Техническая сторона хостинга

Модель OSI

- **7 - Прикладной уровень.** Уровень приложений содержит протоколы для обмена данными между процессами.
- **6 - Уровень представления.** Уровень представления обеспечивает общее представление данных, передаваемых между службами уровня приложений. Обеспечивает преобразование кодов (например, побайтная перекодировка из KOI8-P в Windows 1251), форматов файлов, сжатие и распаковку, шифрование и дешифрование данных. Пример протокола — SSL (Secure Socket Layer), обеспечивающий конфиденциальность передачи данных в стеке TCP/IP.
- **5 - Сеансовый уровень.** Сеансовый уровень предоставляет услуги уровню представления для организации его диалога и управления обменом данными.
- **4 - Транспортный уровень.** Транспортный уровень определяет службы для сегментирования, передачи и повторной сборки данных для индивидуальной связи между конечными устройствами. Является также пограничным уровнем между верхними и нижними уровнями.

Модель OSI

- **3 - Сетевой уровень.** Сетевой уровень предоставляет функции для обмена отдельными частями данных по сети между указанными оконечными устройствами.
- **2 - Канальный уровень.** Протоколы канального уровня описывают способы обмена кадрами данных при обмене данными между устройствами по общей среде передачи данных.
- **1 - Физический уровень.** Протоколы физического уровня описывают механические, электрические, функциональные и процедурные средства для активации, поддержания и деактивации физических соединений для передачи бит к и от сетевого устройства.

TCP/IP модель

TCP/IP — сетевая модель (**стек** протоколов), которая описывает способ передачи данных от компьютера-источника информации к компьютеру-получателю.
В модели предполагается прохождение информации через четыре уровня

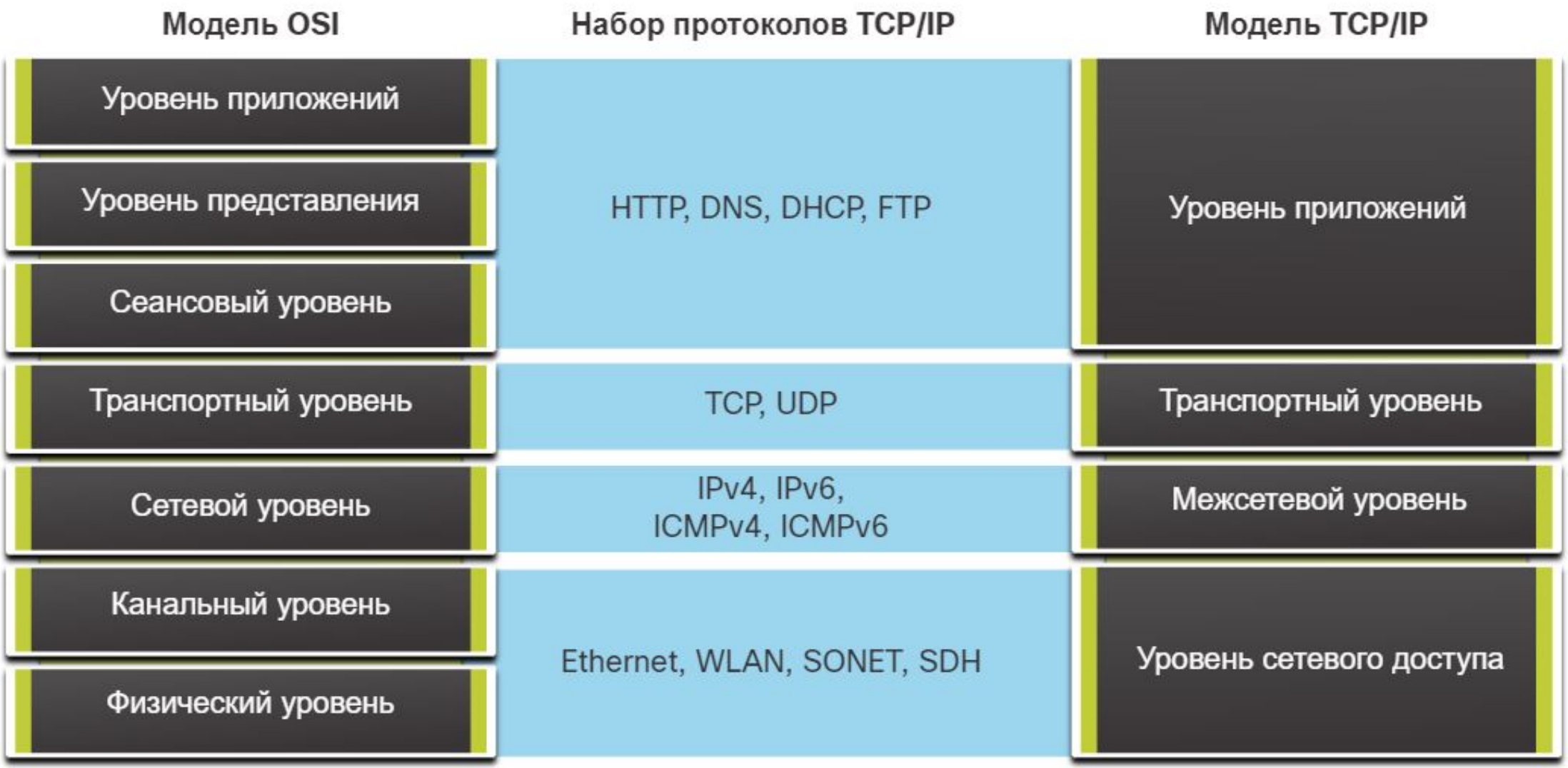
- Прикладной уровень (уровень приложений),
- Транспортный уровень (TCP, UDP),
- Межсетевой уровень (Сетевой уровень)(IPv4, IPv6),
- Уровень сетевого доступа

