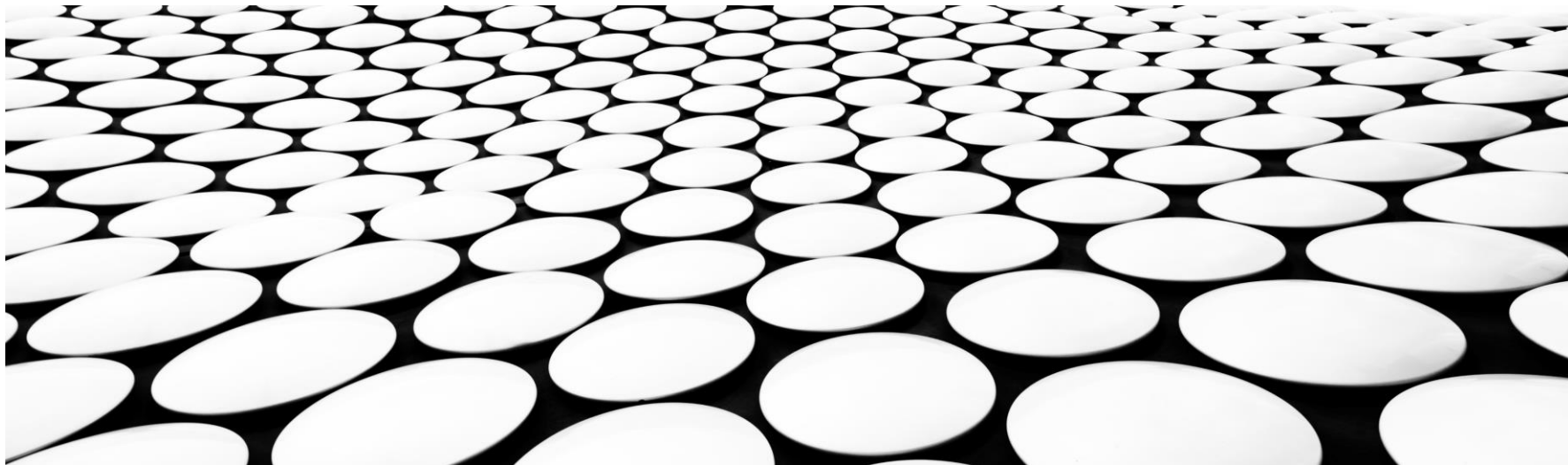
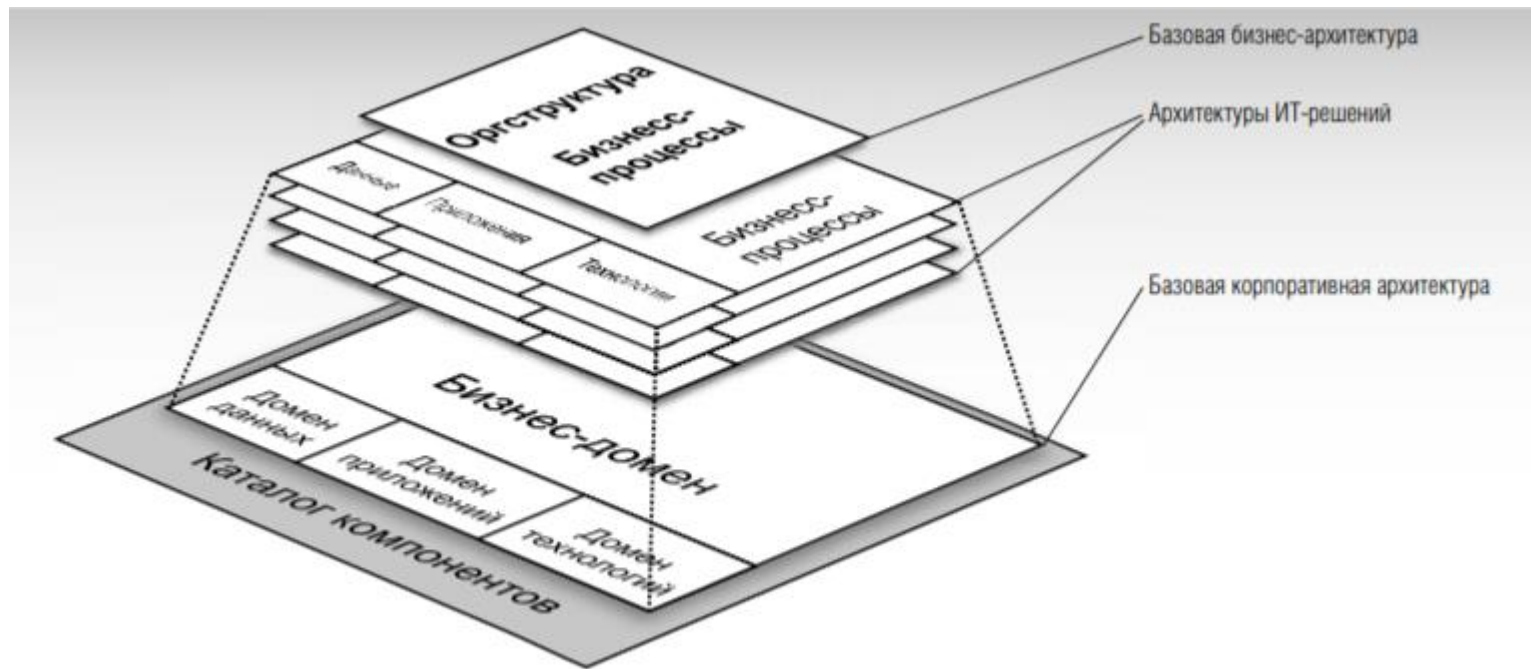


