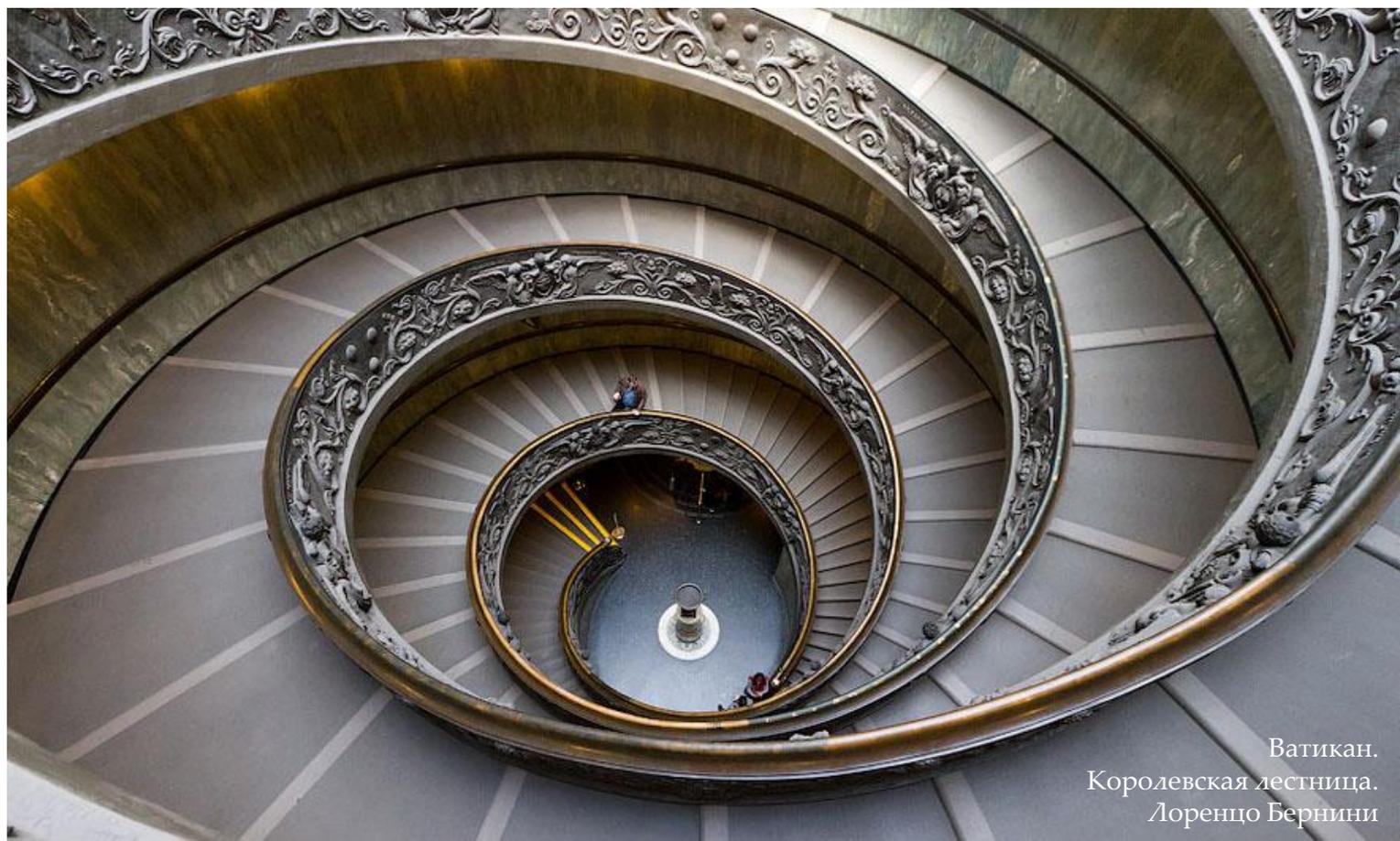




## 4 ВЗГЛЯДА НА СОВРЕМЕННУЮ ИТ-АРХИТЕКТУРУ: КАРТЫ TM-FORUM (eTOM, TAM, SID) + NGOSS ЧЕРЕЗ ПРИЗМУ TOGAF/ARCHIMATE



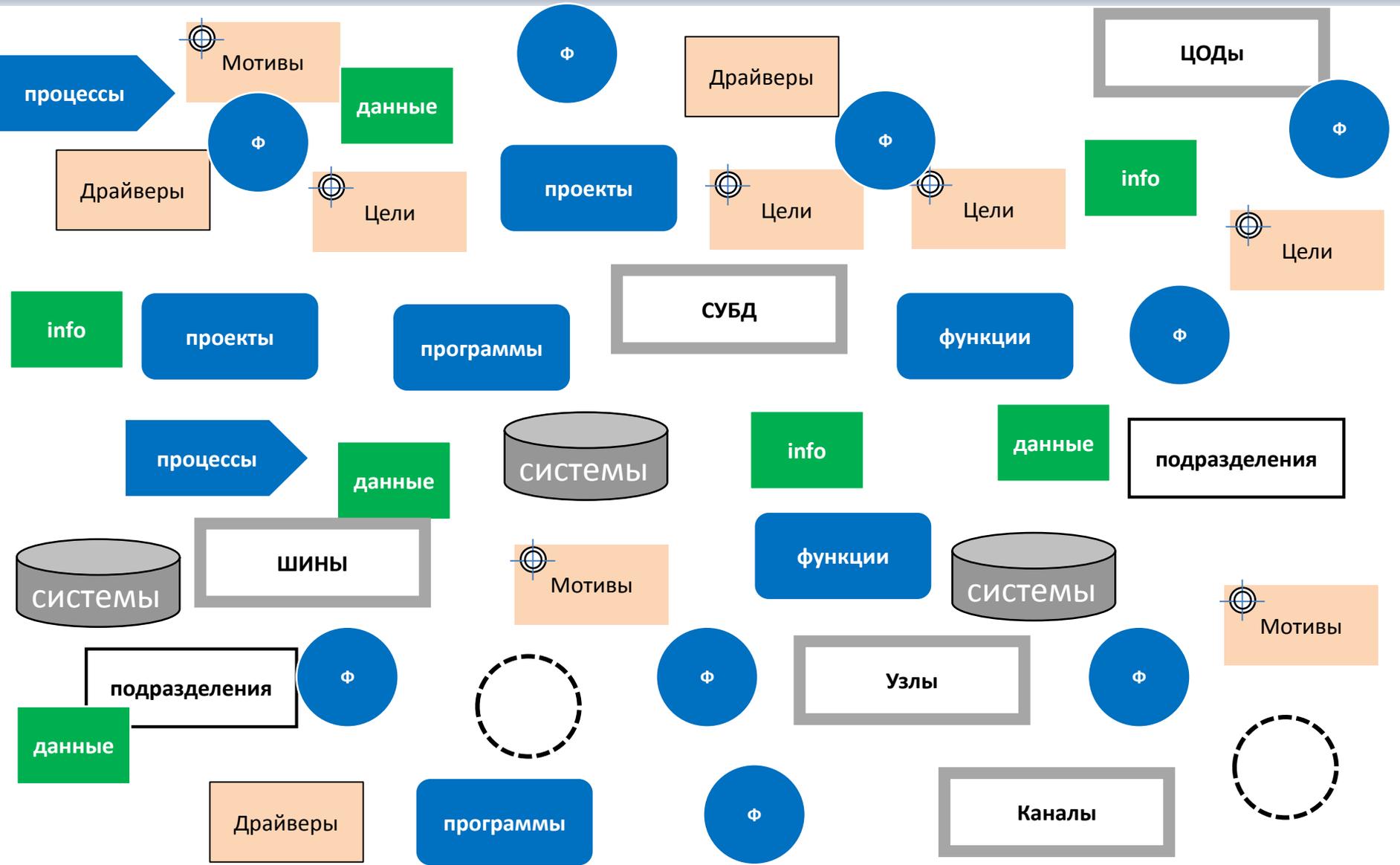


## РАЗДЕЛ 1. ВВЕДЕНИЕ В АРХИТЕКТУРУ

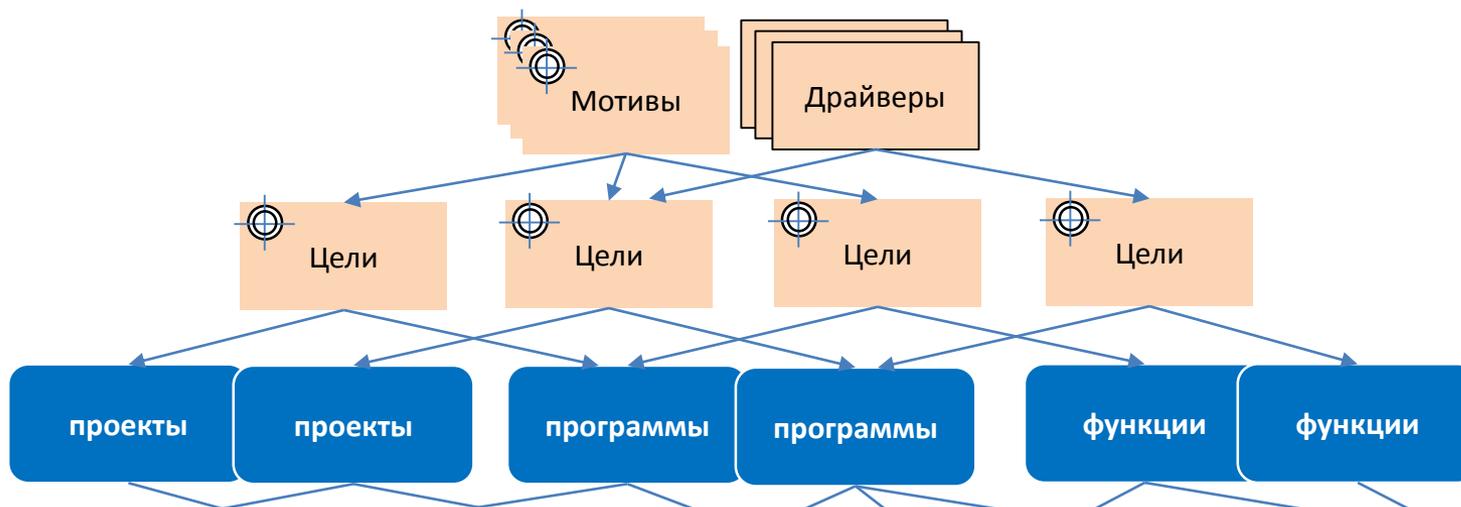
---



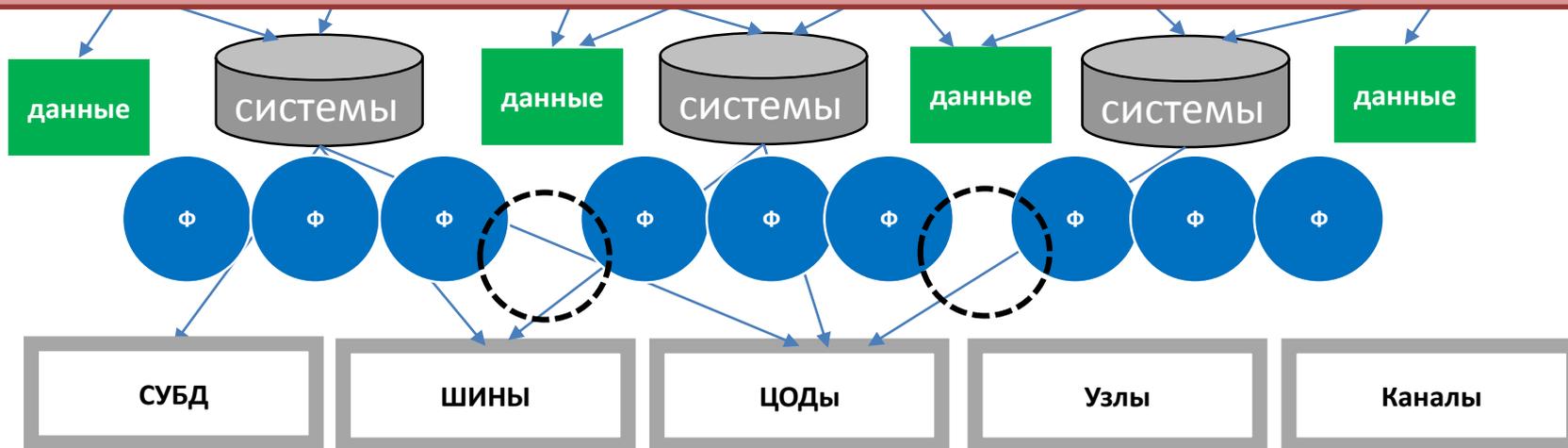
MARCUS AURELIUS LTD



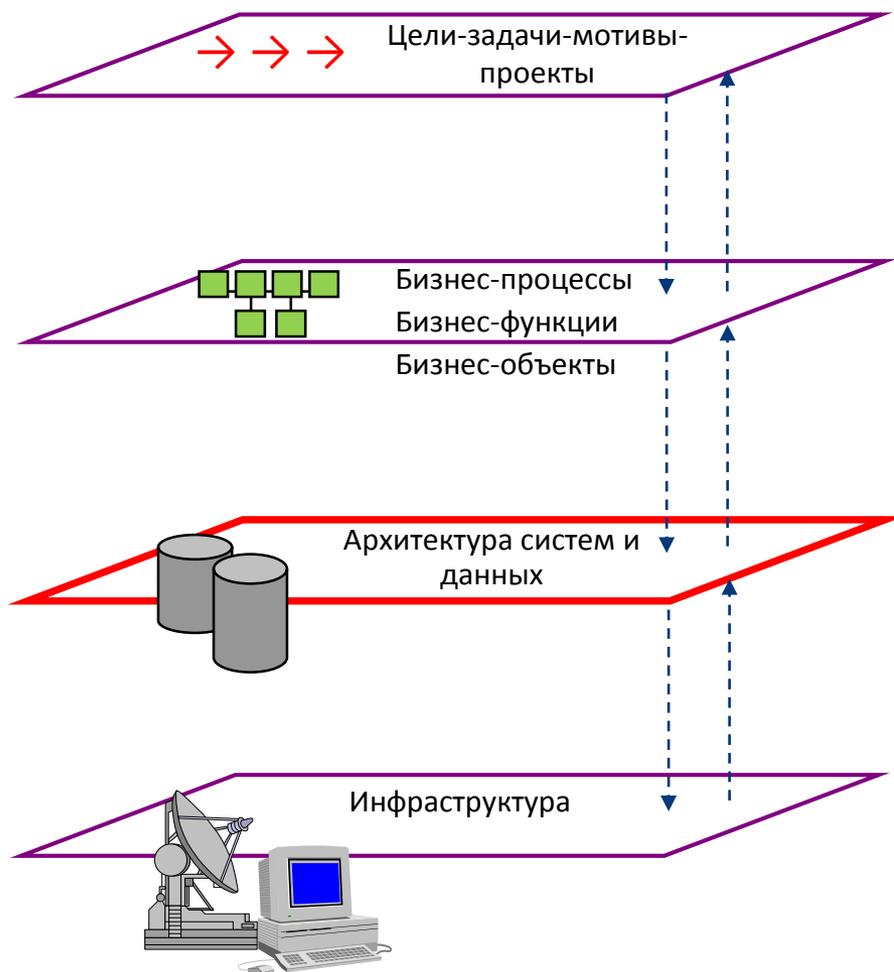
# СИСТЕМНОСТЬ - ЭТО СТРУКТУРНЫЙ ПОРЯДОК



Понимание компонентов системы и их связей лежит в основе управления системой



# АРХИТЕКТУРА: МНОГОУРОВНЕВЫЙ СТРУКТУРНЫЙ ПОРЯДОК В ЭЛЕМЕНТАХ И СВЯЗЯХ МЕЖДУ НИМИ



## Motivation & Delivery:

- Структурирует цели, мотивы и факторы влияния
- Расширение: проекты, программы, плато-гар

## Business layer:

- Структура процессов, функций, подразделений.
- Структурирует бизнес-объекты
- Структура создаваемого продукта

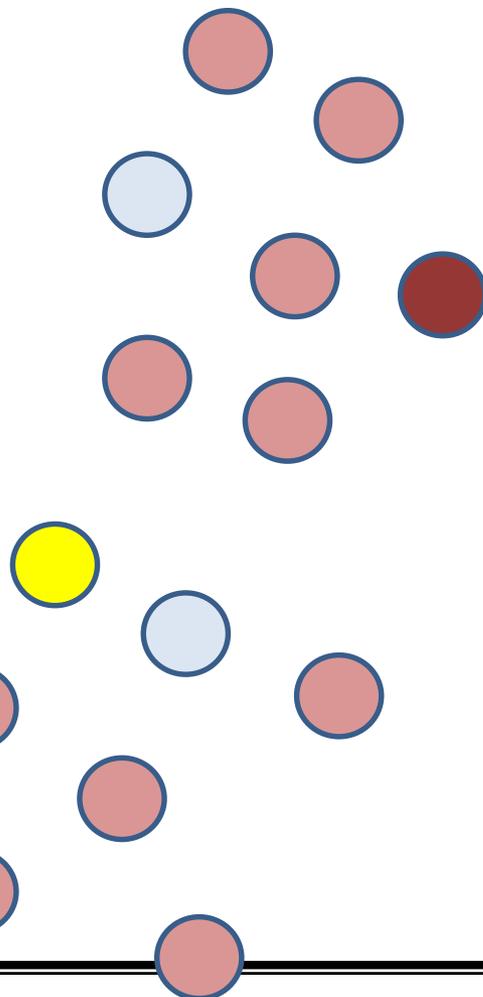
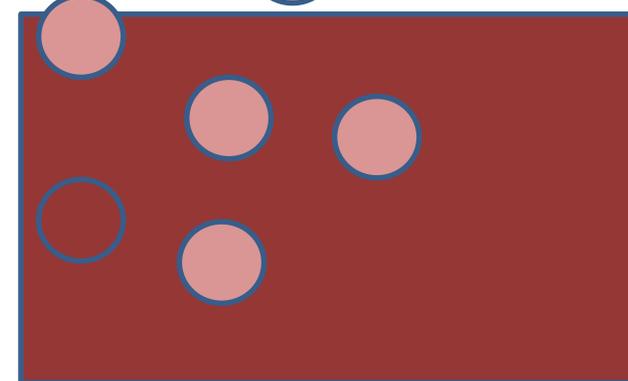
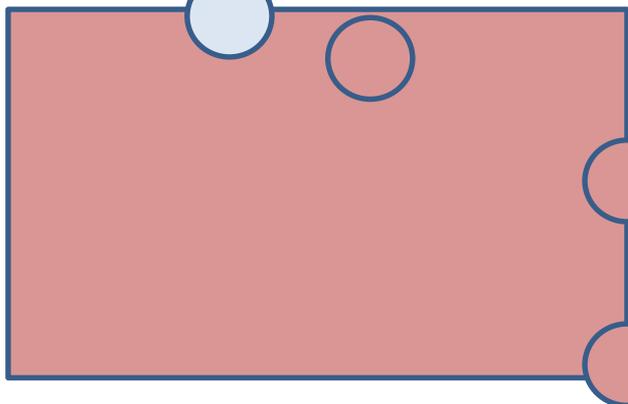
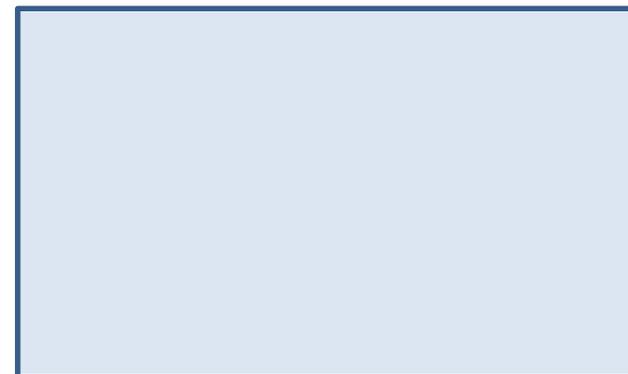
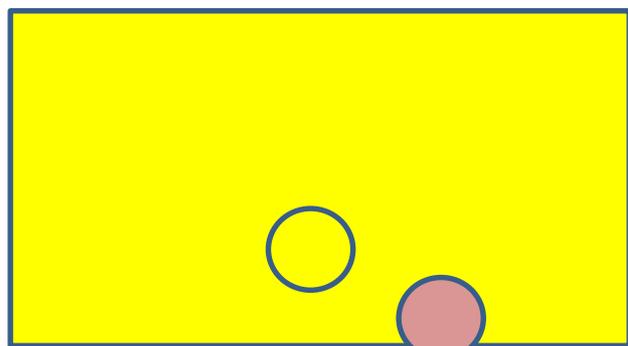
## Application layer: обработка данных

- Структурирует данные
- Структурирует поведение систем
- Структура интеграций

## Infrastructure:

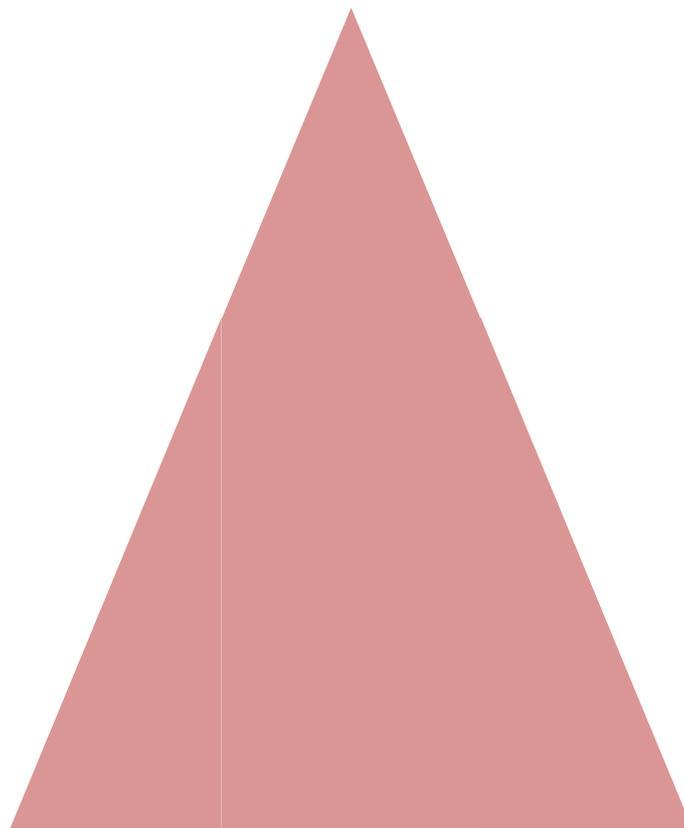
- Структурирует сети.
- Структурирует центры и узлы обработки
- Структурирует системное программное обеспечение

# ТИПОВОЙ ПОДХОД №1: СОБРАТЬ ИЗ ЧАСТЕЙ ЦЕЛОЙ



# АРХИТЕКТУРНЫЙ ПОДХОД №2:

## РАЗДЕЛИТЬ ЦЕЛОЕ НА ЧАСТИ



Типовой подход: от общего к частному, или сверху вниз.

