

Распределенные информационные системы

Единое представление данных:

Поведение

Мир и место в нем

Хранение информации

Вопросы

- Поведение РИС и ее компонентов: где и как хранится
- Окружающий мир
 - Единые модели, внешние интерфейсы
- Хранение состояния
 - Временное
 - Постоянное
 - История

Единое представление данных

- *«Правила игры»*
 - *Единицы измерения, стандарты*
 - *Нормальные условия*
 - *Значения по умолчанию*
- *Когда я? (Время)*
- *Кто я? (структура РИС, именование, функции и поведение РИС и ее компонентов)*
- *Где я?*
 - *Пространство:*
 - *Системы координат*
 - *Определение положения*
 - *Картография*
 - **Окружающий мир:**
 - **Общие модели: атмосфера, движение небесных тел, календари**
 - **Интерфейсы с пользователями, другими ИС**

Метаинформация

- *При анализе, разработке и эксплуатации РИС возникает вопрос о том, насколько полно сама РИС должна обладать информацией о собственном устройстве*
- *К такой информации относятся:*
 - *Структура РИС в целом и связи компонентов друг с другом*
 - *Внешние интерфейсы РИС:*
 - *С пользователями*
 - *С другим ИС*
 - **Поведение РИС и ее компонентов**

Описание поведения

ПОВЕДЕНИЕ РИС И ЕЕ КОМПОНЕНТОВ

Поведение РИС: что

- Под поведением РИС понимается:
 - Процедуры и алгоритмы, реализующие [основные] функции РИС и ее компонентов
 - Бизнес-логика – поведение РИС и ее компонентов в различных условиях, в качестве реакции на [внешние] действия

Поведение РИС: как

- **Согласованность поведения компонентов РИС может достигаться:**
 - Использованием стандартов
 - Описанием поведения в проекте на ЕЯ
 - Хранимыми процедурами и/или алгоритмами
 - Едиными библиотеками
- **Функции РИС могут быть известны другим компонентам**
 - Явно
 - Например, в виде стабов
 - Неявно
 - Выполнение функций предполагается как реакция на действие / приход данных. Напрямую они не вызываются

Как хранится описание поведения?

- Бизнес-логика может быть описана:
 - Явно в каком-либо внутреннем (редактируемом) представлении
 - Например, в системах электронного документооборота
 - Неявно, в виде запрограммированного алгоритма
- Бизнес-логика описывается для отдельных компонентов и для РИС в целом
- Существуют РИС построенные на принципе Агентов

Одноранговые

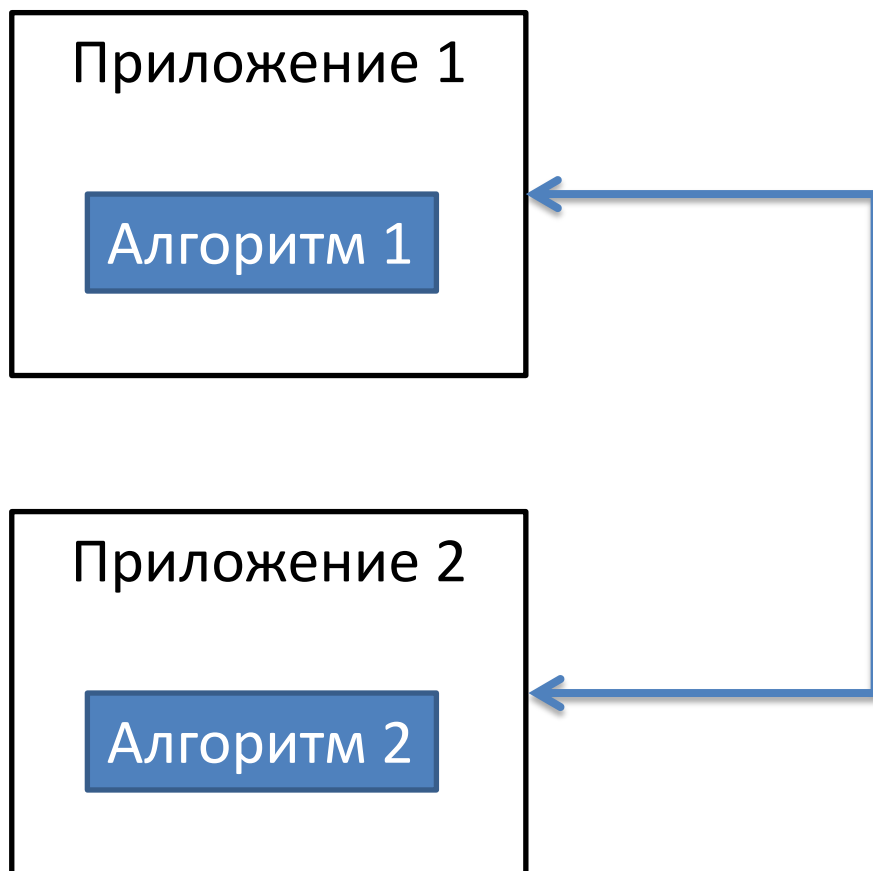
Клиент-серверные

**ОПИСАНИЕ ПОВЕДЕНИЯ РИС В
РАЗНЫХ АРХИТЕКТУРАХ**

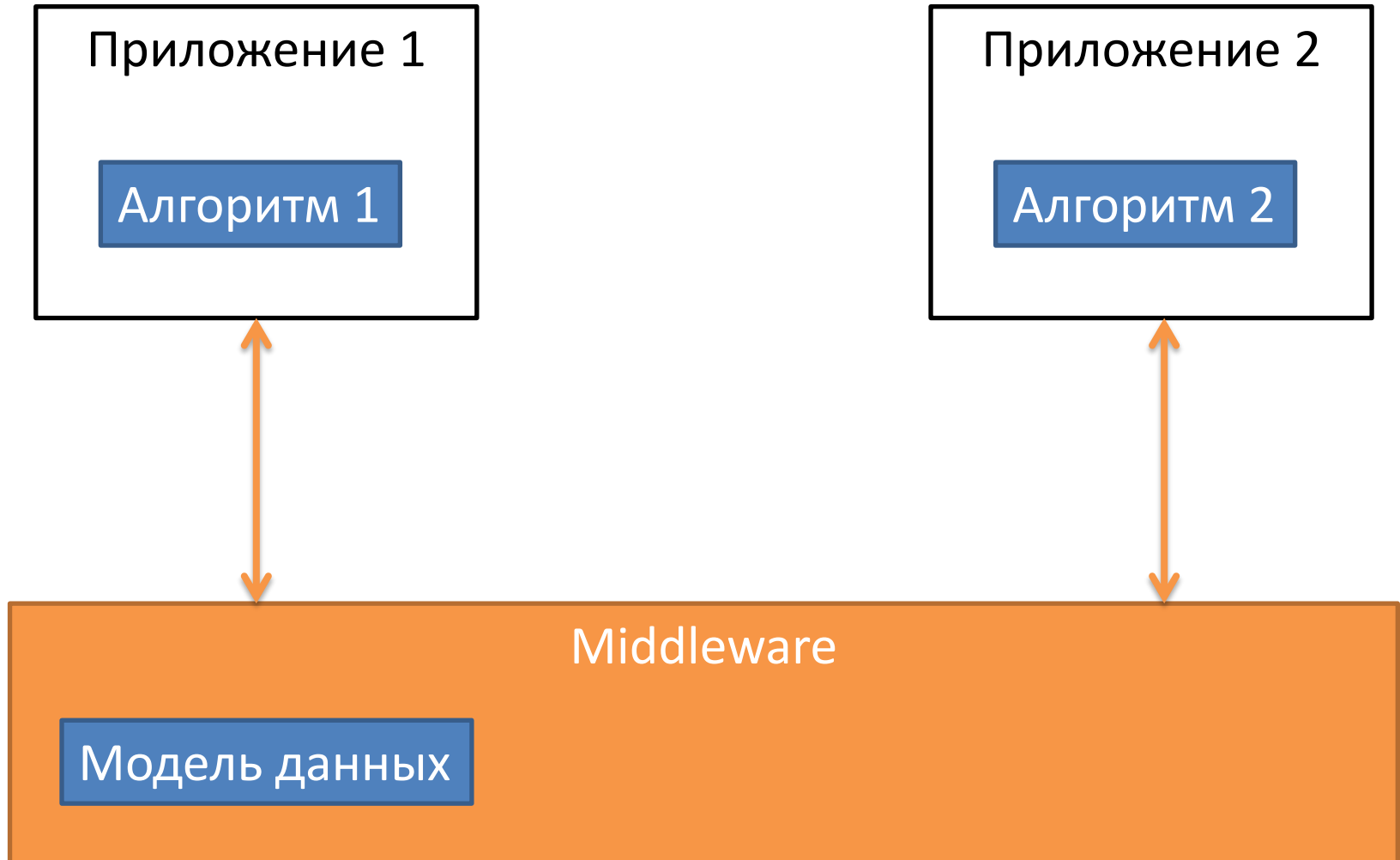
Архитектуры и описание поведения

- По «месту хранения» описания поведения РИС выделяются архитектуры:
 - Одноранговые:
 - С соединениями «точка-точка»
 - С использованием middleware для обмена данными
 - С использованием middleware для удаленного вызова процедур
 - Клиент серверные:
 - Файловый сервер
 - Сервер СУБД
 - Web сервер
 - Intranet

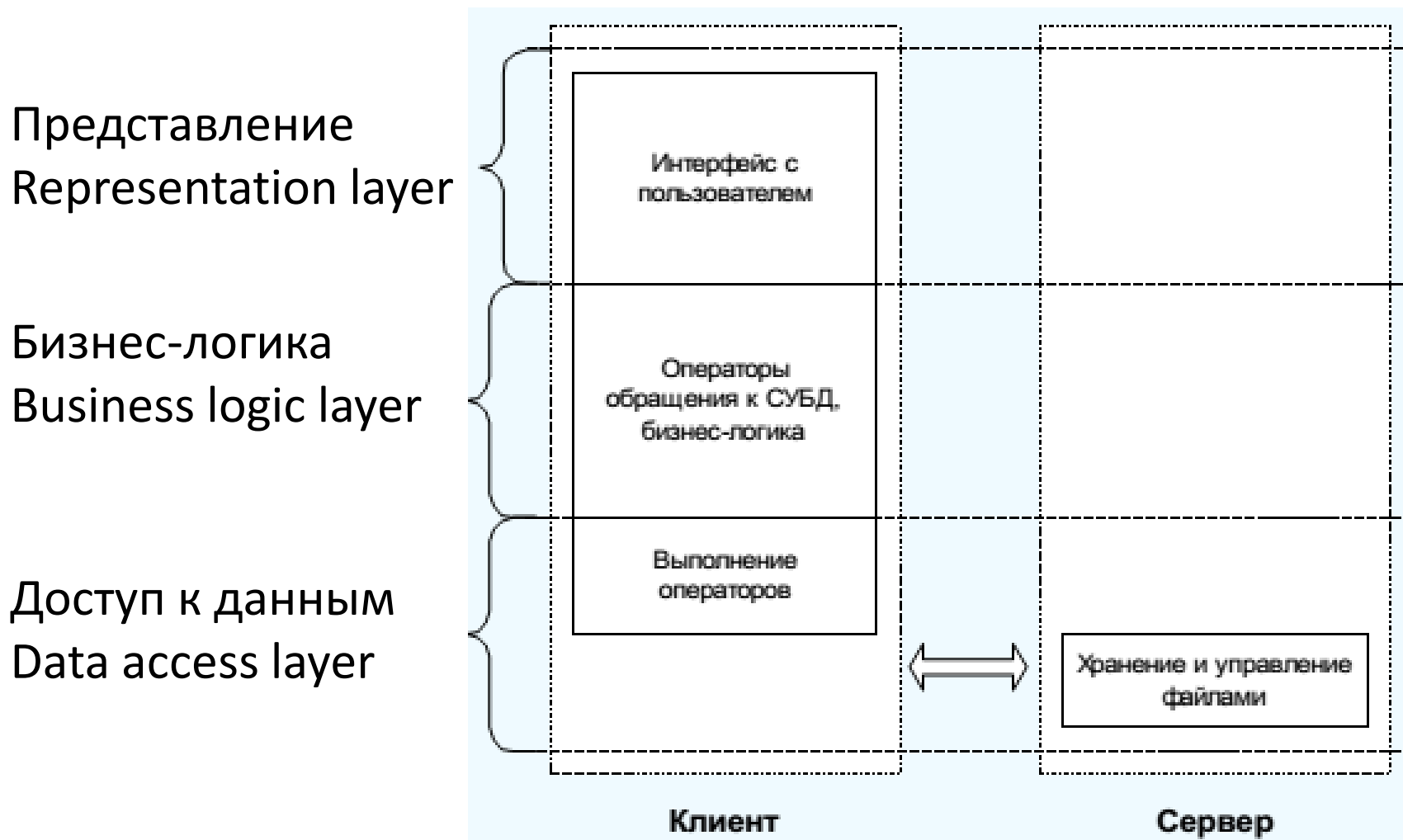
Поведение РИС: «без никто»