Каждый уровень прохождения информации описывается правилом (протоколом передачи).
Протокол - это заранее согласованный стандарт (правило), который позволяет двум компьютерам обмениваться данными.

1. На *прикладном уровне* работает большинство сетевых приложений (например, браузер, FTP- клиент, почтовый клиент и многие другие).

   Представляет данные пользователю, а также обеспечивает кодирование и управление диалоговыми окнами.

2. Протоколы *транспортного уровня* (в TCP/IP) определяют, для какого именно приложения предназначены данные (посредством сокета – пары IP:Порт), и гарантируют правильную последовательность прихода данных. На транспортном уровне также происходит сегментация данных.
3. *Межсетевой уровень* разработан для передачи данных из одной сети в другую. Описывает протоколы, определяющие пути передачи данных в сети. Основная единица данных на этом уровне – пакет.
4. *Канальный уровень* описывает способ кодирования данных для передачи единицы данных на физическом уровне (то есть специальные последовательности бит, определяющих начало и конец потока данных, а также обеспечивающие помехоустойчивость).


Модель OSI
Набор протоколов TCP/IP
Модель TCP/IP
Уровень приложений
Уровень представления
Сеансовый уровень
HTTP, DNS, DHCP, FTP
Уровень приложений
Транспортный уровень
TCP, UDP
Транспортный уровень
Сетевой уровень
IPv4, IPv6,
ICMPv4, ICMPv6
Межсетевой уровень
Канальный уровень
Физический уровень
Ethernet, WLAN, SONET, SDH
Уровень сетевого доступа

PDU – protocol data unit

Данные — общий термин для обозначения PDU, используемой на уровне приложений.

Сегмент — PDU транспортного уровня.

Пакет — PDU сетевого уровня.

Кадр (фрейм) — PDU канального уровня.

Биты — PDU физического уровня, используемая при физической передаче данных по средству подключения.

Примечание: Если заголовок транспорта TCP, то это сегмент. Если заголовком транспортного уровня является UDP, то это датаграмма.

Инкапсуляция

По мере того, как данные приложений передаются по стеку протоколов до перемещения через средство сетевого подключения, различные протоколы добавляют в них информацию на каждом из уровней. Это называется процессом **инкапсуляции**.

