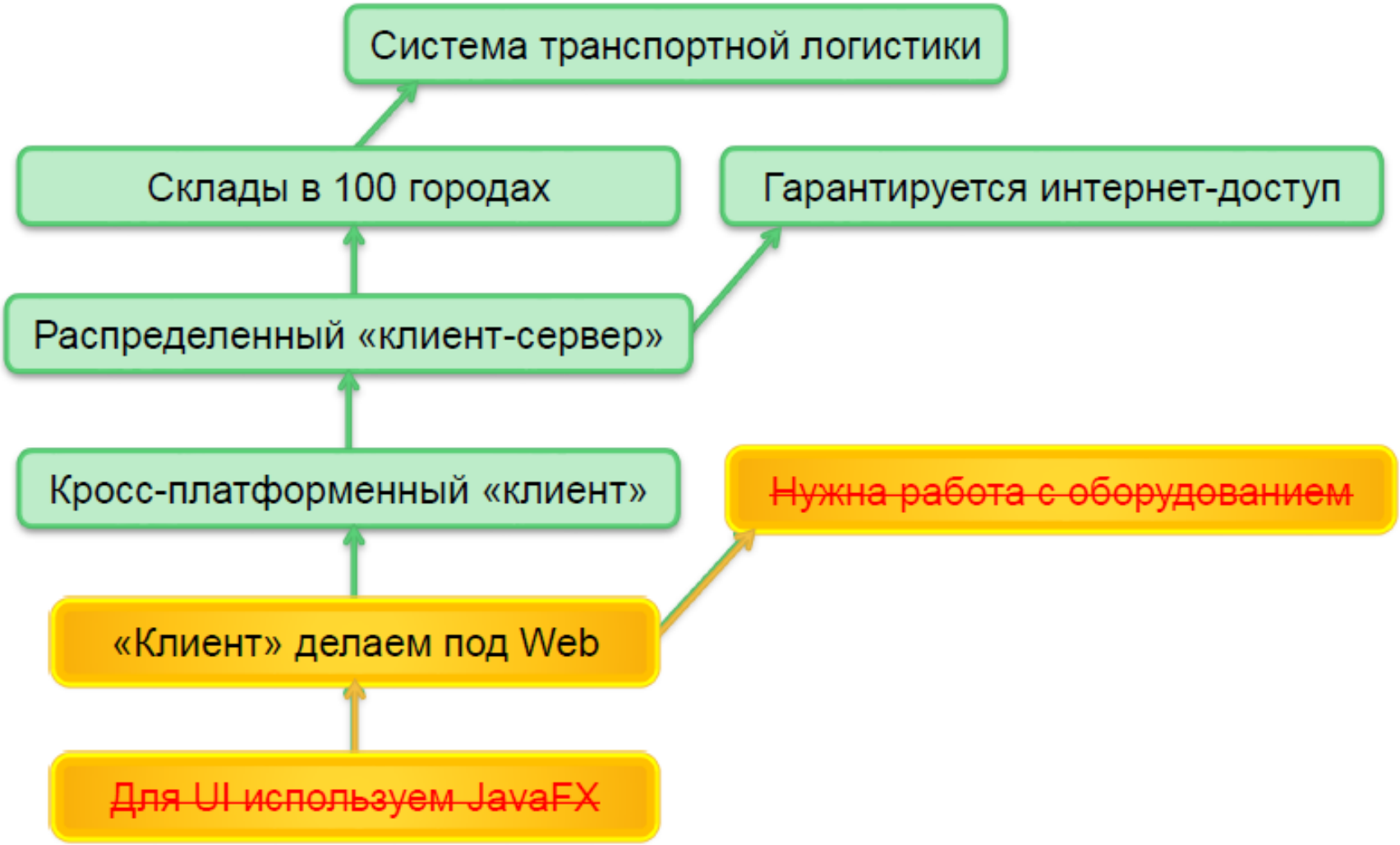
Зависимость проектных решений – 2/7

- 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
- Фундаментальные решения (содержание архитектуры) – решения, от которых зависит слишком много и изменение которых обойдется слишком дорого