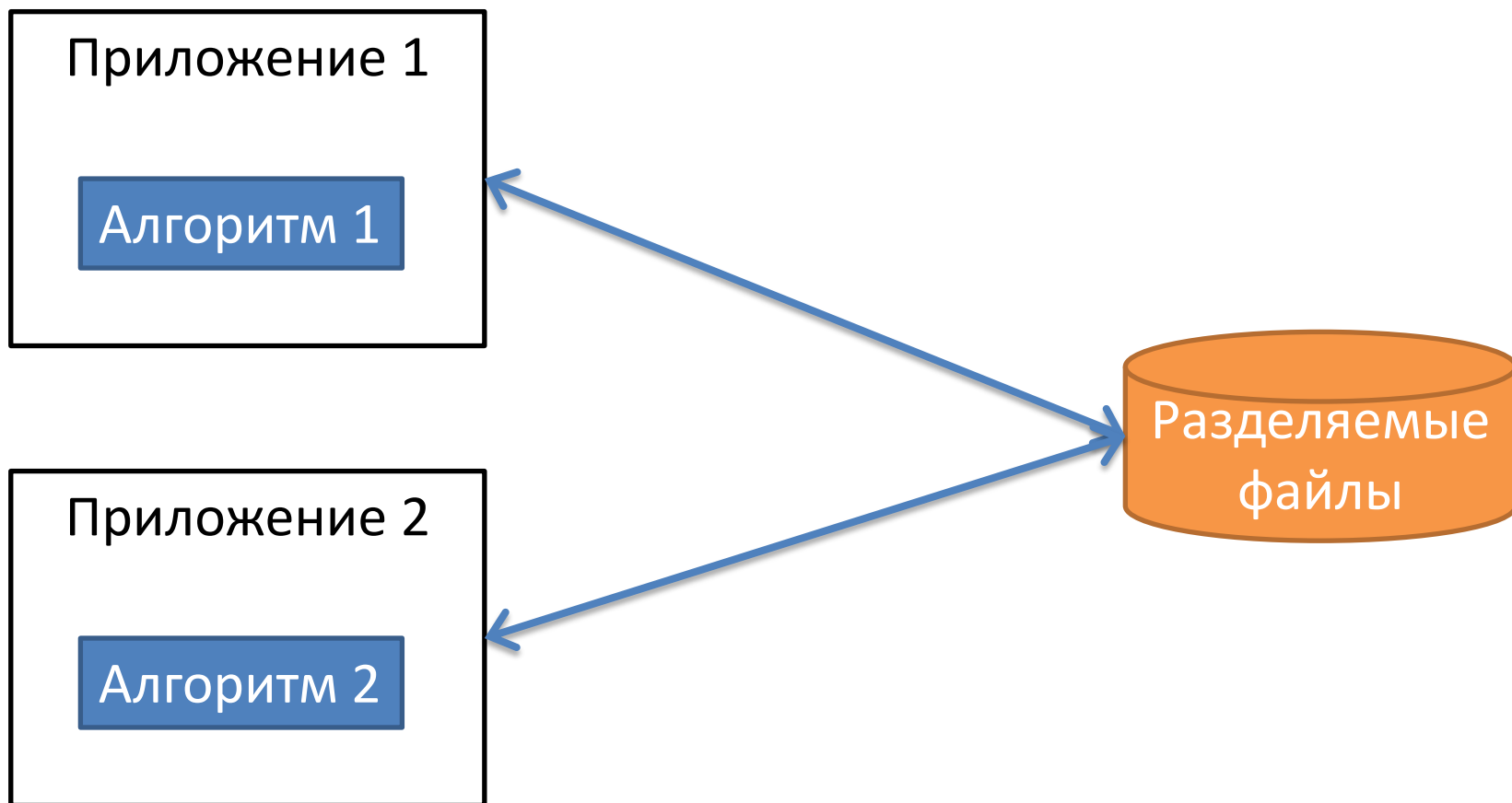
Поведение РИС: middleware для обмена данными