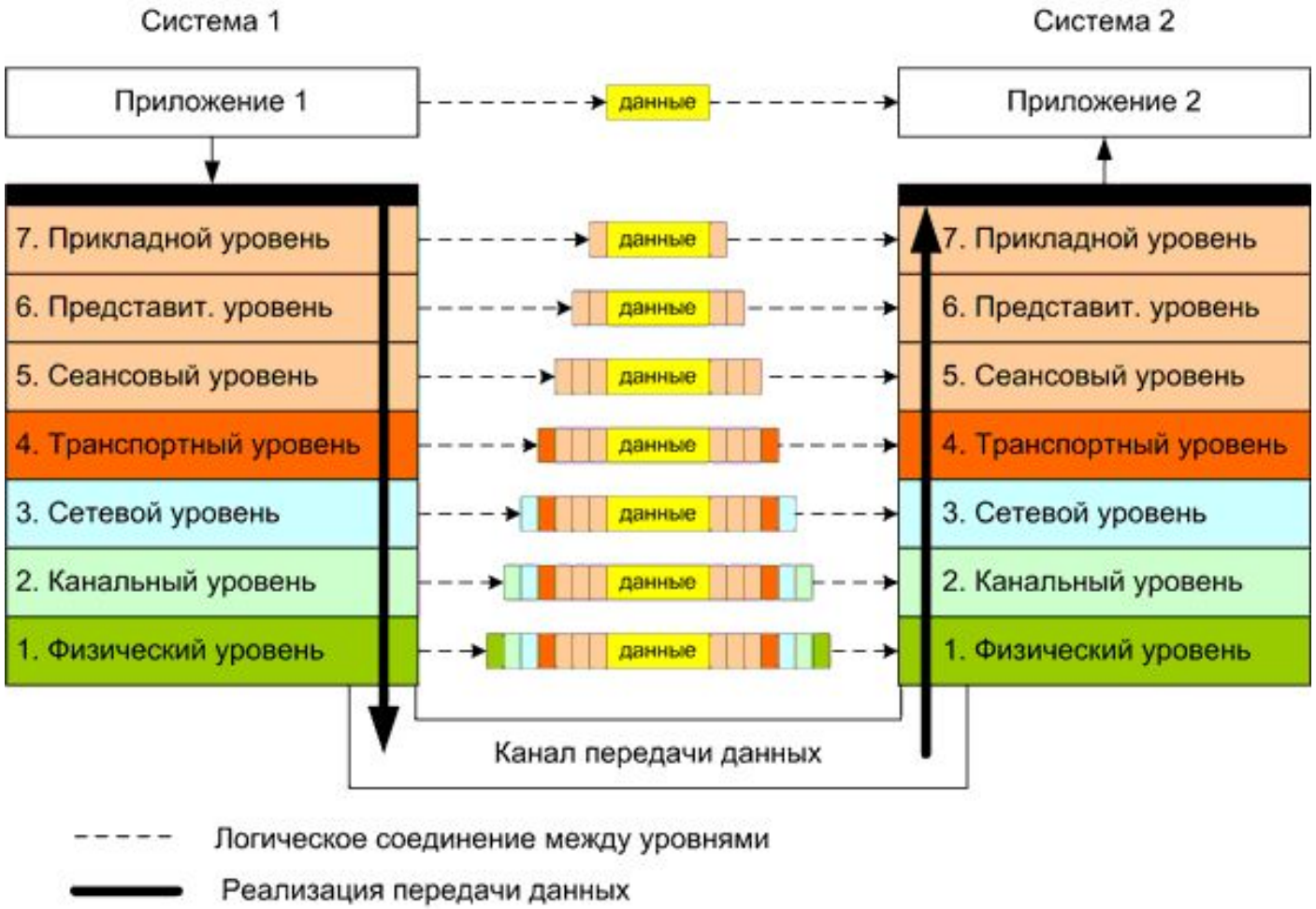
Инкапсуляция происходит сверху вниз (т.е. с 7 уровня до 1-го).

Декапсуляция происходит снизу вверх (с 1-го до 7-го уровня).

Инкапсуляция характерна для исходящих данных.

Декапсуляция – для входящих.

Инкапсуляция



IP-адрес и маска подсети

- **IPv4 адрес** - это уникальный 32 битный адрес хоста.
- **Маска подсети** - используется для определения сетевой части и части хоста адреса IPv4. Имеет длину 32 бита и представляет собой набор единиц в сетевой части и нулей в хостовой.
- В примере этой картинки маска будет следующая:

  11111111.11111111.11111111.00000000 (255.255.255.0)

  Крат

Отличие коммутаторов от маршрутизаторов

Коммутатор (switch) – сетевое промежуточное устройство **второго** уровня. Предназначено для связи множества устройств между собой в единую сеть. Осуществляет передачу **кадров** (**фреймов**) на основе адресации второго уровня (т.е. MAC-адресов).
Коммутаторы не предназначены для передачи данных между сетями. Они работают только в локальных сетях.