# TOGAF: ИСТОЧНИК ЗНАНИЙ ПО АРХИТЕКТУРЕ

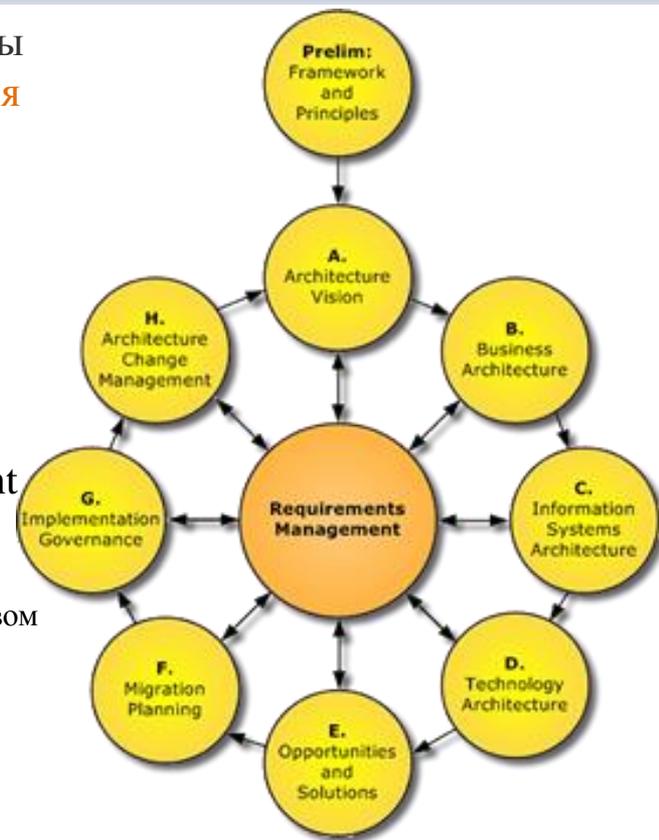
**TOGAF®** - стандарт Open Group – зрелая методология архитектуры предприятия, используемая многими предприятиями-лидерами для улучшения эффективности бизнеса.

**TOGAF** (The Open Group Architecture Framework) - это архитектурный фреймворк, представляющий инструментарий для содействия в принятии, производстве, использовании и обслуживании корпоративных архитектур.

Ключевым компонентом TOGAF является Architecture Development Method – ADM (**TOGAF 9 Part II: ADM**).

**ADM** описывает процесс создания архитектуры предприятия. ADM является руководством для архитекторов на нескольких уровнях:

- ADM предусматривает циклический ряд этапов разработки архитектуры: бизнес-архитектуры, архитектуры информационных систем, технологической архитектуры.
- ADM предоставляет описание каждой архитектурной фазы в виде целей, подхода, входов, шагов и выходов. Входы и выходы предоставляют собой определенные архитектурные артефакты.
- ADM предоставляет механизм корреляции и контроля всех фаз через центральный процесс ADM - управление требованиями.





TOGAF® - (The Open Group Architecture Framework) - стандарт Open Group, предлагаемый в качестве архитектурной методологии для предприятий для целей улучшения эффективности бизнеса.

**Архитектура** – фундаментальная организация системы (например, бизнес-системы), встроенная или присущая ее компонентам, отношение компонентов друг к другу и к внешней среде, а также принципы управления дизайном и эволюцией этой организованности.

**Архитектура** – формальная организация системы или детальный план системы.

**Архитектура** – структура компонентов, их отношения друг к другу, а также принципы методология управления их дизайном и эволюцией во времени.

**Архитектура** – организационная логика, лежащая в основе бизнес-процессов и ИТ-инфраструктуры компании, отражающая требования интеграции и стандартизации, вытекающие из операционной модели компании.

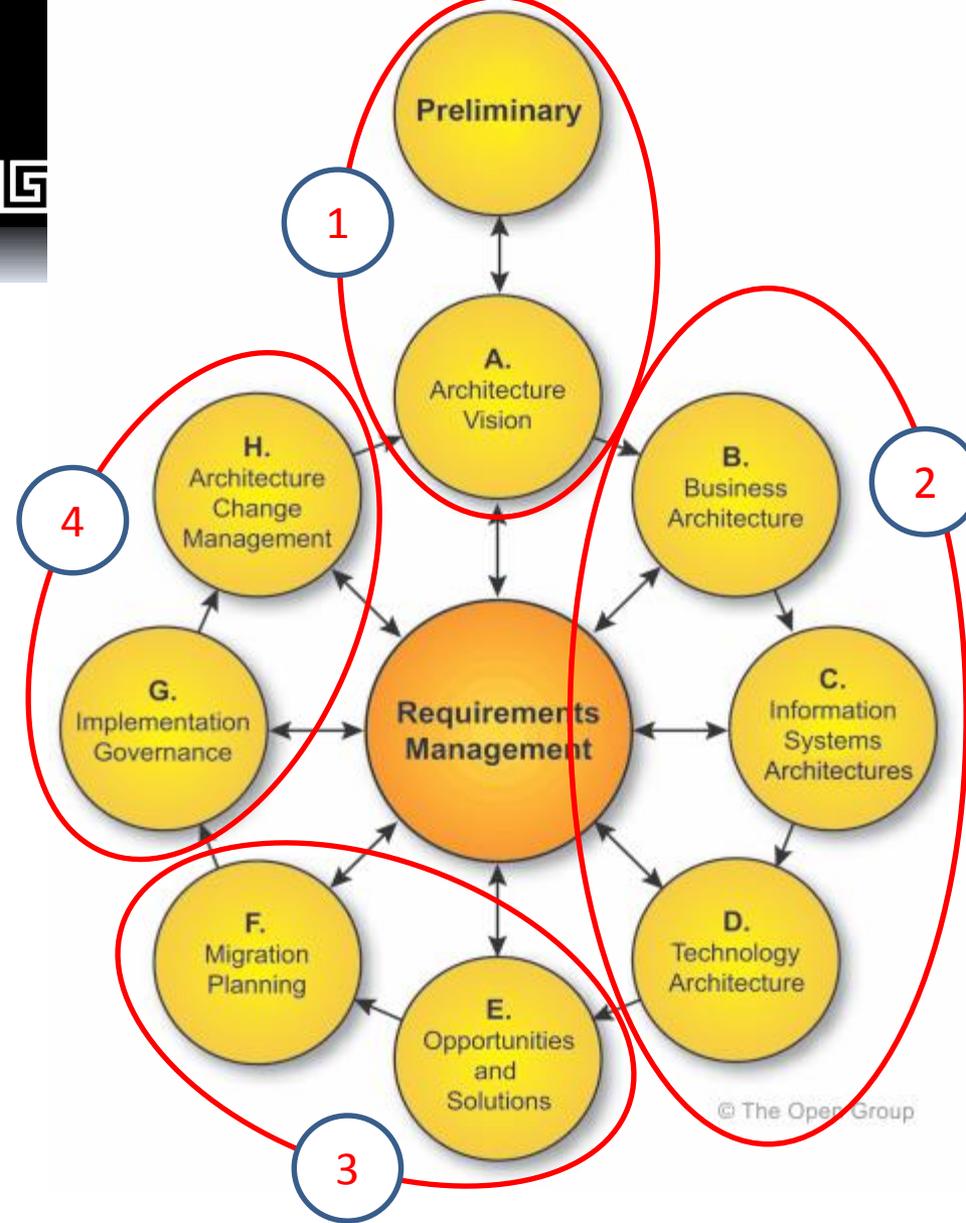
# ARCHITECTURE DEVELOPMENT METHOD (ADM)

1. КОНТЕКСТ АРХИТЕКТУРЫ

2. ДИЗАЙН АРХИТЕКТУРЫ

3. ПОИСК РЕШЕНИЙ И ПЛАНИРОВАНИЕ

4. ВНЕДРЕНИЕ

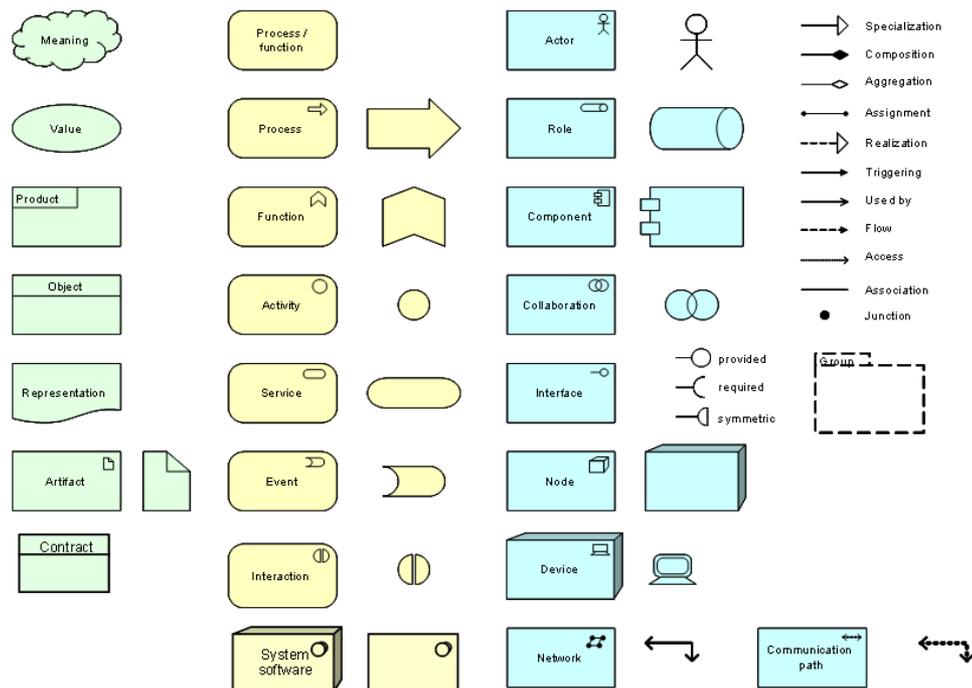


# ARCHIMATE 3.0

Спецификация ArchiMate (уже 3.0 !) - это стандарт, разработанный консорциумом The Open Group в качестве языка архитектурного моделирования.

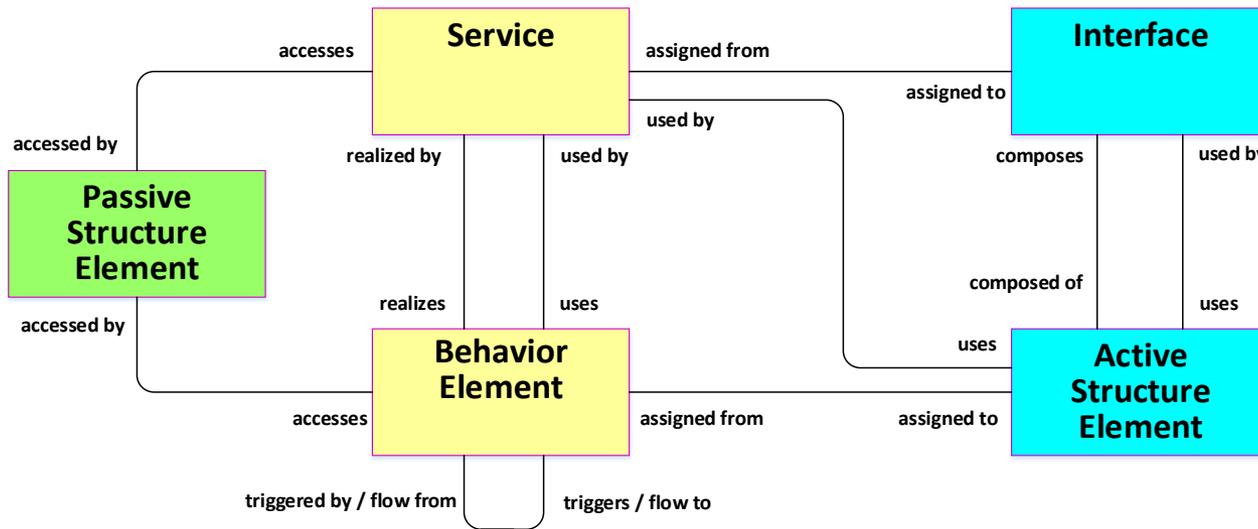
Спецификация содержит формальное определение всех строительных блоков архитектуры, их графические способы отображения, спецификацию связей между элементами, а также набор рекомендуемых viewpoints для типовых заинтересованных лиц.

На сегодня Archimate поддерживается большинством поставщиков ПО для архитектурного описания предприятий.

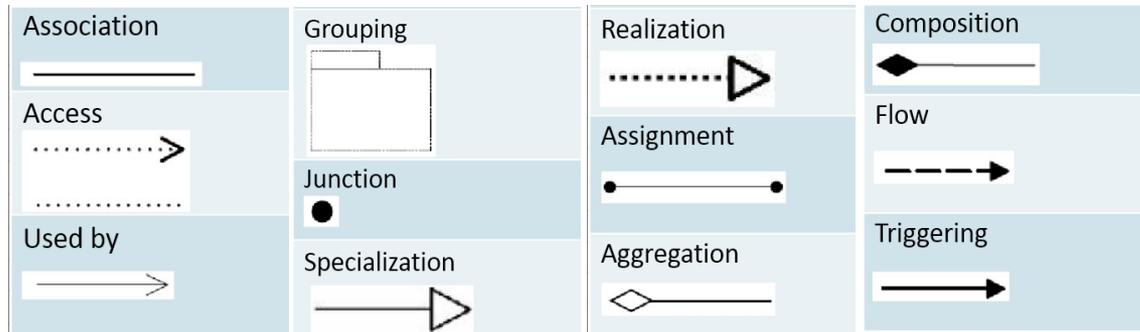


ARCHIMATE – ЭТО ЯЗЫК, НА КОТОРОМ РЕШАЮТСЯ ЗАДАЧИ УРОВНЯ ПРЕДПРИЯТИЯ

# АРХИТЕКТУРА: БАЗОВЫЙ ПАТТЕРН И ВИДЫ СВЯЗЕЙ

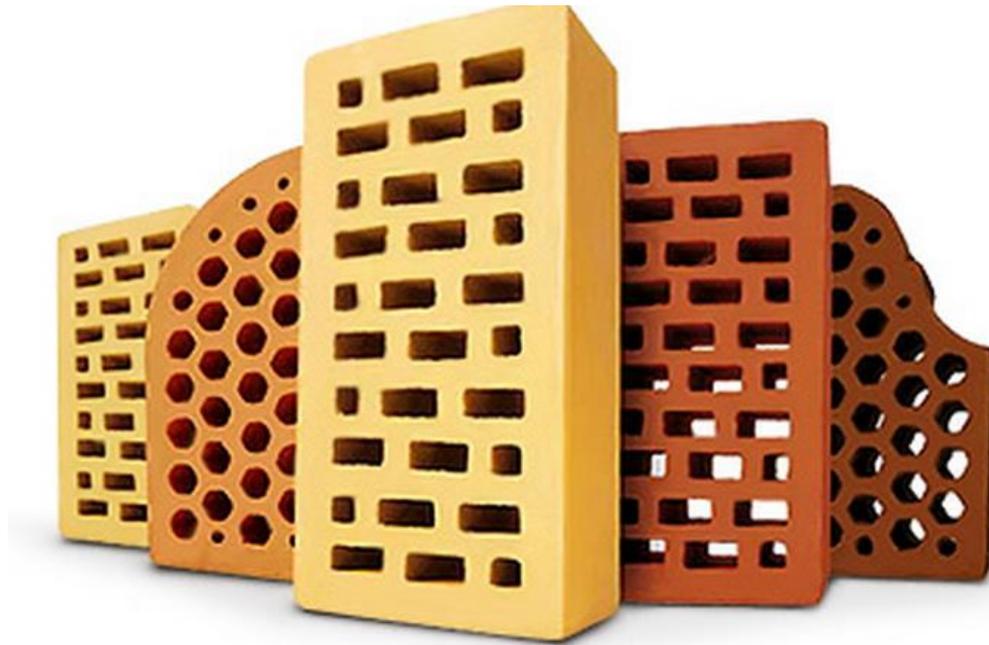


МОДЕЛИРОВАНИЕ СВЯЗЕЙ И ОТНОШЕНИЙ ТАКЖЕ ВАЖНО, КАК И МОДЕЛИРОВАНИЕ САМОЙ СТРУКТУРЫ





# КЛЮЧЕВОЙ МОМЕНТ ПОСТРОЕНИЯ АРХИТЕКТУРЫ – ЭТО ВЫБОР СТРОИТЕЛЬНЫХ БЛОКОВ (КИРПИЧИКОВ) – АВВ\*

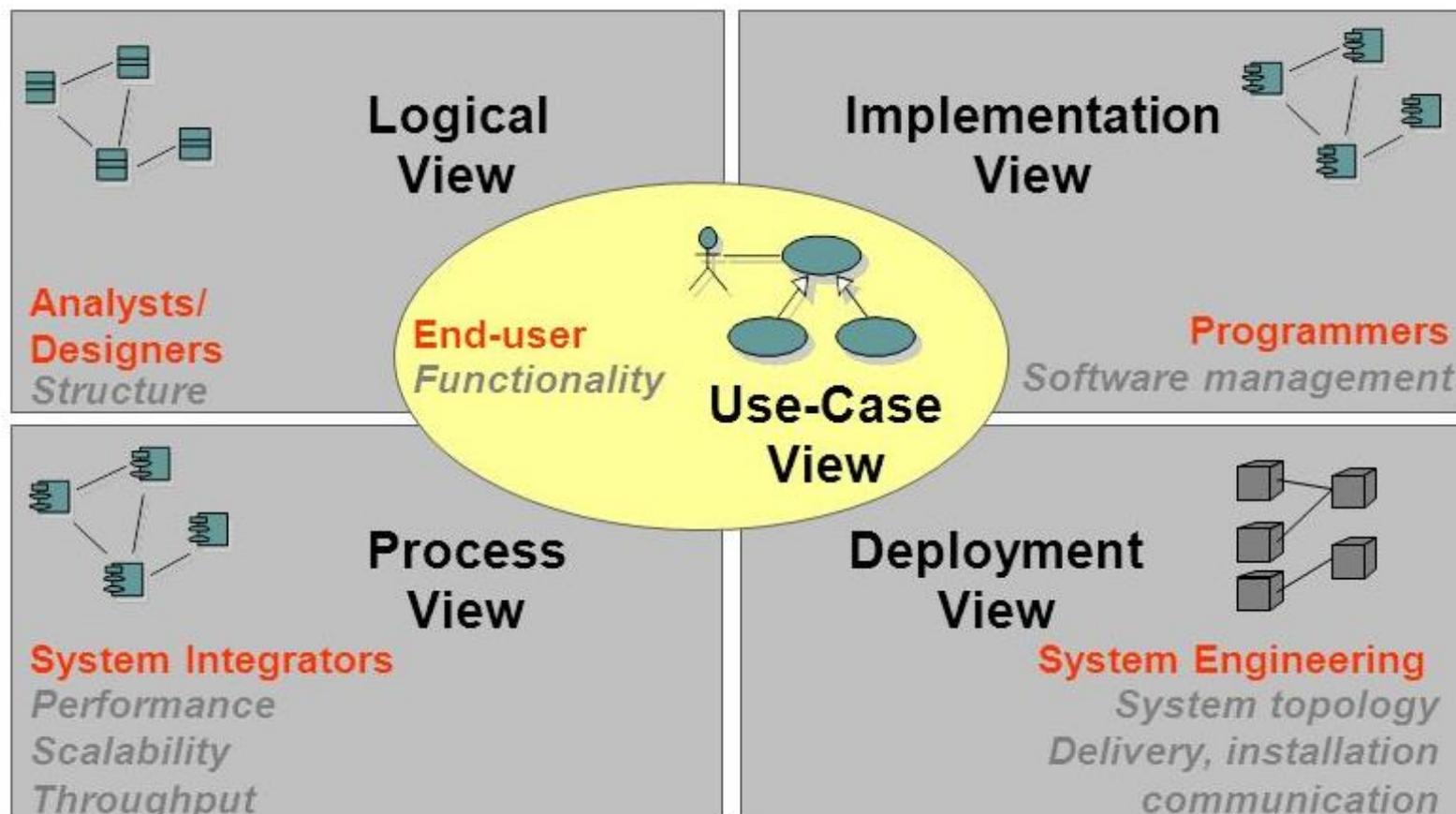


Чем больше диапазон АВВ\* (кирпичиков), тем больше нюансов может учесть ваша архитектурная модель.

Выбор ограниченного набора кирпичиков и связей между ними является далеко не самой простой задачей архитектурных проектов и определяет МЕТАМОДЕЛЬ вашей архитектуры. Наиболее сложное в метамоделе: добиться от участников проекта одинакового понимания architecture building blocks\*.



# ПРЕДСТАВЛЕНИЕ АРХИТЕКТУРЫ: МОДЕЛЬ 4+1



❖ Точка зрения

❖ Заинтересованное лицо

❖ Интерес

❖ Способ представления/моделирования



Если количество кирпичиков (АВВ) и точек зрения ничем не ограничено,  
то как их правильно подобрать?



## РАЗДЕЛ 2. О РАССЛОЕНИИ

---

РАССЛОЕНИЕ, КАК БАЗОВАЯ КОНЦЕПЦИЯ АРХИТЕКТУРЫ

- 
- ❖ В 1967 году Ивар Якобсон, будучи сотрудником Эрикссон, предложил использование софтверных компонент в новом поколении программноконтролируемых коммутаторов (телефонных станций), которые тогда разрабатывала компания Ericsson. В ходе этой работы Якобсон **изобрел sequence diagrams, и разработал collaboration diagrams**. Он также использовал state transition diagrams для описания потока сообщений (message flows) между компонентами (components).
  - ❖ Якобсон увидел необходимость неких шаблонов (blueprints) для разработки программного обеспечения. Он был одним из первых разработчиков of the Specification and Design Language (SDL). В 1975 язык SDL стал стандартом в телекоммуникационной индустрии.
  - ❖ После защиты диссертации Якобсон работал в R&D компании Ericsson над разработкой компьютеризированных коммутаторов АКЕ и АХЕ. С 1987 по 1992 год **в рамках проектов для Ericsson Якобсон разработал** метод Object-Oriented Software Engineering (OOSE) и частью этой работы стало изобретение **Use Cases, как способа спецификации функциональных требований к программному обеспечению**.
  - ❖ В 1995 Ericsson продаёт бизнес, связанный с Objectory, в пользу компании Rational Software и **Якобсон начинает работу с Гради Буч (Grady Booch) и Джеймсом Рамбо (James Rumbaugh)**, известных все вместе, как "Три товарища" (Three Amigos). В Rational они **создают UML** и Objectory-процесс, позднее доработанный до Rational Unified Process **под руководством Филипп Крачтен (Philippe Kruchten)**.
  - ❖ В 1995-1997 году развитие UML переходит под контроль OMG, в чем заинтересованы производители CASE-инструментов и другие игроки, не желающие контроля за стандартом со стороны Rational Software.
  - ❖ Филипп Крачтен выполнял обязанности директора of Process Development (RUP) в Rational Software и сохранял эту позицию до 2003 года, когда IBM купил Rational. Якобсон решил покинуть компанию, но оставался при этом до мая 2004 года в качестве главного технического консультанта. В 2004 году Rational покинул и Филипп Крачтен.