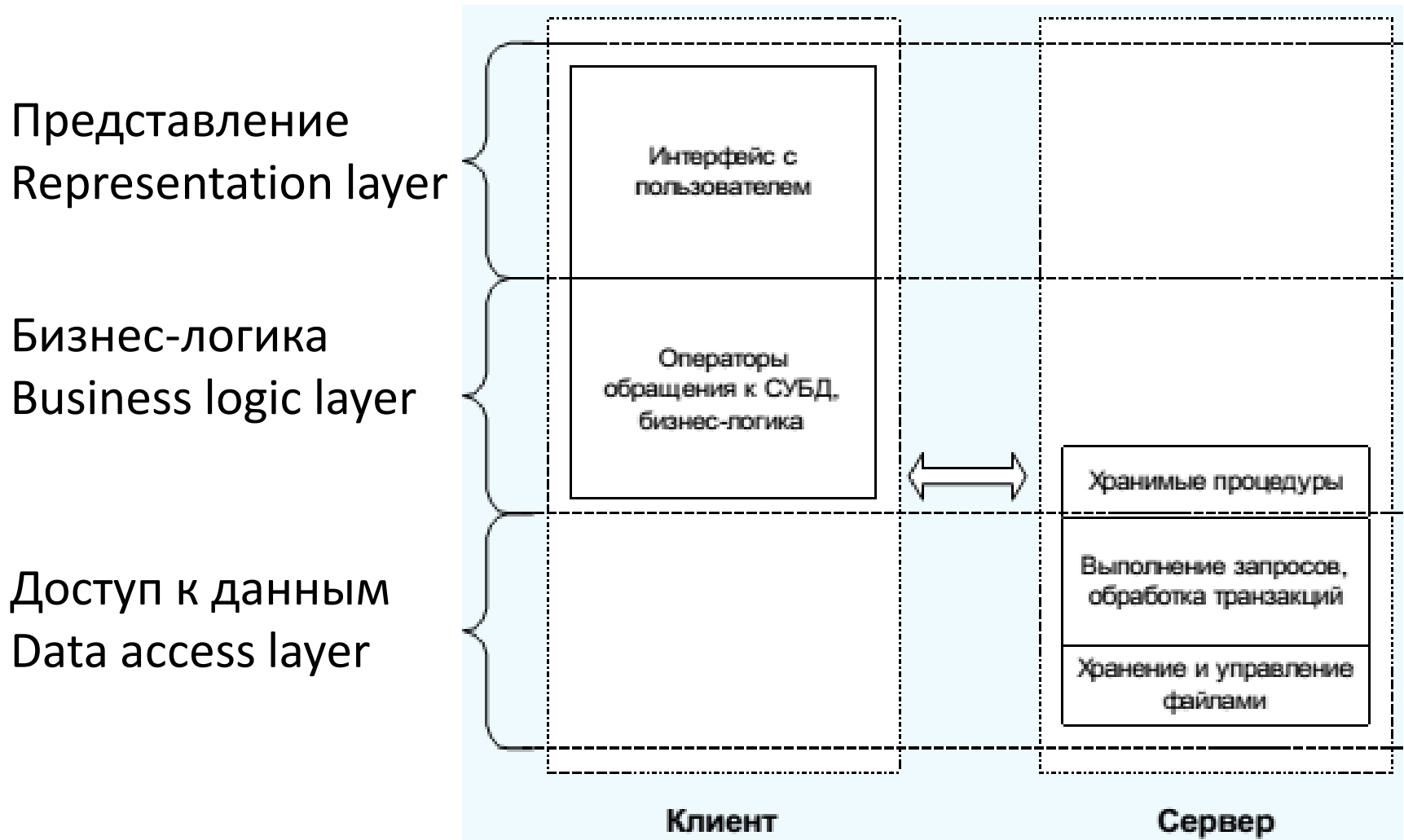
Поведение РИС: файловый сервер



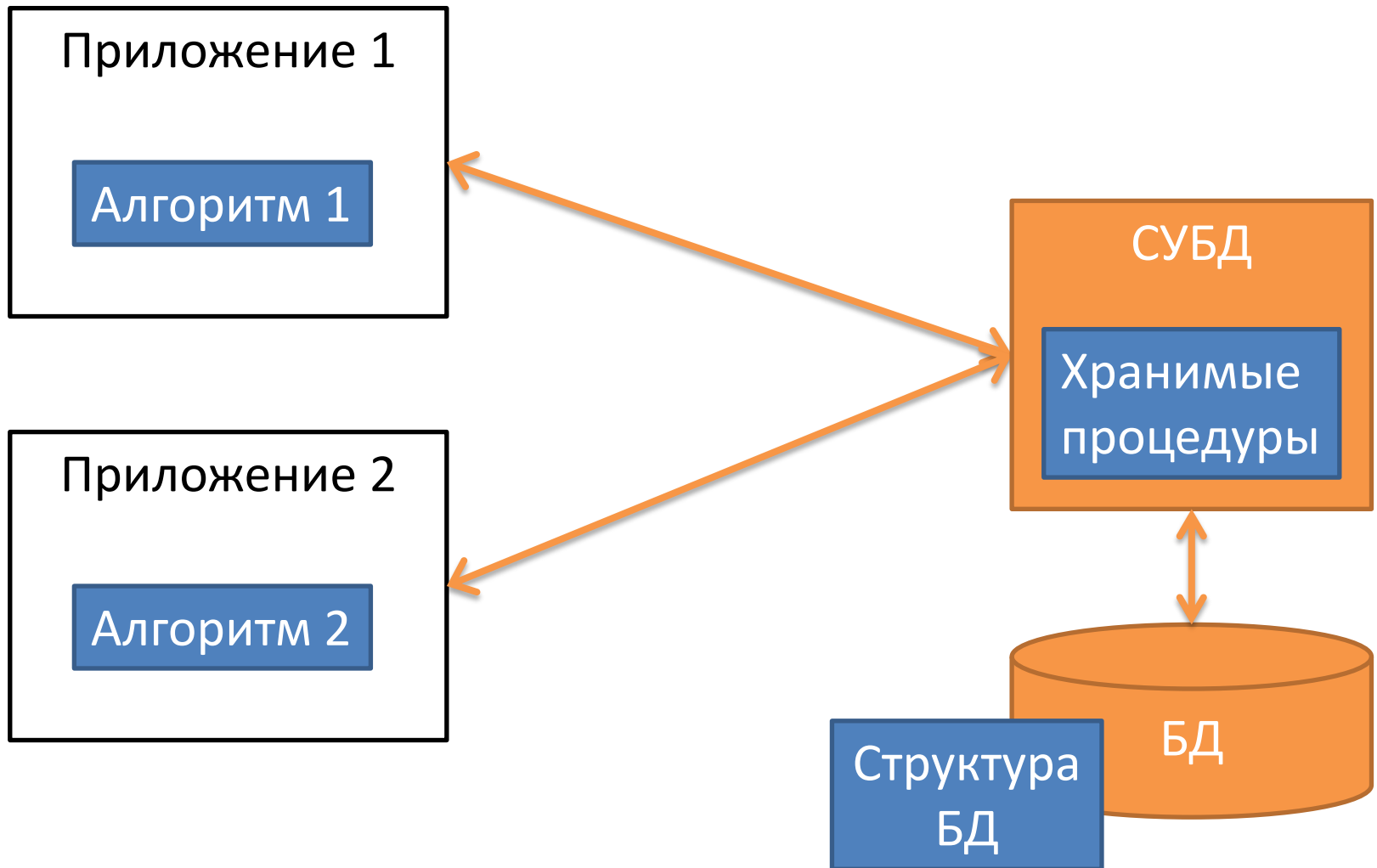
Поведение РИС: файловый сервер



Поведение РИС: сервер СУБД



Поведение РИС: сервер СУБД



Поведение РИС: сервер приложений

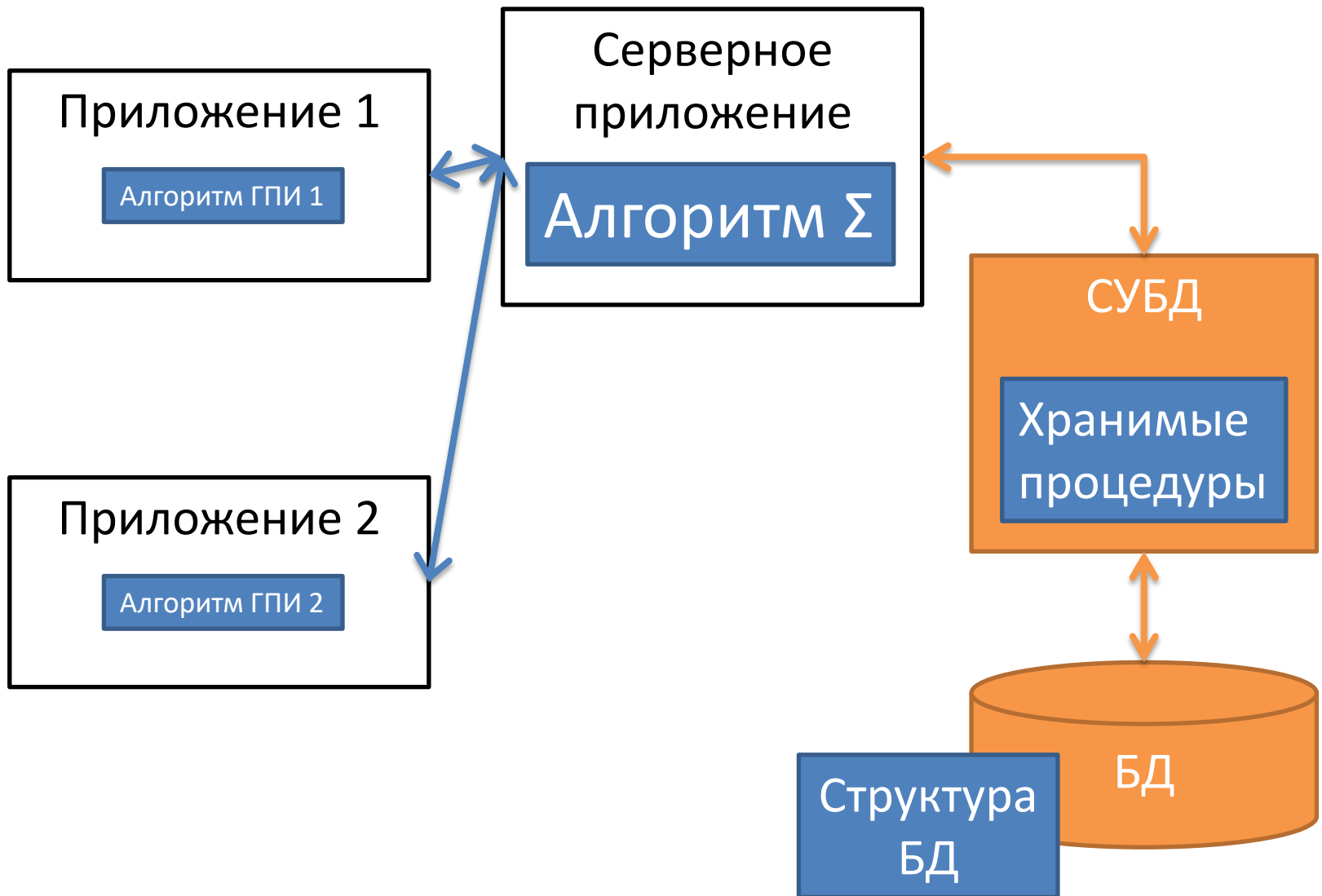
Представление
Representation layer

Бизнес-логика
Business logic layer

Доступ к данным
Data access layer



Поведение РИС: сервер приложений

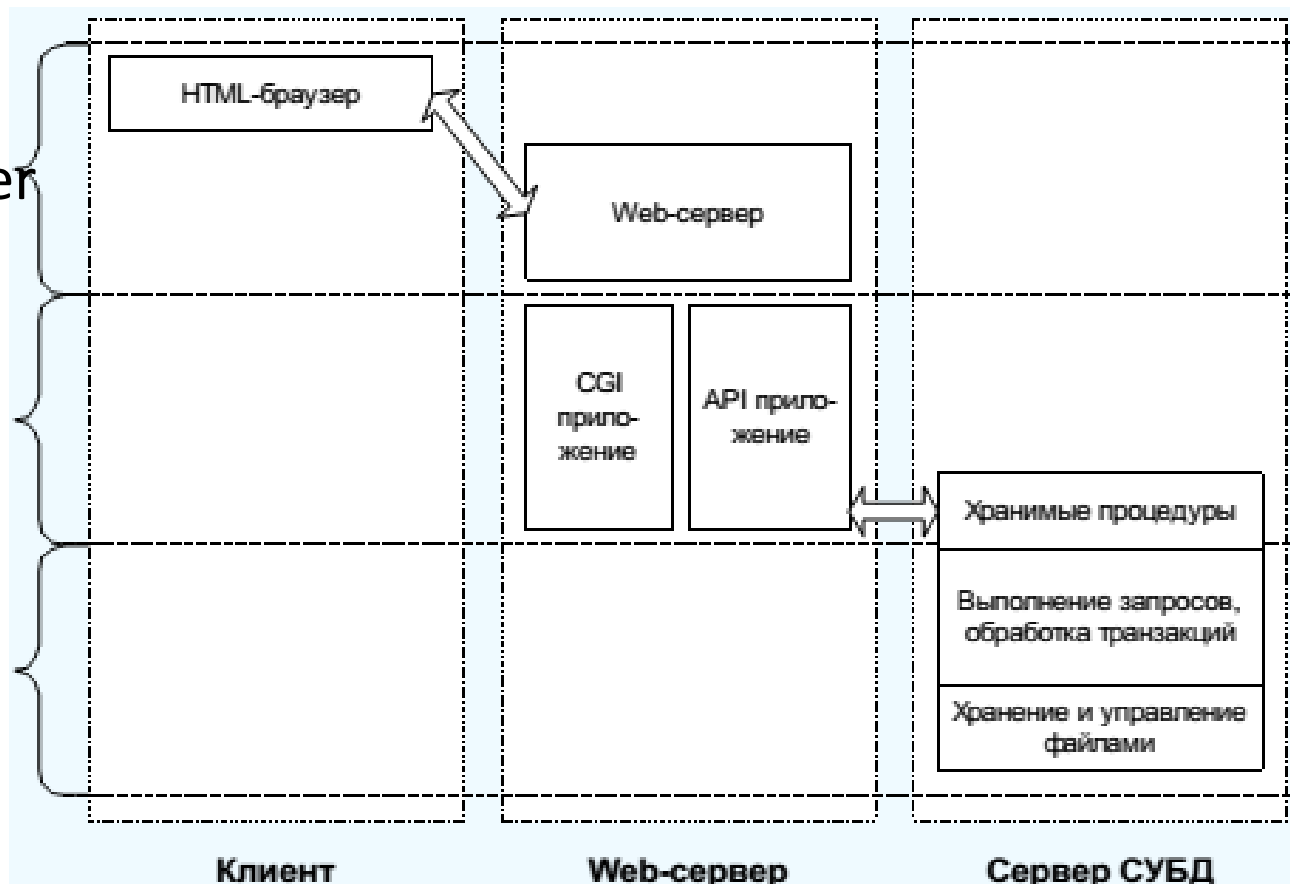


Поведение РИС: WWW сервер

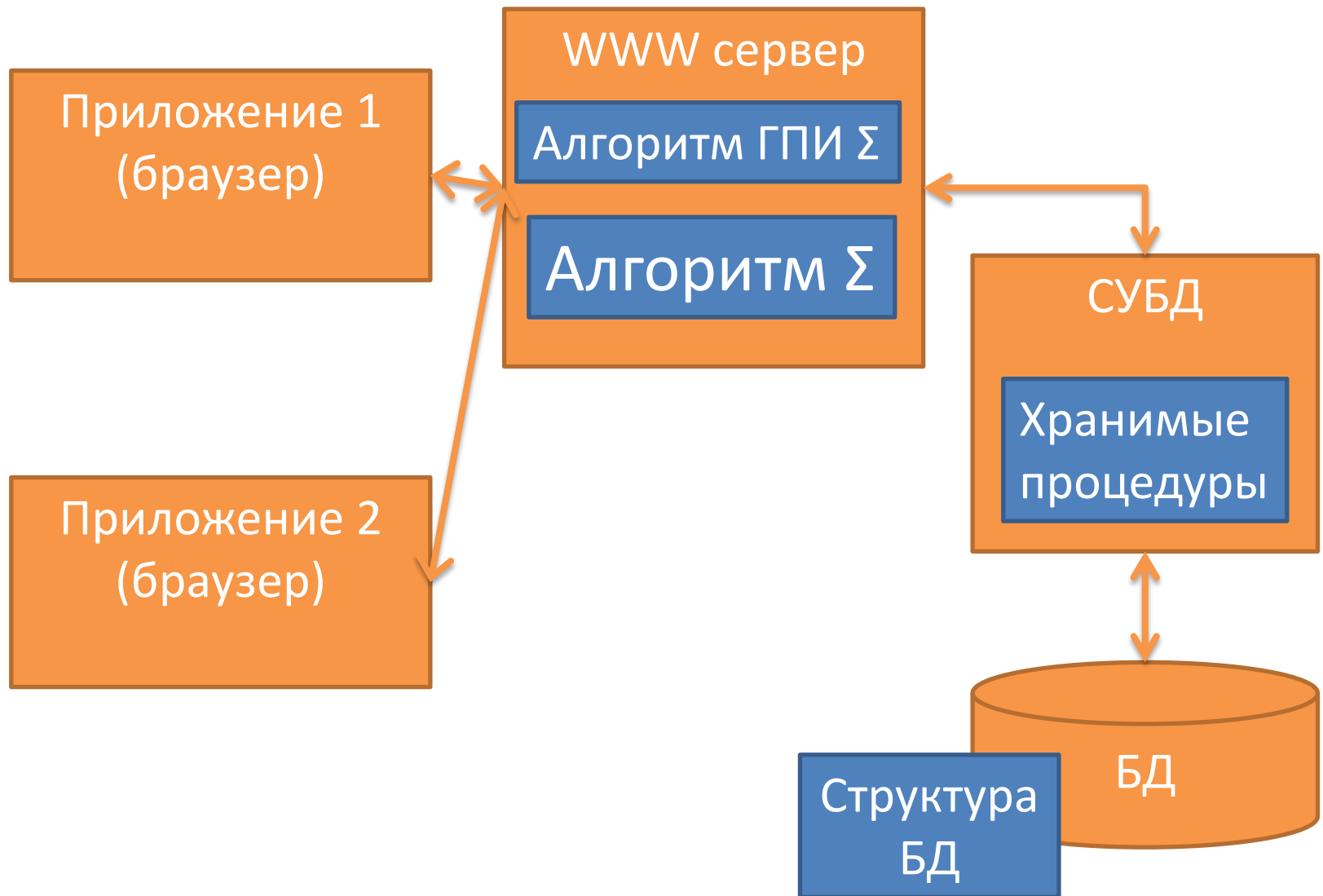
Представление
Representation layer

Бизнес-логика
Business logic layer

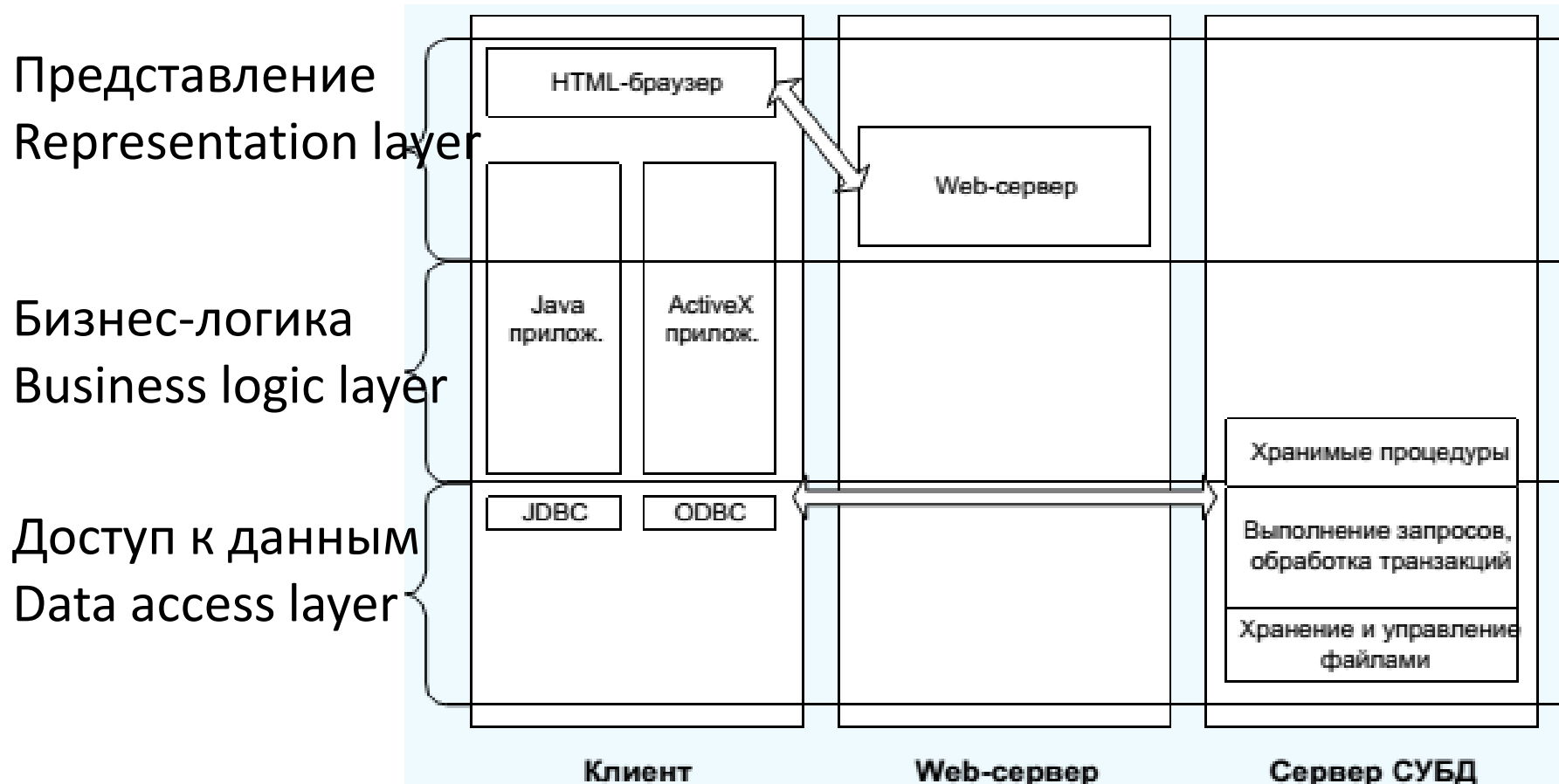
Доступ к данным
Data access layer



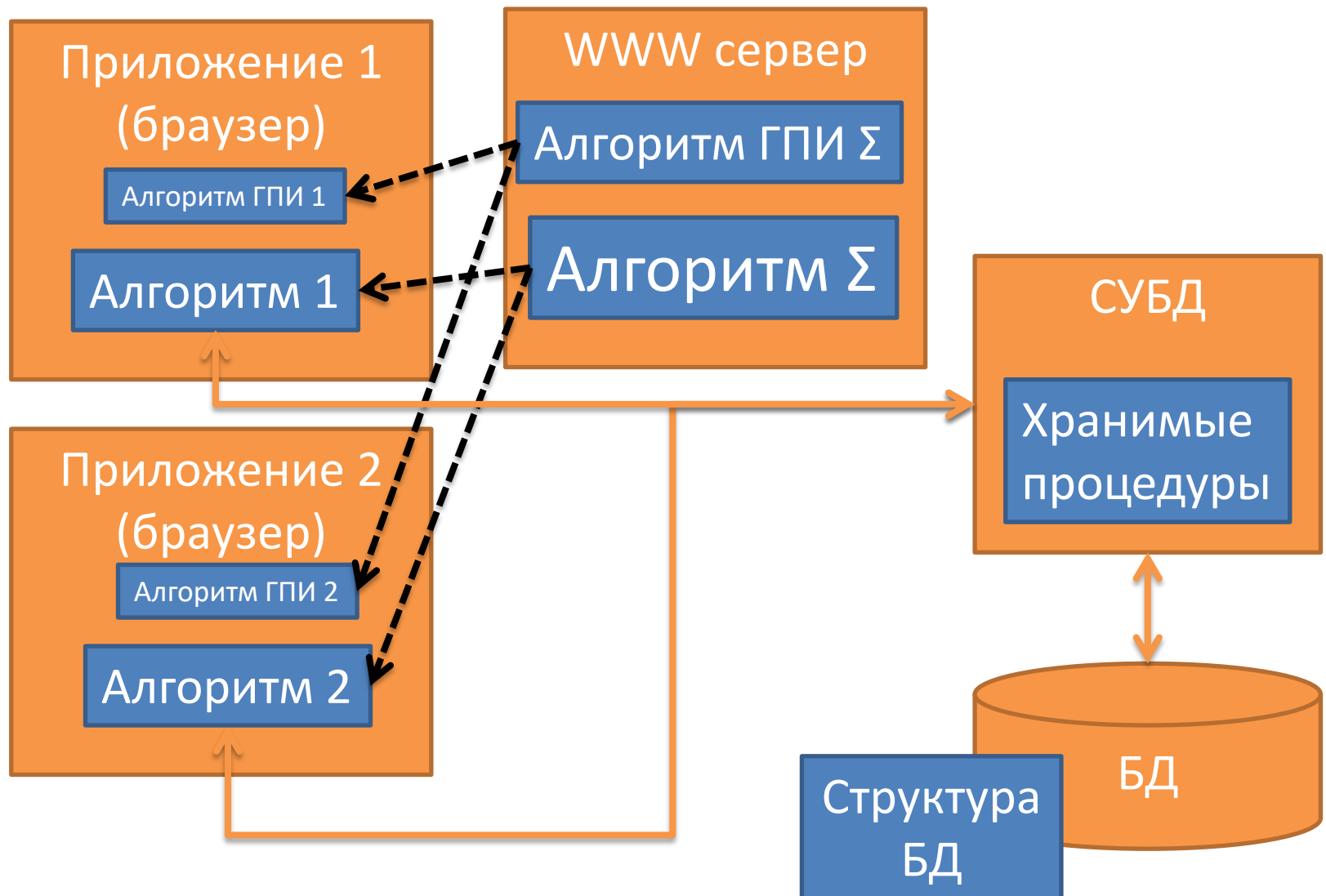
Поведение РИС: WWW сервер



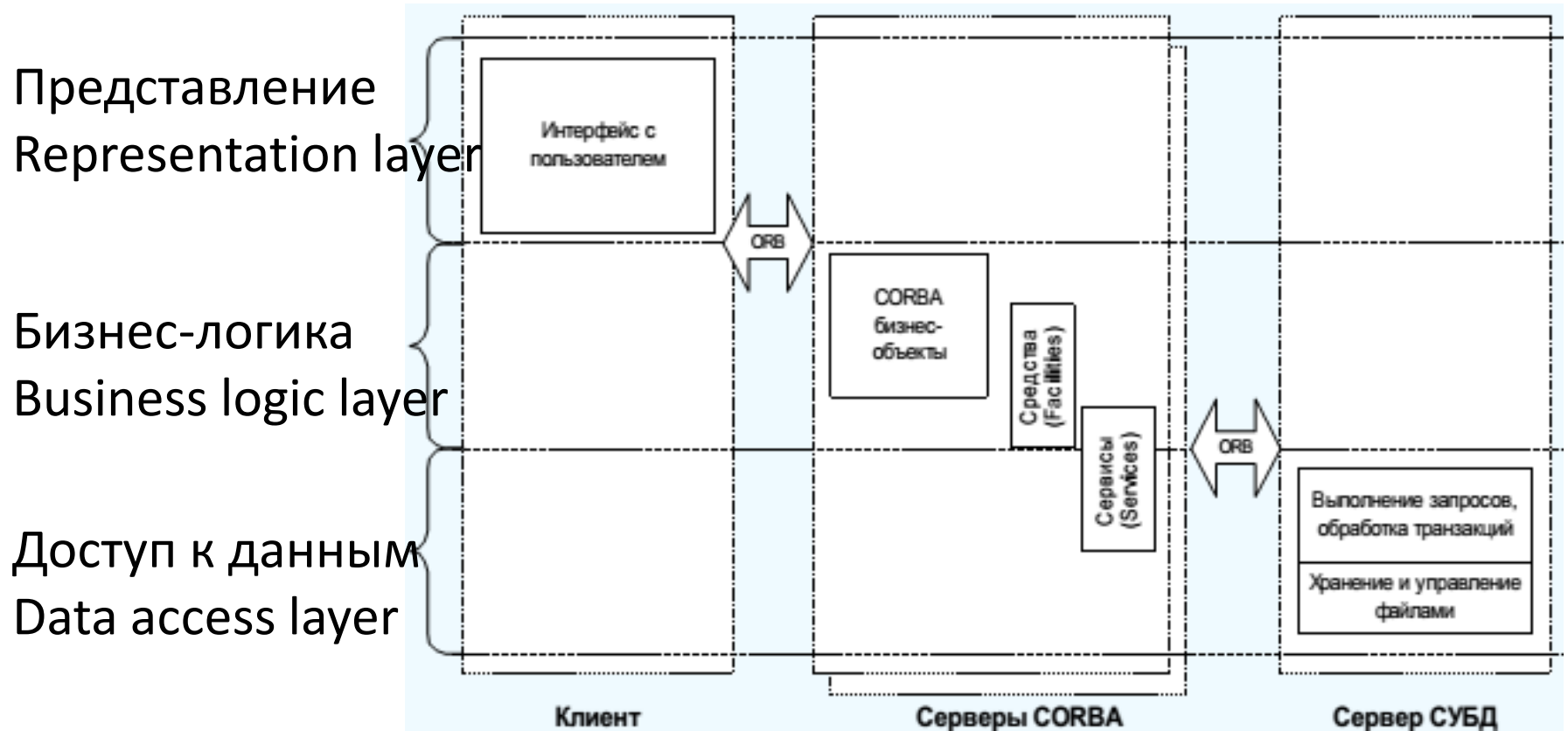
Поведение РИС: Intranet



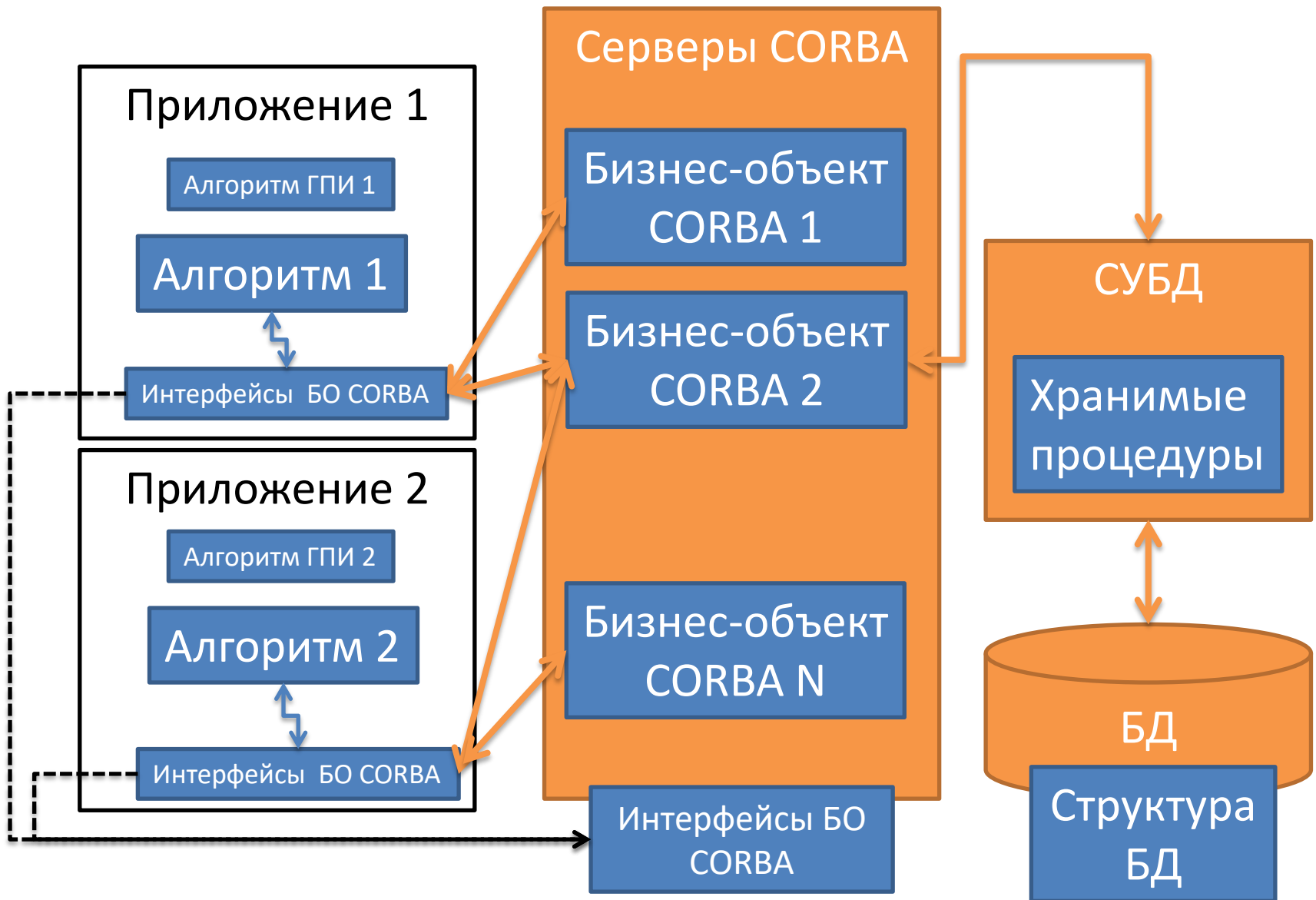