Маршрутизатор (router) – сетевое промежуточное устройство **третьего** уровня, предназначенное для передачи данных между сетями. Является также пограничным устройством в большинстве случаев. Передача **пакетов** осуществляется на основе адресации третьего уровня (т.е. по IP-адресам). Каждый интерфейс(порт) роутера принадлежит своей сети.
Основными функциями маршрутизатора являются определение наилучшего пути для пересылки пакетов на основе информации, содержащейся в таблице маршрутизации, и пересылка пакетов к месту назначения.

Система разрешения доменных имён

- **DNS** - Domain Name System.
- Основная цель DNS — это отображение доменных имен в IP адреса и наоборот — IP в DNS.
- Доменная структура DNS представляет собой древовидную иерархию, состоящую из узлов, зон, доменов, поддоменов и др. элементов.
- **Зона** — это любая часть дерева системы доменных имен, размещаемая как единое целое на некотором **DNS-сервере**.
- **Домен** — это именованная ветвь или поддерево в дереве имен DNS, то есть это определенный узел, включающий в себя все подчиненные узлы.

  Обратите внимание на различие между **зоной** (zone) и **доменом** (domain): *домен groucho.edu затрагивает <u>все</u> машины в университете Groucho Marx, в то время как зона groucho.edu включает только хосты, которые работают в непосредственно компьютерном центре, например в отделе математики. Хост в отделе физики принадлежат другой зоне, а именно physics.groucho.edu.*

FQDN

- **FQDN** (англ. *Fully Qualifed Domain Name*, полностью определённое имя домена) — это имя домена, **однозначно** определяющее доменное имя и включающее в себя имена всех родительских доменов иерархии DNS, **в том числе и корневого**. Своеобразный аналог абсолютного пути в файловой системе.

```
mail.k-max.name.
 |     |  |  | |
 |     |  |  | +-корневой домен
 |     |  |  +---домен первого уровня
 |     |  +------точка, разделяющая домены/части FQDN
 |     +---------домен второго уровня
 +---------------поддомен/домен третьего уровня, возможно - имя хоста
```

Ресурсные записи и делегирование

- **Ресурсная запись** — это то, собственно ради чего в конечном счете и существует DNS.
  Ресурсная запись — это единица хранения и передачи информации в DNS. Каждая такая запись несет в себе ***информацию соответствия*** какого-то *имени* и *служебной информации* в DNS, например соответствие имени домена — IP адреса.
- **Делегирование ответственности** — это операция *передачи ответственности за часть дерева доменных имен (зону)* другому лицу или организации.

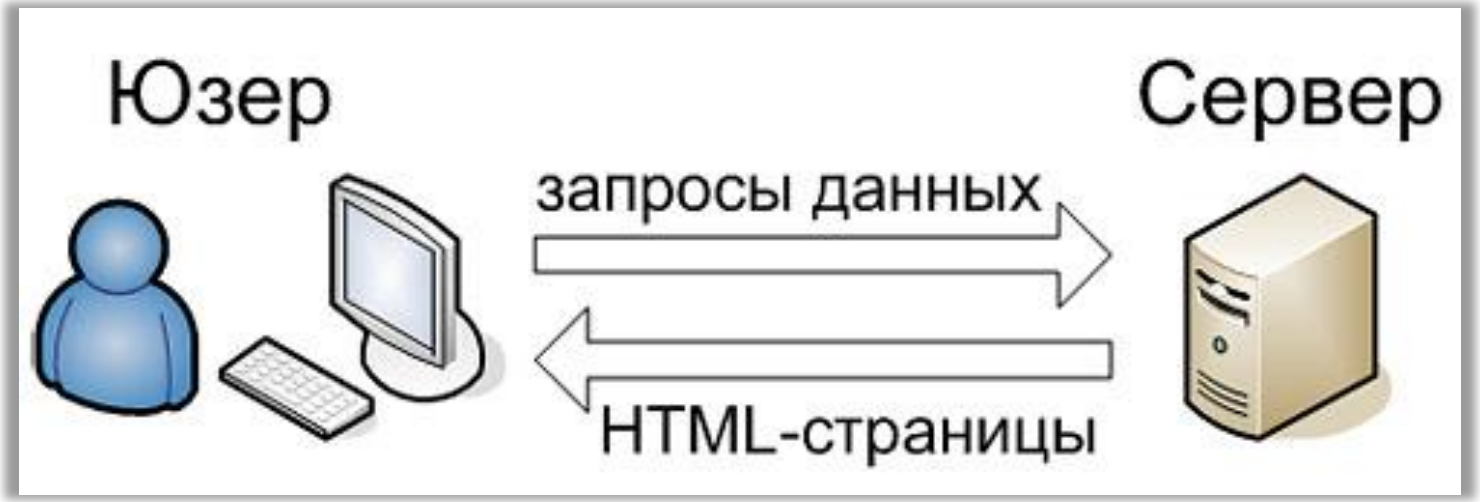
  Технически, **делегирование** заключается в выделении какой-либо части дерева в отдельную **зону**, и размещении этой зоны на **DNS-сервере**, принадлежащем другому лицу или организации. При этом, в родительскую зону включаются «склеивающие» **ресурсные записи** (NS и A), содержащие указатели на авторитативные DNS-сервера дочерней зоны, а вся остальная информация, относящаяся к дочерней зоне, хранится уже на DNS-серверах дочерней зоны.