Зависимость проектных решений – 3/7

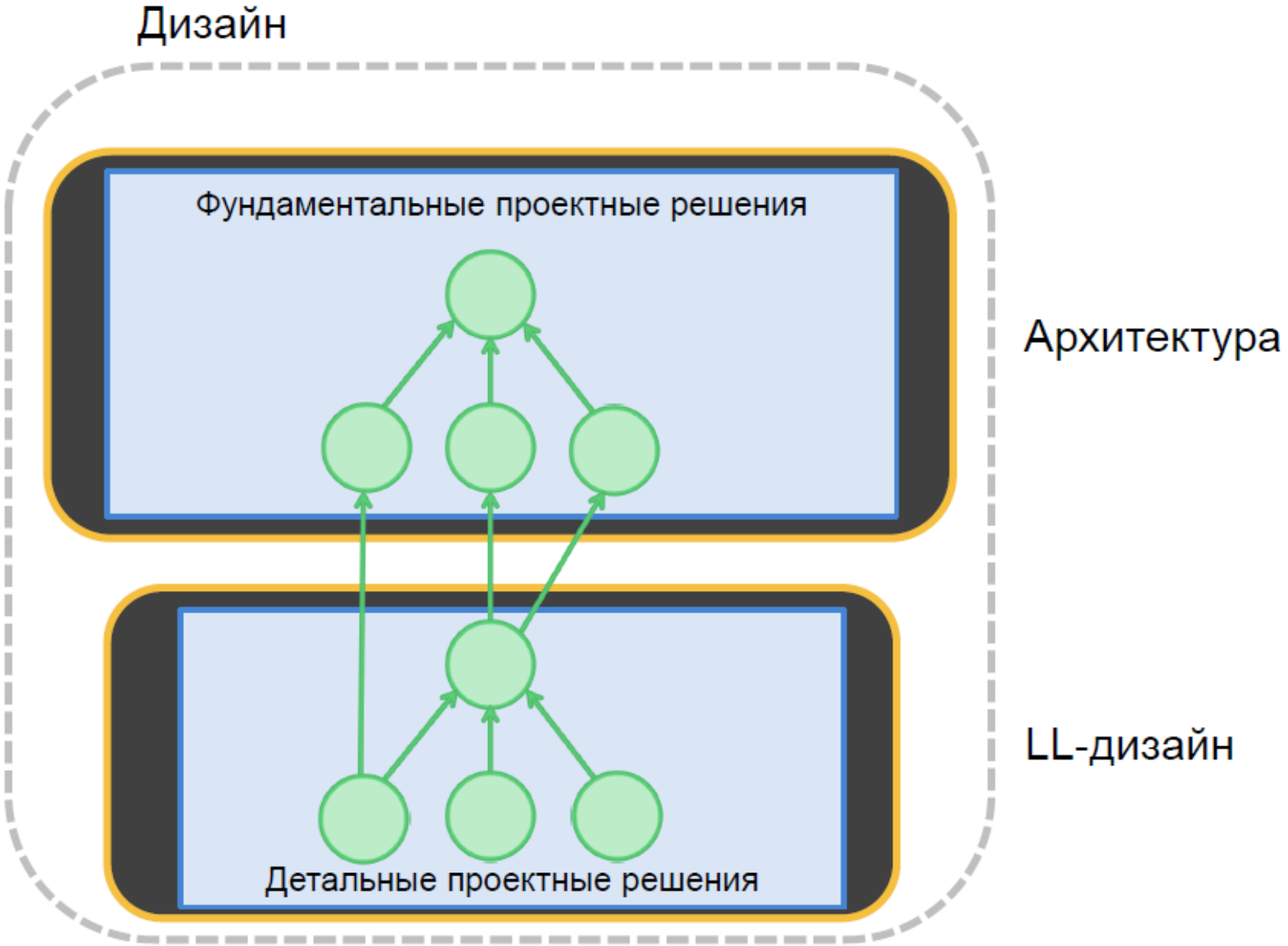


Зависимость проектных решений – 4/7

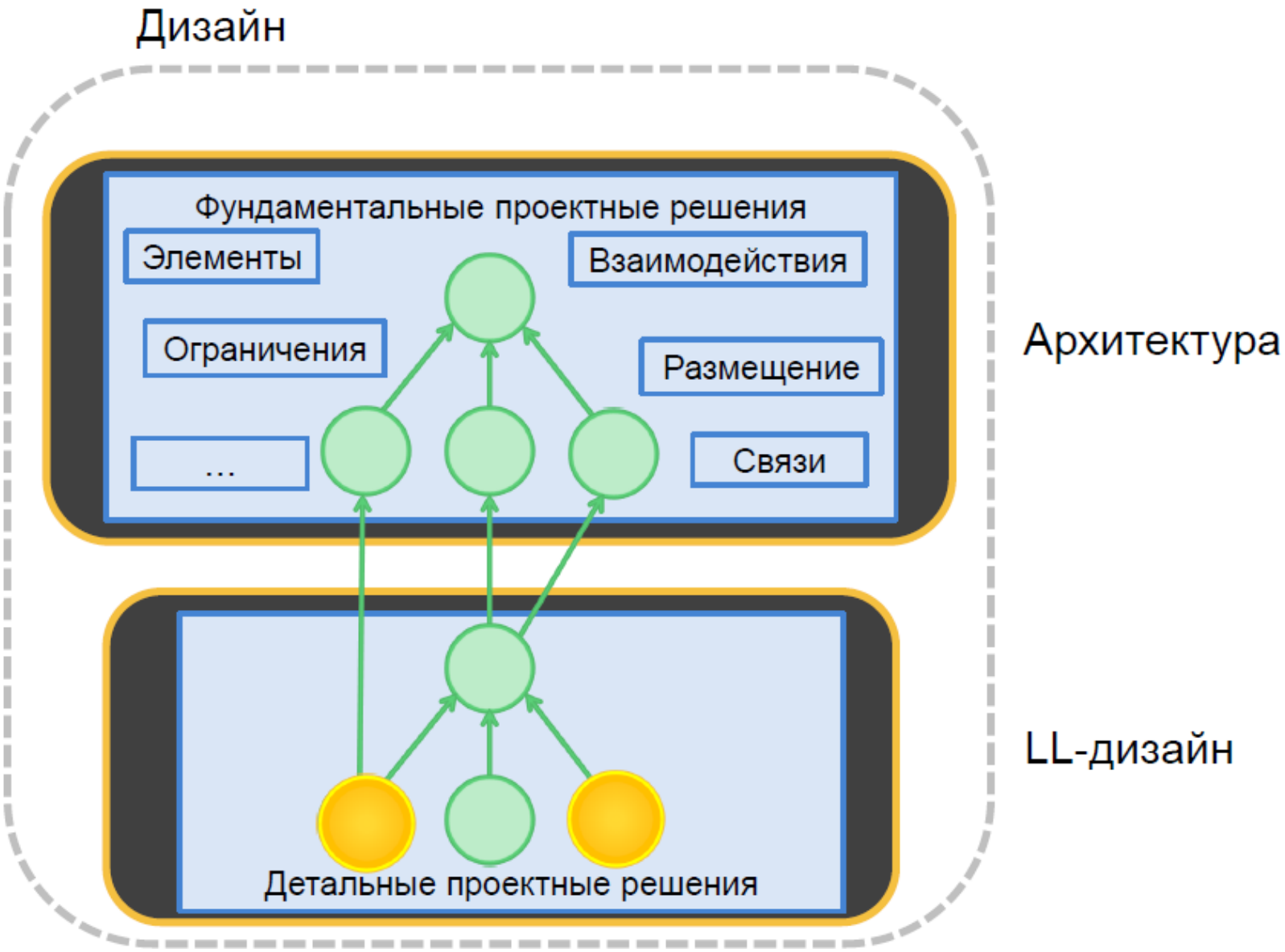


Зависимость проектных решений – 5/7

Зависимость проектных решений – 6/7



Зависимость проектных решений – 7/7



Значимые элементы – 3/3

- Относительная стабильность архитектуры несмотря на изменения, является признаком хорошей архитектуры, хорошо отлаженного процесса разработки и хорошего разработчика
- Если архитектура требует постоянного пересмотра при относительно небольших изменениях, это плохой признак

Баланс интересов

Обоснование решений

ЦЕЛИ СОЗДАНИЯ АРХИТЕКТУРНОГО ОПИСАНИЯ

ЗАИНТЕРЕСОВАННЫЕ СТОРОНЫ

Согласование запросов СХ – 1/4

- Архитектура создается для удовлетворения потребностей СХ, их больше одного. Часто все пожелания нельзя выполнить одновременно. Поэтому принимаются компромиссные решения
- Примерный список СХ с требованиями:
  - Конечный пользователь
    - Интуитивно понятное и корректное поведение
    - Производительность
    - Надежность
    - Удобство использования
    - Доступность
    - Безопасность

Согласование запросов СХ – 2/4

- Примерный список СХ (продолжения):
  - Системный администратор
    - Интуитивно понятное поведение
    - Управляемость, наличие инструментов мониторинга
  - Специалист по маркетингу
    - заинтересован в конкурентноспособных функциях, времени до выхода программы, позиционировании среди других продуктов и в стоимости;
  - Клиент:
    - Цена
    - Стабильность
    - Возможность планировать [использование]

Согласование запросов СХ – 3/4

- Примерный список СХ (продолжения):
  - Разработчик
    - Понятные требования
    - Простой и непротиворечивый принцип проектирования
    - Интеграция со средствами разработки
  - Руководитель проекта
    - Предсказуемость хода проектирования
    - Планирование
    - Продуктивное использование ресурсов и бюджета

Согласование запросов СХ – 4/4

- Примерный список СХ (продолжения):
  - Специалист по сопровождению
    - Сквозное документирование
    - Легкость внесения изменений
- Не только функциональные требования! Нефункциональные требования формируют свойства или ограничения системы. Нефункциональные требования очень часто являются самыми значимыми требованиями, поскольку в них заинтересован разработчик архитектуры