**Logical view:** логический взгляд отражает функциональность, которую система предоставляет для конечных пользователей. Используются UML-диаграммы типа [activity diagrams](#), [class diagrams](#), и [state diagrams](#).

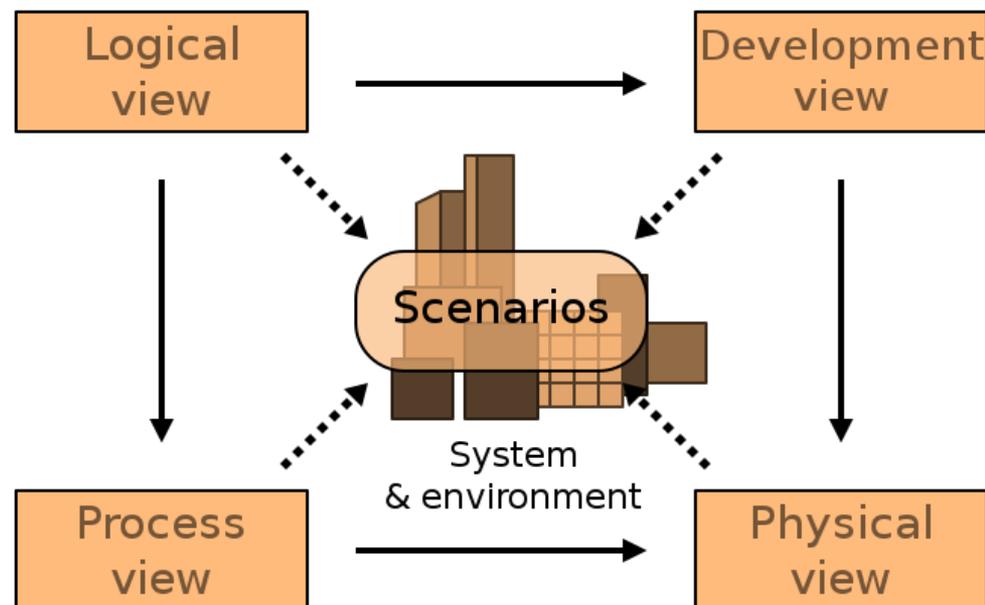
**Process view:** процессный взгляд помогает отразить динамические аспекты системы, разъясняет системные процессы и как они взаимодействуют, фокусируется на поведении системы в режиме runtime. Процессный взгляд адресует потребности в понимании параллельности, распределенности, интеграциях, производительности, масштабируемости и т.п. Используемые UML-диаграммы - [activity diagram](#).

**Development view:** взгляд на систему с точки зрения программистов, заинтересованных в управлении программным кодом. Этот взгляд также известен как **implementation view**. Используются UML-диаграммы типа [Component diagram](#) для описания состава системных компонент, а также [Package diagram](#).

**Physical view:** физический взгляд изображает систему с точки зрения системных инженеров. Он передаёт топологию программных компонент на физическом уровне, а также физические связи между этими компонентами. Этот вид также известен как **deployment view**. Используемые UML-диаграммы - [deployment diagram](#).

**Сценарии:** описание архитектуры иллюстрируется с использованием небольшого количества [use cases](#) (или сценариев), образующих пятый взгляд. Сценарии описывают последовательность взаимодействий между объектами и процессами. Они используются для демонстрации совместной работы всех архитектурных элементов и проверки всего архитектурного дизайна в целом. Сценарии являются отправной точкой в тестировании архитектурного прототипа. Этот взгляд известен также как **Use Case view**.

Ф.Кратчен : «Подход 4+1 является общей идеей и не ограничен какой либо нотацией, инструментом или методологией. Вы можете использовать и другие точки зрения, я лишь описываю те, которые хорошо зарекомендовали себя в нашей практике».

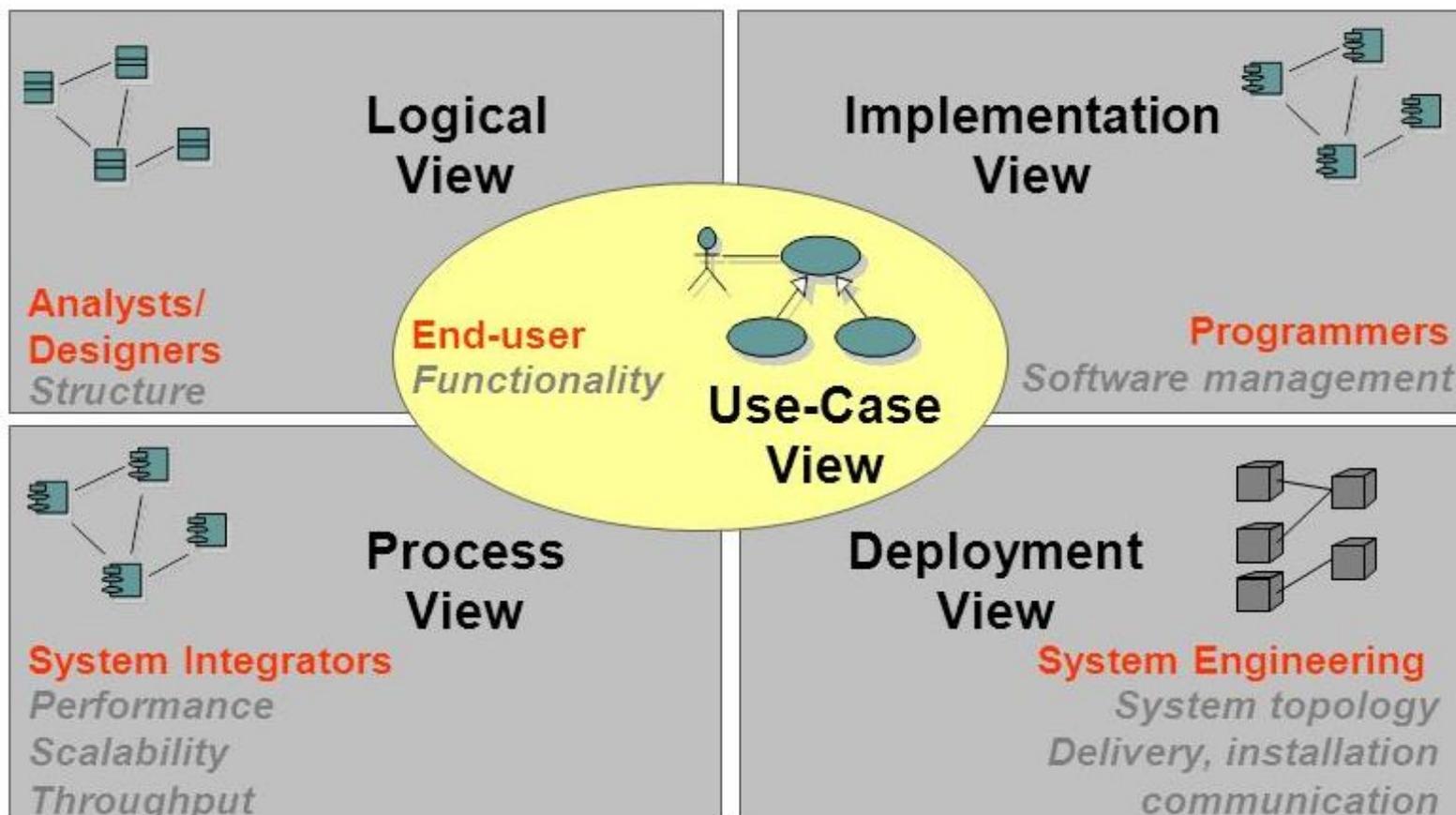


Филипп Кратчен ([Philippe Kruchten](#)) – канадский (а точнее международный) эксперт в области построения программных систем.

4+1 это модель точек зрения, разработанная Филиппом Кратченом для описания архитектуры программного насыщенных систем. Филипп предложил рассматривать такие системы одновременно с нескольких точек зрения. Эти точки зрения предназначены для описания системы с точки зрения различных стейкхолдеров, таких как конечные пользователи, разработчики, проектные менеджеры.

Изначально было предложено 4 взгляда: логический, процессный, физический, разработческий. Дополнительно было отобран и введен взгляд через призму [use cases](#) или сценариев, чтобы проиллюстрировать как работает архитектура. Поэтому модель получила название 4+1.

# ПРЕДСТАВЛЕНИЕ АРХИТЕКТУРЫ: МОДЕЛЬ 4+1



❖ Точка зрения

❖ Заинтересованное лицо

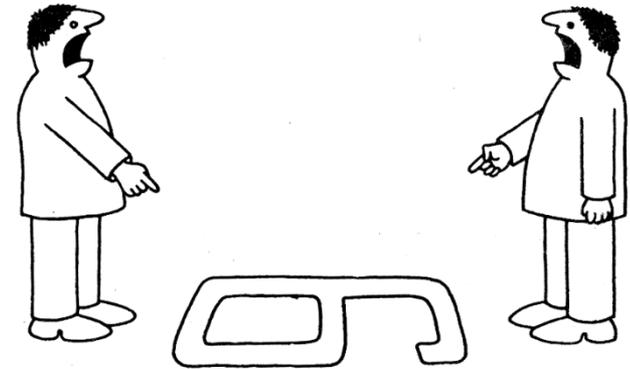
❖ Интерес

❖ Способ представления/моделирования

# КАКИЕ ЕЩЕ ВЗГЛЯДЫ ВОЗМОЖНЫ?

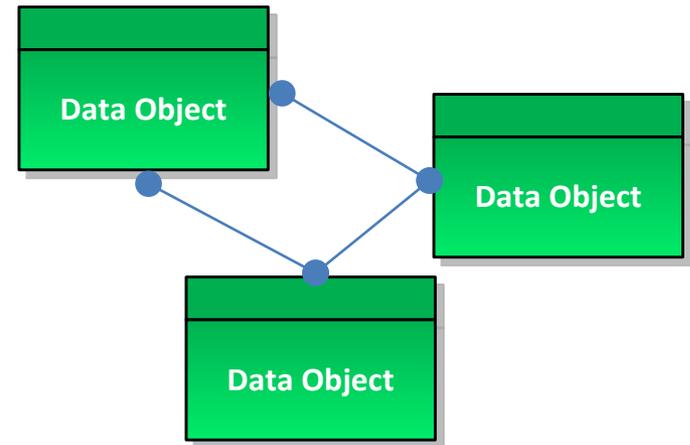
Анализ деятельности предприятия, поиск причин его не эффективной работы или поиск новых драйверов роста заставляет рассматривать предприятие с различных точек зрения:

- ❖ Предприятие, как **совокупность процессов**
- ❖ Предприятие, как **совокупность система**
- ❖ Предприятие, как **совокупность индивидов**
- ❖ Предприятие, как **иерархия структур и функций**
- ❖ Предприятие, как **совокупность информации и данных**
- ❖ Предприятие, как **совокупность синхронизированных информационных потоков**
- ❖ Предприятие, как **набор целей и механизмов их достижения**
- ❖ Предприятие, как **целеустремленная система с набором драйверов, мотивов, рисков и ограничений**
- ❖ Предприятие, как **часть более широкого социального или культурного контекста**



Ни одна точка зрения не даёт универсального представления о предприятии и только совокупность представлений даёт полную картину, хотя, зачастую, субъективную ввиду ограниченных познавательных способностей аналитика или ошибочной интерпретации в коллективе моделей и модельных элементов.

# С ЧЕГО СЛЕДУЕТ НАЧАТЬ?



Эдвард Йордон, 1980-е годы:

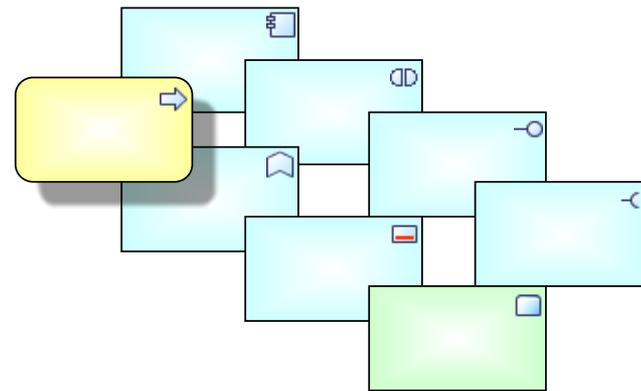
если в прошлом (!) мы применяли функциональную декомпозицию «сверху вниз», учтите, что ООТ (объектно-ориентированные технология) отличается от нее кардинально!

Теперь вы должны мыслить категориями построения системы  
«от центра вовне», в виде взаимодействия объектов.



## СТАНДАРТНЫЕ ЭЛЕМЕНТЫ

- **Application Component** – автоматизированная информационная системы.
- **Функция системы** – элемент поведения системы, отражающий определенный паттерн обработки данных или контроля за ходом бизнес-процесса.
- **Интерфейс системы** – «механизм» предоставления определенного поведения (функции) системы наружу.
- **Application Interaction** – взаимодействие систем, выраженное передачей данных между системами или вызовом из одной системы функции в другой системе.
- **Процесс'ный элемент** – действие, являющееся частью взаимоувязанной цепочки действий.
- **Функция подразделения** – предписанный подразделению вид деятельности.
- **Node** – центры обработки данных.
- **Data Object** – минимальная информационная единица.
- **Location** – место установки или использования системы.



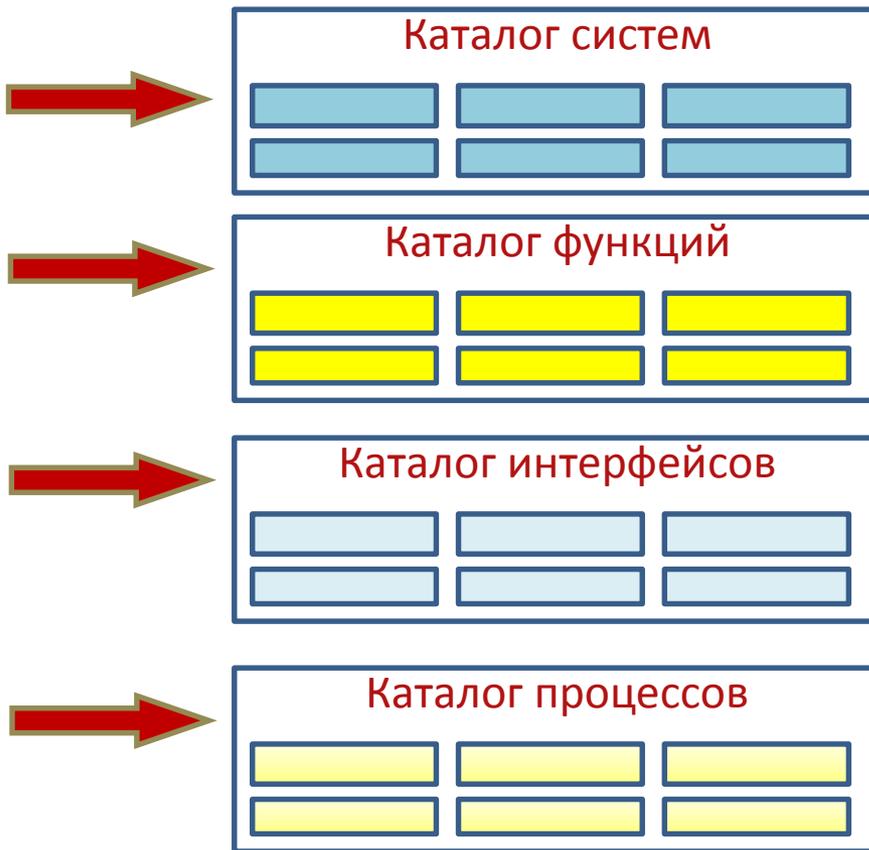
- **Actor** – активный действующий элемент – сотрудник или подразделение.
- **Role** – роль, в которой выступает активный действующий элемент.
- **Значение** – значимость информационного элемента для той или иной роли.
- **Цель** – будущее состояние или результат, к которому стремится система или активный элемент.
- **Stakeholder** – лицо, заинтересованное в результатах деятельности.

# КАЖДЫЙ ЭЛЕМЕНТ ОБРАЗУЕТ/ПОРОЖДАЕТ ОДИН АНАЛИТИЧЕСКИЙ СЛОЙ



Каждый элемент (стереотип) лежит в основе соответствующего каталога элементов.  
Каждый каталог элементов – это аналитический слой.

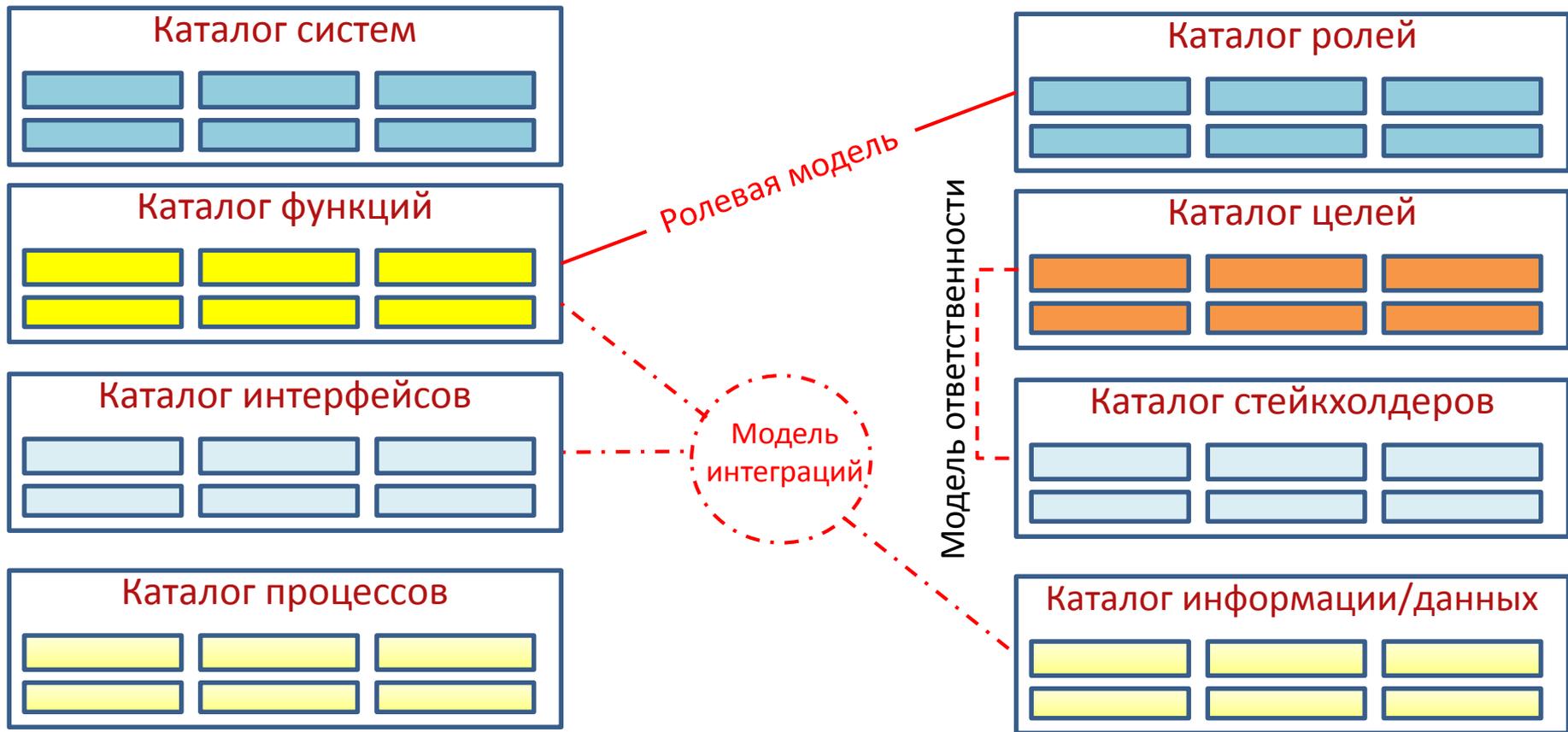
- **Application Component** – автоматизированная информационная системы.
- **Функция системы** – элемент поведения системы.
- **Интерфейс системы** – «механизм» предоставления определенного поведения (функции) системы наружу.
- **Процесс'ный элемент** – действие, являющееся частью взаимоувязанной цепочки действий.