Чаще всего в работе мы сталкиваемся с протоколами прикладного уровня:

- **HTTP** - протокол использует браузер для открытия сайтов (80 порт).
- **HTTPS** - протокол использует браузер для открытия сайтов по зашифрованному соединению (443 порт).
- **FTP** - протокол для передачи файлов (21 порт).
- **SFTP** - защищенный протокол передачи файлов (22 порт over SSH).
- **SSH** - позволяет производить удалённое управление операционной системой (22 порт).
- **POP3** - для получения почты с удалённого сервера по TCP-соединению. Перемещает письма с сервера (110 порт).
- **IMAP** - для получения удалённого доступа к хранилищу писем на сервере. Синхронизирует письма с сервером (143 порт).
- **SMTP** - для передачи электронной почты (25 порт)
- **WHOIS** - сетевой протокол прикладного уровня, базирующийся на протоколе TCP (порт 43). Получение регистрационных данных о владельцах доменных имён, IP-адресов и автономных систем.
- **DNS** – система разрешения доменных имен (53 порт).

Веб-сервер

Веб-сервер - сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.



Nginx vs Apache2

Apache предоставляет несколько модулей мультипроцессинга (multi-processing modules, MPM), которые отвечают за то как запрос клиента будет обработан. Это позволет администраторам определять политику обработки соединений.

MPM: **mpm_event, mpm_worker, mpm_prefork.**

Модули создают процессы и потоки. Потоков на процесс может быть как несколько, так и один.

Nginx изначально был спроектирован на базе асинхронных неблокирующих event-driven алгоритмов. **Nginx** создает процессы-воркеры каждый из которых может обслуживать тысячи соединений. Они работают, отвлекаясь на обработку соединения только когда появляется событие.

Каждое соединение помещается в event loop вместе с другими соединениями. В этом цикле события обрабатываются асинхронно, позволяя обрабатывать задачи в неблокирующей манере. Когда соединение закрывается оно удаляется из цикла.