Поведение РИС: сервер Intranet



Поведение РИС: Corba



Поведение РИС: CORBA



Определения, уровни

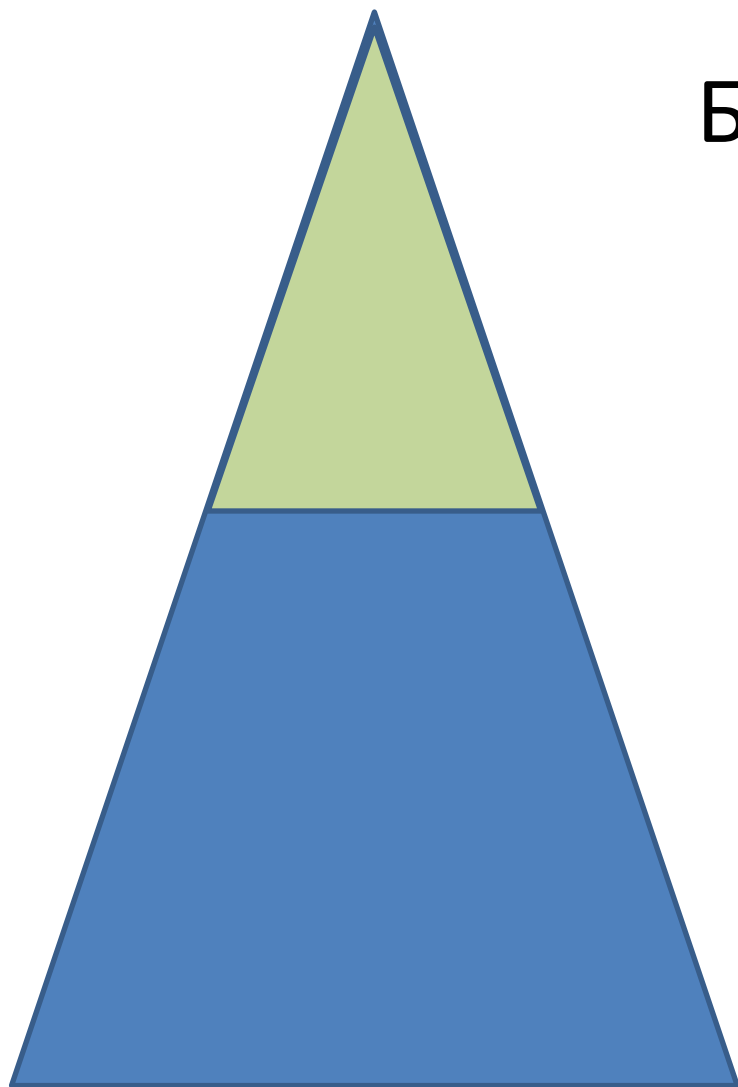
Описание бизнес-логики

DATABASE-CENTRIC АРХИТЕКТУРА

Database-centric Architecture – 1/8

- Архитектура приложений и РИС, в которой БД играет ключевую, центральную роль. Ее характерные особенности:
 - Архитектура клиент-сервер
 - На сервере выполняется реляционная СУБД
 - Логика поведения РИС основывается на содержимом таблиц
 - Используются хранимые процедуры, выполняющиеся на серверах СУБД
 - Для взаимодействия между компонентами РИС (клиентами) используется совместный доступ к СУБД

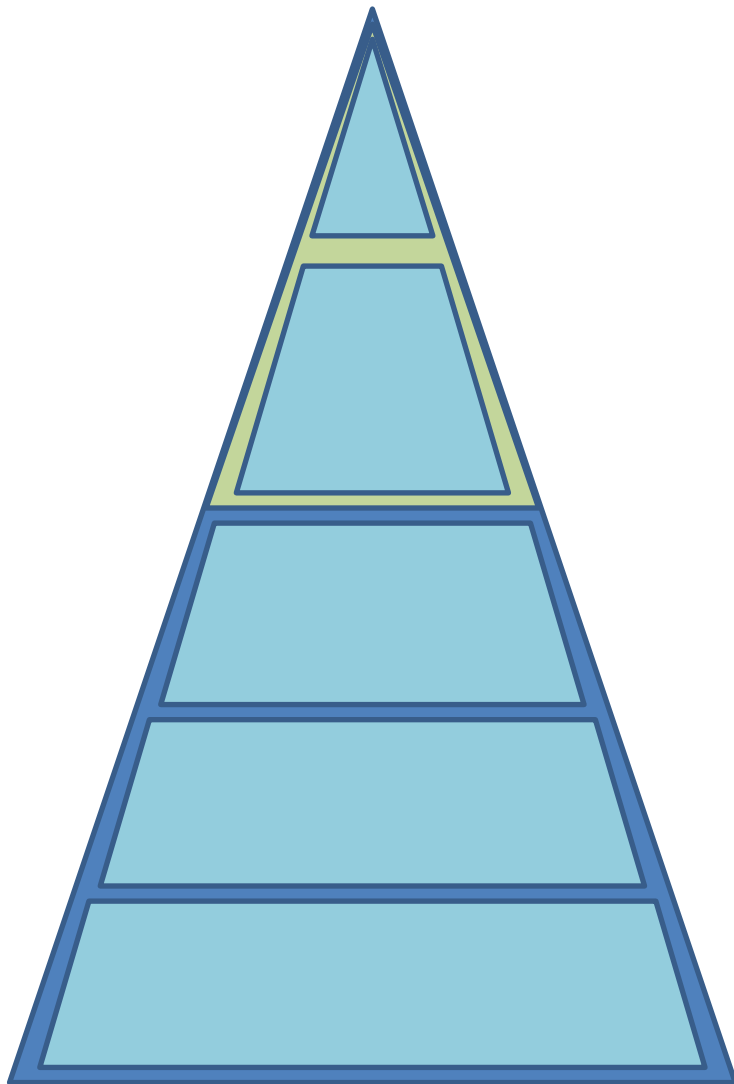
Database-centric Architecture – 2/8



Бизнес уровень

Уровень данных

Database-centric Architecture – 3/8



Пользовательский интерфейс
User Interface Layer

Сервисные приложения
Business Logic Layer

Хранимые процедуры
