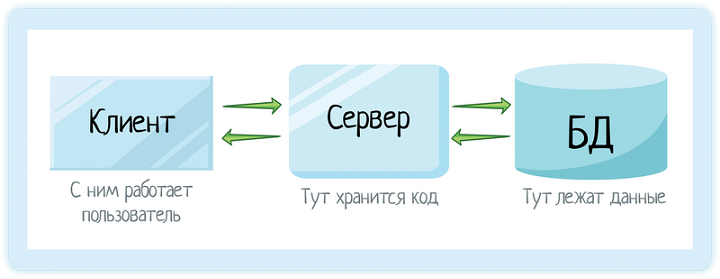
**Клиент-серверная архитектура**

  
  
***Клиент*** – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

***Сервер*** – очень мощный компьютер или специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Особенности такой модели заключаются в том, что пользователь отправляет определенный запрос на сервер, где тот системно обрабатывается и конечный результат отсылается клиенту. В возможности сервера входит одновременное обслуживание сразу нескольких клиентов.

Если одновременно поступает более одного запроса, то такие запросы устанавливаются в определенную очередь и сервером выполняются по очереди.

***Параметры, которые могут реализоваться на стороне сервера:***

1. Хранение, защита и доступ к данным;
2. Работа с поступающими клиентскими запросами;
3. Процесс отправки ответа клиенту.

***Параметры, которые могут реализоваться на стороне клиента:***

1. Площадка по предоставлению пользовательского графического интерфейса;
2. Формулировка запроса к серверу и его последующая отправка;
3. Получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

***Архитектура системы клиент***-сервер формулирует принципы виртуального общения между локальными компьютерами, а все правила и принципы взаимодействия находятся внутри протокола.

***Сетевой протокол*** – это особый набор правил, на основании которого выполняется точное взаимодействие между компьютерами внутри виртуальной сети.

**ВИДЫ СЕТЕВЫХ ПРОТОКОЛОВ**

TCP/IP – совокупность протоколов передачи информации. TCP/IP – это особое обозначение всей сети, которая функционирует на основе протоколов TCP, а также IP.

TCP – вид протокола, который является связующим звеном для установки качественного соединения между 2 устройствами, передачи данных и верификации их получения.

IP – протокол, в функции которого входит корректность доставки сообщений по выбранному адресу. При этом информация делится на пакеты, которые могут поставляться по-разному.

MAC – вид протокола, на основании которого происходит процесс верификации сетевых устройств. Все устройства, которые подключены к сети Интернет, содержат свой оригинальный MAC-адрес.

ICMP – протокол, который ответственен за обмен данными, но не используется для процесса передачи информации.

UDP – протокол, управляющий передачей данных, но данные не проходят верификацию при получении. Этот протокол функционирует быстрее, чем протокол TCP.

HTTP – протокол для передачи информации (гипертекста), на базе которого функционируют все сегодняшние сайты. В его возможности входит процесс запрашивания необходимых данных у виртуально удаленной системы (файлы, веб-страницы и прочее).

FTP – протокол передачи информации из особого файлового сервера на ПК конечного пользователя.

POP3 – классический протокол простого почтового соединения, который ответственен за передачу почты.

SMTP – вид протокола, который может устанавливать правила для передачи виртуальной почты. Он ответственен за передачу и верификацию доставки, а также оповещения о возможных ошибках.

**КОНЦЕПЦИИ ПОСТРОЙКИ КЛИЕНТ-СЕРВЕРНОЙ СИСТЕМЫ**

**Слабый клиент** – производительный сервер.  
При такой модели весь процесс обработки информации перенесен на мощности сервера, а у пользователя права доступа очень строго ограничены. Сервер начинает отправлять ответ, который вообще не требует дополнительной работы по обработке. Клиент взаимодействует с пользователем: создает и отправляет запрос, принимает входящие итоги и выводит данные на экран пользователя.

**Сильный клиент.**  
Концепция, при которой часть обработки данных предоставляет клиенту. В такой ситуации сервер является простым хранилищем информации, а вся деятельность по обработке и предоставлению данных переносится на ПК пользователя.

Продукт или система, которая говорит о том, что часть обрабатывающей информации предоставляется пользователю.  
При такой ситуации сервер выступает особым хранилищем информации, а вся деятельность по обработке и предоставлению данных может переноситься на ПК пользователя.

**ЕСТЬ 2 ВИДА КЛИЕНТ-СЕРВЕРНЫХ АРХИТЕКТУР:**

1. ***Двухуровневая***, состоящая сразу из 2 узлов:

* сервер, который ответственен за получение входящих запросов и отправку ответа пользователю, применяя при этом собственные ресурсы системы;
* клиент, который может предоставлять пользовательский графический интерфейс.

Особенности работы заключаются в том, что на сервер приходит определенный запрос, потом его обрабатывают и дают напрямую, без дополнительного применения группы внешних ресурсов.

2. ***Трехуровневая*** система состоит из использования таких компонентов:

* предоставление информации – графический пользовательский, прикладной объект в виде сервера приложения;
* менеджмент ресурсов – сервер БД, который может предоставлять данные.

Особенность работы состоит в том, что сразу несколько серверов могут обрабатывать клиентские запросы. Процесс распределения операций может существенным образом снизить нагрузку на используемый сервер.

Трехуровневая архитектура может трансформироваться до многоуровневой, возможностью установки группы дополнительных серверов. Подобная виртуальная архитектура позволяет существенным образом повысить эффективность функционирования информационных систем, а также выполнить оптимизированное распределение части ее программно-аппаратных ресурсов.

Нюансы взаимодействия системы клиент-сервер позволяют пользователям разделять определенный функционал и вычислительную нагрузку между подключенными клиентскими веб-продуктами и серверными приложениями при разнообразных [процессах тестирования](https://testmatick.com/ru/our-qa-services/) (от тестирования БД до замеров общей производительности системы).

Понимание программистами и тестировщиками внутренней архитектуры приложения позволяют им более качественно не только создавать продукт, но и тестировать его, выполняя полноценные проверки от кросбраузерного соответствия до регрессионных тестов.