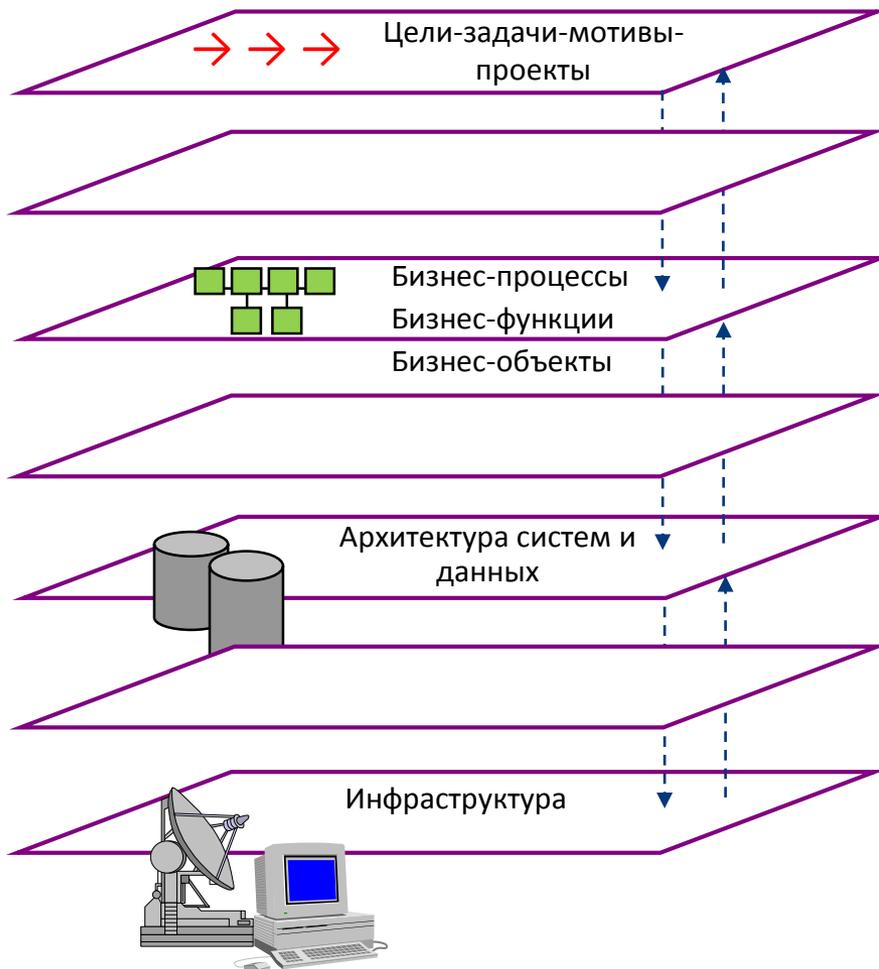
# СВЯЗИ МЕЖДУ СЛОЯМИ – ЭТО КОМПЗИТНЫЙ ВЗГЛЯД НА СИСТЕМУ



Между любыми слоями/каталогами строятся связи, позволяя создавать бесконечное количество комплексных точек зрения на предприятие



# АРХИТЕКТУРА: МНОГОУРОВНЕВЫЙ СТРУКТУРНЫЙ ПОРЯДОК В ЭЛЕМЕНТАХ И СВЯЗЯХ МЕЖДУ НИМИ



## Motivation & Delivery:

- Структурирует цели, мотивы и факторы влияния
- Расширение: проекты, программы, плато-гар

## Business layer:

- Структура процессов, функций, подразделений.
- Структурирует бизнес-объекты
- Структура создаваемого продукта

## Application layer: обработка данных

- Структурирует данные
- Структурирует поведение систем
- Структура интеграций

## Infrastructure:

- Структурирует сети.
- Структурирует центры и узлы обработки
- Структурирует системное программное обеспечение



## РАЗДЕЛ 3. МЕТОДИЧЕСКИЕ ИНСТРУМЕНТЫ TM FORUM

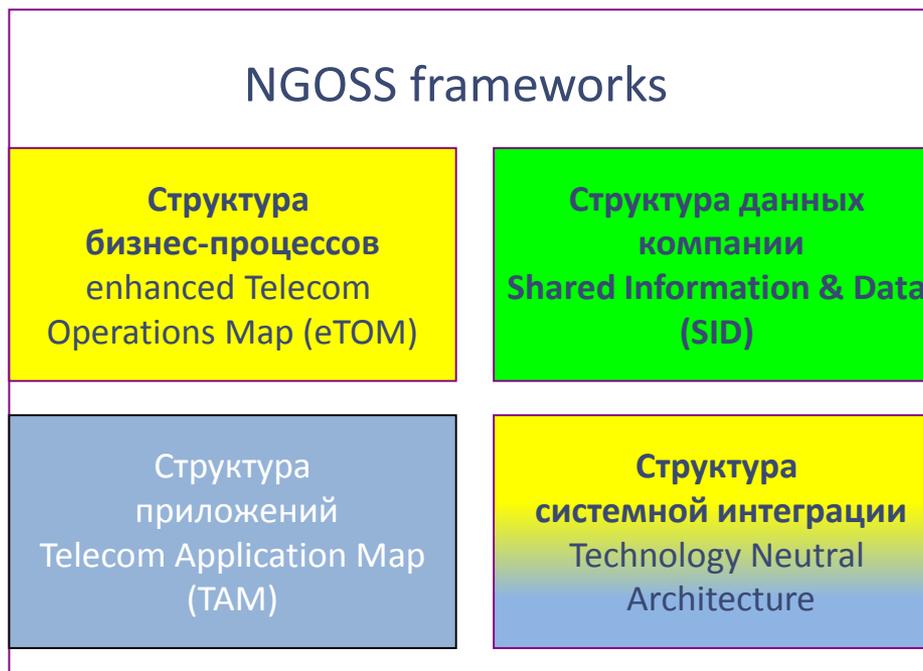
---

eTOM, TAM, SID, NGOSS, API



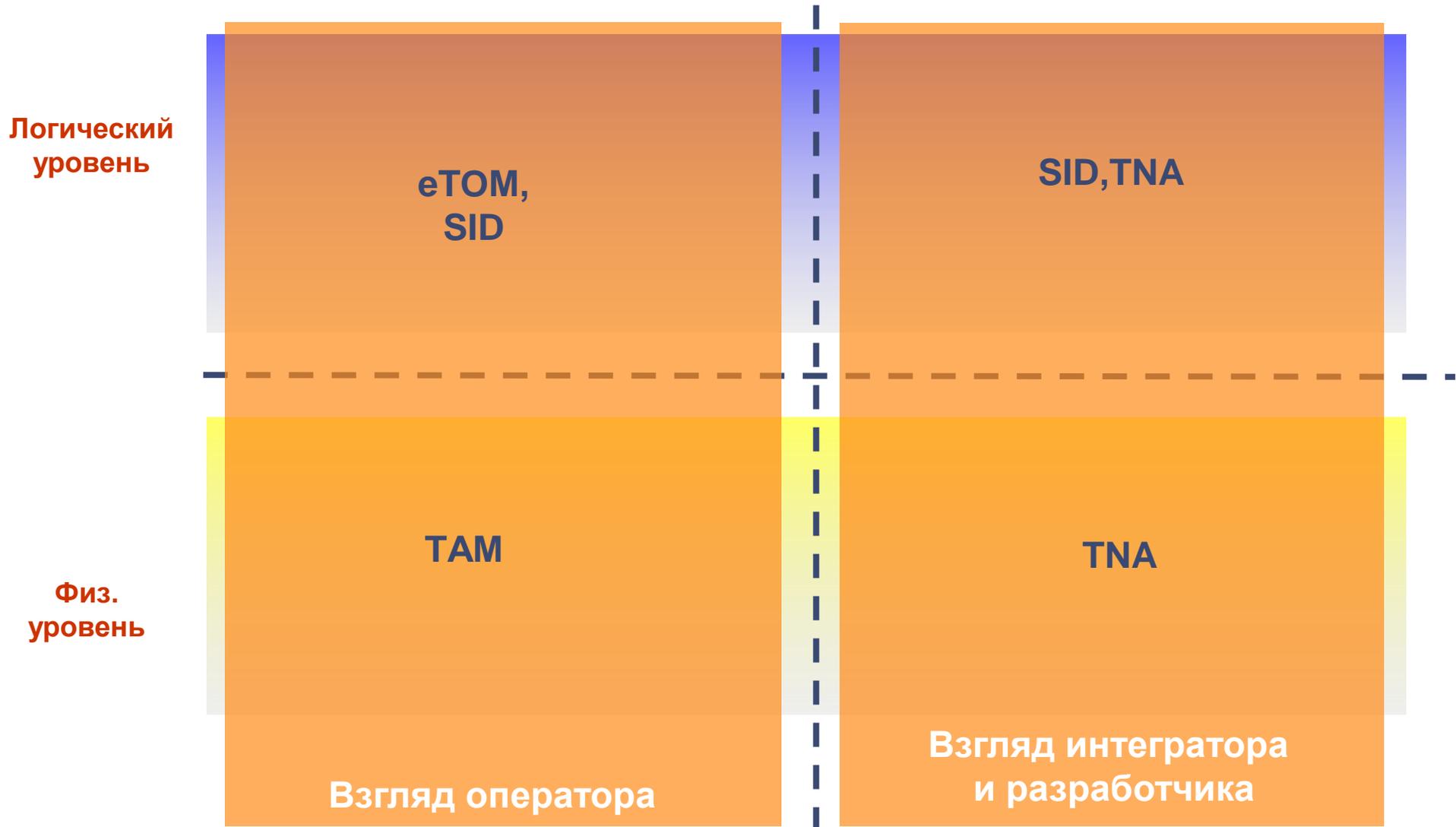
MARCUS AURELIUS LTD

# В ОСНОВЕ NGOSS ЛЕЖАТ ЧЕТЫРЕ КЛЮЧЕВЫХ ФРЕЙМВОРКА:

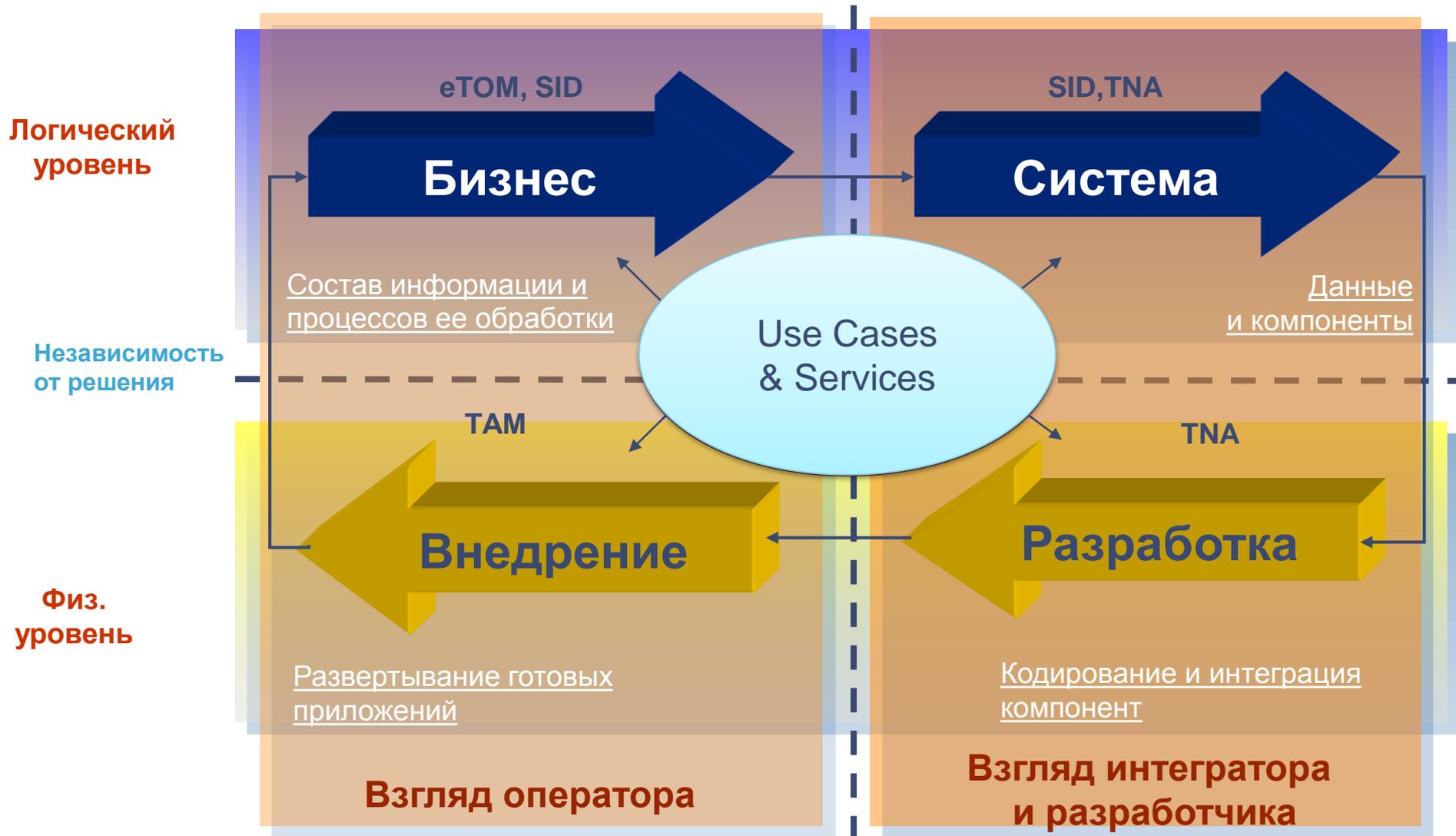


- ❖ eTOM – расширенная карта операций телекоммуникационной компании
- ❖ SID – модель информации и данных
- ❖ TAM – карта приложений телекоммуникационных компаний
- ❖ TNA – технологически нейтральная архитектура или методология интеграции приложений

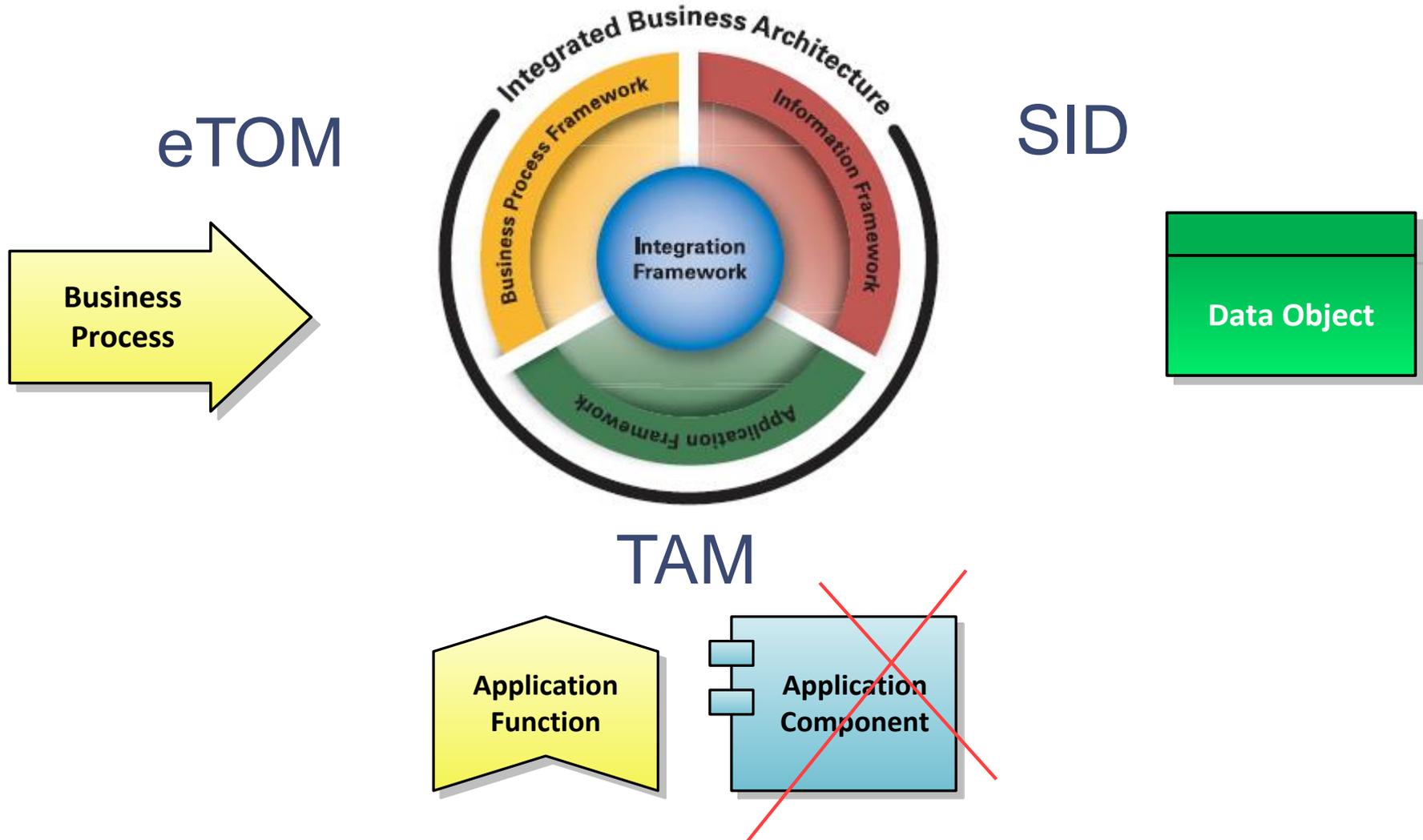
# NGOSS. О всестороннем взгляде при проектировании



# NGOSS. О всестороннем взгляде при проектировании

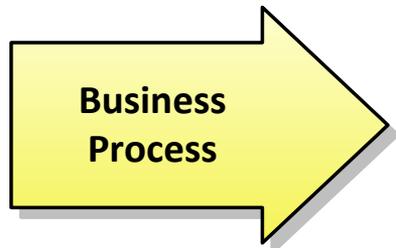


## tmforum Framework



**Application** – кластер когерентных функций или других Applications. Application – абстрактное логическое описание функций, тогда как программный продукт и его способности - это реализация функциональности.