Data Access Layer

Представления данных
Logical Data Layer

Собственно данные
Physical Data Layer

Database-centric Architecture – 4/8

- Собственно данные (таблицы и индексы)
 - 3 КНФ
 - Главные ключи
 - Ограничения на внешние ключи
 - Общие ограничения бизнес-логики

Database-centric Architecture – 5/8

- Представления данных (Views)
 - Собирает данные из нескольких таблиц
 - Дает приложениям одинаковый «взгляд» на данные
 - «Разнормализует» данные
 - Содержит преобразованные данные:
 - Вычисленные из других
 - Переведенные (во всех смыслах)
 - Позволяет предохранить некоторые данные [от изменения] «спрятав» их и не включая их в отчеты

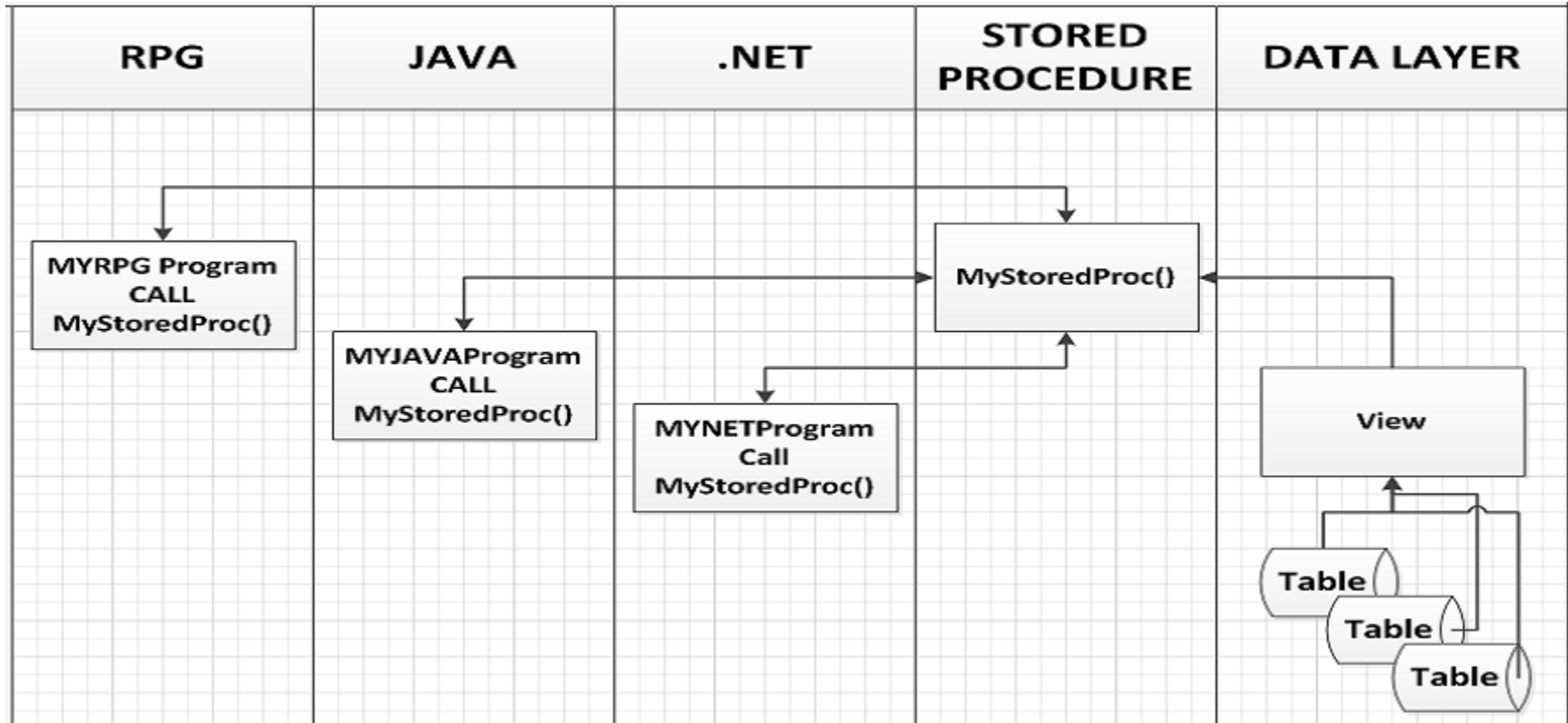
Database-centric Architecture – 6/8

- Хранимые процедуры отвечают за:
 - Выборку
 - Обновление / изменение
 - Добавление
 - Удаление
 - В качестве результата могут возвращать наборы данных