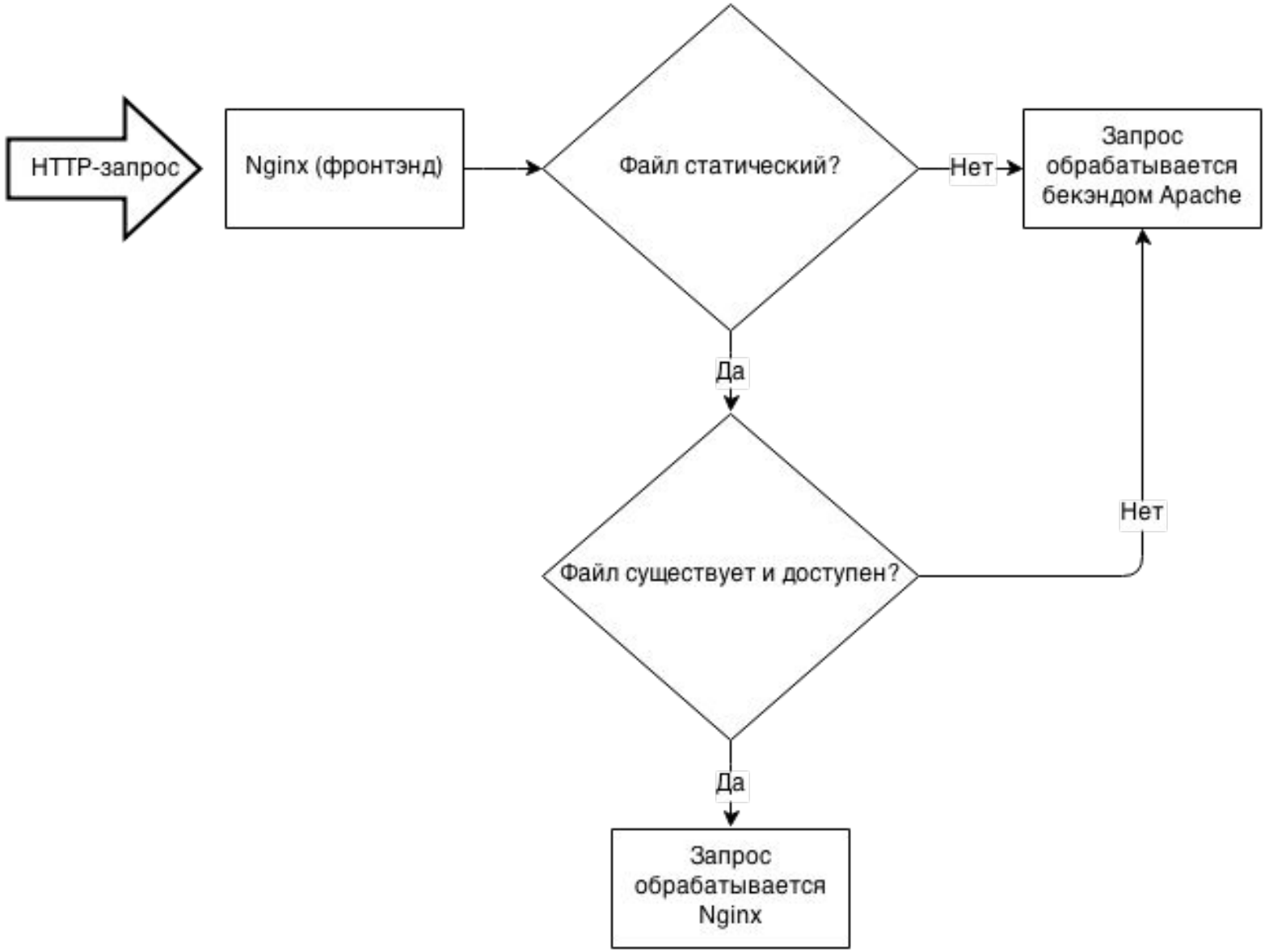
Apache может раздавать динамический контент, встраивая интерпретатор нужного языка в каждого воркера. Это позволяет обрабатывать запросы к динамическому содержимому средствами самого веб-сервера и не полагаться на внешние компоненты. Интерпретаторы языков могут быть подключены к **Apache** с помощью динамически загружаемых модулей.

Возможность обрабатывать динамический контент средствами самого **Apache** упрощает конфигурирование. Нет необходимости настраивать взаимодействие с дополнительным софтом, динамический модуль может быть легко отключен в случае

Nginx не имеет возможности самостоятельно обрабатывать запросы к динамическому контенту. Для обработки запросов к **PHP** или другому динамическому контенту **Nginx** должен передать запрос внешнему процессору по протоколу http (или FastCGI, SCGI, uWSGI, memcache) для исполнения, подождать пока ответ будет сгенерирован и получить его.

Статический контент будет возвращен клиенту простым способом. Apache можно настроить так же, но тогда он лишается своего главного преимущества.

NGINX + Apache



Файл .htaccess – конфигурация на уровне директорий

.htaccess - это конфигурационный файл веб-сервера Apache, который позволяет задавать различные настройки для работы веб-сервера, применяемые для конкретных каталогов пользователя.