# Модель eTOM



# Модель SID

Data Object

## Рынок / Продажи

Рыночная стратегия и план

Маркетинговая кампания

Канал продаж

Контакт/Интерес/Перспектива

Сегмент рынка

Конкурент

Статистика продаж

## Продукт

Продукт

Стратегический план линейки продуктов

Производительность продукта

Спецификация продукта

Предложение продукта

Статистика использования продукта

## Клиент

Клиент

Заказ клиента

Проблема клиента

Тариф клиента

Детализация счетов

Взаимодействие с клиентом

Статистика клиента

SLA клиента

Сбор платежей

## Услуга

Услуга

Приложения услуга

Производительность услуга

Стратегия и план услуги

Тестирование услуги

Спецификация услуга

Конфигурация услуга

Использование услуга

Проблемы услуги

## Ресурс

Ресурс

Топология ресурса

Производительность ресурса

План и стратегия ресурса

Тестирование ресурса

Спецификация ресурса

Конфигурация ресурса

Использование ресурса

Проблемы ресурса

## Поставщик / Партнер

П/П Производительность

П/П Счет

П/П Исследование счетов

Поставщик / Партнер

П/П Взаимодействия

П/П Заказ

П/П Проблема

П/П План

П/П Продукт

П/П SLA

П/П Статистика

П/П Платеж

## Предприятие

## Общие бизнес-сущности

Участник

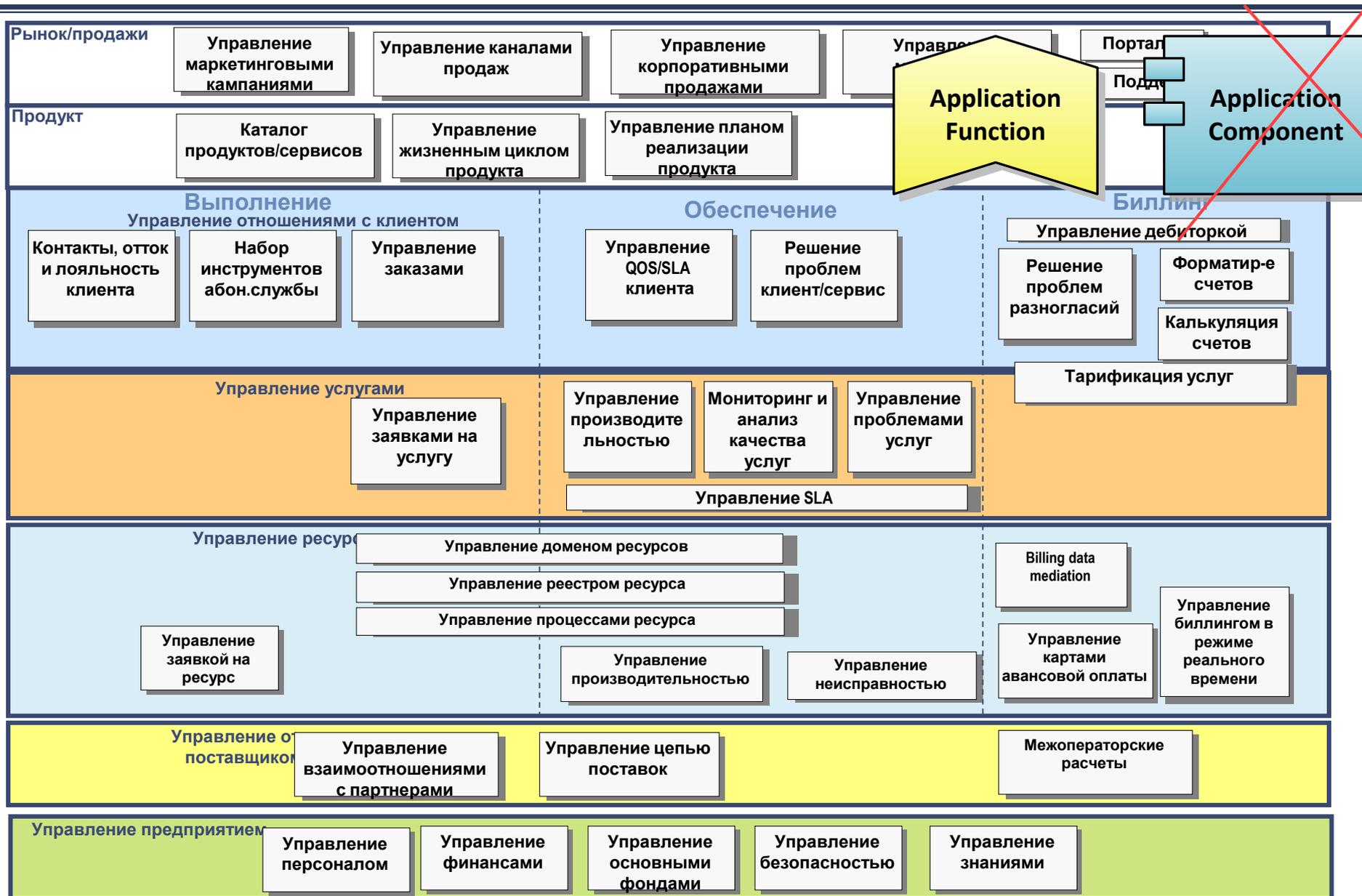
Бизнес-взаимодействие

Местоположение

Политика

Соглашение

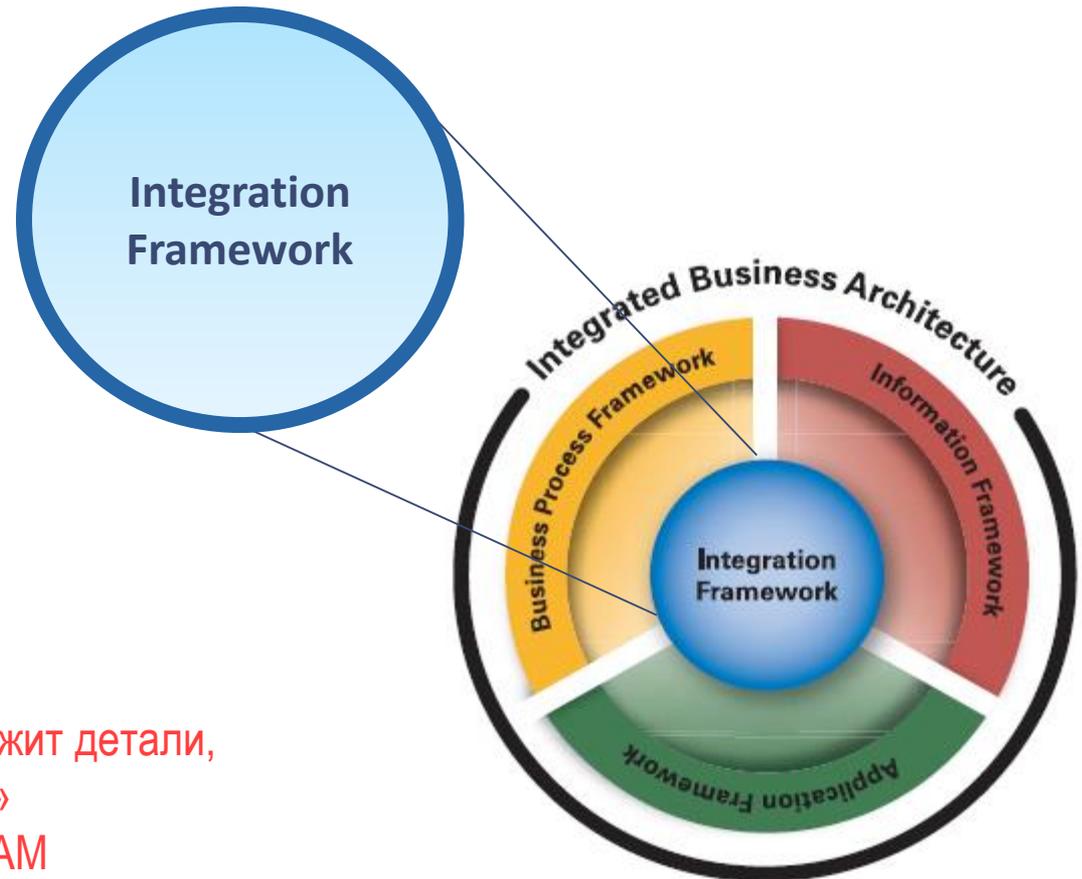
# Telecom Application Map



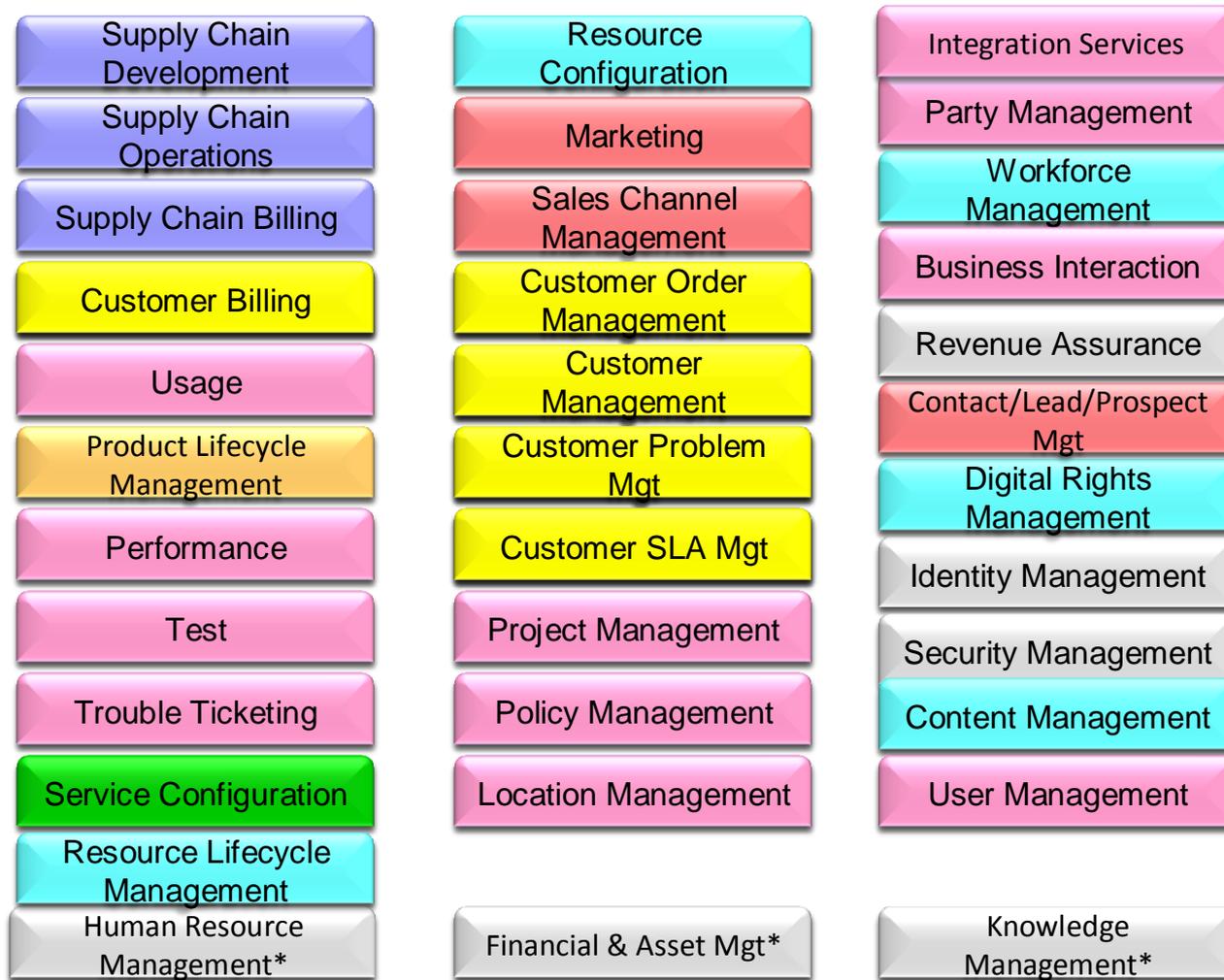
# INTEGRATION FRAMEWORK. ЧТО ЭТО?



Integration – ответ на вопрос, как eTOM, TAM, SID работают вместе.



TM Forum Integration Framework содержит детали, поясняющие как происходит «склейка» строительных блоков из eTOM, SID, TAM



**Market / Sales**  
**Product**

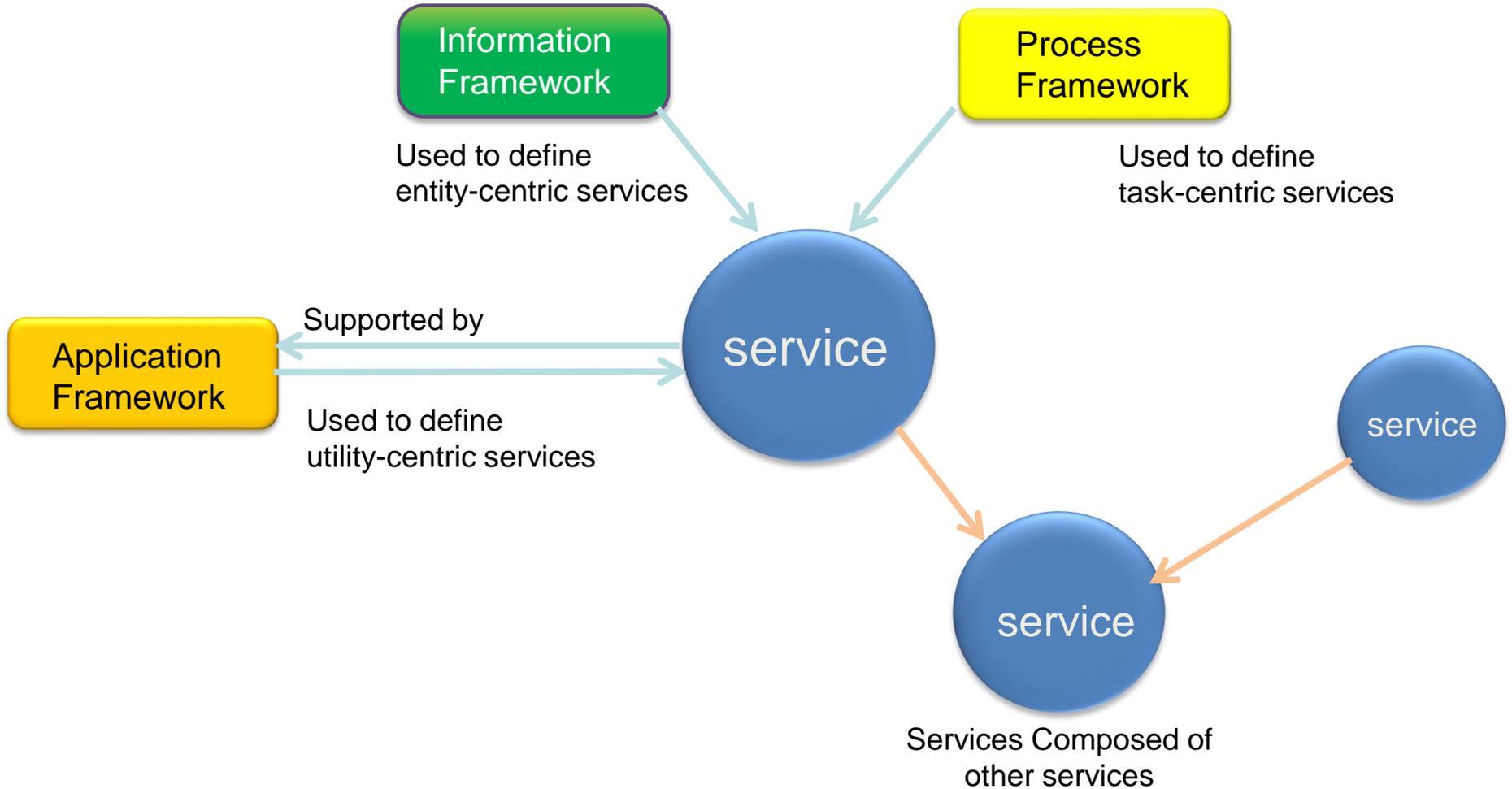
**Customer Service**

**Resource Supplier / Partner**

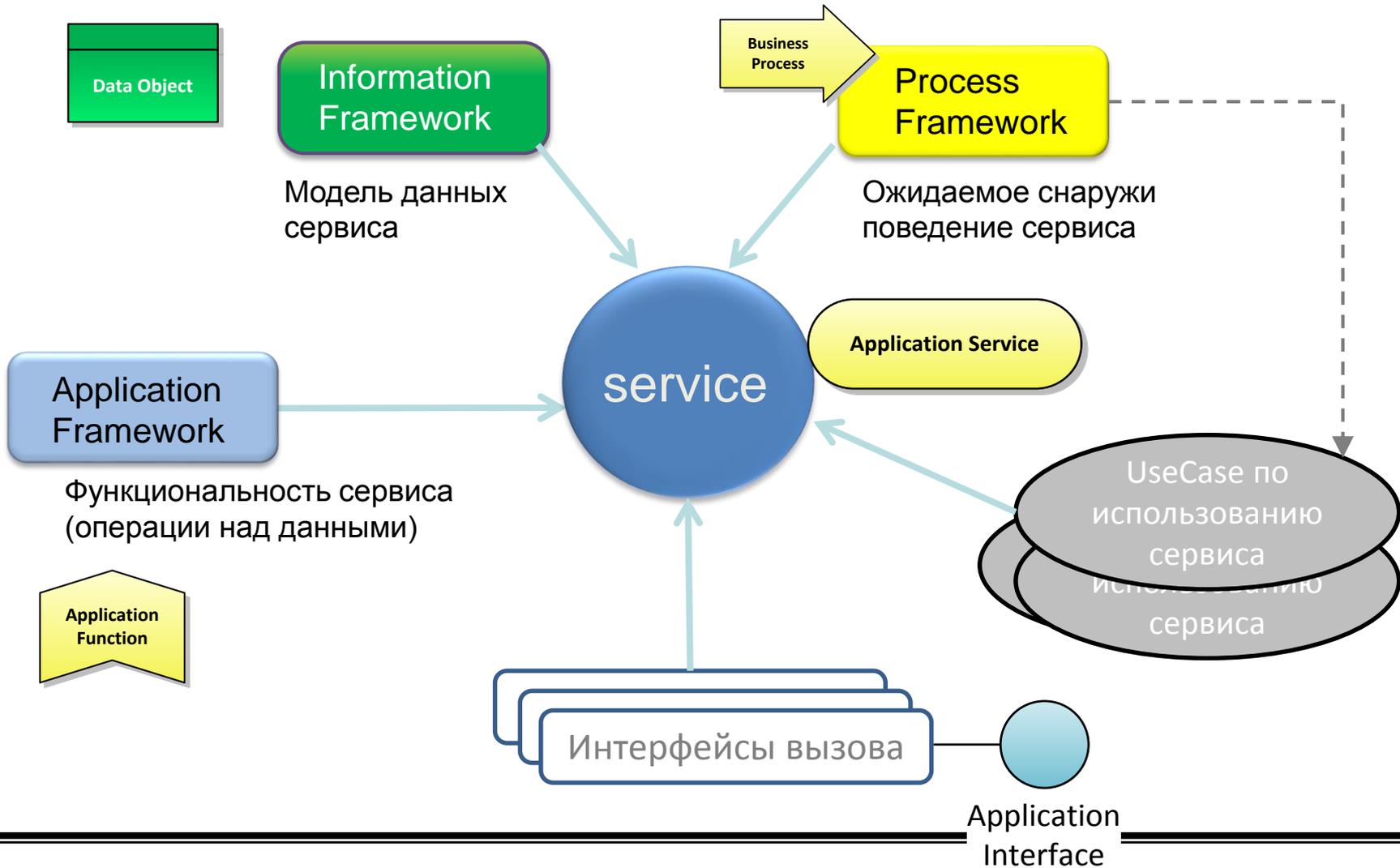
**Enterprise Common Business**

\* Indicates out of scope

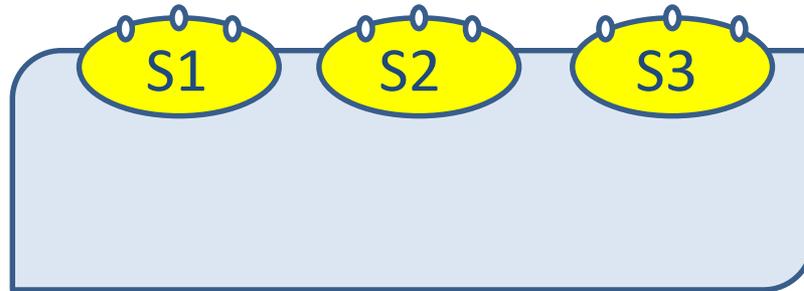
# ВИДЫ БИЗНЕС-СЕРВИСОВ



# МЕТАМОДЕЛЬ БИЗНЕС-СЕРВИСА



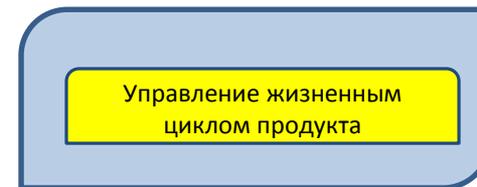
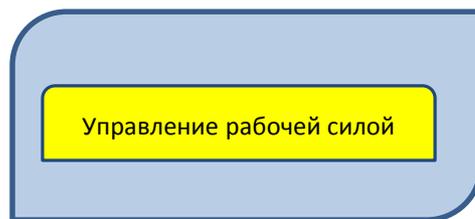
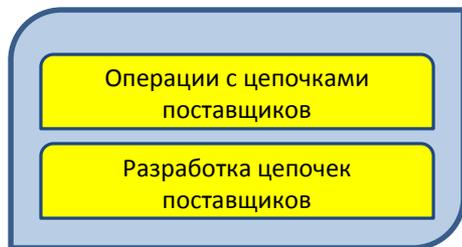
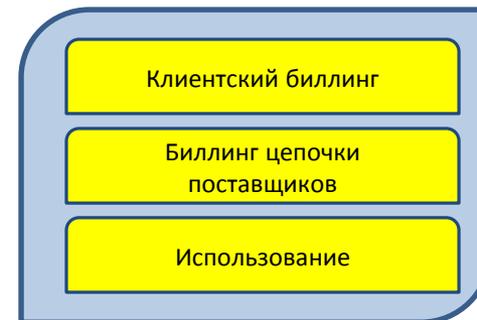
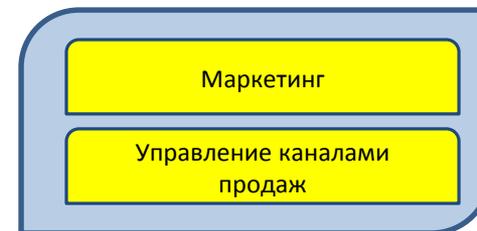
# КОНЦЕПЦИЯ ПЛАТФОРМЫ (GB 945)



Платформа (вариант: «черный ящик»)

- ❖ Платформа представляет собой группировку сервисов, людей и ролей
- ❖ Платформы – это строительные блоки архитектуры предприятия
- ❖ Объединяют наборы сервисов, которые:
  - ❖ Отражают концентрацию на задачах предприятия
  - ❖ Задают верхнеуровневый подход к предоставлению услуг

# ПРЕДПОЛАГАЕМЫЕ ПЛАТФОРМЫ



### TMForum Open APIs for Enabling Ecosystems

Product Ordering  
API

Product Catalog  
API

Product Inventory API

Billing  
API

Service Catalog  
API

Performance  
API

SLA Management  
API

Trouble Ticket  
API

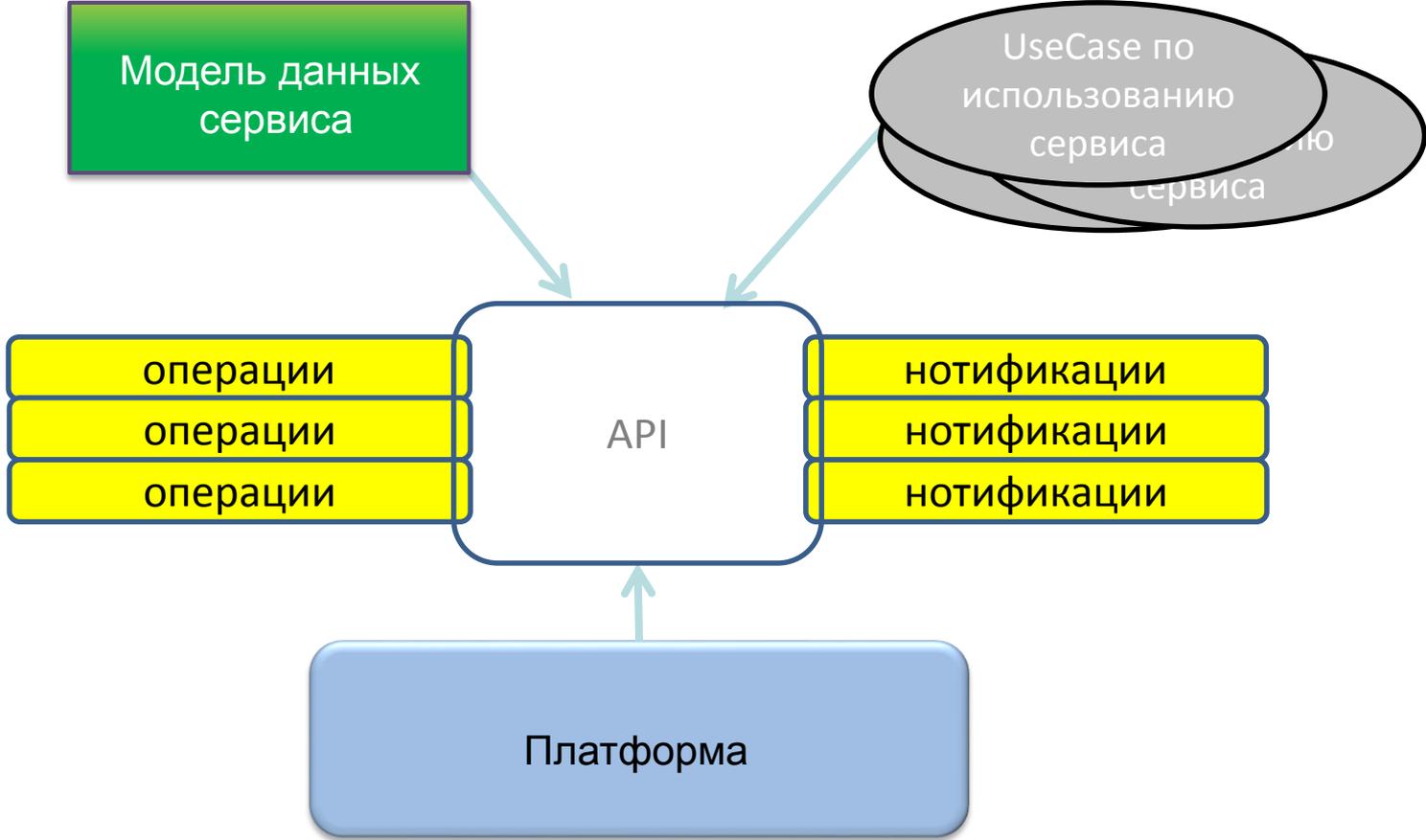
Onboarding  
API

Privacy  
API

Service Activation  
& Provisioning  
API

В настоящий момент TM FORUM сосредоточен на новом взгляде на бизнес оператора связи: ведение бизнеса в экосистеме равных игроков. Чтобы оператору связи стать в новых экосистемах равноправным игроком, а не просто инфраструктурным провайдером, нужны новые стандарты интеграции, позволяющие участвовать в фабриках по производству и доставке цифровых сервисов.