# ИТ-АРХИТЕКТУРА

## ЛЕКЦИЯ 4 И 5



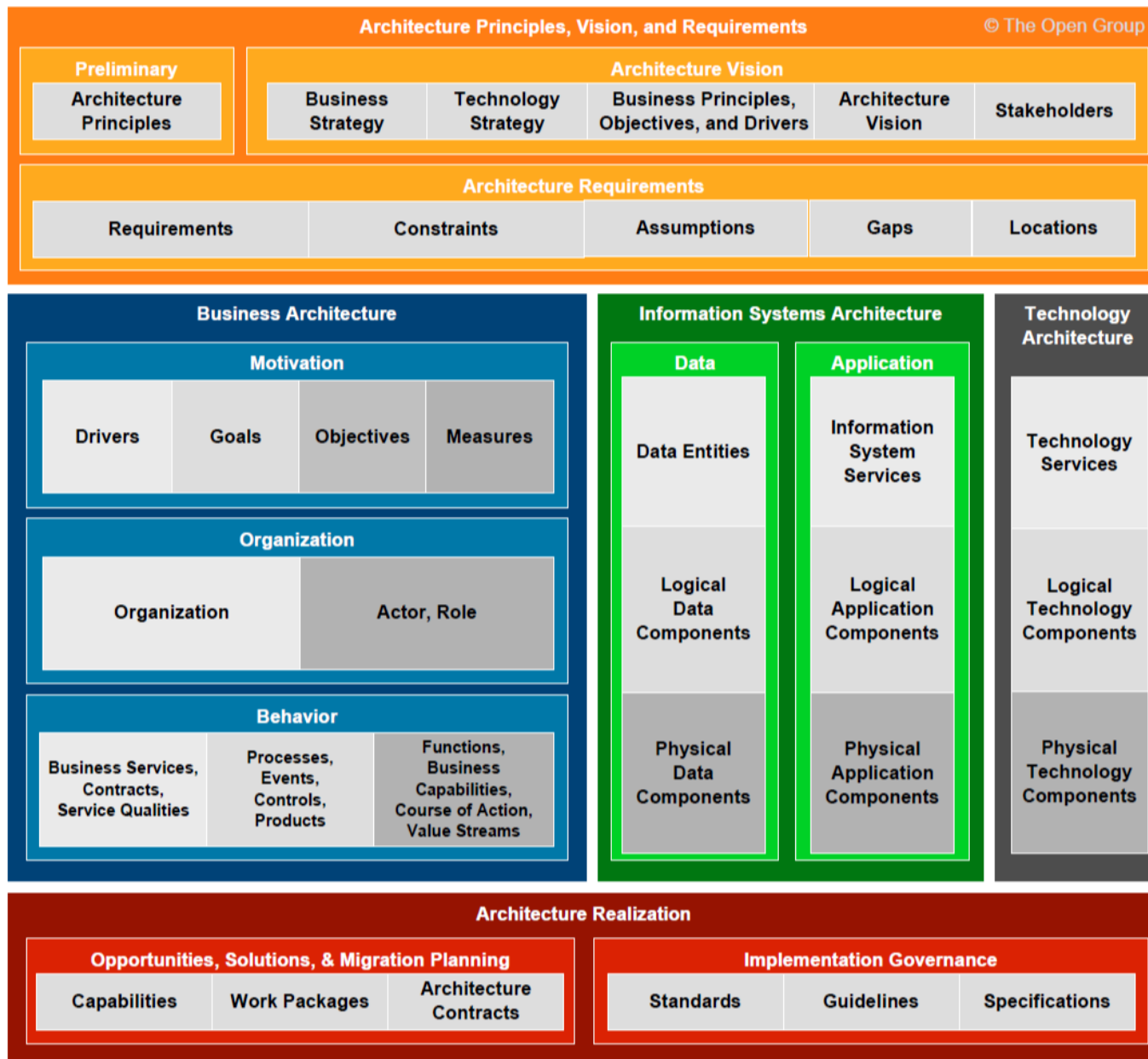


## ИТ-АРХИТЕКТУРА

# ИТ- АРХИТЕКТУРА

TOGAF

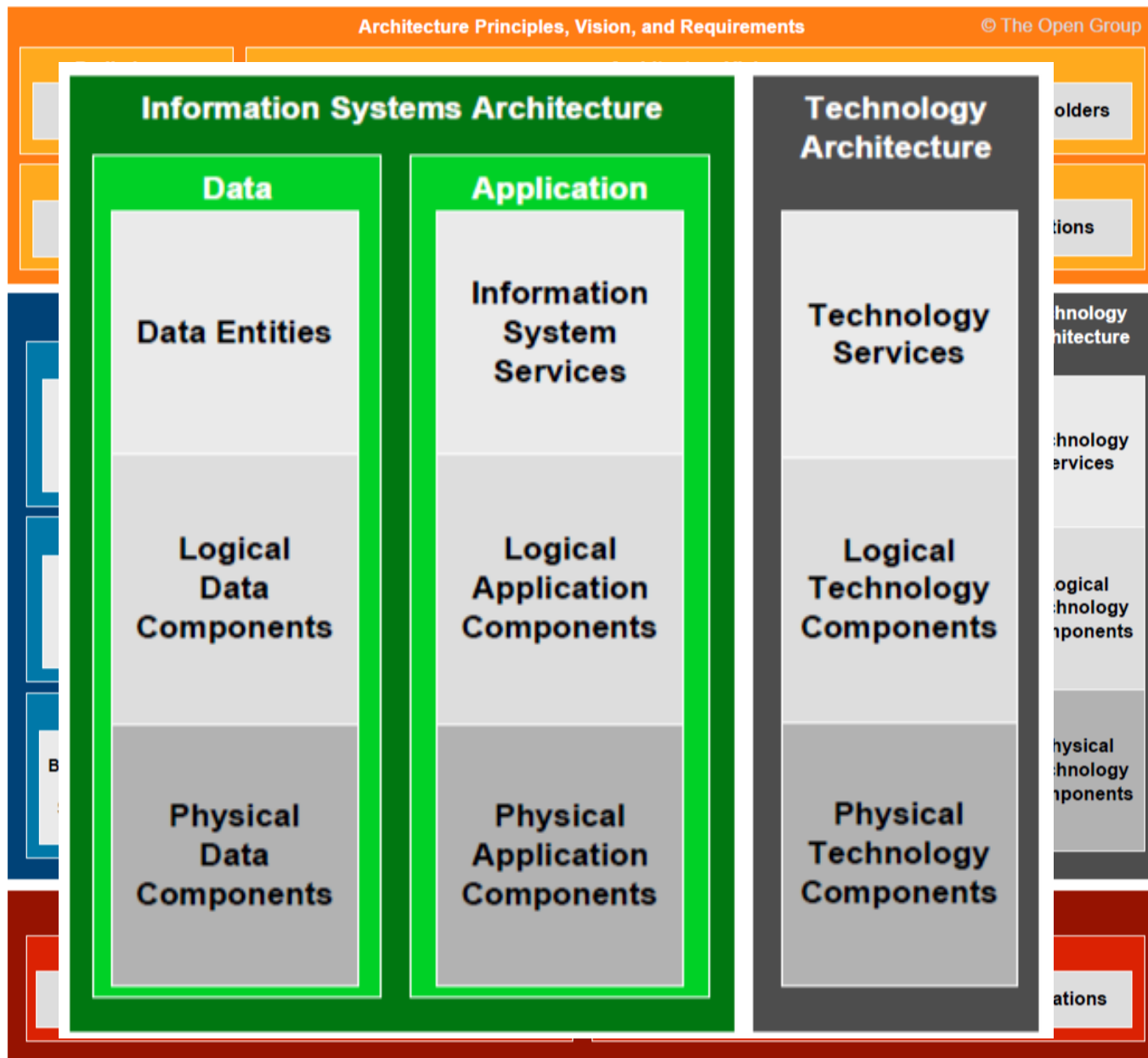
структура ИТ-компонентов компании, их взаимодействия, а также принципов и руководств по управлению их разработкой и развитием

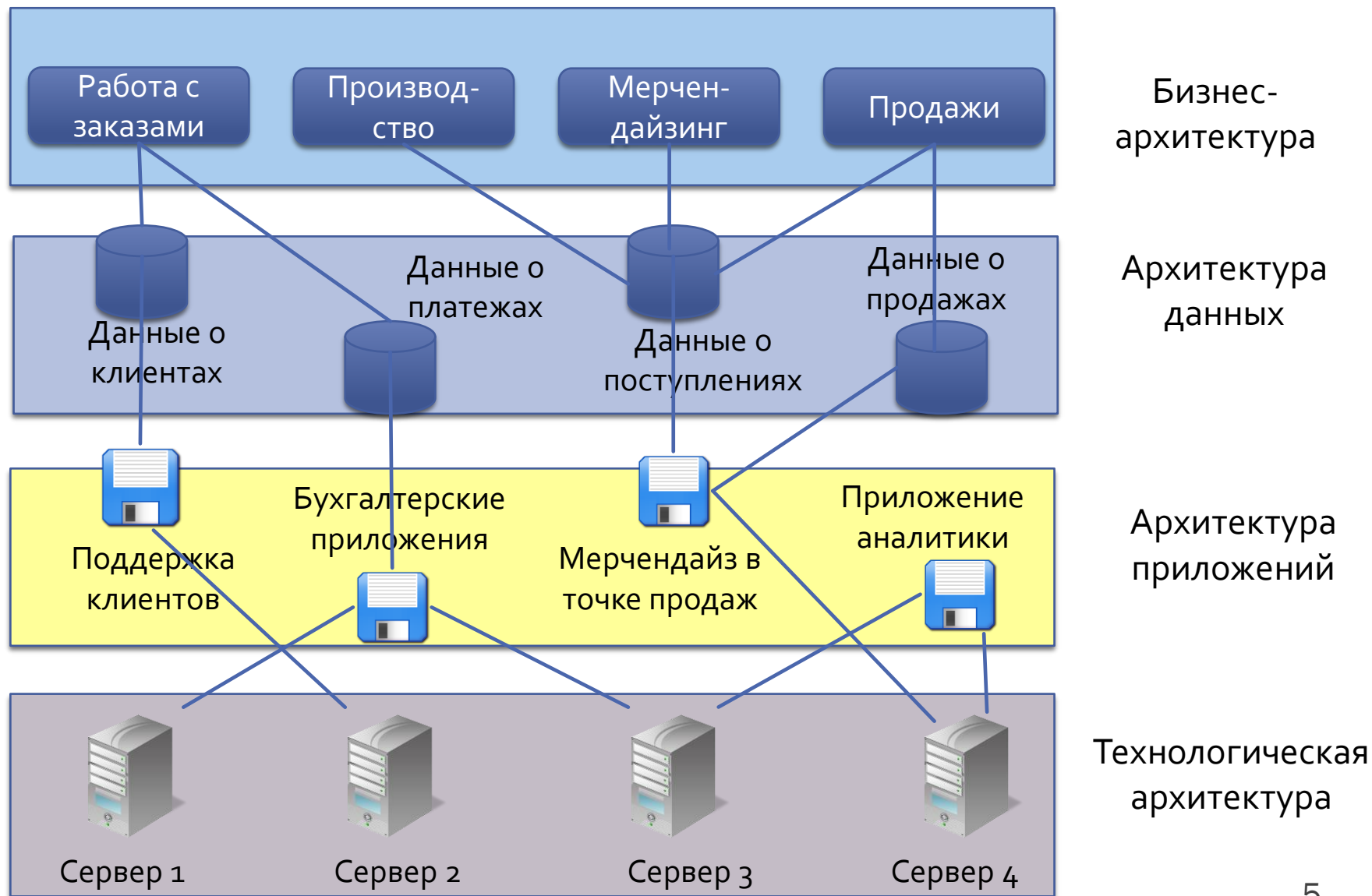


# ИТ- АРХИТЕКТУРА

TOGAF

структура ИТ-компонентов компании, их взаимодействия, а также принципов и руководств по управлению их разработкой и развитием







# АРХИТЕКТУРА ДАННЫХ



# АРХИТЕКТУРА ДАННЫХ

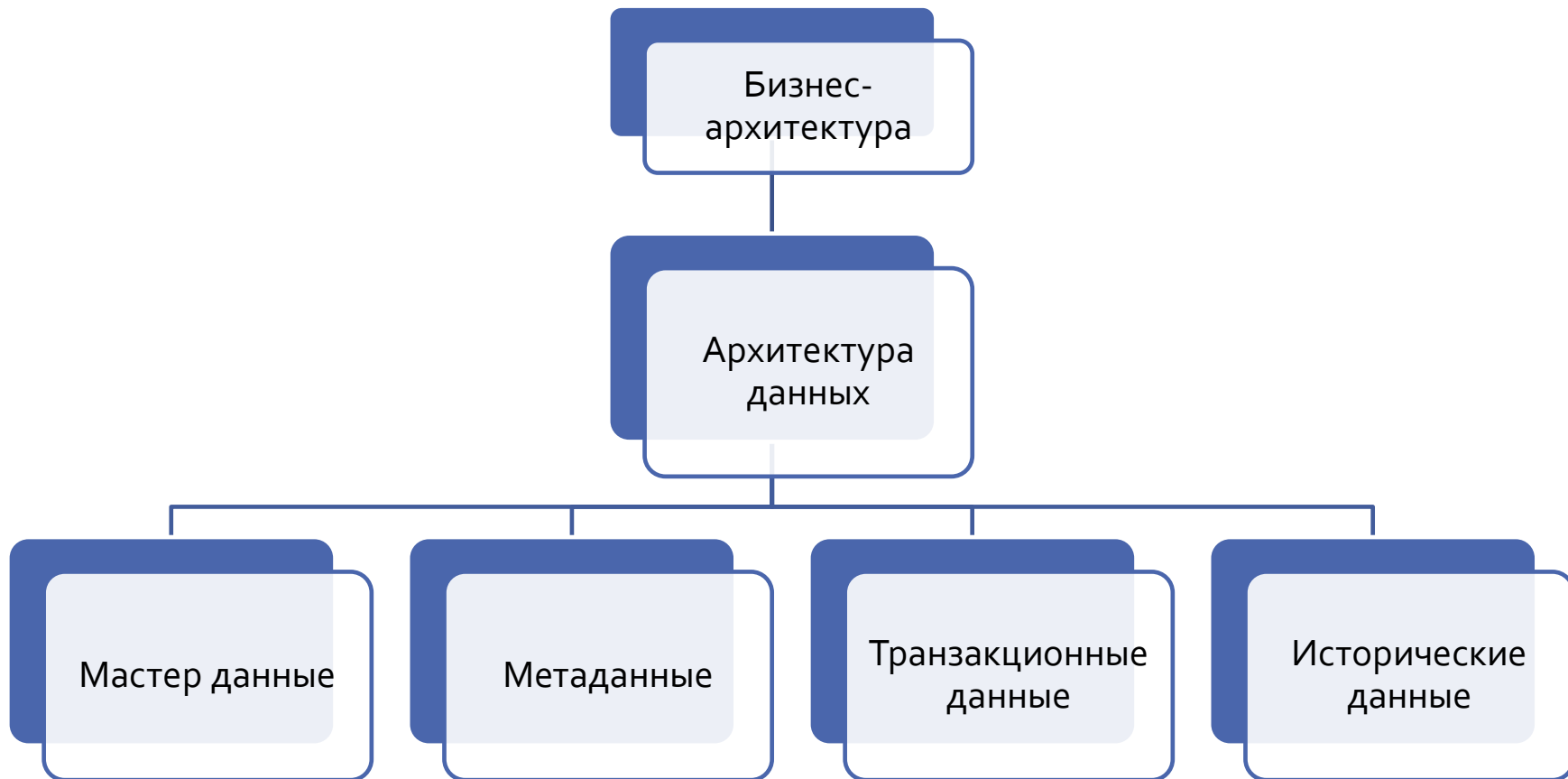
**Архитектура данных описывает данные, используемые предприятием**

**Выделяется три элемента архитектуры данных:**

- Структура данных
- Требования к переносу данных
- Требования по управлению данными



# ТИПЫ ДАННЫХ





# ТИПЫ ДАННЫХ



## ТРЕБОВАНИЯ К УПРАВЛЕНИЮ ДАННЫМИ

Этот элемент описания архитектуры данных описывает ресурсы, которые есть на предприятии для проведения преобразований данных.