Database-centric Architecture – 7/8

- Сервисные приложения
 - Специфические для РИС правила редактирования и бизнес-правила
 - Набор утилит
- Пользовательский интерфейс

Database-centric Architecture – 8/8



Что есть внешний мир

Внешние интерфейсы

**ПРЕДСТАВЛЕНИЕ О ВНЕШНЕМ
МИРЕ**

Объекты внешнего мира

- Объект наблюдения / управления (ОНУ)
 - Доступен для наблюдения
 - Доступен для влияния
- Взаимодействующие пользователи
 - Вместе с РИС образуют систему более высокого уровня
 - Не обязательно согласованно действуют (друг с другом и с целями РИС)
- Среда функционирования
 - Определяет правила поведения конкретных объектов
 - Влияет на ОНУ неконтролируемым образом
 - Зависит от поведения ОНУ

С чем взаимодействует РИС

- [\rightarrow РИС] Собственно физический внешний мир в целом (окружающая среда)
- Объект наблюдения / управления:
 - [\rightarrow РИС] Датчики
 - [РИС \rightarrow] Акторы
- [\rightarrow РИС \rightarrow] Взаимодействующие с РИС:
 - Пользователи
 - Другие ИС

Объект наблюдения / управления – 1/2

- Доступен РИС опосредованно:
 - Датчики
 - Акторы
 - Ввод данных пользователями
 - Воздействие на объект пользователями по рекомендации системы
- Представлен в идеализированном виде (в виде модели):
 - Данных о состоянии ОНУ
 - Понимания «общей логики» его функционирования

Объект наблюдения / управления – 2/2

- Обратная связь
 - Состояние / поведение ОНУ может изменяться в зависимости от действий / поведения РИС
 - Осуществляется через Акторы или опосредованно через пользователей/другие системы

Внешние интерфейсы РИС

- Конфигурационный интерфейс:
 - Конфигурационные файлы
 - GUI настройки
- Пользовательский интерфейс
 - Например, GUI
- Входные интерфейсы:
 - Каналы получения информации от датчиков
 - Входные интерфейсы для данных, подлежащих обработке
- Выходные интерфейсы для результатов
- Интерфейсы мониторинга и диагностики

Внешний мир

- Окружающая среда
 - Солнечная система (движение планет, солнечный ветер, космический мусор, и т.д.)
 - Земля (атмосфера, космические лучи, распространение радиоволн, приливы, и т.д.)
 - Районы Земли (климат, солнечная освещенность, и т.д.)
- Правила поведения
 - Законы физического мира
 - Бизнес-правила взаимодействия ОНУ с внешним миром
 - Прогнозирование результатов воздействий
- Структура и состояние внешнего мира

Солнечная система, Земля, РФ

ФИЗИЧЕСКИЙ МИР

Физический мир: стандарты – 1/5

- Международные:
 - ISO/IEC 18025, Environmental Data Coding Specification (EDCS)
 - ISO/IEC 18026, Spatial Reference Model (SRM)
- Солнечная система:
 - ГОСТ Р 25645.167-2005 Космическая среда (естественная и искусственная). Модель пространственно-временного распределения плотности потоков техногенного вещества в космическом пространстве

Физический мир: стандарты – 2/5

- Солнечная система (продолжение):
 - ГОСТ 25645.135-86 Поле магнитное межпланетное. Пространственная модель регулярного поля
 - ГОСТ 25645.136-86 Ветер солнечный. Состав, концентрация частиц и скорость
- Земля:
 - ГОСТ 25645.127-85 Магнитосфера Земли. Модель магнитного поля магнитосферных токов
 - ГОСТ 4401-81 Атмосфера стандартная. Параметры

Физический мир: стандарты – 3/5

- Земля:
 - ГОСТ 24631-81 Атмосферы справочные. Параметры
 - ГОСТ 24728-81 Ветер. Пространственное и временное распределение характеристик
 - ГОСТ 25645.146-89 Ионосфера Земли. Модель глобального распределения концентрации, температуры и эффективной частоты соударений электронов
- Районы Земли:
 - ГОСТ 26351-84 Модель влажности воздуха над территорией СССР

Физический мир: стандарты – 4/5

- Районы Земли:
 - ГОСТ 24482-80 Макроклиматические районы земного шара с тропическим климатом. Районирование и статистические параметры климатических факторов для технических целей
- Законы физического мира:
 - ГОСТ 22312-77 Таблица аэродинамическая. Динамические давления и температуры торможения воздуха за прямой ударной волной для скорости полета от 4000 до 7000 км/ч

Физический мир: стандарты – 5/5

- Законы физического мира:
 - ГОСТ 25431-82 Таблица динамических давлений и температур торможения воздуха в зависимости от числа Маха и высоты полета

Международная стандартная атмосфера

- Международная организация гражданской авиации (ICAO) определяет международную стандартную атмосферу (International Standard Atmosphere, ISA):
 - Условное вертикальное распределение температуры, давления и плотности воздуха в атмосфере Земли
 - Рассчитывается по барометрической формуле с определёнными в стандарте параметрами
 - Для уровня моря (геоид):
 - Температура воздуха: 15 °C
 - Атмосферное давление: 101325 Па (101,3 кН/м², 760 мм рт. ст.)
 - Относительной влажностью: 0 %

Хранение данных: где?

Разное представление: какое?

Разное представление: зачем?

Преобразование: как?

СЕРИАЛИЗАЦИЯ И МАРШАЛИНГ

Хранение данных: где? – 1/4

- БД → потому что «ну как же без БД!»
 - Вне приложений
 - Доступ процесса через IPC с процессом СУБД
- Объекты (экземпляры классов) → потому что «ну как же без ООП!»
 - В виртуальной памяти процесса
 - Доступ прямой / через методы классов

Хранение данных: где? – 2/4

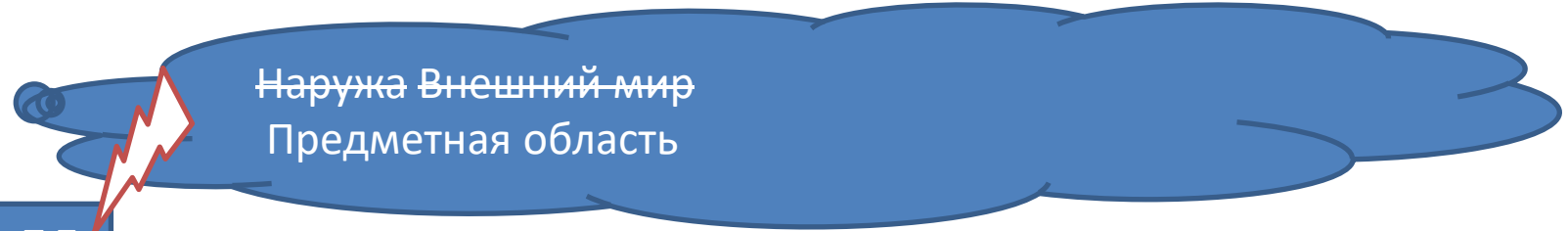
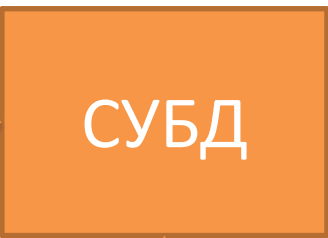
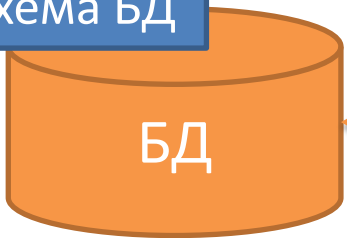
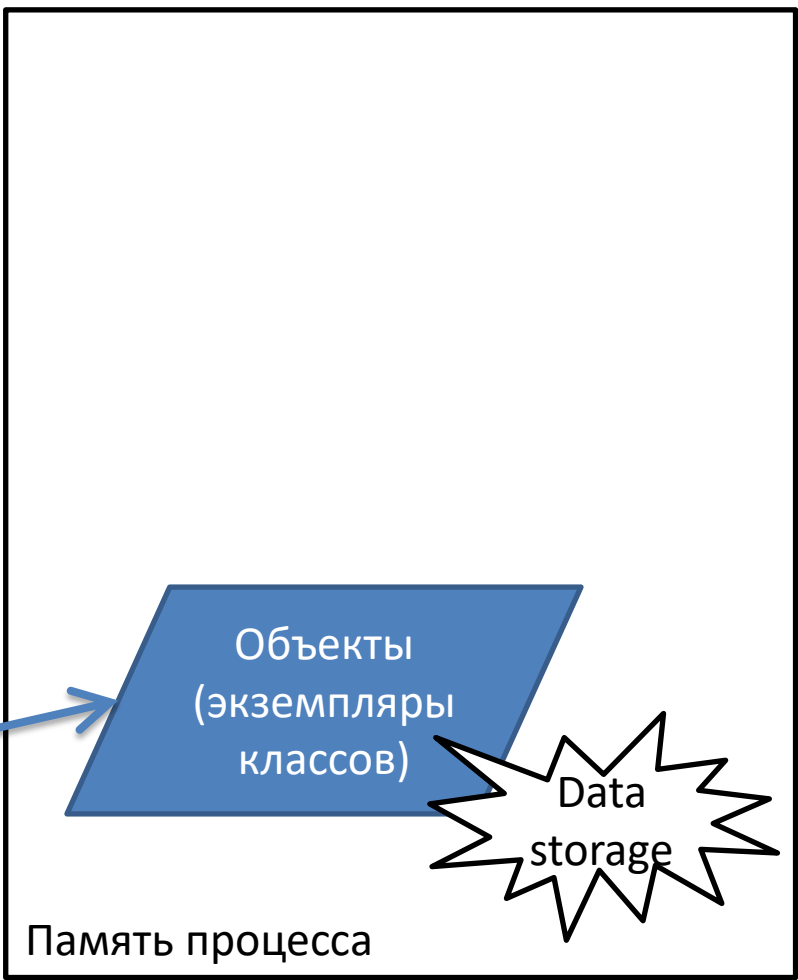