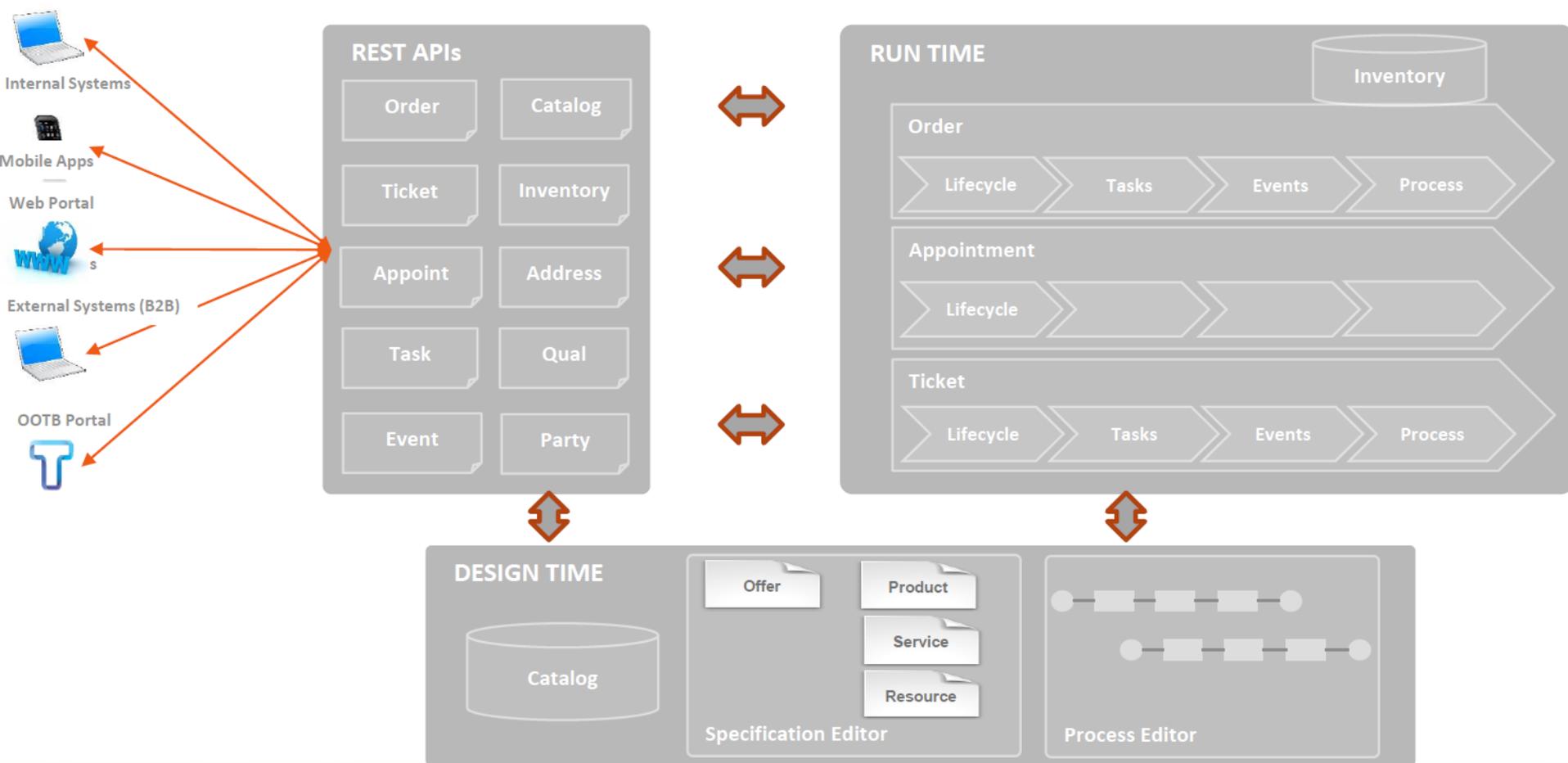
# МЕТАМОДЕЛЬ API



# API: ОРИЕНТАЦІЯ НА REST

## REST APIs Controlling the Telflow Service Delivery Platform

tmforum



## Open API Roadmap

Open API Roadmap	Available	Update Planned
Trouble Ticket API	✓	
Customer Management API	✓	
Product Catalog Management API	✓	
Product Inventory Management API	✓	
Product Ordering API	✓	
Billing Management API	✓	✓
Party Management API	✓	
SLA Management API	✓	
Usage Management API	✓	
Activation and Configuration API	✓	
Performance Management API	✓	✓
Privacy API	✓	
Onboarding API	✓	
Agreement API	✓	
Service Qualification API	✓	
Quote API	✓	
Address API	✓	
Appointment API	✓	

Open API Roadmap	Soon
Service Quality Management API	✓
Test Management API	✓
Change Management API	✓
Resource Order API	✓
Service Ordering API	✓
Resource Catalog API	✓
Resource Inventory API	✓
Entity Provisioning API	✓
Service Catalog API	✓
Service Inventory API	✓
Digital Service Management API	✓
Location API	✓
Loyalty API	✓
Shopping Cart API	✓





## РАЗДЕЛ 4. О КРУПНЫХ АРХИТЕКТУРНЫХ ПРОЕКТАХ

---

- ❖ ПАО «РОСТЕЛЕКОМ»
- ❖ МИНТРАНС РФ



MARCUS AURELIUS LTD

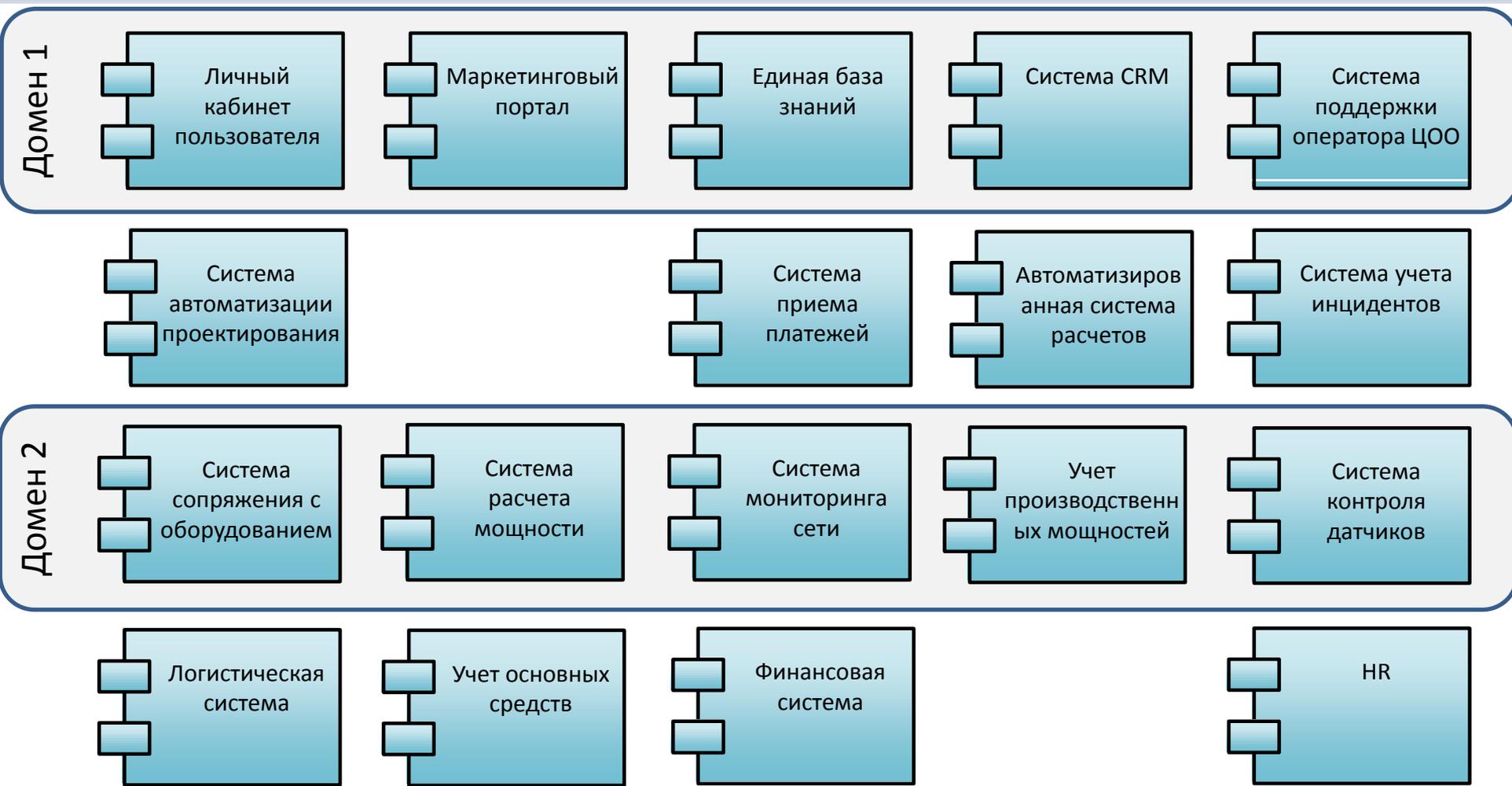
- ❖ Более 20 ЦОД'ов и до 100 мест развёртывания приложений
- ❖ Более 1000 систем и их инсталляций
- ❖ Тысячи системных функций
- ❖ Тысячи интеграционных взаимодействий
- ❖ Десятки вендоров и интеграторов
- ❖ Гигабайты не актуальной документации
- ❖ Десятки не документированных систем собственной разработки
- ❖ Десятки параллельных проектов по модернизации и интеграции систем (более 12 проектных офисов по управлению проектами развития)

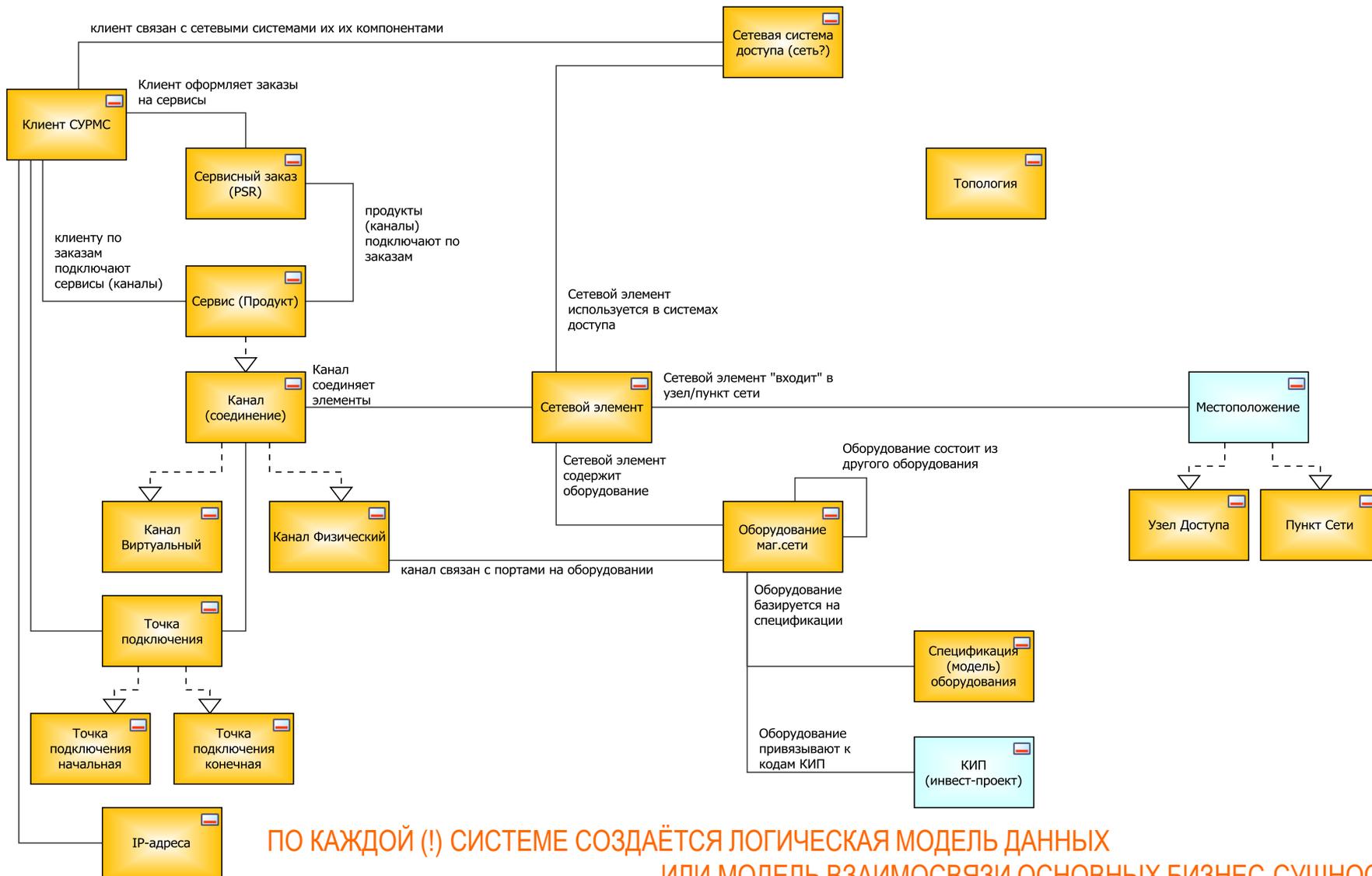
## НАШ ОТВЕТ:

СОЗДАН ОДИН РЕПОЗИТОРИЙ, СОДЕРЖАЩИЙ ВСЁ В ОДНОМ МЕСТЕ В ПОЛНОСВЯЗНОМ ВИДЕ:

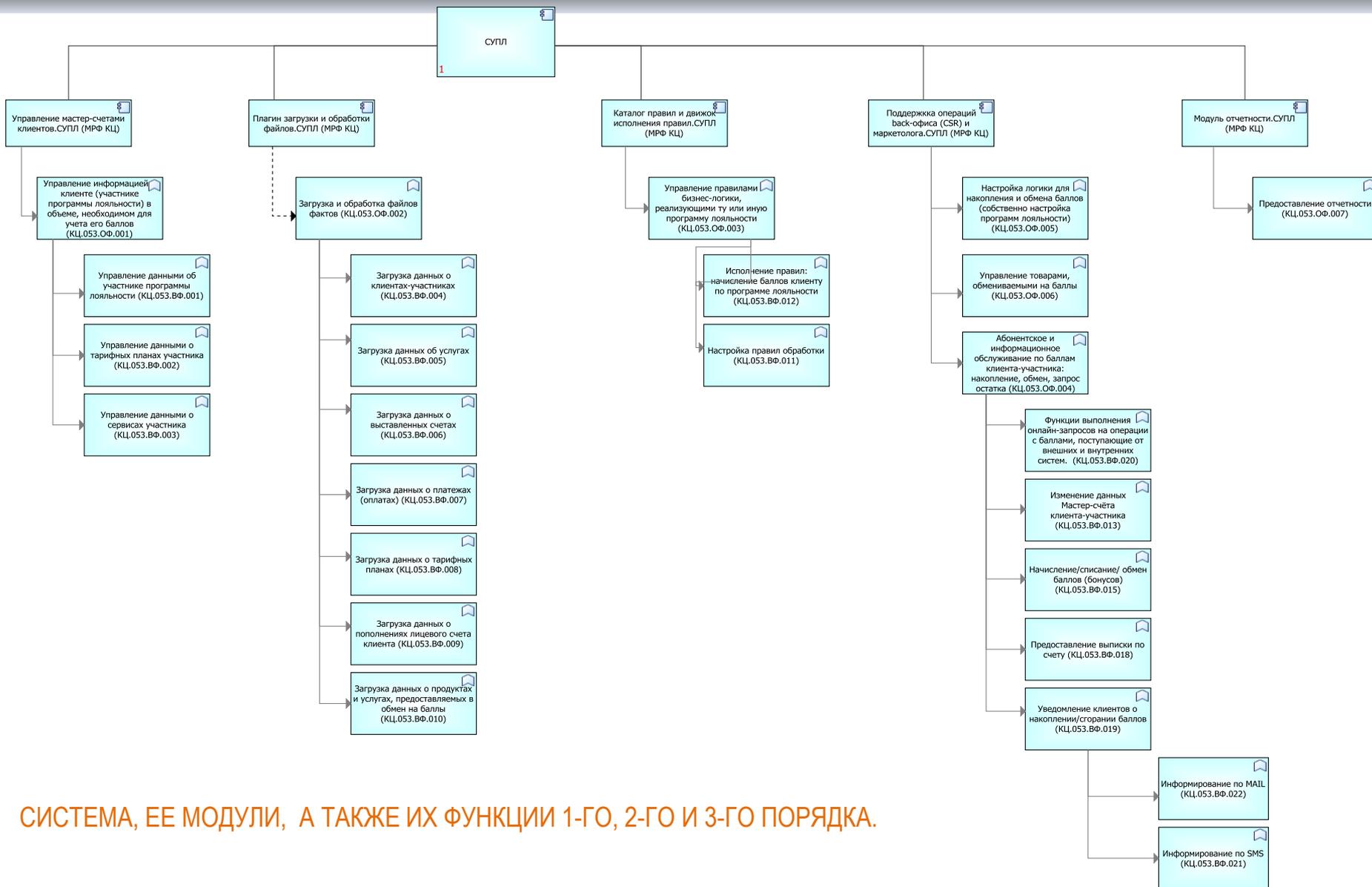
- ❖ Каталог систем
- ❖ Каталог функций
- ❖ Каталог объектов данных
- ❖ Каталог Атрибутов
- ❖ Каталог интеграций
- ❖ Каталог методов
- ❖ Каталог канонических сервисов
- ❖ Каталог ЦОДов

# СИСТЕМНЫЙ ЛАНДШАФТ: СЛОЙ ИНФОРМАЦИОННЫХ СИСТЕМ





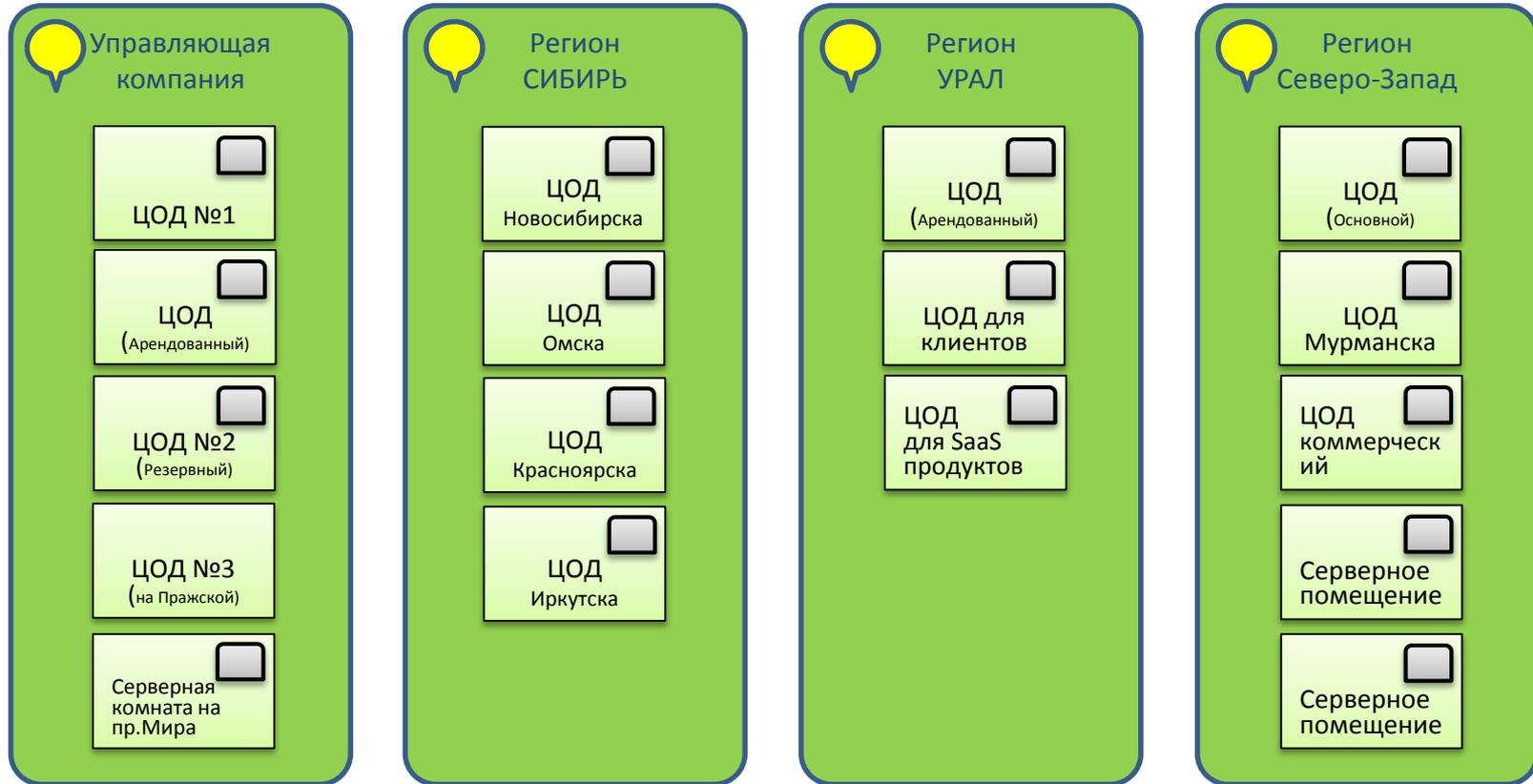
ПО КАЖДОЙ (!) СИСТЕМЕ СОЗДАЁТСЯ ЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ  
ИЛИ МОДЕЛЬ ВЗАИМОСВЯЗИ ОСНОВНЫХ БИЗНЕС-СУЩНОСТЕЙ.



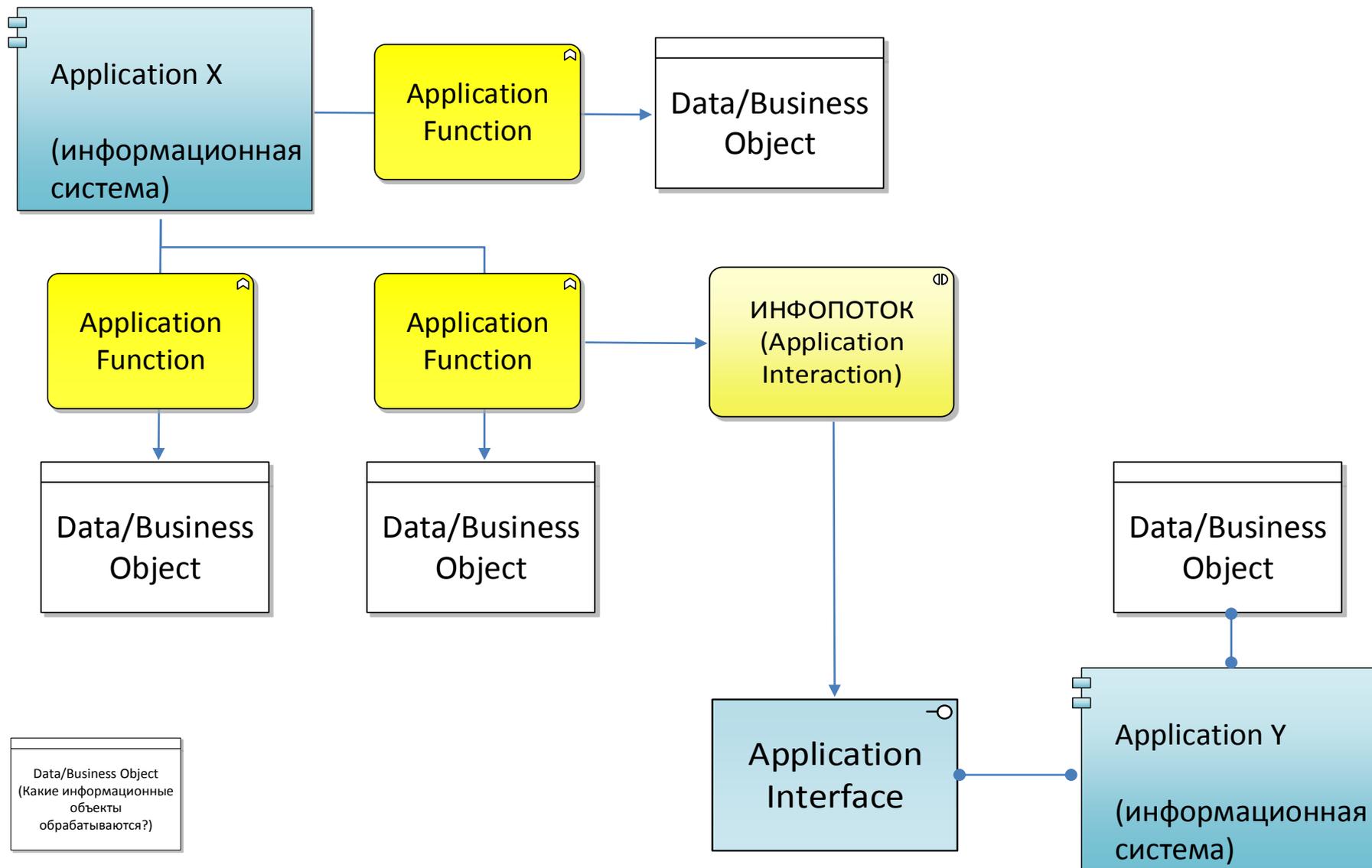
СИСТЕМА, ЕЕ МОДУЛИ, А ТАКЖЕ ИХ ФУНКЦИИ 1-ГО, 2-ГО И 3-ГО ПОРЯДКА.