### Структура

- организационная структура и органы стандартизации управления преобразованием данных

### Люди

- навыки и роли, которые нужны для преобразования данных

### Системы управления данными

- на протяжении их жизненных циклов

### Требования к безопасности данных

- иногда отдельно выделяется в архитектуре

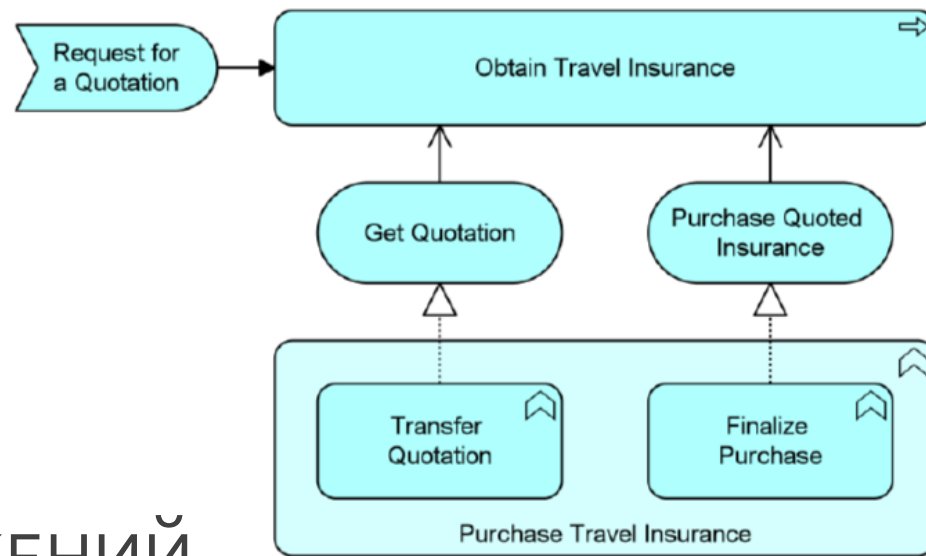
## ЦЕЛИ РАЗРАБОТКИ АРХИТЕКТУРЫ ДАННЫХ

- **Интеграция метаданных** для целостного представления данных из различных источников
- **Сокращение избыточности и фрагментарности** данных для уменьшения затрат на:
  - хранение,
  - стоимости обслуживания
  - повышения качества данных
- **Улучшение качества** данных за счет привлечения бизнес-пользователей к определению данных
- **Улучшение защиты** данных за счет последовательных мер по защите и использованию данных

# АРТЕФАКТЫ АРХИТЕКТУРЫ ДАННЫХ

- **Каталог компонентов данных**
- **Матрицы:**
  - Объект данных / бизнес-функция (показывает, какие данные поддерживают какие функции и какой бизнес-функции принадлежит какие данные)
  - Бизнес-сервис / информация (разработано на этапе бизнес-архитектуры, здесь актуализировано)
  - Приложение / Данные
- **Диаграммы:**
  - Схема концептуальных данных
  - Диаграмма логических данных
  - Диаграмма распространения данных
  - Диаграмма жизненного цикла данных
  - Диаграмма безопасности данных
  - Диаграмма переноса данных

# АРХИТЕКТУРА ПРИЛОЖЕНИЙ



# АРХИТЕКТУРА ПРИЛОЖЕНИЙ

- Описывает приложения, с помощью которых осуществляется автоматизация деятельности организации (бизнес-архитектура) и обработка потоков информации (архитектура данных)

## TOGAF

- **Цель** в архитектуре приложений - определить **основные виды прикладных систем**, необходимых для обработки данных и поддержки бизнеса

## FEA

- Архитектура приложений «определяет приложения, необходимые **для управления данными и поддержки бизнес-функций**»



## ЦЕЛИ АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ

- Разработать **целевые архитектуры ИС**, описывающие, как архитектура информационных систем предприятия позволит реализовать:
  - бизнес-архитектуру
  - концепцию архитектуры таким образом
  - чтобы соответствовать интересам заинтересованных сторон
- Определить **подходящие компоненты «дорожной карты»** архитектуры на основе различий между базовой и целевой архитектурами ИС

## ЭЛЕМЕНТЫ ПРИКЛАДНОЙ АРХИТЕКТУРЫ

- Классификация приложений и/или предоставляемых ими сервисов
- Выделение и описание основных классов приложений или групп сервисов
- Привязка классов приложений к основным бизнес-процессам организации
- Описание взаимодействия и взаимозависимости корпоративных прикладных программ
- Определение функциональных требований: уровень использования и критичность для компании, обрабатываемые данные и поддерживаемые протоколы, требования по надёжности и интенсивности использования
- Приоритеты в совершенствовании и развитии, приобретении и создания ПП

### Принцип построения архитектуры приложений

В виде классов приложений

В виде конкретных систем



# СЕРВИСНО-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА (SERVICE ORIENTED ARCHITECTURE, SOA)

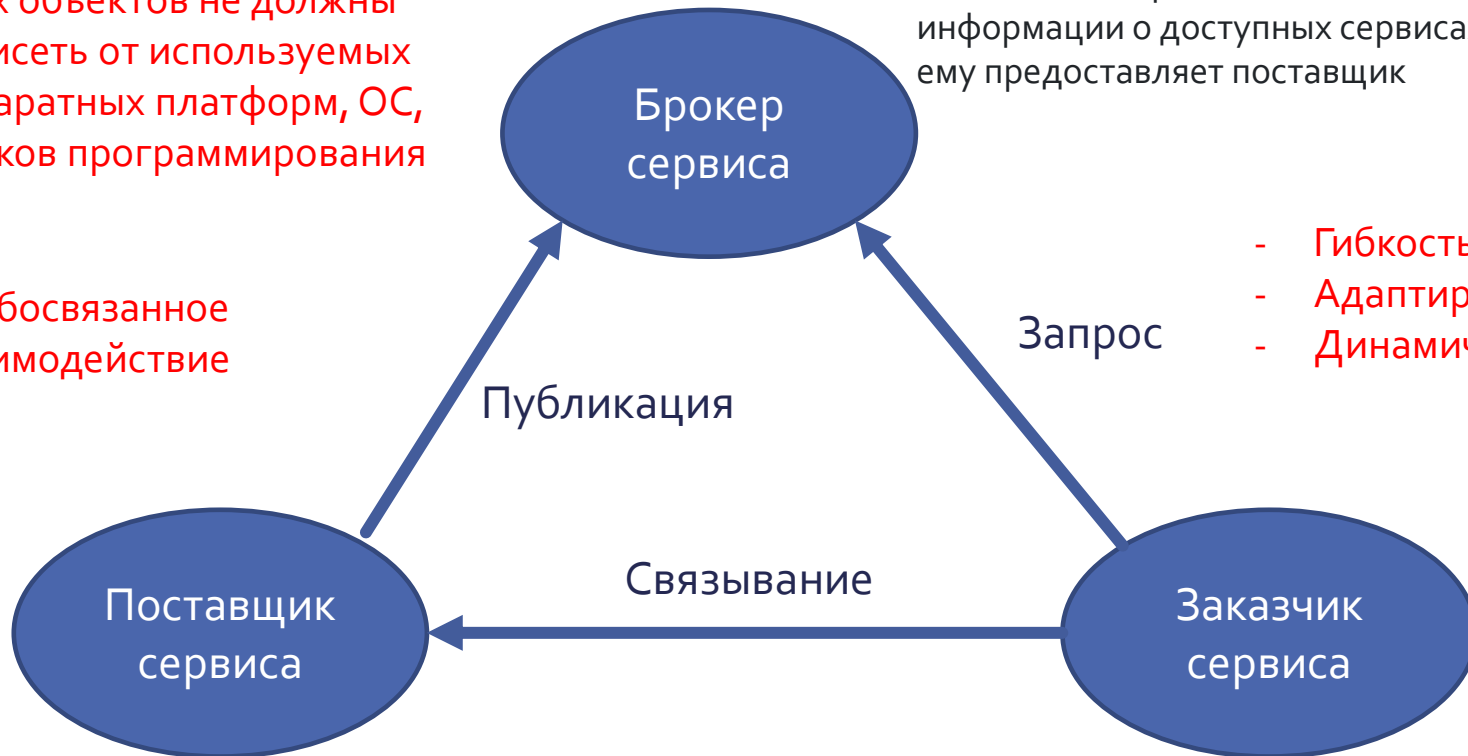
- **SOA** — это стандарт архитектуры предприятия, прежде всего, архитектуры приложений, который позволяет **решить проблемы интеграции** разнообразных программных систем в единую программную среду предприятия
- Стандарт основан на понятии **сервиса, т.е. удовлетворения потребностей одного объекта другим.**
- Gartner: **SOA— это подход к проектированию прикладных информационных систем, который руководствуется следующими принципами:**
  - явное отделение бизнес-логики прикладной системы от логики презентации информации
  - реализация бизнес-логики прикладной системы в виде некоторого количества программных модулей (сервисов), которые доступны потребителям в режиме «запрос-ответ» через чётко определённые формальные интерфейсы доступа
  - потребитель сервиса может быть прикладной системой или другим сервисом и вызывает сервис, используя соответствующие коммуникационные механизмы
- **SOA – эталонная архитектура модели облачных вычислений**

# ЛЮБОЙ КОМПОНЕНТ SOA предоставляет свою функциональность другим компонентам на основании ПРОСТЫХ ПРАВИЛ

Интерфейсы взаимодействия этих объектов не должны зависеть от используемых аппаратных платформ, ОС, языков программирования

Слабосвязанное взаимодействие

Связывает заказчика и поставщика, на основании запроса от заказчика и информации о доступных сервисах, которые ему предоставляет поставщик