Схема БД



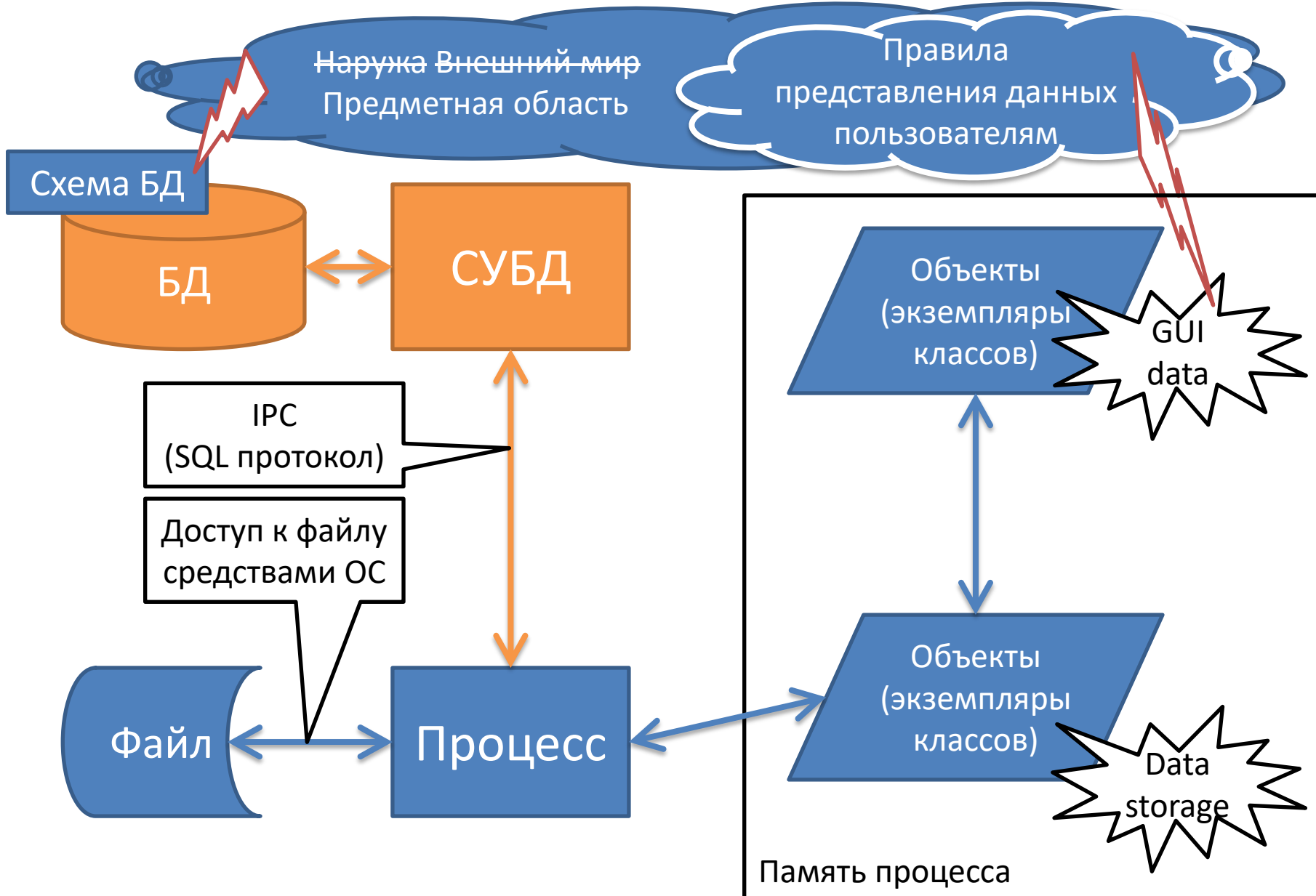
IPC
(SQL протокол)



Хранение данных: где еще? – 3/4

- Специальные объекты ГПИ → потому что «ну как же без GUI!»
 - Как минимум, временно хранят редактируемые пользователем данные
 - А еще набор операций (методов) разный
- Файлы → потому что «ну, пусть тоже будут...»
 - Доступ из приложения через API ОС
 - Дополнительные возможности
 - Например, могут редактироваться без СПО, «обычными» текстовыми редакторами

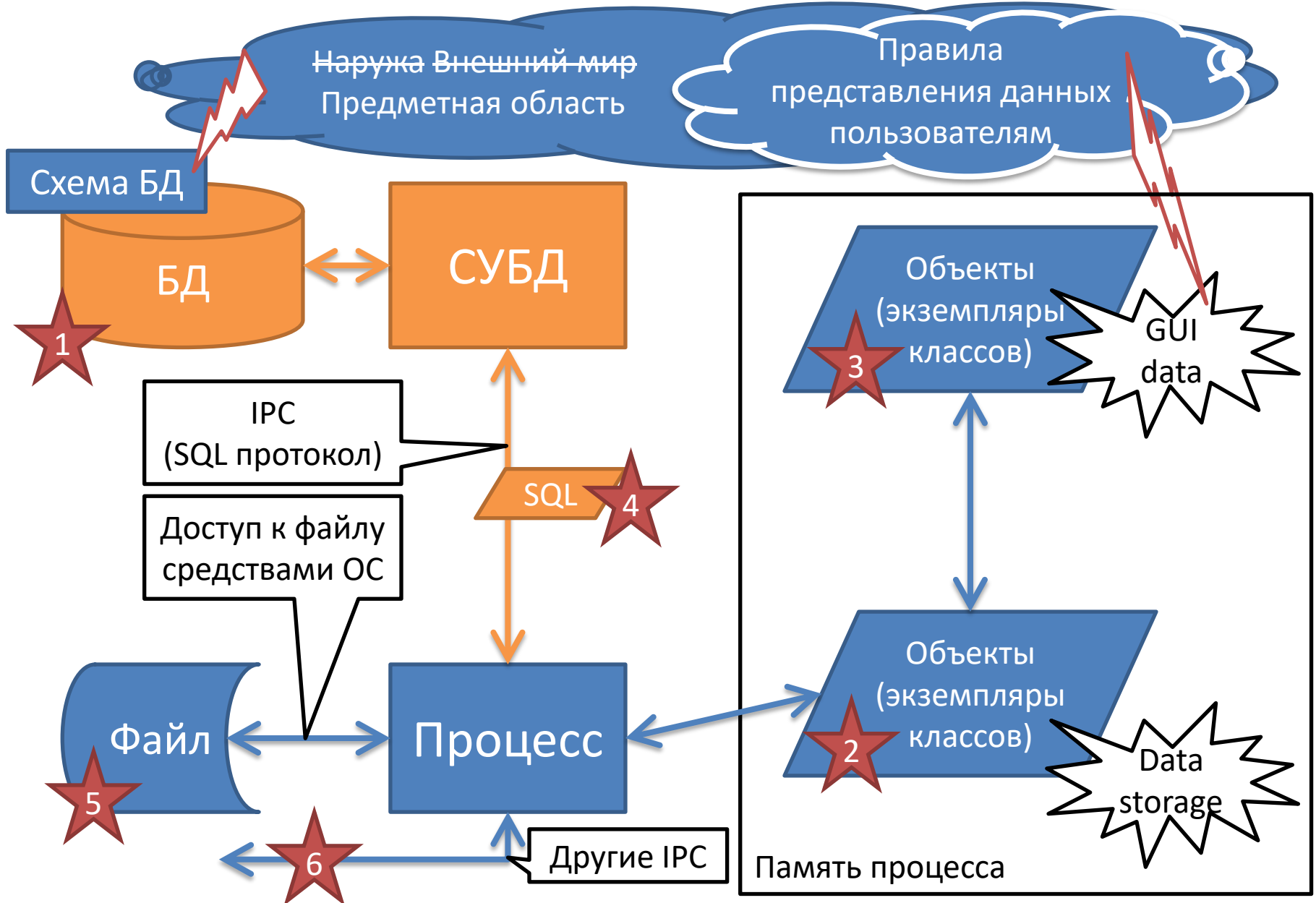
Хранение данных: где? – 4/4



Представление данных – 1/5

- Ой! Забыли промежуточный формат: SQL запрос/ответ, тоже хранящий данные
- Ну тогда и... сетевые протоколы – если есть
- И вообще любой IPC требует описания конкретных форматов данных:
 - Файлов
 - Сообщений при обмене ими
 - Организации памяти при совместном доступе
 - Вызываемых функций при использовании RPC

Представления данных: какие? – 2/5



Представления данных: зачем? – 3/5

- Разные представления данных требуются для достижения большей эффективности:
 - Хранения данных
 - Например: компактность, простота манипуляций, простота проверки и поддержания целостности
 - Обработки данных:
 - Например: поиска, добавления, удаления, изменения
 - Взаимодействия с внешними пользователями (людьми, а также использующими и обеспечивающими системами):
 - Способ представления
 - Протокол обмена через IPC

Представления данных: какие? – 4/5

- Основным представлением данных является внутреннее (в памяти процесса). Обычно оптимизировано для обработки данных
 - Примечание: представление, связанное с ГПИ, оптимизировано для отображения и редактирования пользователем – это тоже обработка, хотя вычислительные затраты и сравнительно невелики

Представления данных: какие? – 5/5

- Другие представления можно разделить на:
 - Представления хранения – используются для хранения данных с простыми манипуляциями: сохранить, загрузить (реже: изменить хранимое)
 - Примечание: Предполагается, что процесс **сам** использует это представление для [со]хранения / загрузки данных
 - Представления обмена – используются для взаимодействия с другими РИС.
 - Зависят от предметной области:
 - Требуемой от РИС функциональности
 - Процедур взаимодействия (прагматики) с другими РИС: обеспечивающим и использующими

Преобразование данных – 1/2

- Использование разных представлений данных требует возможности преобразовывать данные между форматами
- Преобразование идет между внутренним представлением и внешним. Например:
 - Внутреннее → формат для хранения
 - Сетевое представление → внутреннее

Преобразование данных – 2/2

- Преобразование может служить для:
 - Сохранения / восстановления данных (внутреннее \leftrightarrow формат хранения)
 - Один и тот же исполнимый модуль – перед выключением сохранили состояние, дальше продолжили с той же точки
 - Обмена данными
 - Симметрично между процессами, порожденными одинаковыми исполнимыми файлами (внутреннее \rightarrow сетевое или формат хранения \rightarrow внутреннее)
 - Несимметрично, между по-разному устроенными процессами (внутреннее 1 \rightarrow сетевое или формат хранения \rightarrow внутреннее 2)

Сериализация и маршалинг – 1/2

- Маршалинг – процесс преобразования представления объекта в памяти в **формат данных**, пригодный для хранения или передачи
- Сериализация – процесс перевода какой-либо структуры данных в **последовательность битов**

Сериализация и маршалинг – 2/2

- Сложности:
 - Различная последовательность байт на разных платформах
 - Различное представление одинаковых типов данных
 - Различные кодировки строк
- Варианты маршалинга
 - Запись в бинарные файлы
 - Хранение в БД
 - etc.

Примеры форматов маршалинга

- Специфические для связующего ПО (обмен между процессами):
 - Парадигма RPC: Microsoft: COM, .NET, Mozilla: XPCOM (Cross Platform Component Object Model), CORBA, etc.
 - Парадигма обмена сообщениями: IEEE-1516: HLA OMT, OMG DDS: DDSI, etc.
- Специфические для ЯП (хранение/восстановление):
 - Java: JAXB (Jakarta XML Binding): class \leftrightarrow xml
 - Java: методы toString(): class \leftrightarrow строка
 - JavaScript: JSON

Форматы файлов

- Требования:
 - Человеко-читаем и редактируем стандартными редакторами
 - Есть готовые способы манипуляции (библиотеки)
- Решения:
 - INI
 - XML (eXtensible Markup Language)
 - JSON (JavaScript Object Notation)
 - YAML (сначала: Yet Another Markup Language, сейчас: YAML Ain't Markup Language)

Варианты хранения

Пример: DDS Reliability QoS

Пример: DDS Durability QoS

Долговременное хранение

Пример: CORBA

ХРАНЕНИЕ СОСТОЯНИЯ

Состояние

- Объекты в РИС могут быть:
 - С внутренним состоянием (stateful)
 - Атрибуты в определении интерфейса – достаточное, но не необходимое условие
 - Внутренние переменные экземпляра
 - Без внутреннего состояния (stateless)
- Варианты хранения:
 - Долговечными (persistent) – продолжающие существовать и после завершения процесса
 - Временными (transient)

Состояние: примеры

	Stateful	Stateless
Persistent	Служба именования REST клиент Координаты объекта	HTTP протокол REST сервер
Transient	Координаты объекта	Команда управления выполнением