# ОТДЕЛЬНЫЕ ЭЛЕМЕНТЫ СЛОЯ ИНФРАСТРУКТУРЫ: ТЕРРИТОРИАЛЬНЫЕ ЦЕНТРЫ РАЗМЕЩЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ



ПО КАЖДОМУ ЦОД'У ЗАПОЛНЯЕТСЯ ЕГО АТТРИБУТИВНЫЙ СОСТАВ (ФОРМУЛЯР ЦОД'а): ПЛОЩАДЬ, МОЩНОСТЬ, ОТВЕТСТВЕННЫЙ, АДРЕС И Т.П. КАЖДАЯ ИНФОРМАЦИОННАЯ СИСТЕМА СВЯЗАНА С ОДНИМ ИЗ ЦОД'ОВ



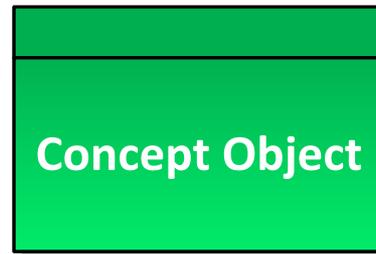
- ❖ Более 100 субъектов взаимодействия
- ❖ Сотни систем
- ❖ Сотни системных функций
- ❖ Сотни бизнес-функций, выполняемых в системах
- ❖ Сотни потоков неструктурированной информации
- ❖ Десятки не документированных систем в постоянном развитии
- ❖ Десятки параллельных проектов по модернизации и интеграции систем
- ❖ Десятки целевых ориентиров и задач государственного уровня

## НАШ ОТВЕТ:

СОЗДАН ОДИН РЕПОЗИТОРИЙ, СОДЕРЖАЩИЙ ВСЁ В ОДНОМ МЕСТЕ В ПОЛНОСВЯЗНОМ ВИДЕ:

- ❖ Каталог подразделений и предприятий
- ❖ Каталог деловых функций
- ❖ Каталог информации
- ❖ Каталог систем
- ❖ Каталог инфопотоков
- ❖ Каталог функций систем
- ❖ Каталог объектов данных
- ❖ Каталог атрибутов

# ОСОБЕННОСТИ МИНТРАНС: МОДЕЛИРОВАНИЕ ДАННЫХ





## РАЗДЕЛ 5. ОСОБЕННОСТИ АРХИТЕКТУРНЫХ ПРОЕКТОВ

---



MARCUS AURELIUS LTD



❖ eTOM – расширенная карта операций телекоммуникационной компании

❖ SID – модель информации и данных

❖ Open API – открытые методы интеграции

## Модули и Приложения

- ❖ Сбор данных
- ❖ Архивирование и резервирование данных
- ❖ Валидация данных
- ❖ Массовые операции
  1. Загрузка данных, парсинг данных
  2. S - поиск значений/данных/сущностей
  3. Фильтрация данных
  4. Выделение объектов в группу, в том числе путем фильтрации
  5. Построение выборки данных из информационной системы по заданным критериям
  6. Построение выборки данных из группы систем
  7. Массовое однотипное изменение над всеми объектами группы, в том числе изменение по формуле или определенному алгоритму, например, вычисление должников или передовиков

## ❖ Учёт информационных единиц (ИЕ).

1. L - загрузка
2. E - ввод
3. G[etting] - получение из внешней системы (или с оборудования)
4. P – проводка изменения или проводка, как метод ввода данных
5. R – чтение/отображение
6. U – изменение
7. UF(forward) - направление/назначение
8. D - удаление
9. Ex[port]
10. Прикрепление файла к объекту информационной системы
11. Поддержка истории изменений объекта (поддержка версионности)

## Модули и Приложения

- ❖ Массовые операции. Выборки и отчеты
- ❖ Создание отчета
- ❖ Представление данных:
  1. Представление данных в виде формы
  2. Представление данных в виде WEB-формы или страницы
  3. Представление данных в виде отчета
  4. Представление данных в виде выгрузки определенного формата
- ❖ Управление внешним устройством: датчиком или приводом
  1. Получение данных от датчика, в том числе в результате опроса датчика
  2. Отправка команд на датчики
  3. Отправка команд на приводы
- ❖ Операции над коллекциями объектов
- ❖ Вычисление значения, атрибута
  1. Расчёт, например, баланса или суммы счета
  2. Расчёт стоимости вызова
  3. Расчет остатка на складе
  4. Вычисление прогнозной величины
  5. Расчет KPI

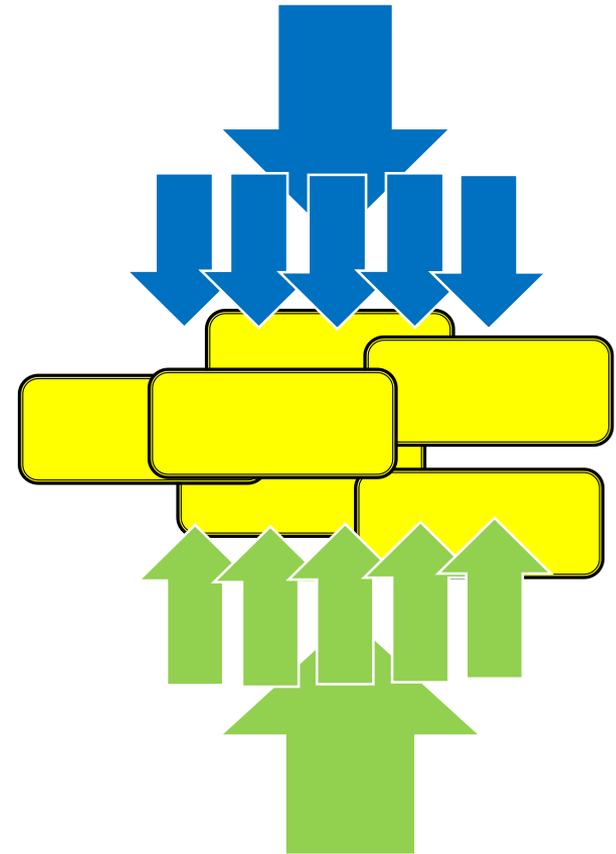
# СОЧЕТАНИЕ ДВУХ ПОДХОДОВ

При создании архитектуры возможно использование двух подходов:

- **Сверху вниз** или от общего к частому. Основные аналитические методы – декомпозиция и детализация.
- **Снизу вверх** или от частного к общему. Основные аналитические методы – агрегация и обобщение.

Обычно у заказчика преобладают:

- на уровне штаб-квартиры: ToBe-проектирование сверху вниз, то есть от новых/изменяющихся целей и задач бизнеса к функциям информационных систем.
- на местах (в регионах): подход снизу вверх, то есть от имеющихся систем и функций к новым возможностям для бизнеса.



**ПРАВИЛЬНОЕ ПРОЕКТИРОВАНИЕ ЗАКЛЮЧАЕТСЯ В КОМБИНАЦИИ ДВУХ ПОДХОДОВ, А НЕ В ОДНОЗНАЧНОМ ПРЕДПОЧТЕНИИ ОДНОГО ИЗ НИХ.**

# ПОЧЕМУ АРХИТЕКТУРОЙ НЕ ЗАНИМАЮТСЯ?

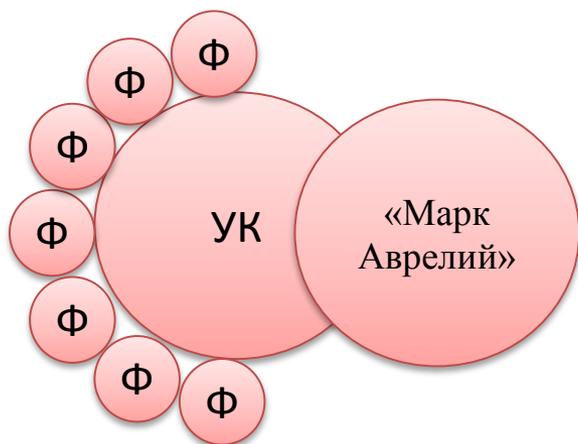


- ❖ Не было и нет инструментов для такого моделирования.
- ❖ Не было и нет регулярной архитектурной практики.
- ❖ Гипертрофированная роль целевых моделей (планирование будущего) в ущерб AsIs-моделям (понимание настоящего): будущее не приходит, а настоящее постоянно усложняется заплаточными методами.
- ❖ Нет специалистов, способных разработать метамодель.
- ❖ Нет понимания как актуализировать результаты больших исследовательских работ.
- ❖ Сложно найти баланс между желаниями по быстрым выгодам от архитектурного проекта и необходимостью длительных, скрупулезных и трудоемких исследований текущего хаоса/порядка в компании.
- ❖ Прозрачность – не самое желанное качество текущей ситуации. Хотя и декларируется чаще всего.

# АРХИТЕКТУРНЫЙ ПРОЕКТ: ЧИСЛО ЗАДЕЙСТВОВАВАННЫХ РЕСУРСОВ

## От управляющей компании, штаба или корп.центра

- по одному человеку на макрорегион/домен (до 7 человек)
- по одному специалисту от каждого проектного офиса (до 10 человек)
- администратор системы (1 человек на 50%)



## От консультантов «Марк Аврелий»:

- Управление проектом
- Обучение рабочей группы
- Методическое руководство
- Разработка и поддержание метамодели
- Архитектор интеграций
- Консультант по процессным моделям
- Специалисты по предметным областям

## От компаний холдинга:

- один ответственный в каждом макрорегионе или федеральном агентстве (до 8 человек)
- специалисты по каждой системе

**Итого для крупных компаний требуется команда не менее 10 человек!**



## РАЗДЕЛ 6. ПОРЯДОК АКТУАЛИЗАЦИИ

---

Архитектура всегда находится в квазиактуальном состоянии, так как нет возможности поддерживать модель в полном соответствии с постоянно изменяющимися реалиями бизнеса. Это понимают все и это изначально заложено в архитектурные методологии, такие, например, как TOGAF.



Все выделенные нами структурные компоненты архитектуры: системы, функции, интерфейсы, интеграции - достаточно стабильны во времени и легко подлежат периодической или даже постоянной актуализации. Оценочно для актуализации требуется ресурс в количестве 1 человек на макрорегион.

# 24 ДОМЕНА ДЛЯ ПРОЕКТИРОВАНИЯ – ХОРОШИЕ КАНДИДАТЫ ДЛЯ ПЛАНИРОВАНИЯ СКОУПА АКТУАЛИЗАЦИИ



1. Архитектура систем (системы и функции первого порядка)

2. Архитектура функций и распределение информации по системам  
(бизнес-объекты и функции второго порядка)

3. Архитектура данных (сущности информационных систем и их атрибуты)

4. Архитектура взаимодействий (интеграции систем)

Продажи  
B2C

Продажи  
B2B

Расчеты с  
клиентами

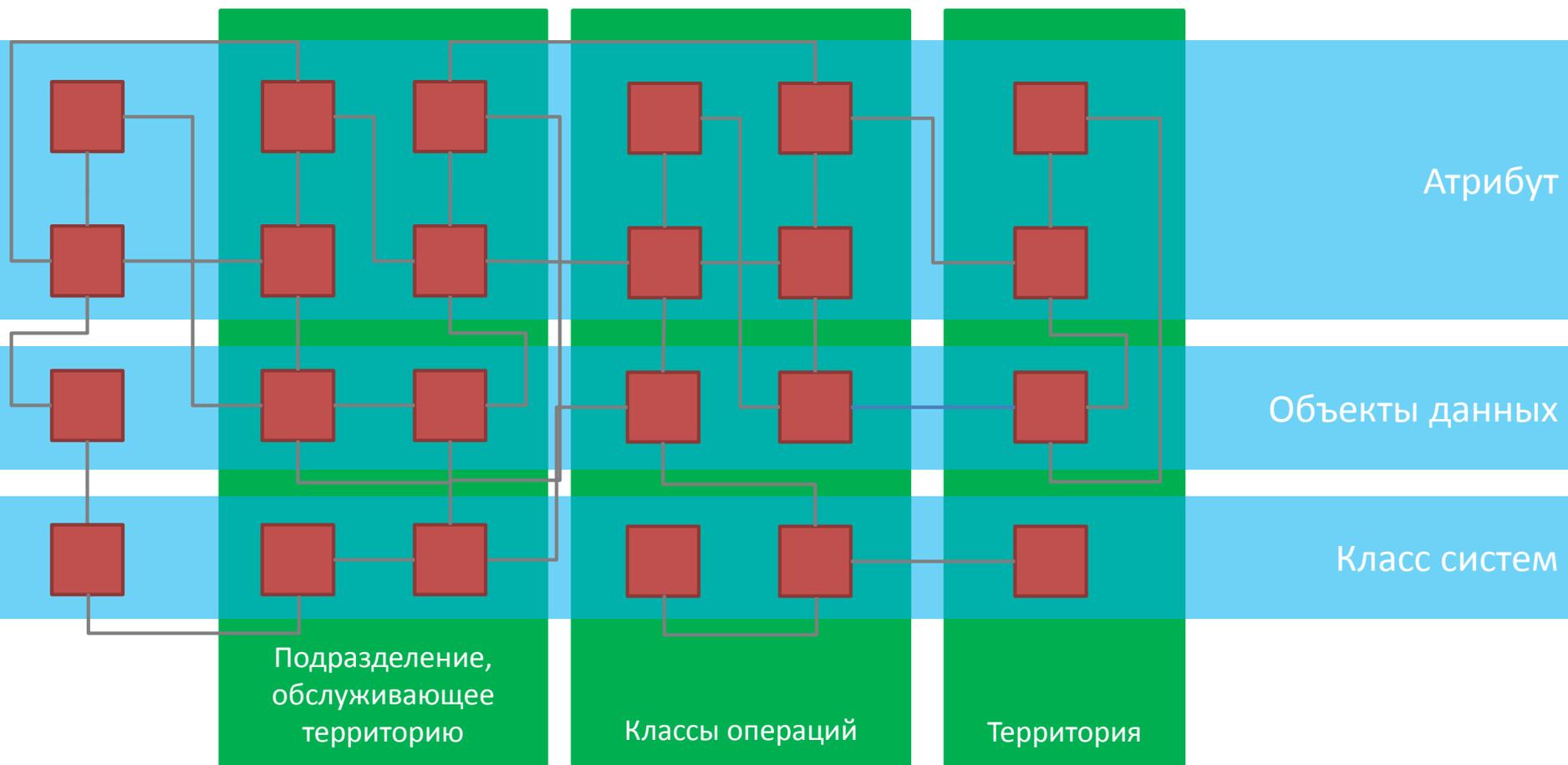
Тех.поддержка  
B2C

Претензионная  
работа

Взыскание  
задолженности

Актуализация может выполняться по-процессно или по слоям. Или даже в отдельном небольшом домене. Это зависит от поставленной со стороны бизнеса или ИТ задачи.

# ПОРЯДОК АКТУАЛИЗАЦИИ АРХИТЕКТУРЫ





## РАЗДЕЛ 7. ВОЗМОЖНОСТИ ИНСТРУМЕНТА

---

Возможности программного инструмента Enterprise Architect финской компании QPR Software PLC. Отличительной особенностью продукта является интуитивная простота работы (как с VISIO) с возможностью получения интерактивных диаграмм, а также неограниченная свобода в использовании на схемах любых элементов, не предусмотренных нотацией моделирования. Это делает схемы настолько выразительными, насколько вы умеете управлять визуальным восприятием сложных картинок высокой степени детализации.

# ИНСТРУМЕНТАЛЬНАЯ СРЕДА МОДЕЛИРОВАНИЯ

ENTERPRISE ARCHITECT. QPR SOFTWARE PLC



The screenshot displays the QPR Enterprise Architect software interface. At the top, the title bar reads "ArchiSurance - QPR EA based ODM.qpream - QPR EnterpriseArchitect". The main menu includes "Home", "View", "Model", and "Help". A ribbon contains various tool groups: Clipboard (Cut, Copy, Paste, Format Painter), Label (Font Color), Arrange (Order, Alignment), Tools (User Rights, Run Script, Auto Numbering Wizard), Web Publishing (Published to Web, Web Styles), Editing (Undo, Delete, Find, Select), and Model Comparison (Select Model, Change Direction, Close).

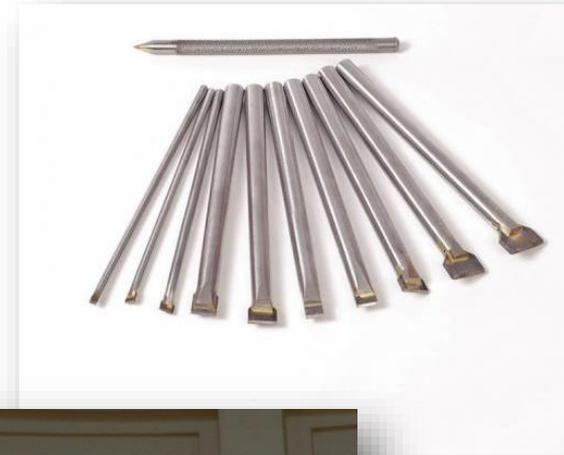
The central workspace shows a diagram titled "QPR Enterprise Architecture based Operational Development". It is organized into three main layers:

- Business Management Diagrams (Orange background):**
  - Background: 00. Principles, 01. Strategies, 02. Business Models, 09. Work Packages, 04. Organization Structures
- Business Architecture Diagrams (Blue background):**
  - 20. Capability Areas, 21. Process Areas, 22. Process Integrations
- Data Architecture Diagrams (Blue background):**
  - 30. Conceptual Data, 31. Conceptual Data Domains, 32. Interface Data, 33. Logical Data
- Application Architecture Diagrams (Blue background):**
  - 41. Application Usage Landscapes, 42. Logical Application Structures, 43. Application Cooperations, 44. Application Integrations

On the left, the "Diagram Explorer" pane shows a "Diagram Hierarchy" tree with folders for 00. Principles through 50. Technology Domains. A "Preview" window at the bottom left shows a detailed view of a diagram element. The bottom status bar includes "Quick properties", "Related elements", and a 100% zoom level.

# ВОЗМОЖНОСТИ ENTERPRISE ARCHITECT

- ❖ Поддержка стандарта Archimate 2.0-3.0
- ❖ Поддержка стандарта BPMN
- ❖ Возможность разработка любой (!) нотации
- ❖ Возможность доработки встроенных нотаций
- ❖ Конструирование любой метамоделли
- ❖ Управляемые панели инструментов
- ❖ Возможность устанавливать связи между моделями
- ❖ Связывание объектов атрибутивно или графически
- ❖ Коллективная работа на общем сервере
- ❖ Локальная работа на ноутбуке с копией сервера в режиме редактирования и без связи с сервером
- ❖ Vbasic-образный язык разработки скриптов для автоматизации рутинный операций
- ❖ Поддержка TOGAF



# QPR ENTERPRISE ARCHITECT



QPR Enterprise Architect - среда коллективной работы с моделями:

- ❖ Содержит карточки всех систем
- ❖ Содержит карточки всех интеграций
- ❖ Содержит все справочники, нужные для стандартизации ввода данных
- ❖ Содержит все представления (отчеты)
- ❖ Содержит различные подложки для отображения данных
- ❖ Содержит интерактивные диаграммы, созданные в полной взаимосвязи с реестрами систем, функций, интеграций
- ❖ Содержит взаимосвязи между всеми элементами, включая возможность навигации между формулярами объектов и схемами
- ❖ Предоставляет единый механизм поиска артефактов моделирования: как на схемам, так и в реестрах

Enterprise Architect – единая интегрированная среда, в которой интегрированы: системы, функции, интерфейсы, инфопотоки, процессы, цели, задачи и т.п. зафиксированные в виде артефактов системы и иллюстрированные интерактивными диаграммами.