- Гибкость
- Адаптируемость
- Динамичность

Может предоставить запрошенный функционал

Хочет получить определенный функционал, не зная, кто может его предоставить

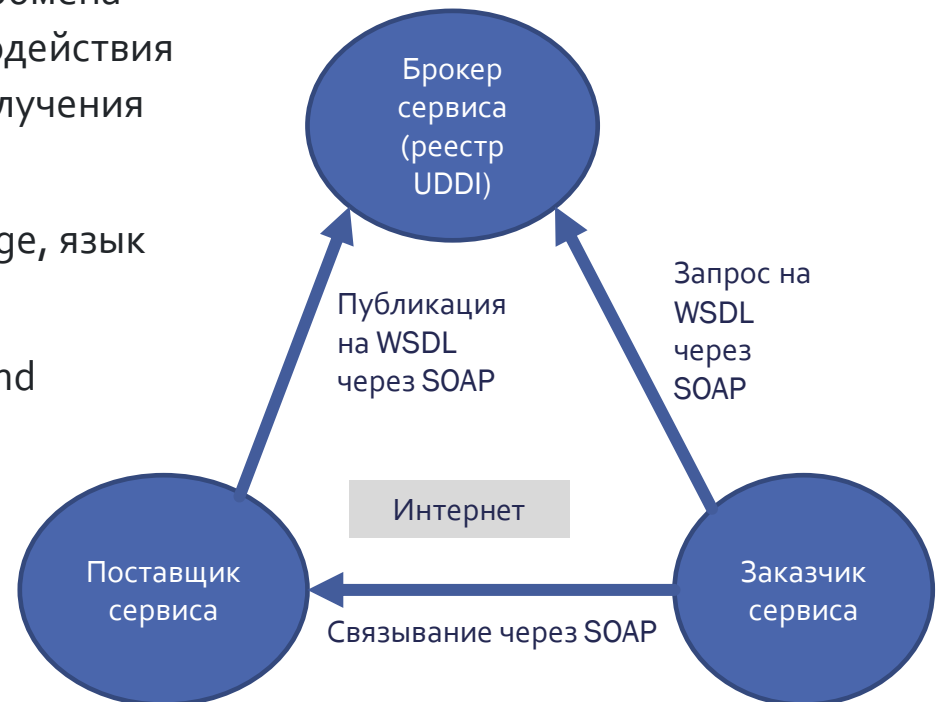
## WEB-СЕРВИСЫ, КАК ПОДТИП SOA

2004 г., IBM и Microsoft

Три стандарта web-сервисов:

- **SOAP** — Simple Object Access Protocol, простой протокол доступа к объектам; протокол обмена сообщениями, используемый для взаимодействия между участниками предоставления / получения сервиса
- **WSDL** — Web Services Description Language, язык описания веб-сервиса и доступа к нему
- **UDDI** — Universal Discovery, Description and Integration, универсальный интерфейс распознавания, описания и интеграции;

каталог веб-сервисов и сведений о компонентах, предоставляющих веб-сервисы, описанных на WSDL



## MDA – MODEL DRIVEN ARCHITECTURE АРХИТЕКТУРА, УПРАВЛЯЕМАЯ МОДЕЛЯМИ

- Архитектура, управляемая моделями – стандарт архитектуры приложений
- Может использоваться одновременно в SOA
- 2001 г., Международный консорциум OMG (Object Management Group)
- Назначение – проектирование программных систем
- состоит из спецификаций, написанных в основном на UML, и руководств по их использованию
- Особенность – система может строиться как последовательность взаимосвязанных моделей, которые должны последовательно детализировать систему, начиная с независимой от технологий модели CIM, через платформенно-независимую модель PIM, затем – платформенно-зависимую модель PSM и к конкретной программной реализации
- Стандарт MDA содержит правила перехода CIM – PIM – PSM для некоторых платформ, таких как: CORBA, J2EE, .NET

# АРХИТЕКТУРА, УПРАВЛЯЕМАЯ МОДЕЛЯМИ (MDA)



# АРТЕФАКТЫ АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ

## ■ Каталоги:

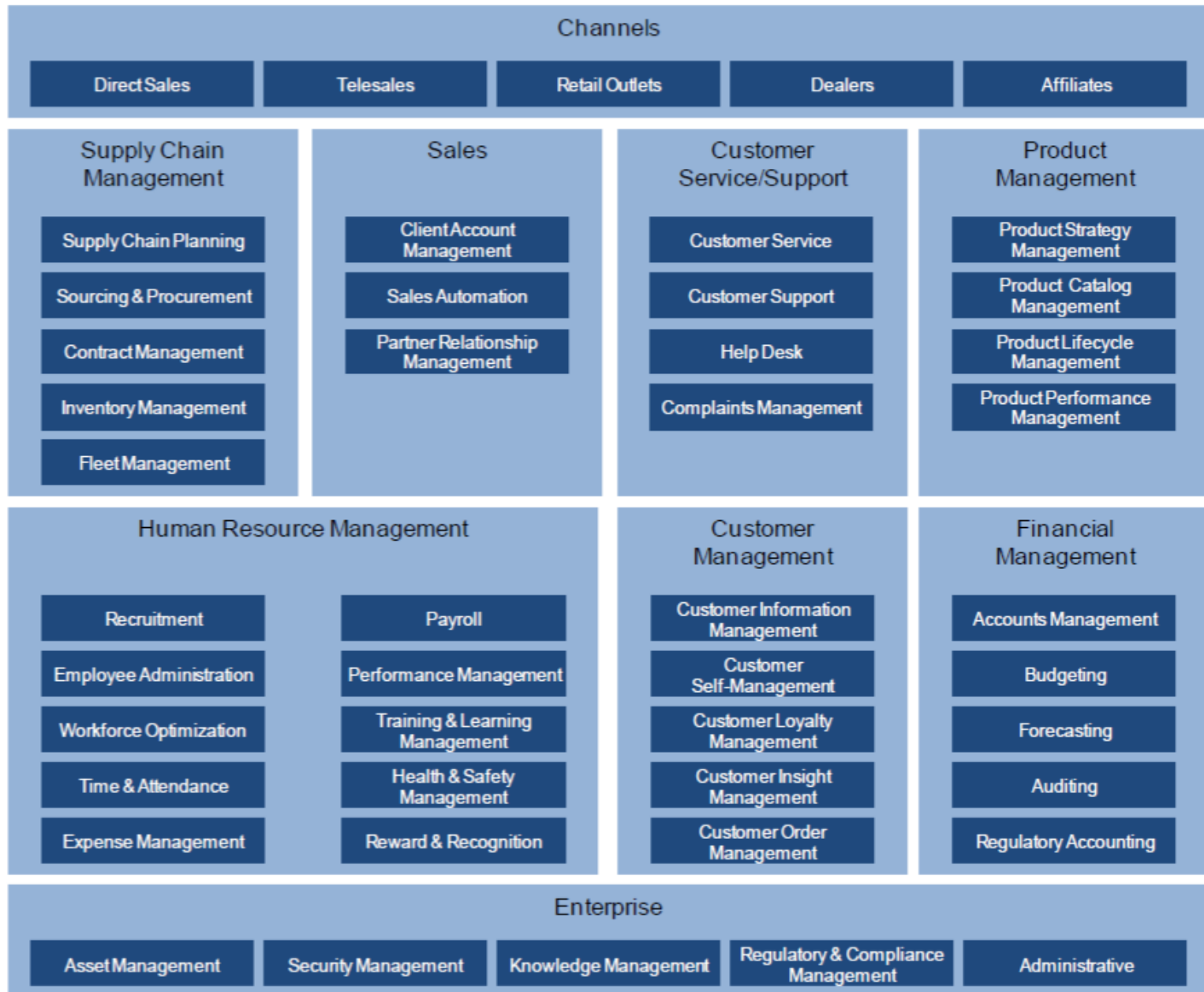
- Портфолио приложений
- Реестр информационных систем
- Каталог интерфейсов

## ■ Матрицы:

- Приложение / Организационная матрица
- Роль / матрица приложений
- Матрица взаимодействия приложений
- Приложение / Функциональная матрица

## ■ Диаграммы:

- Диаграмма связей приложений
- Схема расположения приложения и пользователя
- Диаграмма реализации процесса / приложения
- Диаграмма миграции приложений
- Схема распространения программного обеспечения
- Схема разработки программного обеспечения
- Диаграмма вариантов использования приложения



# КЛАССИФИКАЦИЯ ПРИЛОЖЕНИЙ В ПОРТФЕЛЕ

## ■ По глубине:

- покрытие всех областей АЕ
- Обеспечивается прослеживаемость изменений в приложениях с бизнес-операциями (возможности, сервисы, объекты, роли) и мотивацией (цели, драйверы, задачи)

## ■ По ширине, используя следующие характеристики:

- Бизнес-сфера / бизнес-возможности
- Критичность для предприятия
- Взаимодействие между приложениями
- Общность базовой технологии и / или общей инфраструктуры
- Географическое положение
- Или любое сочетание этих критериев

## ■ По критичности:

- Сумма полученного дохода
- Сумма защищенного дохода
- Важность безопасности для людей, активов и других ресурсов
- Соответствие нормативным требованиям и обязательная отчетность
- Безопасность
- Любая комбинация вышеперечисленных мер



## КЛАССИФИКАЦИЯ «THE 5RS»

Действие	Ценность для бизнеса	Состояние
Удалить (Retire)	Низкая	Плохое
Перенести, заменить, например на SaaS (Replace)	Низкая	Плохое
Перенести на другую платформу, например PaaS или IaaS (Re-platform)	Низкая	Хорошее
Перестроить приложение (Re-architect)	Высокая	Плохое
Расширение, развитие (Refactor)	Высокая	Хорошее

## МАТРИЦА ОЦЕНКИ ПРИКЛАДНЫХ СИСТЕМ

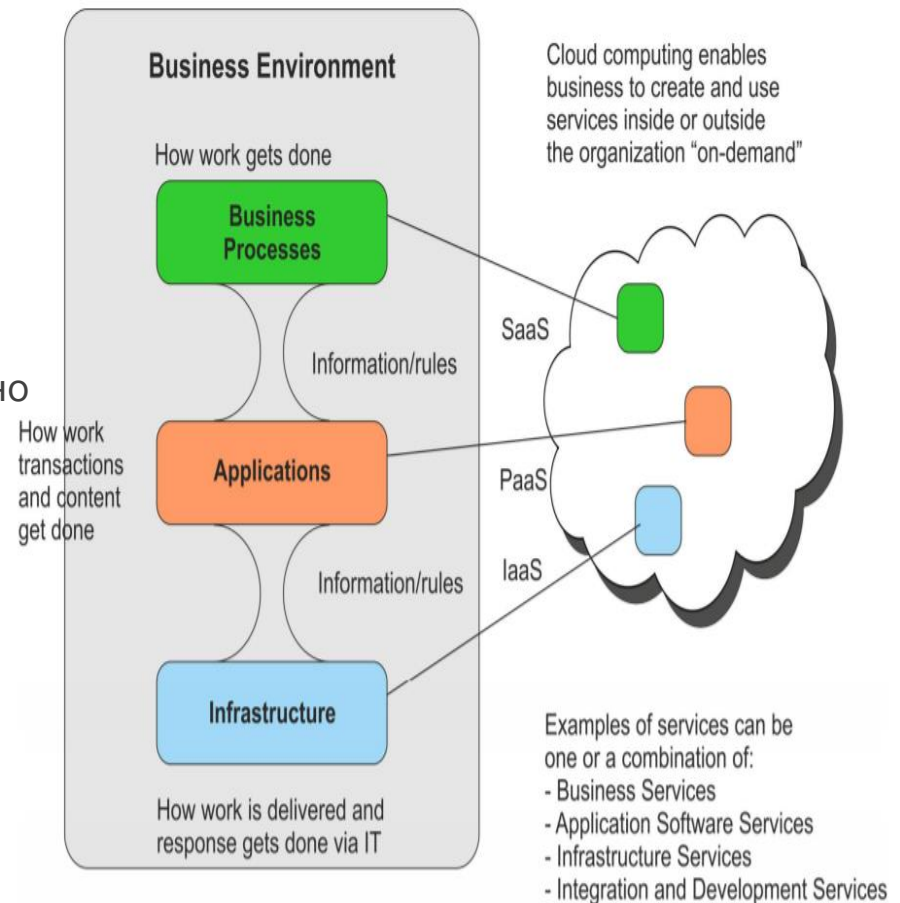
<b>Превосходное техническое состояние</b>	<b>Перепозиционирование и оценка</b>  Недавно введенные в эксплуатацию ИС, которые не достигли поставленных целей	<b>Обеспечение сопровождения и развития</b>  Самые перспективные системы. Критически важны для успеха бизнеса
<b>Неудовлетворительн ое техническое состояние</b>	<b>Вывод из эксплуатации / замена / консолидация</b>  Ситуация наличия унаследованных систем	<b>Обновление инфраструктуры прикладной системы</b>  Возможен постепенный переход на более современные решения
	<b>Низкая ценность для бизнеса</b>	<b>Высокая ценность для бизнеса</b>



# ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ

# ОПРЕДЕЛЕНИЕ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

- **Облачные вычисления** – информационно-технологическая концепция, которая подразумевает удобный сетевой доступ по требованию к общему пулу конфигурируемых вычислительных ресурсов, причем ресурсы могут оперативно предоставляться и освобождаться с мин. затратами
- 5 основных характеристик
- 3 модели обслуживания
- 4 модели развертывания



# ХАРАКТЕРИСТИКИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

## Самообслуживание по запросу

- потребитель может в одностороннем порядке использовать вычислительные возможности, например в качестве серверного времени и сетевого хранилища по мере необходимости автоматически

## Широкий сетевой доступ

- возможности доступны по сети и доступны через стандартные механизмы, которые способствуют использованию разнородными платформами тонких или толстых клиентов (мобильные телефоны, ноутбуки, ...)

## Объединение ресурсов

- вычислительные ресурсы (хранилище, обработка, память, пропускная способность сети и виртуальные машины) поставщика объединены для обслуживания нескольких потребителей, при этом использующие разные физические и виртуальные ресурсы, которые динамически назначаются в соответствии со спросом. Клиент, как правило, не контролирует или не знает точное местоположение предоставленных ресурсов, но может указать местоположение на более высоком уровне абстракции (страна, штат или ЦОД).

## Быстрая эластичность

- в некоторых случаях возможности могут быть быстро и эластично предоставлены автоматически

## Измеряемый сервис

- облачные системы автоматически контролируют и оптимизируют использование ресурсов путем измерения на некотором уровне абстракции, соответствующем типу услуги (например, хранилище, обработка, пропускная способность и активные учетные записи пользователей) Так обеспечивается прозрачность как для поставщика, так и для потребителя услуг

# МОДЕЛИ РАЗВЕРТЫВАНИЯ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ

## 1. Частное облако

- исключительно для организации
- управляется организацией / третьей стороной
- существует локально / за пределами организации

## 2. Облако сообщества

- используется несколькими организациями и поддерживает конкретное сообщество
- управляется организациями / третьей стороной
- существует локально / за пределами помещения

## 3. Публичное облако

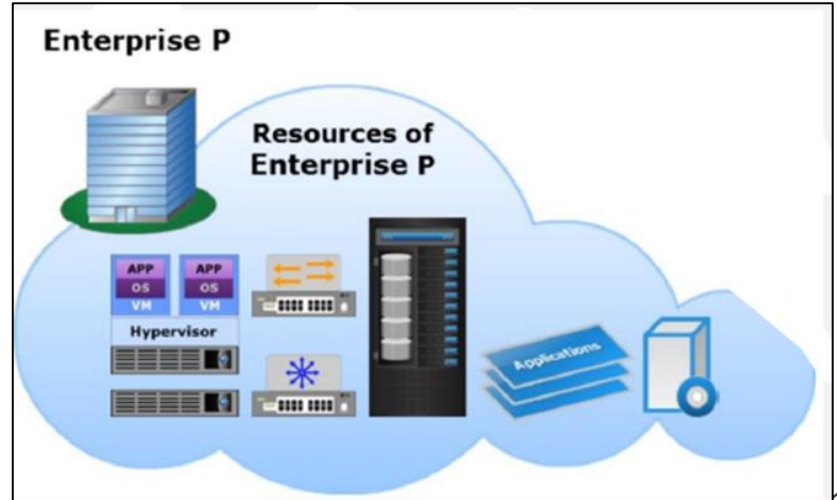
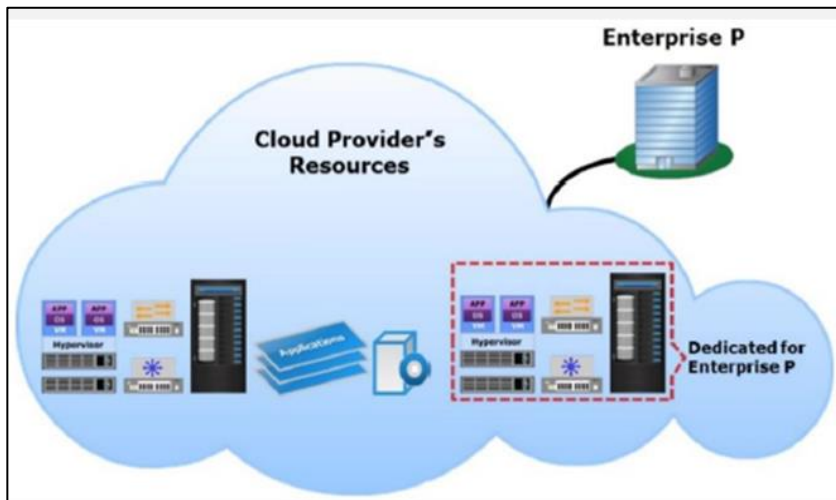
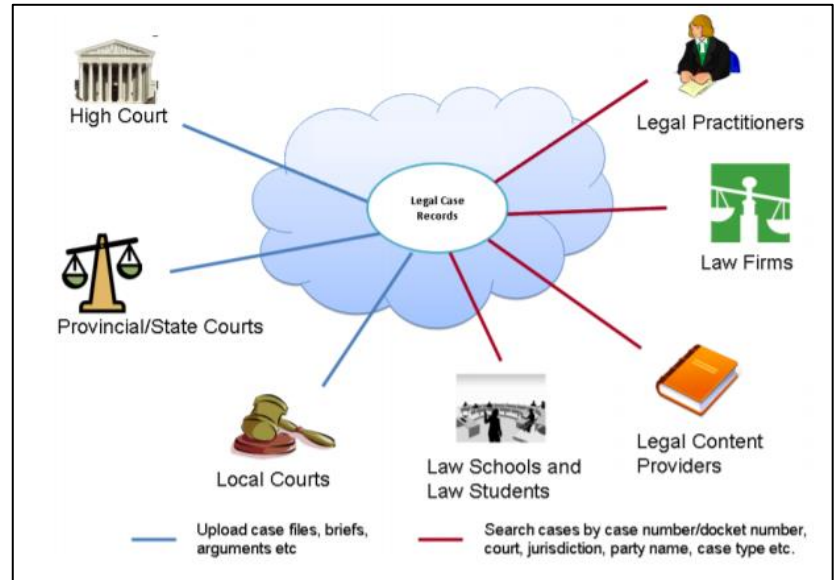
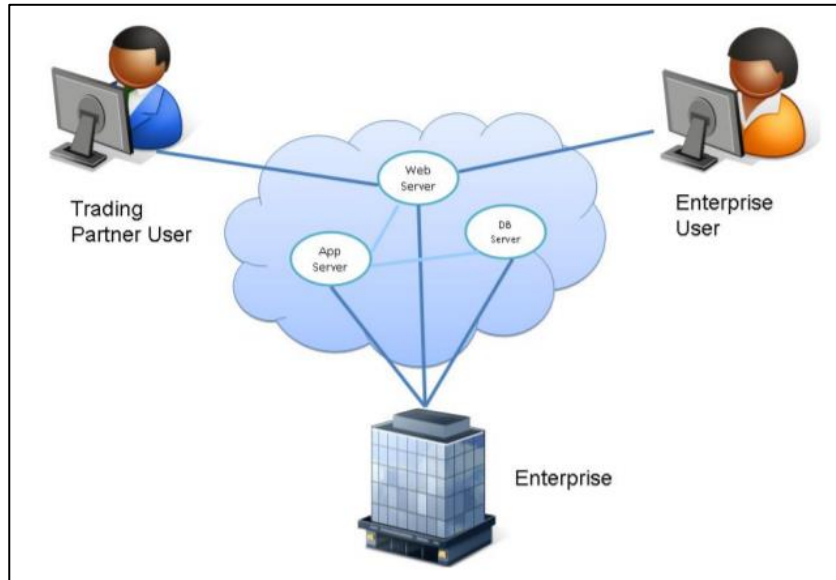
- доступна широкой публике / крупной отраслевой группе
- принадлежит организации, продающей облачные сервисы

## 4. Гибридное облако

- композиция из двух или более облаков, которые остаются уникальными объектами
- Используют технологии, обеспечивающие перенос данных и приложений

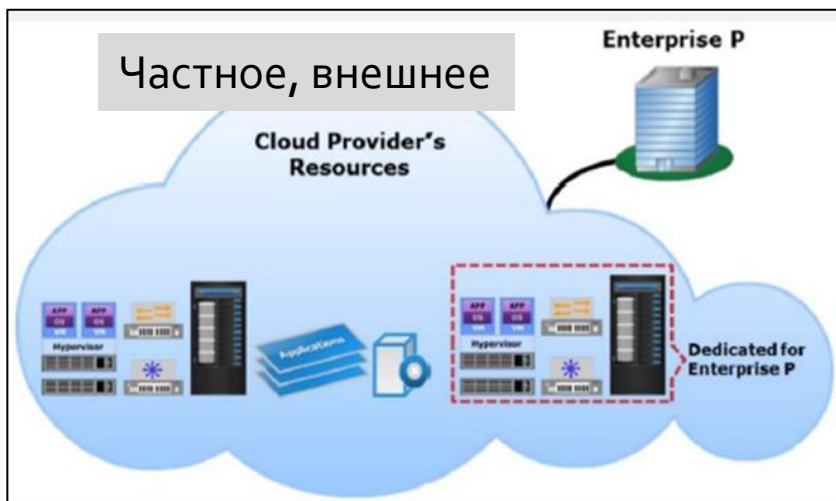


# ГДЕ КАКАЯ МОДЕЛЬ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ?





# ГДЕ КАКАЯ МОДЕЛЬ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ?





# SAAS – Software as a Service, ПО как услуга

## Провайдер предоставляет:

- Использование приложения провайдера, работающие в облачной инфраструктуре
- Приложения доступны из различных клиентских устройств через интерфейс тонкого клиента