Параметры, указанные в .htaccess, не затрагивают настройки главного конфига Apache. Они имеют силу только для каталога, в котором этот файл размещен, и его подкаталогов. Для подкаталогов можно создавать дополнительные .htaccess-файлы (настройки .htaccess, заданные в подкаталоге, переопределят настройки, назначенные для него в файле .htaccess на уровне выше).

С помощью .htaccess можно настраивать перенаправления, управлять опциями PHP, устанавливать разрешения и права доступа к файлам, задавать собственные страницы ошибок и др.

Shared (первая схема)

- Тарифы – **Year+, Optimo+, Century+, Millenium+, 1Сайт, Старт CMS**.
- В этом режиме общий процесс (на каждую версию PHP свой) Apache с большим количеством обработчиков обрабатывает виртуальные хосты всех пользователей на сервере.

- Плюсы:
  - простота и удобство сопровождения,
  - экономия оперативной памяти,
  - большая вместимость сервера.

Обработчики Apache сообщают серверу, как управлять файлами. Какой программой должен быть обработан файл с определенным расширением.

Dedicated (вторая схема)

- Тарифы - **Eterno, Premium, Pro CMS, аренда с администрированием**.
- Для каждого пользователя для каждой версии на сервере запускается отдельный процесс Apache с ограниченным количеством обработчиков.
- 6 обработчиков для PHP 7.1 и по 3 обработчика для других версий PHP

- Плюсы:
  - более высокая скорость работы сайтов отдельного клиента.
  - изоляция клиентов друг от друга, нагрузка на одном из них не сказывается на остальных.

Существуют разные виды сайтов (по их назначению и содержанию):

- Сайт-визитка
- Корпоративный сайт
- Портал
- Интернет-магазин
- Лендинг
- Социальная сеть
- Форум
- Блог
- и др.

Современный веб-сайт состоит из двух основных частей: интерфейсной части и серверной части.

- Внешний интерфейс - это часть, которую посетители сайта видят в браузере: посты в блогах, изображения, видео, страницы, формы для рассылок и т. д. Текстовая часть отображается на стандартном языке разметки под названием HTML, а дизайн добавляется с помощью CSS и JavaScript.
- Серверная часть состоит из базы данных и файлов сайта. Содержимое сайта сохраняется в базе данных и передаётся от внутреннего интерфейса к внешнему, когда пользователь запрашивает веб-страницу. Файлы сайта связаны между собой и образуют структуру сайта (код). Структура может быть написана на разных языках программирования, таких как PHP, Python, JavaScript и другие.

CMS - система управления контентом

Это приложение, которое запускается в браузере.
Система управления контентом позволяет использовать редактор контента для создания постов, страниц, интернет-магазинов и размещения контента в интернете. Параметры можно с помощью раскрывающихся меню, флажков и других элементов управления.

При использовании CMS не нужно писать ни внешний, ни внутренний код, чтобы сформировать структуру сайта.
CMS - это программа с открытым кодом. Это значит, что можно настроить и изменить код в соответствии с требованиями владельца сайта.

Самые распространенные и известные CMS:

- Wordpress
- Joomla
- OpenCart
- Drupal
- ...

Сайты, которые созданы на основе CMS также состоят из файлов (они образуют структуру - скелет сайта) и базы данных (в ней хранится наполнение сайта).

MySQL-сервер

MySQL - это система управления реляционными базами данных с моделью клиент-сервер.

- СУБД - это программное обеспечение, используемое для создания и управления базами данных.
- База данных - это набор структурированных данных; место, в котором хранятся данные.
- Реляционная модель - данные представлены в виде таблиц, связанных между собой.
- Модель клиент-сервер. Клиенты - компьютеры, которые устанавливают и запускают СУБД. Когда им нужно получить доступ к данным, они подключаются к серверу, на котором установлена СУБД.

phpMyAdmin - это приложение, представляющее собой веб-интерфейс для администрирования СУБД MySQL.