# КОНТАКТЫ ДЛЯ СВЯЗИ



*Не отказывайся от помощи, особенно  
когда это связано с исполнением долга.  
Многое из того, что не удаётся сделать  
в одиночку, может быть легко  
достигнуто, если действовать сообща...*

*Марк Аврелий*

Рудь Виктор  
Директор по консалтингу  
ООО «МАРК АВРЕЛИЙ»

<http://www.consulo.ru>

Е-mail: **v.rud @consulo.ru**

Телефон: **+7 (495) 922-12-40**



*Люди существуют друг для  
друга, чтобы друг друга  
улучшать и возвышать.*

*Марк Аврелий*





## Виды услуг компании:

- Управление архитектурой систем и предприятия
- Концептуальное проектирование
- Реинжиниринг процессов
- Дизайн информационных систем
- Бизнес-анализ
- Дизайн и поддержание автоматизированных каталогов услуг
- Проведение тендеров на выбор программного обеспечения
- Обучение по архитектурным методологиям
- Управление требованиями
- Разработка Требований, Тех.Заданий, Архитектурных решений и концепций
- Организационный дизайн
- Процессное управление
- Проработка KPI процессов и подразделений

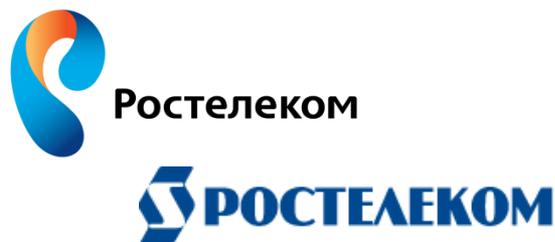
## Виды помощи в больших проектах:

- Управление проектом
- Планирование проекта и ресурсов
- Создание и контроль процессной архитектуры
- Создание и контроль функциональной архитектуры
- Создание и контроль информационной архитектуры
- Разработка учебных материалов
- Обучение ключевых пользователей
- Нормирование численности подразделений и оптимизация орг.штатной структуры
- Реинжиниринг процессов
- Подготовка процессов и функций к передаче в аутсорсинг

# КЛИЕНТЫ КОМПАНИИ «МАРК АВРЕЛИЙ»



РУССКИЕ БАШНИ





## ПРИЛОЖЕНИЯ

---



MARCUS AURELIUS LTD

# КАРТОЧКА ИНФОРМАЦИОННОЙ СИСТЕМЫ



	Название поля
1	ID системы
2	Полное наименование системы
3	Краткое имя для схем
4	Краткое описание системы
5	Место текущей физической установки ИС
6	Уровень унификации решения
7	Количество установок
8	Аналитик
9	Класс ИС
10	Целевая архитектура (признак)
11	Ответственный департамент в УК
12	Локальный разработчик ИС
13	Вендор ИС
14	Интегратор ИС

	Название поля
15	Язык разработки ИС или СУБД
16	Оценка имеющейся документации
17	Штат специалистов по ИС
18	Статус реализации ИС: <ul style="list-style-type: none"><li>• промышленная эксплуатация</li><li>• архивная система</li><li>• система на стадии внедрения</li><li>• выведена из эксплуатации</li></ul>
19	Куда выгружается отчетность?
20	Ответственное лицо по ИС <ul style="list-style-type: none"><li>• ФИО</li><li>• Телефон/Email</li><li>• Подразделение ответственного лица</li></ul>
21	Контактное лицо по ИС <ul style="list-style-type: none"><li>• ФИО</li><li>• Телефон/Email</li></ul>
22	Заказчик ИС <ul style="list-style-type: none"><li>• ФИО</li><li>• Телефон/Email</li><li>• Подразделение ответственного лица</li></ul>

# IT-ЛАНДШАФТ: ФОРМУЛЯР ИНФОРМАЦИОННОЙ СИСТЕМЫ



Element - Модуль по работе с дебиторами (МРД)

General Information Items Connectors Instances Custom Attributes

Name  
Модуль по работе с дебиторами (МРД)

Symbol  
MO131444484

Element Type  
Информационная система

Modeling direction  
Horizontal

Description  
Модуль автоматизации процессов по взысканию дебиторской задолженности. Замещаемая ИС. В настоящее время ведется централизованный проект по внедрению ЕССДЗ

OK Cancel Delete Help

Общее описание системы

Общее описание системы

Связи системы

Вхождения системы в различные диаграммы

Атрибуты системы: СУБД, Вендор, Статус и т.п.

# ПОСТРОЕНИЕ СВЯЗЕЙ: ФУНКЦИИ И ИНФОПОТОКИ



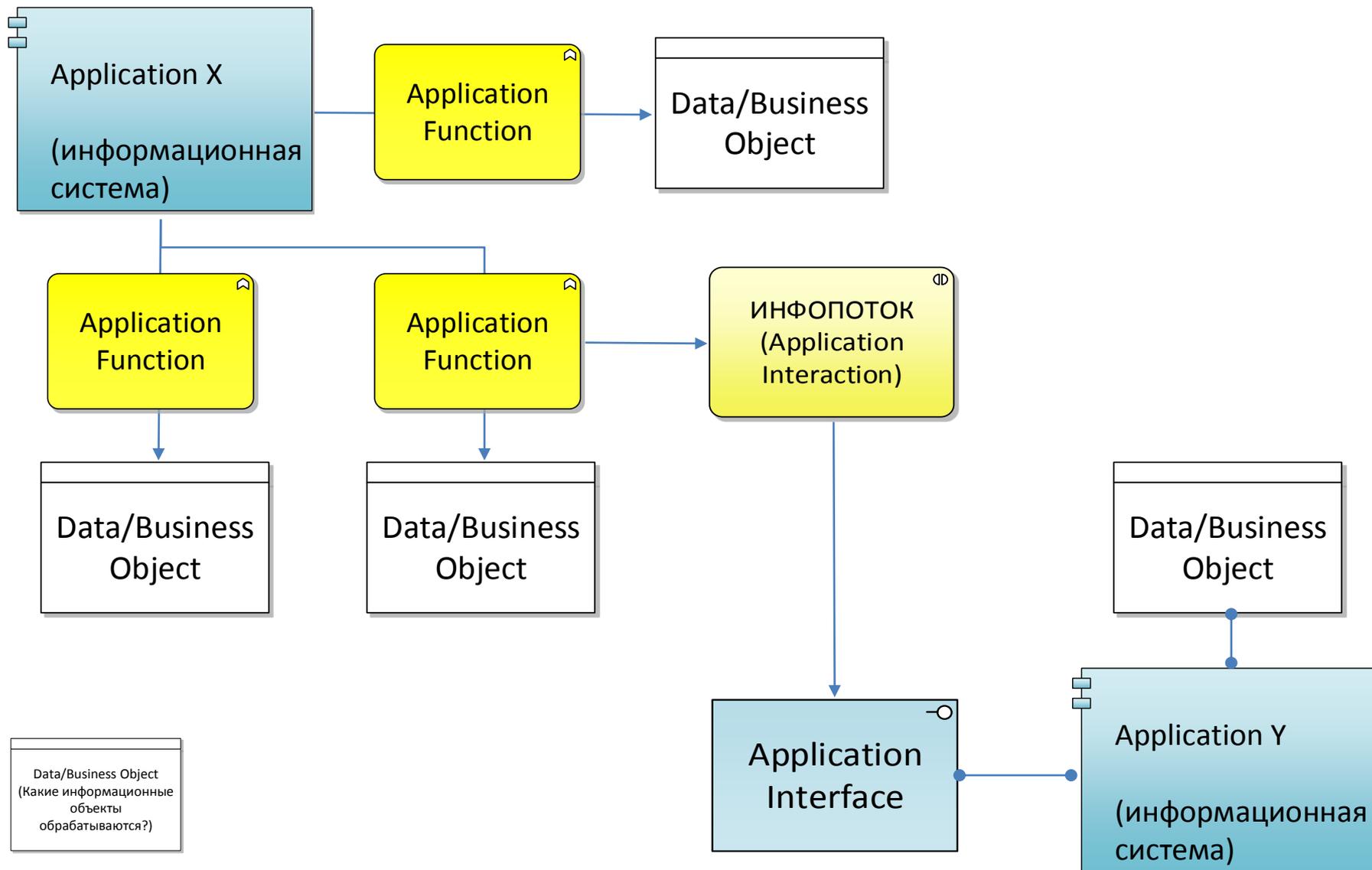
01. Системы, их функции и интеграции

Название ИС	AC ID	Short Name	Класс системы
Единая база знаний (ЕБЗ)	КЦ.038	ЕБЗ	ИСО и ЛК
IS.ЕБЗ (МРФ КЦ)			
Авторизация внешних пользователей			
Авторизация внешних пользователей			
Загрузка дополнительной информации о тарифах			
Загрузка дополнительной информации о тарифах			
переход на страницу Базы знаний			
переход на страницу Базы знаний			
Модуль обучения и тестирования пользователей			
Поиск курсов по заданным параметрам (КЦ.038.ОФ.014)	КЦ.038.ОФ.014		
Просмотр статистики тестирования пользователей (КЦ.038.ОФ.013)	КЦ.038.ОФ.013		
Прохождение обучения в МОТП (КЦ.038.ОФ.015)	КЦ.038.ОФ.015		
Публикация учебных курсов (КЦ.038.ОФ.010)	КЦ.038.ОФ.010		
Создание теста для работников продаж и обслуживания (КЦ.038.ОФ.011)	КЦ.038.ОФ.011		
Создание учебных курсов для работников продаж и обслуживания (КЦ.038.ОФ.009)	КЦ.038.ОФ.009		
Управление обучением пользователей (КЦ.038.ОФ.012)	КЦ.038.ОФ.012		
Формирование отчетов о прохождении обучения (КЦ.038.ОФ.016)	КЦ.038.ОФ.016		
Ядро системы			
Блокировка/разблокировка абонента (СЗ.170-01.ОФ.004)	СЗ.170-01.ОФ.004		
ведение географических расположении пунктов сети (СЗ.190.ОФ.004)	СЗ.190.ОФ.004		
Ведение заявок на материалы (СЗ.172.ОФ.002)	СЗ.172.ОФ.002		

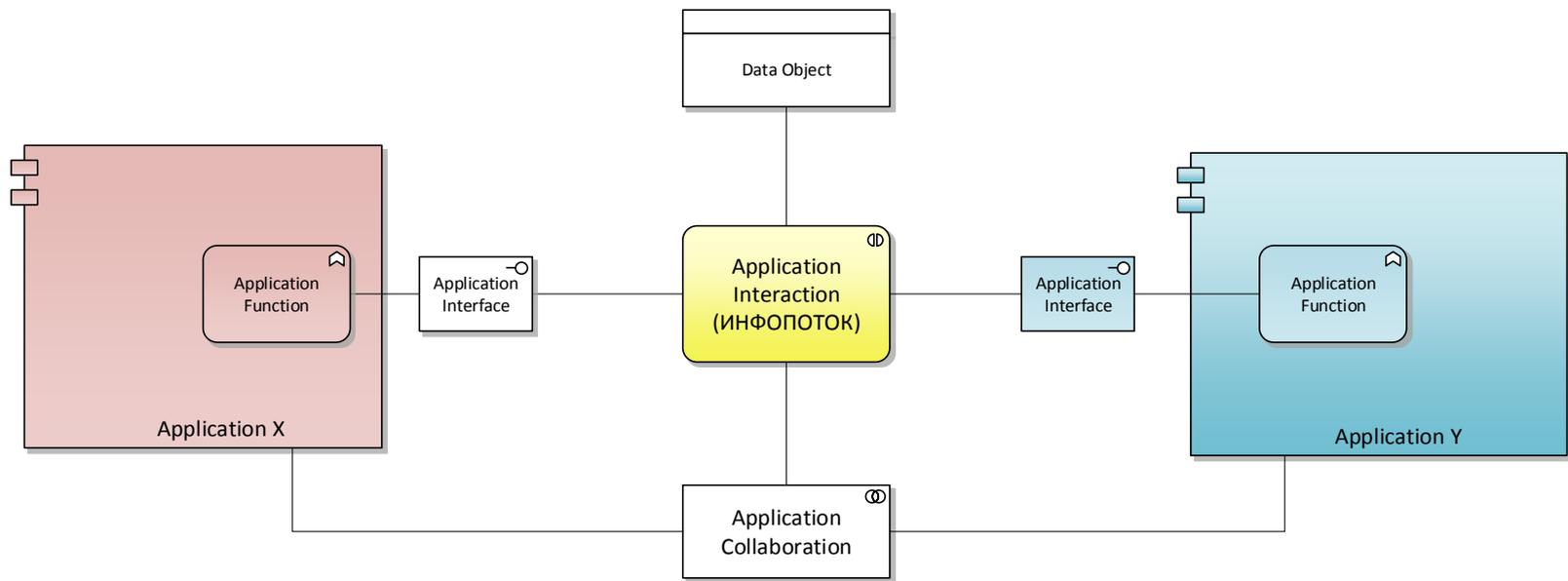
ИНФОПОТОКИ

ФУНКЦИИ СИСТЕМЫ

СВЯЗИ МОЖНО УСТАНОВЛИВАТЬ КАК ГРАФИЧЕСКИ, ТАК И ТАБЛИЧНО



# МЕТАМОДЕЛЬ ИНТЕГРАЦИОННЫХ ВЗАИМОДЕЙСТВИЙ



Приложение X в контексте выполнения своей функции вызывает приложение Y через интерфейс, который предоставляет приложение Y. При этом возникает поток данных, логически связанный с одним из объектов информационной модели предприятия.

# ФОРМУЛЯР ИНФОПОТОКА:

## ОПИСАНИЕ ИНТЕГРАЦИОННОГО ВЗАИМОДЕЙСТВИЯ ДВУХ СИСТЕМ

	Название поля	Тип поля
1	Номер инфопотока	Текст
2	Краткое наименование инфопотока	Текст
3	Полное наименование инфопотока	Текст
4	Описание инфопотока или комментарии по реализации	Текст
5	Активный? (используется или нет)	Boolean
6	Статус описания (в работе; описание завершено; заблокировано от изменений)	Список выбора
7	Ответственный аналитик	Список выбора
8	Вызывающая система	Ссылка
9	Вызываемая система	Ссылка
10	Состав передаваемых данных или название запрошенной функции	Текст
11	Релевантные информационные объекты	Ссылка
12	Наименование API и метода, предоставляемого вызываемой системой	Текст
13	Метод интеграции	Список выбора
14	Протокол интеграции	Список выбора

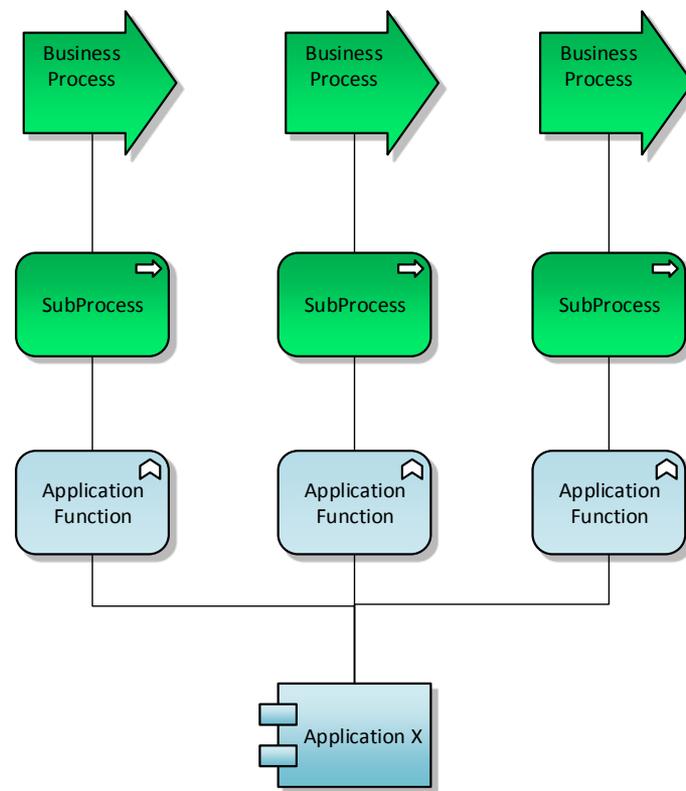
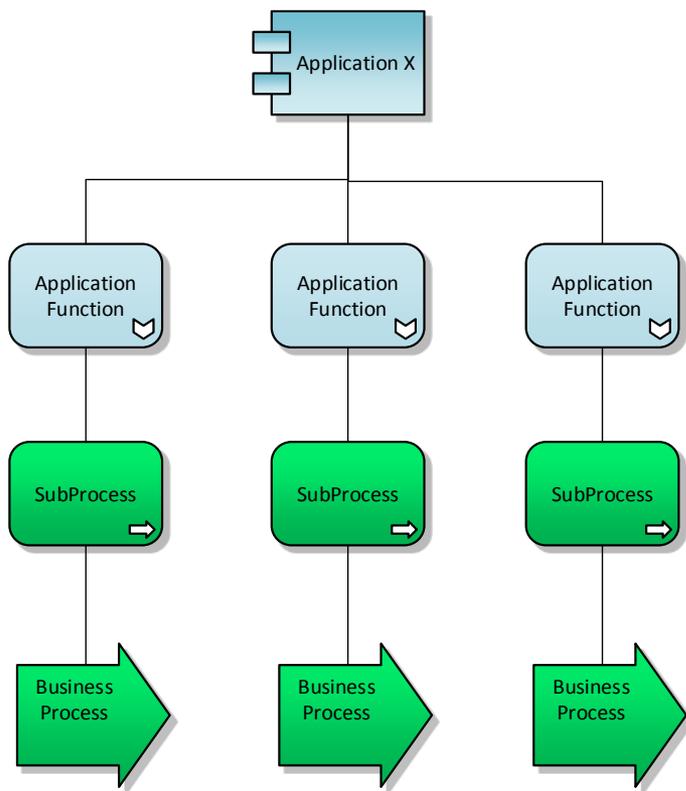
# КАРТОЧКА ИНФОПОТОКА (ПРОДОЛЖЕНИЕ)



	Название поля	Тип поля
14	Иницилирующее событие в системе №1	Текст
15	Функция системы №1 (вызывающей системы), в контексте которой происходит вызов системы 2	Ссылка
16	Функция системы №2 (вызываемая система), в контексте которой происходит «прием» инфопотока или собственно вызываемая функция	Текст
17	Интегратор	Текст
18	Контакты интегратора	Текст
19	ИТ-специалист, ответственный за интеграцию	Текст
20	Документация, описывающая интеграцию	Текст

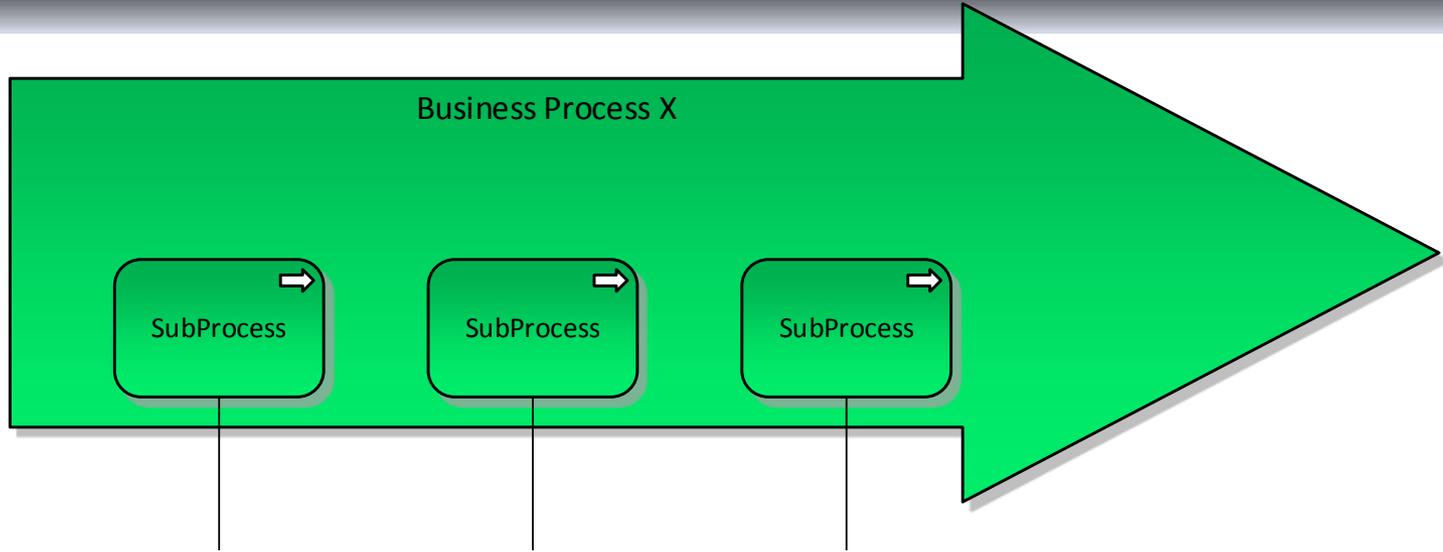
# ВЗАИМОСВЯЗЬ ГРУППЫ СИСТЕМ

## В КОНТЕКСТЕ ОДНОГО ПРОЦЕССА

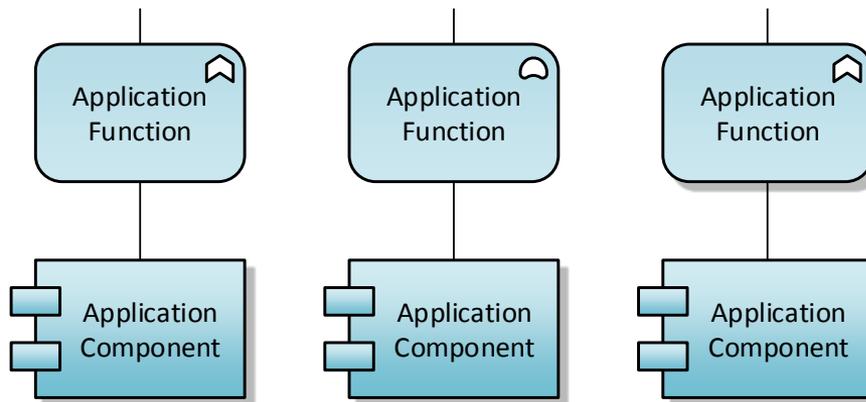


ДАННЫЕ ДИАГРАММЫ ИЛЛЮСТРИРУЮТ, КАКИЕ БИЗНЕС-ПРОЦЕССЫ ЗАВИСЯТ ОТ СИСТЕМЫ ИЛИ В КАКИХ БИЗНЕС-ПРОЦЕССАХ И КАКИМИ СВОИМИ ФУНКЦИЯМИ СИСТЕМА УЧАСТВУЕТ.

# ЗАВИСИМОСТЬ ПРОЦЕССА ОТ СИСТЕМ

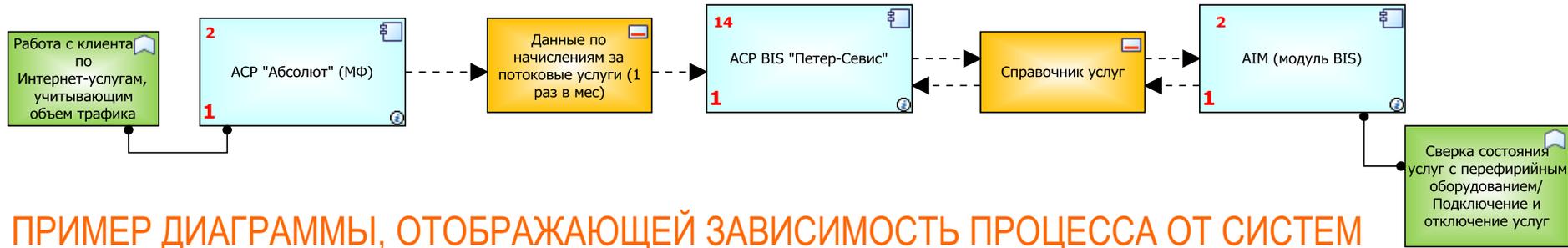


НА КАКИЕ СИСТЕМЫ ОПИРАЕТСЯ БИЗНЕС-ПРОЦЕСС? ОТ КАКИХ СИСТЕМ ОН ЗАВИСИТ?



# ПОСТРОЕНИЕ СВЯЗЕЙ: ФУНКЦИИ И ДАННЫЕ

## ПРИМЕР ДИАГРАММЫ, ОТРАЖАЮЩЕЙ СВЯЗЬ ПРИЛОЖЕНИЙ ПО ДАННЫМ



## ПРИМЕР ДИАГРАММЫ, ОТОБРАЖАЮЩЕЙ ЗАВИСИМОСТЬ ПРОЦЕССА ОТ СИСТЕМ