## Потребитель:

- не управляет базовой облачной инфраструктурой
- не контролирует ее
- Возможное исключение – ограниченная пользовательская конфигурация приложения и его настройка



# PAAS – Platform as a Service, платформа как услуга

## Провайдер предоставляет:

- Платформу для разработки
- СУБД
- контроль базовой облачной инфраструктуры, включая сеть, серверы, операционные системы, хранилище

## Потребитель:

- Развертывание в облачную инфраструктуру своих приложений
- Контроль над развернутыми приложениями
- Использование инструментов, поддерживаемых провайдером

## Пример:

- Ростелеком, Microsoft Azure, Google, Amazon Web Services, ЦОД IT Energy Китай-город



# IAAS – Infrastructure as a Service, инфраструктура как услуга

## Провайдер:

- управляет базовым облаком
- контролирует его инфраструктуру

## Потребителю предоставляются:

- возможность обработки, хранения, использование сетей и других основных вычислительных ресурсов
- возможность развертывать и запускать произвольное программное обеспечение (системное и прикладное)
- возможность контроля ОС, хранилища, приложениями
- ограниченное управление отдельными сетевыми компонентами



# IAAS – Infrastructure as a Service инфраструктура как услуга

## Лидеры и нишевые игроки рынка IaaS

Провайдер

- управля
- контрол

Потребитель

- ВОЗМОЖИ  
ИСПОЛЬЗ  
ВЫЧИСЛИ
- ВОЗМОЖИ  
произвс  
обеспеч
- ВОЗМОЖИ  
хранили
- огранич  
сетевым



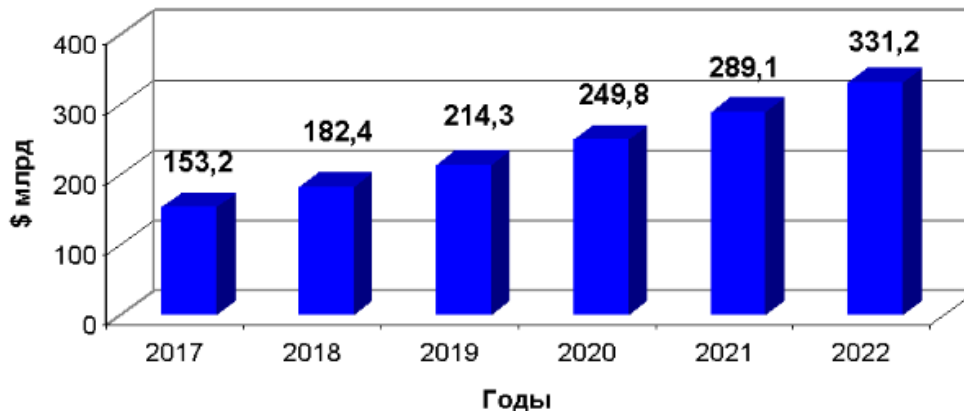
Источник: Gartner, 2019

# ОБЛАЧНЫЕ УСЛУГИ

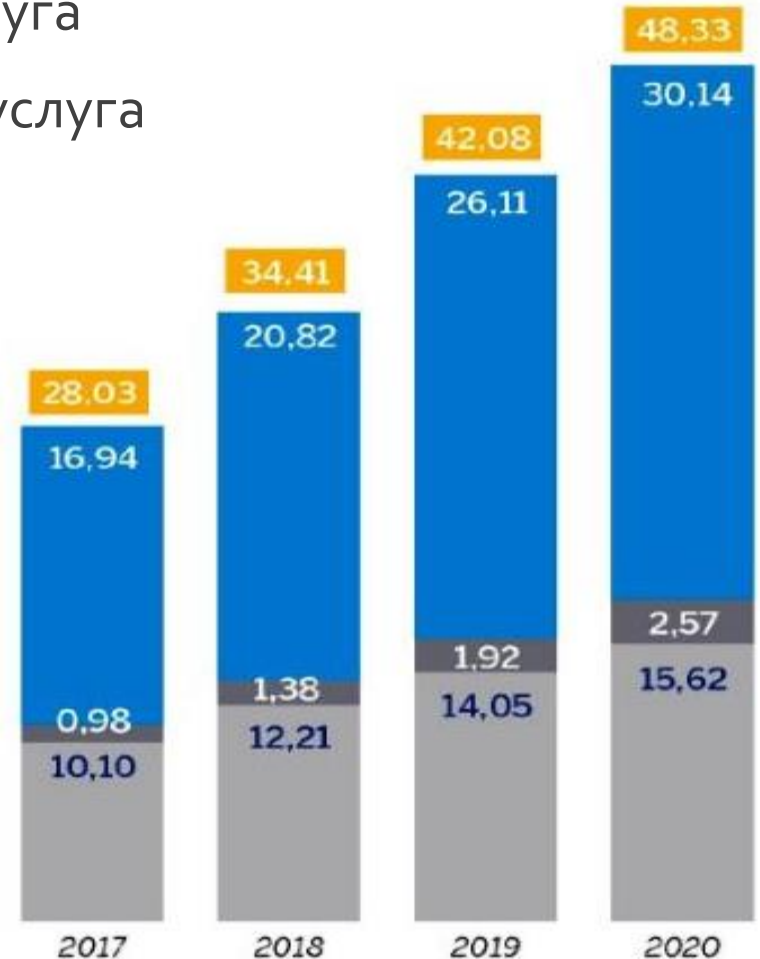
- SaaS – Программное обеспечение как услуга
- IaaS – Инфраструктура как услуга
- PaaS – Бизнес-платформа как услуга

прогноз

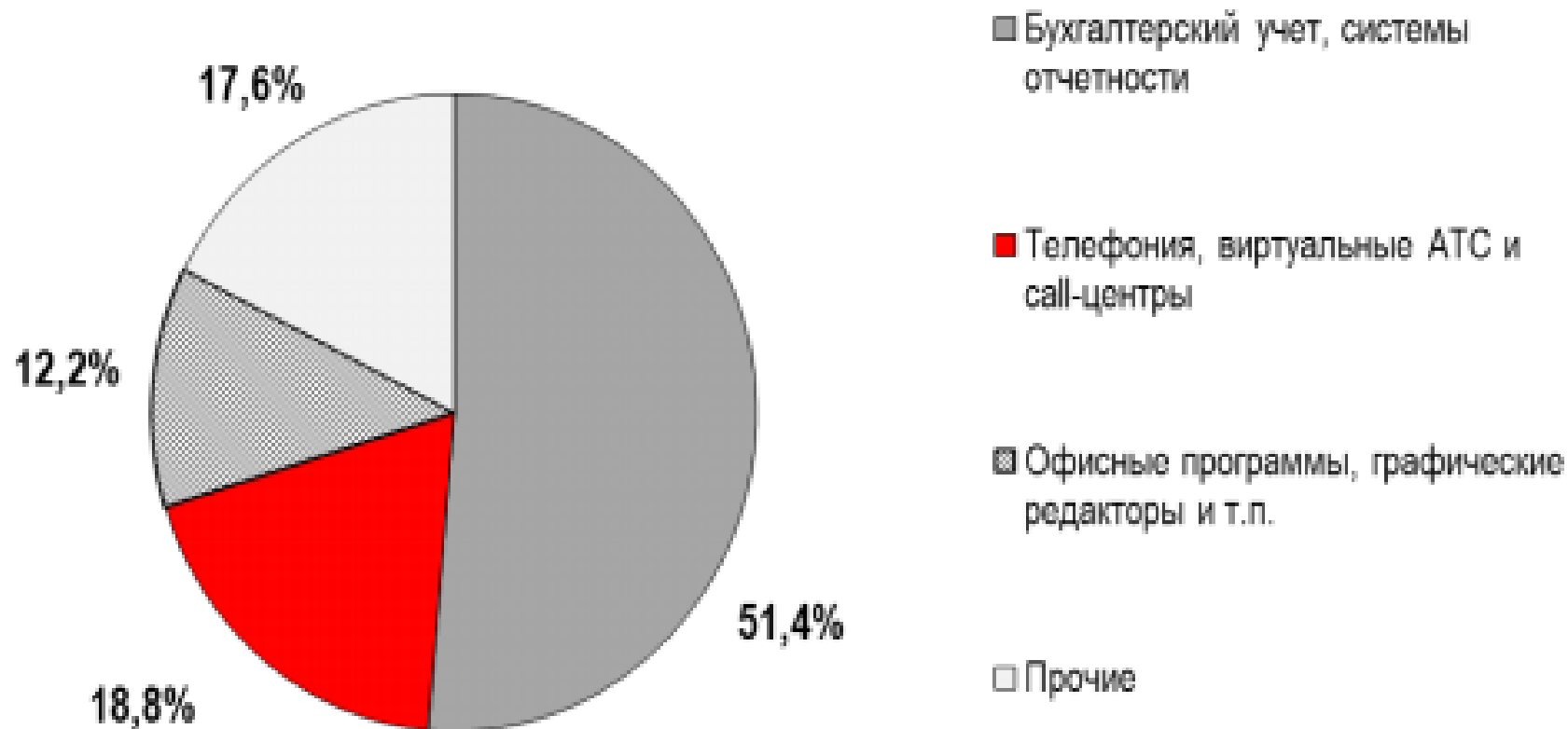
Динамика мирового рынка облачных сервисов