Код ответа сайта / Ошибки в работе сайта

- Код ответа 2** сообщает о успешном выполнении запроса.
- Код ответа 3** сообщает что тут страницы нет, простыми словами это редирект - 301,302
- Ошибки HTTP 4** (код ошибки сообщает о проблеме с файлами) - 400, 403, 404. Что могут означать, порядок диагностики:
    1. Проверка А-записи сервера
    2. Проверка наличия файлов сайта и правильного размещения (проверка индексного файла)
    3. Передаем в ОТП
- Ошибки HTTP 5** (код ошибки сообщает о внутренней проблеме сервера, когда он что то не смог обработать)- 500, 502, 503. Что могут означать, порядок диагностики:
    1. Проверка А-записи сервера
    2. Проверка есть ли проблема в Zabbix
    3. Проверка наличия информации о проблемах на сервере в #oip_otp_osa
    4. при 502 переключаем версию PHP объясняем пользователю по количеству обработчиков.
    5. Передаем в ОТП (Зачастую 500 ошибка возникает при неправильном синтаксисе в .htaccess или связаны с ошибкой PHP скрипта)
- Ошибки CMS или кода. Что могут означать, порядок диагностики:
    1. Проверка А-записи
    2. Передаем в ОТП для проверки.
- Парковка
    1. Проверка А-записи сервера - должна совпадать с адресом сервера аккаунта или с дополнительным Ip
    2. Проверка привязки в разделе “Сайты” (перепривяжите домен к сайту)

Работа почты

Любое электронное письмо (e-mail) отправляется с сервера.
Это может быть SMTP-сервер - сервер исходящей почты, на котором создан почтовый ящик. А может быть сервер, на котором размещен сайт. Либо сервер компании, которая специализируется на предоставлении услуг по формированию и осуществлению массовых рассылок.

Нас интересуют 2 первых способа отправки писем. Они используются при создании правила для отправки писем с сайта.

При создании правила для отправки письма с сайта клиент может выбрать один из следующих способов :

1) с помощью php mail()
2) SMTP

PHP mail

Функция PHP mail() предназначена для отправки электронной почты, и ее работа осуществляется на нашем хостинге через агент пересылки почтовых сообщений Exim, который установлен на каждом сервере виртуального хостинга.

Письма отправляются от **user@server.timeweb.ru**, где user - это логин, а server - имя сервера, на котором расположен аккаунт. Доступ к данному электронному ящику невозможен. Также невозможна DKIM-подпись (цифровая подпись, подтверждающая что письмо отправлено именно с этого адреса) для данного вида писем.

Для того, чтобы в качестве адреса отправителя указывался определенный ящик, необходимо помимо заголовка From передавать почтовому серверу в функции mail аргумент –f. Подробно в документации php: https://www.php.net/manual/ru/function.mail.php

SMTP-авторизация

Отправку писем из скриптов на сайтах также можно осуществлять через SMTP-сервер. Фактически происходит подключение к почтовому серверу, на котором размещен ящик. Отправка письма осуществляется через почтовый ящик, почтовый сервер.

Для отправки по SMTP используются следующие реквизиты:

- логин пользователя совпадает с электронным ящиком,
- пароль от электронного ящика,
- сервер для отправки электронной почты - smtp.timeweb.ru, (или другой SMTP-сервер, то на который направляют MX-записи)
- порт - 25 или 2525.

Логи

Логи (лог-файлы) - это файлы, которые содержат системную информацию работы сервера или компьютера. В них заносятся определенные действия пользователя или программы. Логи представляют из себя журнал событий на сервере (компьютере), и иногда их так и называют - журналом.

Предназначение логов - протоколирование операций, выполняемых на машине, для возможности дальнейшего анализа.

Регулярный просмотр журналов позволит определить ошибки в работе системы в целом, конкретного сервиса или сайта (особенно скрытые ошибки, которые не выводятся при просмотре в браузере), диагностировать злонамеренную активность, собрать статистику посещений сайта.

Наши клиенты могут включить автоматическое ведение логов доступа и ошибок Apache (access_log и error_log) для своего сайта в панели управления (раздел “Логи”).

Файлы access_log и error_log будут располагаться на одном уровне с директорией public_html сайта.

Логи сайта за прошедший период (за последние 180 дней) можно заказать в панели управления. Для заказа доступны следующие виды логов:

- Apache (access_log (на основе этих логов можно подключить AWStats – расширение для отслеживания статистики посещаемости сайта) и error_log);
- Qmail (логи отправки писем с сервера);
- логи FTP/SSH-доступа.

Еще несколько типов логов могут быть предоставлены по запросу в службу поддержки (их предоставляет ОТП):

- логи cron;
- логи SMTP/Exim;
- логи входов в панель управления.

Ура!

Теперь мы стали чуть более подкованными в технических вопросах!