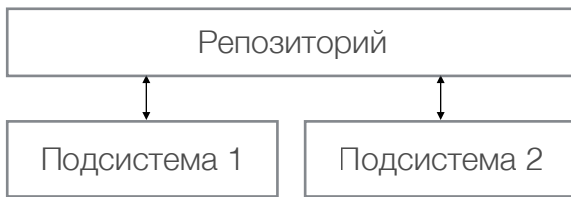


### Модель «Репозиторий»

Предполагает что есть централизованное хранилище и некоторые подсистемы, которые с ними работают.

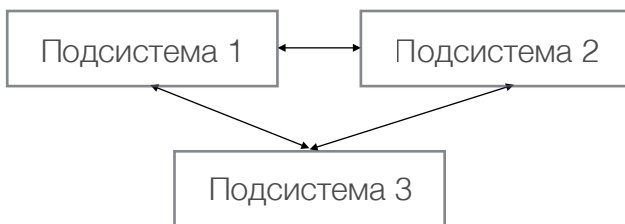


Для каждой из подсистем можно определить входные и выходные данные. Выходные данные одной могут быть входными для следующей.

Организация хранения и передачи данных между системами:

1. Локальное хранилище в каждой подсистеме. Если данным из 2ой подсистемы требуются данные из 1ой между ними необходимо

реализовать передачу данных (каждая с каждой).



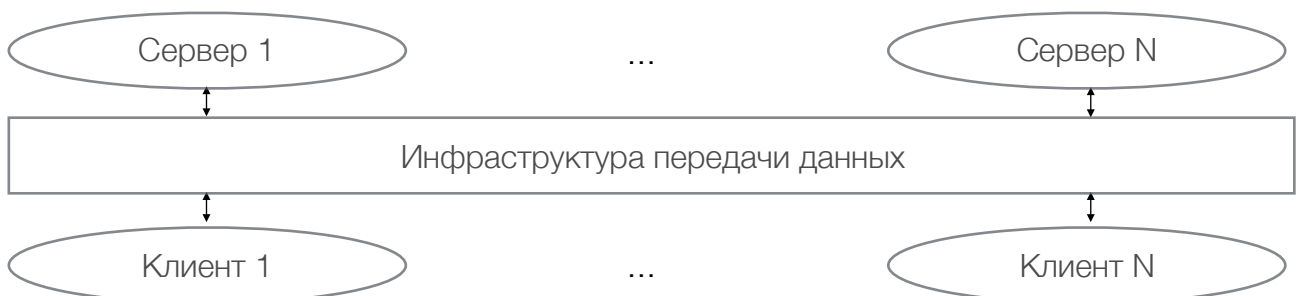
### 2. Использование централизованного хранилища (модель «Репозитория»)

Есть центральная БД, схема которой реализована таким образом, что бы отвечать требованиям. Применяется общая политика безопасности для данных всех подсистем. Управление данными выполняется централизованно и нет необходимости тратить ресурсы подсистем на это.

ПРИМЕР: приложение и одна БД на сервере

- + Не нужно передавать большие объемы данных между локальными подсистемами
- + Надёжность системы в целом т.к. уменьшается связность подсистем
- + Скорость работы
- + Ослабляется связность подсистем, подсистемы, производящие данные не должны знать как они используются
- + Все операции связанные с администрированием выполняются централизованно, не требуя участия подсистем
- + Легко добавить новую подсистему, единственное требование - согласованность с моделью данных («Репозиторий»)
- К данным различных подсистем применяется общая политика безопасности и требования надёжности (плохо ибо разные подсистемы могут работать с разными данными)
- Единая универсальная модель, которая не может быть эффективна для всех подсистем (т.к. не может учитывать все их требования)
- При проектировании БД требуется идти на компромис между требованиями подсистем
- Слишком высокие нагрузки на централизованное хранилище данных, повышенные требования по производительности (решение - распределение)

### Архитектура «Клиент-Серверная»



Распределённая модель, которая предполагает наличие компонентов или подсистем 3х основных типов:

- 1) Набор серверов. Сервер – программная система, предоставляющая сервисы по запросу других компонентов
- 2) Набор клиентов. Клиент – программная подсистема, которая умеет запрашивать сервис у Сервера и получать и интерпретировать результат этого сервиса
- 3) Инфраструктура передачи данных (сеть), посредством которой Клиенты передают Серверам запрос и необходимые данные и получают результат

В общем случае Клиент с Сервер могут быть физически на одном устройстве, но в этом случае теряются основные преимущества.

Для запроса сервиса Клиент должен знать и использовать протокол Сервера (адрес Сервера, запрос который можно отправить, входные аргументы которые необходимо при этом передать, какой результат ожидается в ответ)

+ Распределённость

+ Простая интеграция клиентов и серверов (для сервера нужна публикация протоколов, для клиентов ничего)

+ Максимально слабая связность системы

– Снижение скорости за счет передачи данных между физически удалённым клиентом и сервером

– Надёжность передачи данных по сети - ключевой фактор надёжности системы в целом

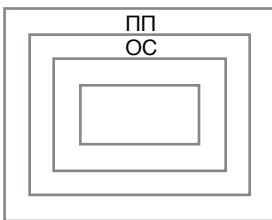
Модель «Абстрактной машины» aka «Многоуровневая модель»

Предлагает компоненты организовывать вложено.

Каждая подсистема предоставляет некоторый

Реализация программной системы в виде вложенных уровней, каждый из которых описывает абстрактную машину, реализующую набор команд на основе которых разработан компонент следующего уровня.

Программа использует Виртуальную машину Java, которая использует ОС, которая использует аппаратное обеспечение.



+ Обеспечивает хорошую переносимость ПП за счет того, что за взаимодействие с аппаратным обеспечением реализуется одним уровнем

+ Каждый компонент взаимодействует только с 2-мя соседними уровнями

– Снижение скорости работы из-за необходимости перенаправления запроса вызовов по уровням с верхнего на нижний уровня

– Более сложная архитектура

– Компонент верхнего уровня не может работать с компонентами более чем на уровень ниже

– Может потребоваться взаимодействие многих уровней, что нарушает архитектуру