Gartner, 2019

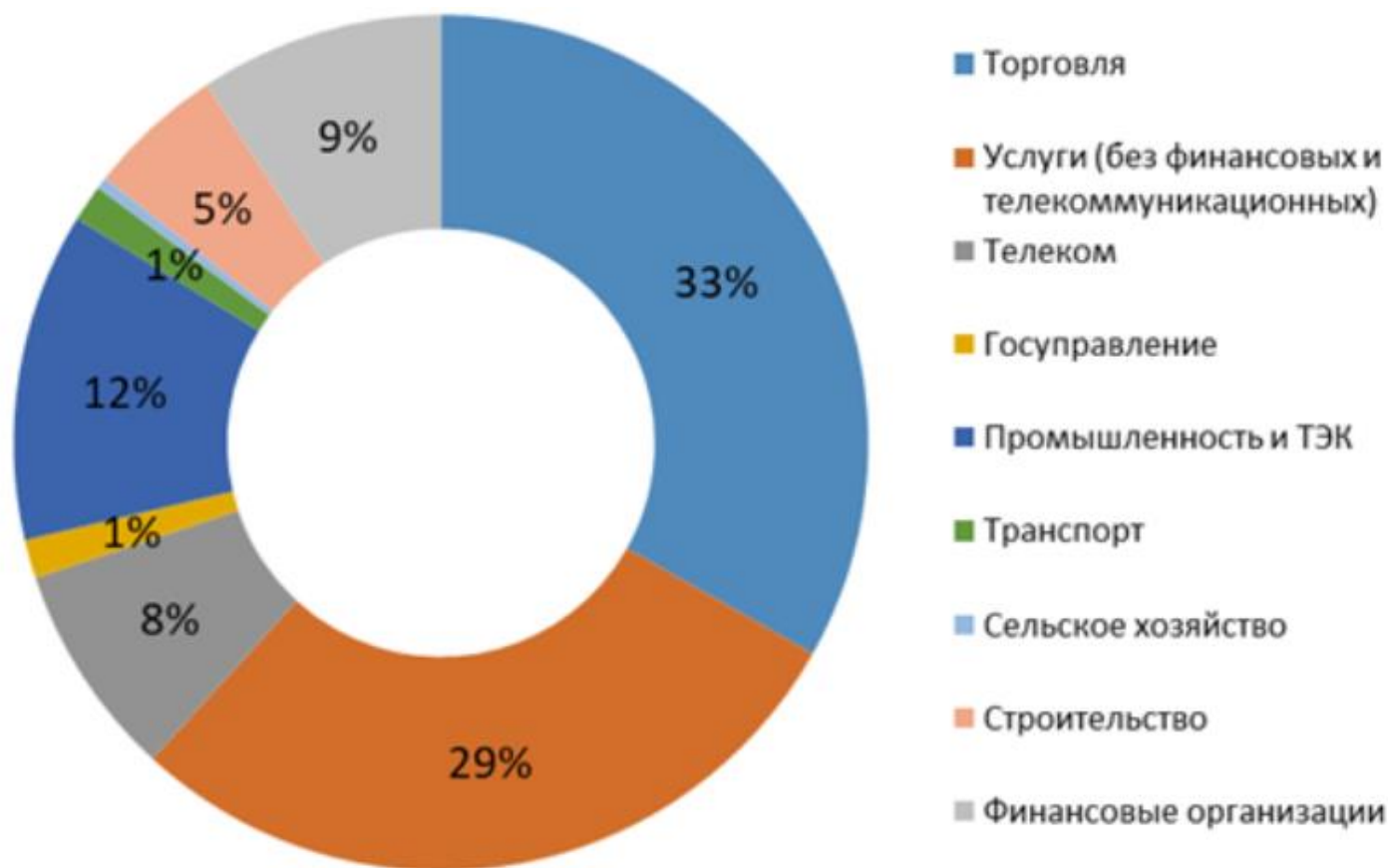


# РЫНОК **SAAS** В РОССИИ



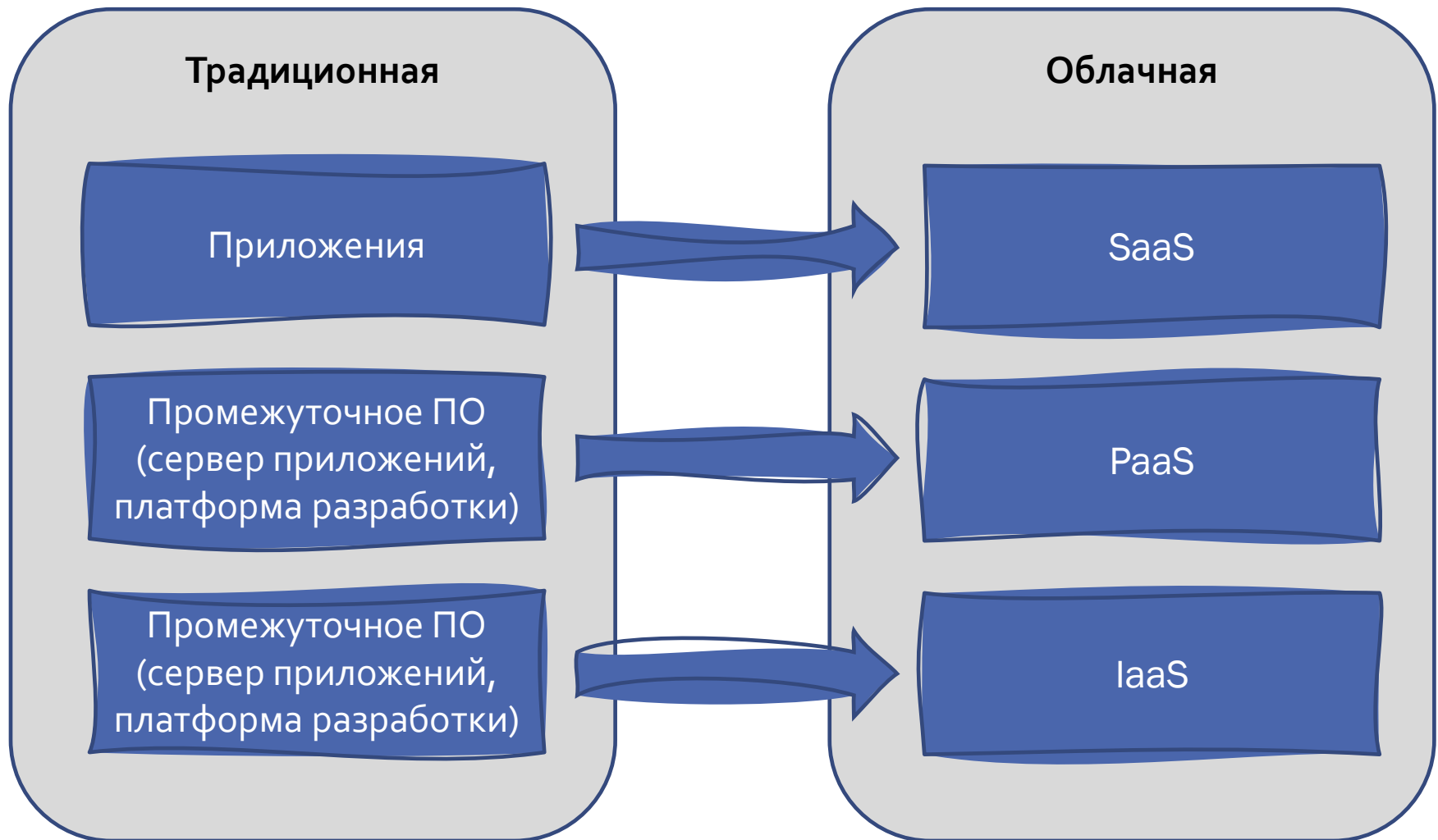
Источник: iKS-Consulting

# ПОЛЬЗОВАТЕЛИ **SAAS**, 2019



Источник: J'son & Partners Consulting

## ТРАДИЦИОННАЯ & ОБЛАЧНАЯ ВЫЧИСЛИТЕЛЬНАЯ ИТ-СРЕДА





## СПЕЦИАЛИЗИРОВАННЫЕ МОДЕЛИ ПРЕДСТАВЛЕНИЯ УСЛУГ

### BPaaS

- Business Platform as a Service, бизнес-платформа как услуга
- Предоставление набора всех необходимых компонентов ИТ-инфраструктуры (бухгалтерия, почта, CRM-системы) для средних и крупных компаний

### DaaS

- Desktop as a Service, рабочий стол как услуга
- Пользователь получает заранее настроенное и готовое к работе виртуальное рабочее место

### SecaaS

- Security as a Service, безопасность как услуга
- Передача провайдеру управления всеми системами обеспечения информационной безопасности компании (firewall, антивирусная защита, системы обнаружения и предотвращения вторжений, и т.д.)

### BaaS

- Backup as a Service, резервное копирование как услуга
- Услуга по резервированию наиболее важных сервисов или данных

### DRaaS

- Disaster Recovery as a Service, катастрофоустойчивость как услуга
- Предоставляет возможность экстренного восстановления сервисов пользователя после возможных аварий и катастроф

## МИГРАЦИЯ В ОБЛАКА


<b>Сильные стороны (Strenghts)</b>	<b>Слабые стороны (Weakness)</b>
Низкие капитальные затраты	Латентность (Задержки сети)
Легкость внедрения	Надежность (Потери данных)
Легкость поддержки	Отсутствие выделенного персонала
Горизонтальное масштабирование	Ограниченные возможности по конфигурации
Вертикальное масштабирование	Ограниченные возможности по кастомизации
Отказоустойчивость данных и сервисов	
<b>Воможности(Opportunities)</b>	<b>Угрозы (Threats)</b>
Удобство использования	Конфиденциальность, целостность и доступность данных
Эластичность	Трудность перехода между поставщиками
Переход от CAPEX к OPEX	Трудность трансграничной обработки данных
Быстрый выход на рынок	Проблемы с гарантированным временем восстановления или политикой возмещения
Гибкое ценообразование (плата за фактическое использование)	Проблемы совместимости
Устойчивость к потере доходов во время кризиса	

## ПРЕИМУЩЕСТВА ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

- + Сокращение разработки и вывода новых продуктов на рынок
- + Снижение капитальных затрат
- + Применение принципа «попробуйте прежде, чем купить»
- + Снижение стоимости «разовых» мероприятий (миграция данных, например)
- + Снижение расходов на тестирование
- + Возможность передачи пиковой нагрузки со своего «дата-центра», сокращение его вместимости
- + Лучшая архитектура и дизайн ИТ-инфраструктуры
- + Консолидация и централизованное администрирование
- + Высвобождение ресурсов для других задач
- + Высокое качество предоставляемых ИТ-услуг 😊
- + ИТ-безопасность
- + Гибкость
- + Улучшенный финансовый контроль
- + Экологичные ИТ 😊

# ПРОБЛЕМЫ ОБЛАЧНЫЙ ВЫЧИСЛЕНИЙ

- ИТ-безопасность и соответствие
- «Молодость» решений
- Отсутствие четкой терминологии и назначения компонент облачных решений
- Лицензирование ПО
- Ясность в предоставлении услуг
- Расчет стоимости услуг
- Доступ к данным
- Интеграция
- **The Open Group** – есть облачная рабочая группа
- **Association for Retail Technology Standards (ARTS)** – Стандарты по облакам для торговли
- **Cloud Audit** – Межотраслевая рабочая группа
- **Cloud Computing Interoperability Forum (CCIF)** – Некоммерческое сообщество
- **Cloud Security Alliance (CSA)** – Некоммерческая организация по продвижению лучших практик
- **Cloud Security Alliance (CSA)** – консорциум по разработке облачных стандартов
- **Open Grid Forum (OGF)** – об эволюции прикладных распределенных вычислений