Reliability QoS – 1/2

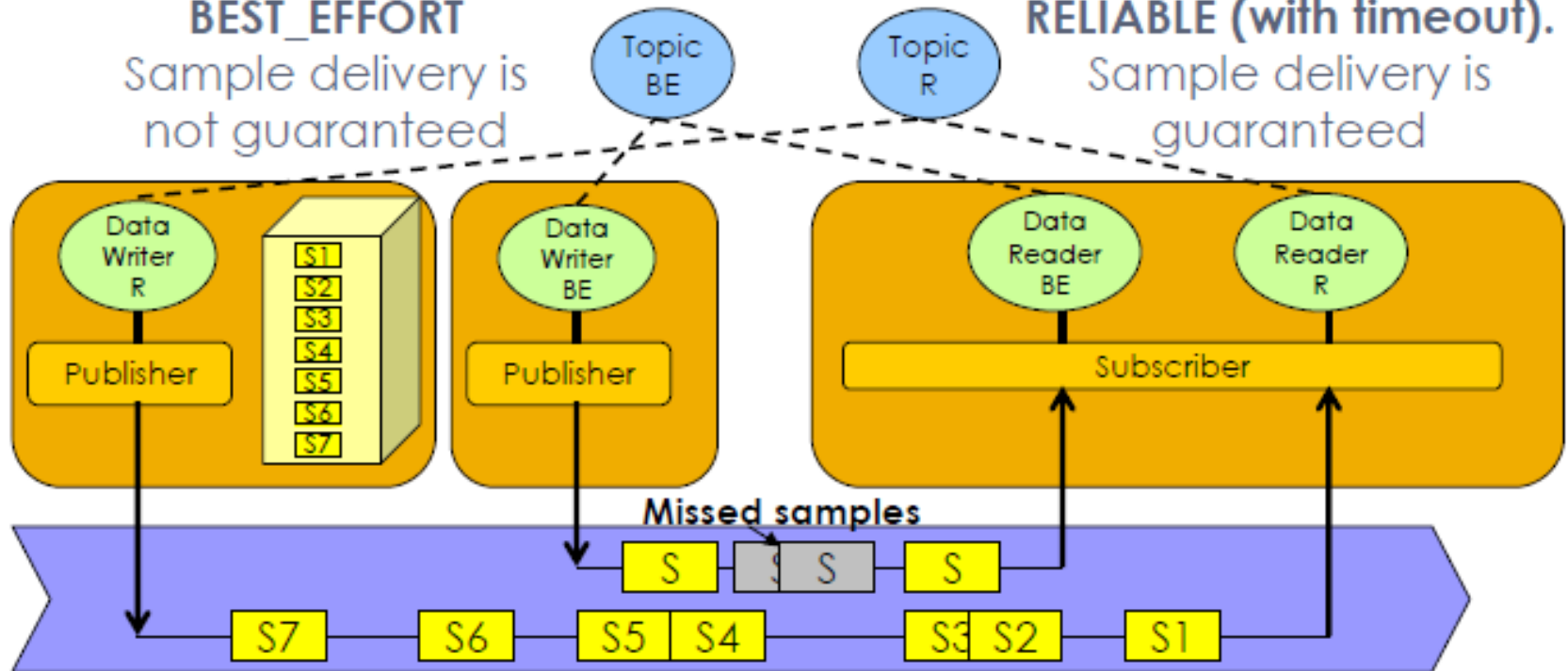
- В DDS (и HLA) имеется QoS описывающий способ доставки сообщений (данных)
 - Best_effort
 - Производится [одна] попытка доставки сообщения, при неудаче сообщение теряется
 - Обычно реализуется поверх протокола UDP
 - Reliable
 - Доставка сообщения гарантируется, сбой доставки расценивается как отказ системы или ее элемента
 - В СПО PCPM реализуется поверх протокола TCP
 - В DDS реализуется поверх протокола multicast UDP

Reliability QoS: DDS – 2/2

RELIABLE > BEST_EFFORT

BEST_EFFORT
Sample delivery is not guaranteed

RELIABLE (with timeout).
Sample delivery is guaranteed



DDS: Durability QoS – 1/2

- В DDS имеется QoS описывающая варианты хранения истории изменений:
 - Volatile (непостоянный, изменчивый)
 - История не хранится
 - Transient_local
 - История хранится локально в памяти отправителя
 - Transient
 - История хранится в сервисе TRANSIENT (в оперативной памяти)
 - Persistent
 - История хранится в постоянной памяти

DDS: Durability QoS – 2/2

PERSISTENT > TRANSIENT > TRANSIENT_LOCAL > VOLATILE

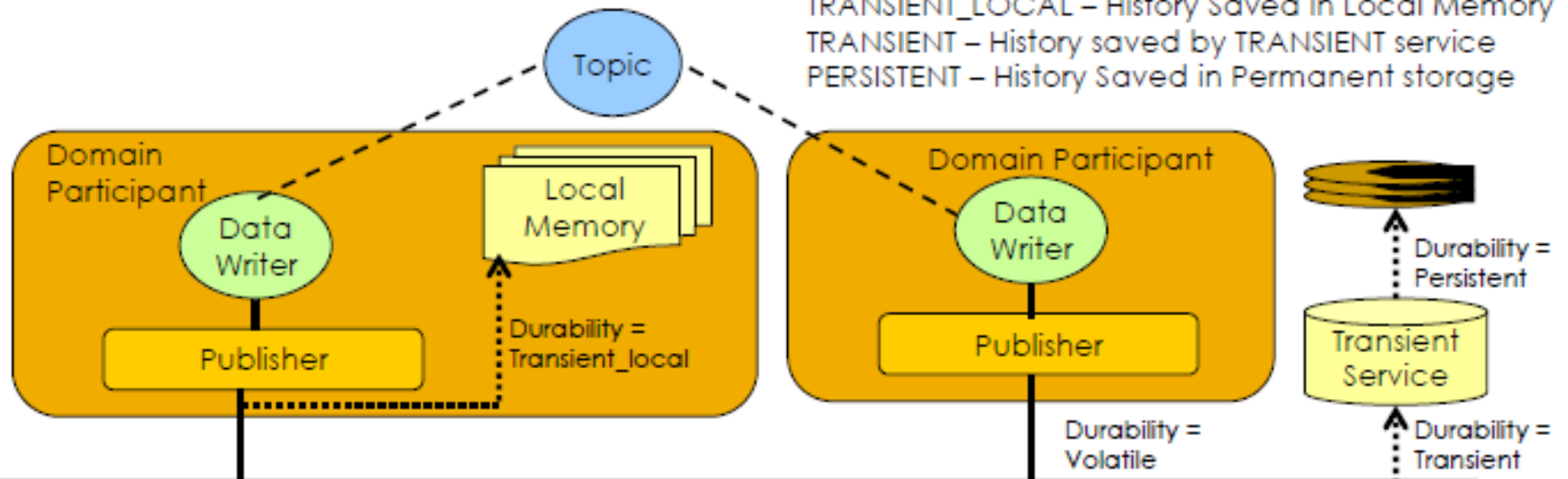
Durability Kind:

VOLATILE – No Instance History Saved

TRANSIENT_LOCAL – History Saved in Local Memory

TRANSIENT – History saved by TRANSIENT service

PERSISTENT – History Saved in Permanent storage



saved in Transient affected by QoS: History and QoS: Resource_Limits

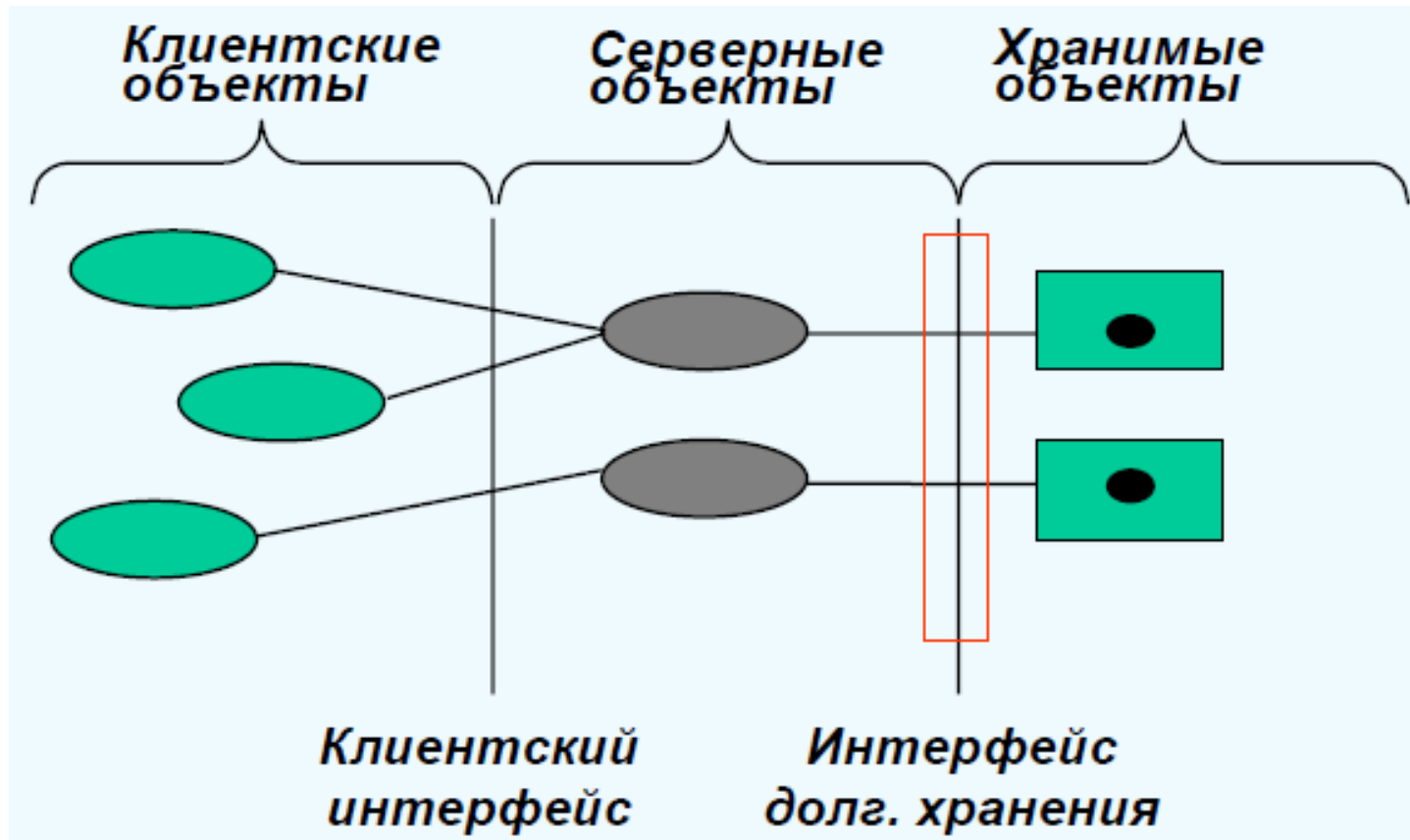
Долговременное хранение – 1/6

- Долговременное хранение - способность объекта переживать процесс, в котором он находится
 - Состояние должно запоминаться на промежуток деактивации и активации
- В качестве хранилища может использоваться:
 - Файловая система
 - СУБД (любого вида: реляционная, объектная)
 - Постоянная оперативная память

Долговременное хранение – 2/6

- За долговременное хранение в РИС может отвечать:
 - Сервис middleware
 - Каждое приложение самостоятельно
 - Выделенное приложение
- Например, в HLA за реализацию долговременного хранения отвечает:
 - Состояние middleware и выполняющейся распределенной модели в целом: сам HLA RTI
 - Состояние внутренних переменных моделей сохраняются самими приложениями

Долговременное хранение: CORBA – 3/6



- Пример долговременного хранения в CORBA

Долговременное хранение: CORBA – 4/6

- Хранилище данных (datastorage) – общая область хранения всех данных
 - Например: БД
- Область хранения (storage home) – отдельный контейнер с долговременные представлениями хранимых объектов
 - Например: таблица
- Объект хранения (storage object) – представление одного хранимого объекта
 - Например: запись в таблице РБД
- Тип хранения (storage type) – определяет интерфейс объекта хранения

Долговременное хранение: CORBA – 5/6

- Воплощение объекта хранения (storage object incarnation) – представление объекта хранения на ЯП
 - NB! представление в памяти зависит от ЯП
- Воплощение области хранения (storage home incarnation) – представление области хранения на ЯП
 - NB! представление в памяти зависит от ЯП
- Ключ – атрибут, однозначно идентифицирующий объект хранения в пределах области хранения
- Сессия – логическая связь между хранилищем данных и процессом, в котором выполняется объект-сервер

Долговременное хранение: CORBA – 6/6