ОБОСНОВАНИЕ РЕШЕНИЙ

Обоснование решений – 1/2

- Обоснование выбора основной архитектуры и всех принятых решений более низкого уровня помогает в согласовании интересов СХ. Кроме этого, обоснование решений важно для:
  - Воплощения системы
    - Разработчик понимает причины выбора архитектуры и операционное окружение системы/компонента
  - Тиражирования выбранных решений
    - Обоснование описывает область применимости решения

Обоснование решений – 2/2

- Обоснование важно для (продолжение):
  - Обслуживания системы
    - Обоснование помогает понять предполагаемый способ использования системы
  - Развития системы
    - Анализ актуальности обоснований позволяет сконцентрироваться на тех элементах архитектуры воплощенной системы, которые надо (можно) поменять

Готовые решения vs. Специфика

Влияние операционного окружения

ПРОЦЕСС ПРОЕКТИРОВАНИЯ

ВЫБОР ОБЩЕЙ АРХИТЕКТУРЫ

Готовые решения – 1/3

- Подобные задачи порождают подобные решения. Архитектуры, построенные для СХ со сходными наборами интересов, похожи. Если группа схожих архитектур велика и имеет специфику (структур, поведения, наложенных ограничений), то она может быть определена как **архитектурный стиль**
  - Шоу и Гарлан: Семейство систем в терминах шаблона организации структуры. Точнее, архитектурный стиль определяет номенклатуру компонентов и типов соединительных звеньев, а также набор условий, в соответствии с которыми они могут соединяться

Готовые решения – 2/3

- Более низкоуровневый вид стиля называется архитектурным шаблоном
  - UML: Шаблон – это общее решение общей проблемы в данном контексте
- Использование архитектурных стилей и шаблонов облегчает разработку, так как для стиля есть:
  - Обоснования его использования → можно меньше обдумывать (!)
  - Описание его структуры и поведения → можно меньше документировать, просто сославшись на стиль / шаблон

Готовые решения – 3/3

- Конкретная система [в конкретных условиях] может наследовать более одного архитектурного стиля и/или накладывать больше ограничений на поведение / структуру в сравнении с «чистым» стилем

ВЛИЯНИЕ ОПЕРАЦИОННОГО ОКРУЖЕНИЯ

Операционное окружение – 1/3

- Система функционирует в некотором окружении, называемом операционным (operational environment), которое влияет на архитектуру:
  - Определяются границы, в которых должна работать система, включая внешние интефейсы системы
  - Миссия бизнеса (mission – глобальное назначение и действия системы в целом) может быть определена только в операционном окружении

Операционное окружение – 2/3

- СХ часто являются артефактами операционного окружения
  - Например, пользователи или обеспечивающие системы
- Внешний мир проявляется как стандарты или принятые процедуры
- Внешние интерфейсы навязывают конкретные технические решения

Операционное окружение – 3/3

- Также наоборот, архитектура влияет на свое окружение. Например:
  - Появление новых сервисов (очевидно!)
  - Добавление навыков и процедур, доступных в пределах организации
- В случае программных систем, следует учитывать, что аппаратное обеспечение для ПО является ОО

Организационно-штатная структура

Есть в каждой системе

ВЛИЯНИЕ АРХИТЕКТУРЫ

АРХИТЕКТУРА И СТРУКТУРА КОЛЛЕКТИВОВ

Люди как часть системы

- Стандарты IEEE 12207-1995 (Software Life Cycle Processes) и «A configuration of the Rational Unified Process® for Systems Engineering» (RUP SE) добавляют в описание системы понятие ресурсов (информационных и человеческих). Таким образом, в систему включается обслуживающий персонал и необходимо учитывать влияние архитектуры на организационно-штатную структуру предприятия

Орг-штатная структура – 1/2

- [Очевидно]: Автоматизация функций уменьшает число [неквалифицированных] рабочих мест
- [Менее очевидно]: Архитектура определяет функциональные модули. Например:
  - Система обработки заказов может включать подразделения:
    - ввода заказов, ведения счетов, работы с клиентами, выполнения заказа, интеграции с внешними системами, персистентности и безопасности
  - Разбиение, отличное от существующего бизнес-процесса меняет бизнес-процесс и штат

Орг-штатная структура – 2/2

- [Совсем не очевидно]: Архитектура системы отражает структуру коллектива разработчиков:
  - Такое разбиение может быть неэффективно
  - С другой стороны, если назначать реализацию конкретных компонентов (функций) системы, группам разработчик, согласуясь с их навыками, то результат будет хорош

АРХИТЕКТУРА ЕСТЬ ВЕЗДЕ

Наличие архитектуры

- Каждая система имеет архитектуру, даже если эта архитектура формально не документирована или система слишком проста (например, состоит из одного элемента)
- Документирование очень ценно:
  - Документированные архитектуры более продуманы → эффективными
  - Облегчается доказательство соответствия требованиям (в том числе по удобству обслуживания, заимствованию передового опыта и так далее)