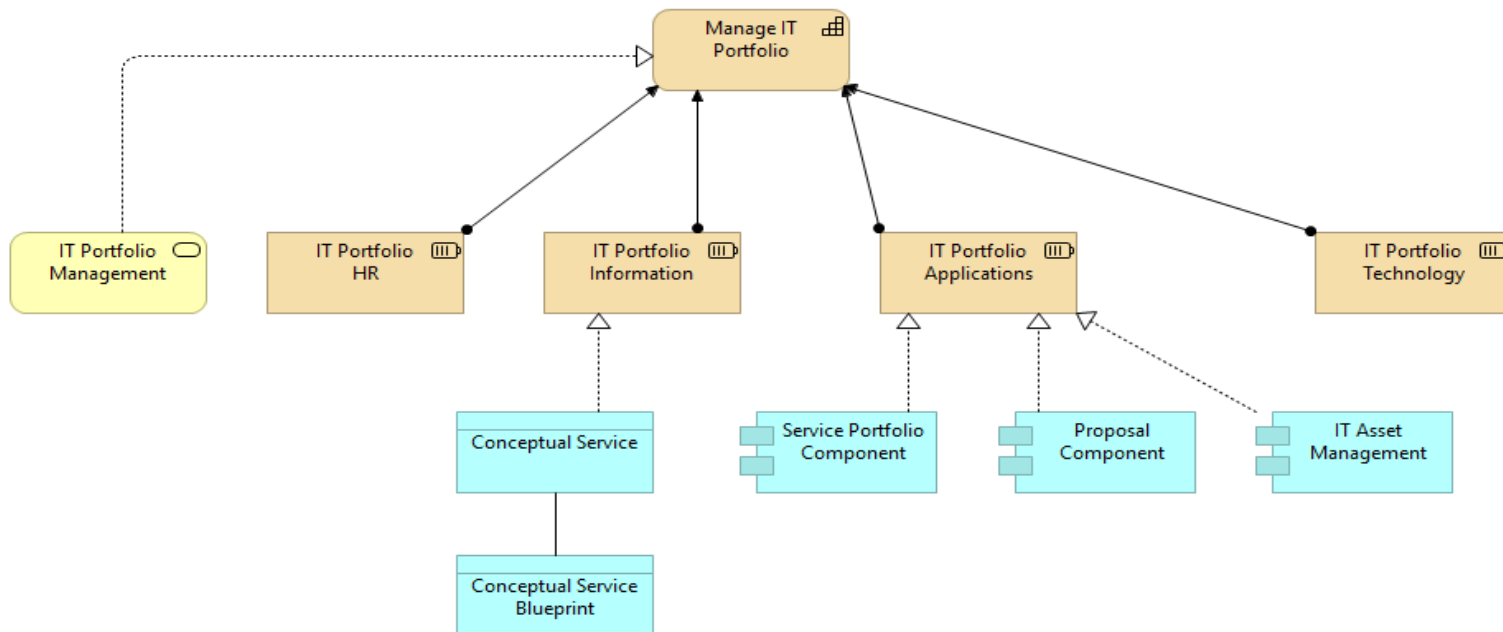
Представления отражают то, что  
архитектор должен учитывать при работе  
в области архитектуры приложений

## РАЗНЫЕ ПРЕДСТАВЛЕНИЯ В ОБЛАСТИ АРХИТЕКТУРЫ ПРИЛОЖЕНИЙ



# ПРЕДСТАВЛЕНИЕ ВОЗМОЖНОСТЕЙ

- отражает соответствие приложений бизнес-возможностям
- Глядя на область архитектуры приложений с точки зрения поддерживаемых ею бизнес-возможностей, вы получаете хорошее представление о том, как эти приложения напрямую поддерживают бизнес
- Функциональный тип приложения (CRM, HCM и т. д.)



## ТЕХНОЛОГИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ

- Это представление отражает базовую технологию, из которой состоят приложения
- «сложность равна стоимости», особенно при попытке снизить эту сложность с помощью стандартизации

### Состав:

#### ■ Программное обеспечение:

- само приложение
- ПО, на котором оно функционирует (веб-серверы, серверы приложений)

#### ■ Инфраструктура:

- базовое оборудование
- сетевые компоненты, необходимые для приложения и поддерживающие прикладное ПО

#### ■ Тип:

- тип приложения (смешанное, трехуровневое и т.п.)

#### ■ Разработка:

- компоненты разработки, которые являются частью самого приложения
- закрепленные за разработкой веб-службы, API и т. д.
- сам процесс обслуживания

#### ■ Интеграция:

- интерфейсы, которые приложение предоставляет для интеграции
- интеграция с другими приложениями и источниками данных, которые необходимы приложению для работы

## ПРЕДСТАВЛЕНИЕ ОРГАНИЗАЦИИ

- Это представление отражает **человеческий аспект приложения**

- организации состоят из людей, и эти люди используют приложения для выполнения своей работы

- Стоит ли определять домен архитектуры приложения, не принимая во внимание организацию, которая будет его использовать / финансировать / изменять, - все равно что пытаться спроектировать автомобиль, не задумываясь о том, кто будет его водить Оно включает:

### Состав:

- Право собственности: **Кто «владеет»** приложением?
- Финансирование: **Кто финансирует** как приобретение / создание, так и текущее обслуживание (финансирование создания / изменения / эксплуатации)?
- Изменение: **кто может / запрашивает изменения** в приложении и каковы последующие действия?
- Использование: **кто использует приложение**, как часто и как они его используют?
- Поддержка: **какая организация** отвечает за **текущую** поддержку приложения?



# ИНФОРМАЦИОННОЕ ПРЕДСТАВЛЕНИЕ

- «информация движет предприятием»
- Управление, создание и организация этой информации - основные функции корпоративных приложений
- Это представление отражает то, как приложения связаны с информацией (как домен архитектуры приложения относится к домену информационной архитектуры)

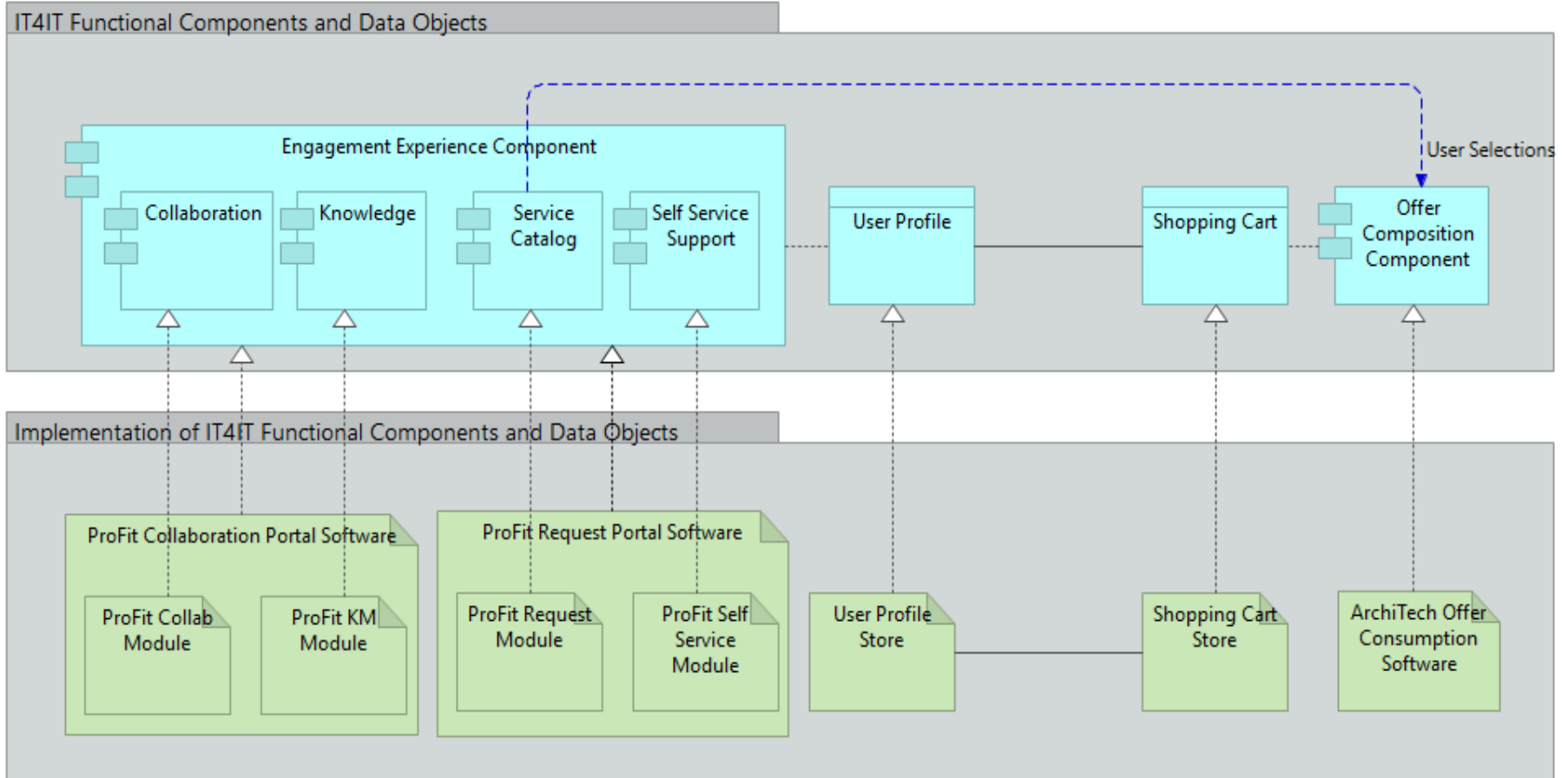
## Состав

- **Доступ:** приложение - это механизм, с помощью которого конечные пользователи получают доступ к информации:
  - основное приложение (например, CRM)
  - приложение типа доступа к информации (например, BI)
- **Создание:** приложения создают данные для предоставления информации конечным пользователям
  - заказ, который будет использоваться конечным пользователем как часть процесса выполнения (пример)
- **Потребление:** описывает данные, необходимые приложениям для работы
  - Идентификатор продукта требуется приложению закупок для создания заказа (пример)

## ПРЕДСТАВЛЕНИЕ МОДЕЛИ ДОСТАВКИ

- Организации ищут способы доставки своих приложений для:
  - для снижения затрат,
  - для повышения гибкости
- С точки зрения ЕА, где и как развертывается приложение оказывает влияние на общую архитектуру предприятия:
  - вопросы интеграции
  - требования SLA
  - вопросы безопасности и соответствия
- Это представление отражает способ доставки приложений конечным пользователям, т.е. **варианты развертывания приложений**

# КАКОЙ ТИП ПРЕДСТАВЛЕНИЯ?



## ОСНОВНЫЕ ИСТОЧНИКИ

- NIST // <http://csrc.Nist.Gov/groups/sns/cloud-computing/cloud-def-v15.Doc>
- Cloud computing explained // The Open Group, 2011
- Maximizing the value of cloud for small-medium enterprises // The Open Group, 2012
- A Practical Approach to Application Portfolio Consolidation using the TOGAF® Standard // The Open Group, 2018
- TOGAF 9.2
- FEA
- Учебник 4CIO
- Gartner.com (публикации 2020 г)
- Tadviser
- IBS
- Данилин А., Слюсаренко А. «Архитектура и стратегия» // ИНТУИТ.РУ, 2018

СПАСИБО ЗА ВНИМАНИЕ! ВОПРОСЫ?

