


# Теория 1

## Уровни OSI

Сетевая модель OSI (The Open Systems Interconnection model) - сетевая модель стека протоколов OSI/ISO. Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определенные функции при таком взаимодействии.

### Небольшие сноски:

1. Open System Interconnection model - Модель взаимосвязи открытых систем.
2. ISO (International Organization for Standardization) - Международная организация по стандартизации.
3. Протокол - Набор правил, позволяющий осуществлять какие либо действия между включенными в сеть устройствами.

 **В 1990-х годах с моделью стала конкурировать TCP/IP, которая и стала широко использоваться в сетях, а OSI используется в основном в качестве эталона для обучения и документации.**

### Небольшие сноски:

1. TCP/IP (Transmission Control Protocol / Internet Protocol) - сетевая модель передачи данных, представленных в цифровом виде. Название происходит из двух важнейших протоколов семейства TCP (Протокол управление передачей) и IP (Интернет протокол). Будет расписываться подробнее позже.

Вклад OSI в том, что она стала толчком к всеобщей стандартизации, для обеспечения совместимости с различным оборудованием. До неё большая часть сетей поддерживала различные наборы протоколов и не всегда в наборах нескольких сетей находились общие.

### Стандарты

Модель OSI, которая была определена в серии стандартов ISO/IEC 7498, состоит из следующих частей:

- ISO/IEC 7498-1 - Базовая модель.
- ISO/IEC 7498-2 - Архитектура безопасности;
- ISO/IEC 7498-3 - Наименования и адресация;
- ISO/IEC 7498-4 - Система менеджмента.

### **Небольшие сноски:**

1. International Electrotechnical Commission (IEC) - Международная комиссия электротехники

## Основные принципы

Протоколы связи позволяют структуре на одном хосте взаимодействовать с соответствующей структурой того же уровня на другом хосте.

На каждом уровне N два объекта обмениваются блоками данных PDU с помощью протокола данного уровня на соответствующих устройствах. Каждый PDU содержит блок служебных данных (SDU), связанный с верхним или нижним протоколом

### **Небольшие сноски:**

1. Protocol Data Unit (PDU) - Блок данных протокола. Единица информации, передаваемая между одноранговыми объектами сети. Как правило состоит из управляющей информации, специфичных для протокола, и пользовательских данных.
2. Service Data Unit (SDU) - Блок служебных данных. Блок данных, который был передан с уровня или подуровня OSI на более низкий уровень.

Обработка данных двумя взаимодействующими OSI - совместимыми устройствами происходит следующим образом:

1. Передаваемые данные составляются на самом верхнем уровне передающего устройства (уровень N) в PDU.
2. PDU передается на уровень N-1, где он становится SDU.
3. На уровне N-1 SDU объединяется с верхним, нижним или обоими уровнями, создавая PDU слоя N-1. Затем этот PDU передается в слой N-2.
4. Процесс продолжается до достижения самого нижнего уровня, с которого данные передаются на принимающее устройство.
5. На приемном устройстве данные передаются от самого низкого уровня к самому высокому в виде серии SDU, последовательно удаляясь из верхнего или нижнего колонтитула (блок, размещенный на краю набора) каждого слоя, до достижения самого верхнего уровня, где принимаются последние данные.

### **Важно**

1. Любой протокол модели OSI должен взаимодействовать либо с протоколами своего уровня, либо с протоколами на единицу выше и/или ниже своего уровня.

2. Взаимодействия с протоколами своего уровня называются горизонтальными, а с уровнями на единицу выше или ниже - вертикальными.
3. Любой протокол модели OSI Может выполнять только функции своего уровня и не может выполнять функции другого уровня;
4. Уровни взаимодействуют между собой внутри одной системы при помощи интерфейсов.

Каждому уровню соответствует свой операнд - логически неделимый элемент данных:

На физическом уровне мельчайшая единица - бит;

На канальном - кадр;

На сетевом - пакет (датаграмма);

На транспортном - сегмент.

Любой фрагмент данных, логически объединённых для передачи считается сообщением.

**Именно сообщения в общем виде являются операндами сеансового, представления и прикладного уровней.**

## Общая модель:

Уровни	Тип данных (PDU)	Функции	Примеры
7. Прикладной (application)	Данные	Доступ к сетевым службам	HTTP, FRP, POP3, SMTP, WebSocket
6. Представления (presentation)		Представление и шифрование данных	ASCII, EBCDIC, SSL, gzip
5. Сеансовый (session)		Управления сеансом связи	RCP, PAP, L2TP, gRPC
4. Транспортный (transport)	Сегменты (segment) / Датаграммы (datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, Порты
3. Сетевой (network)	Пакеты (packet)	Определение маршрута и логичная адресация	IPv4, IPv6, IPsec, AppleTalk, ICMP
2. Канальный (data link)	Биты (bit) / Кадры (frame)	Физическая адресация	PPP, IEEE, 802.22, Ethernet, DSL, ARP, сетевая карта
1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, RJ

## 7. Прикладной уровень

### Основные функции

Обеспечивает взаимодействие пользовательских приложений с сетью:

- Позволяет приложениям использовать сетевые службы:
  - Удалённого доступа к файлам и базам данных;
  - Пересылке электронной почты;
- Отвечает за передачу служебной информации;
- Предоставляет приложениям информацию об ошибках
- Формирует запросы к уровню представления.

### ☰ Примеры протоколов

- **RDP** - Протокол удалённого рабочего стола;
- **HTTP** - Протокол передачи гипертекста;
- **SMTP** - Простой протокол передачи почты;
- **SNMP** - Простой протокол сетевого управления;
- **POP3** - Протокол почтового отделения;
- **FTP** - Протокол передачи файлов;
- **XMPP** - Протокол мгновенного обмена сообщениями и информацией о присутствии;
- **OSCAR** - Открытая система общения в реальном времени;
- **Modbus** - Открытый коммуникационный протокол;
- **SIP** - Протокол передачи данных;
- **TELNET** - Протокол для реализации текстового терминального интерфейса по сети.

## 🔗 6. Уровень представления

### ☰ Основные функции

- Преобразование данных, такое как изменение порядка битов, замена метасимволов или преобразование целого числа в числа с плавающей точкой;
- Перевод символов из одной кодировки в другую;
- Сжатие данных для увеличения пропускной способности канала;
- Шифрование и расшифрование.

### ☰ Примеры протоколов

- **AFP** - Протокол предоставляющий доступ к файлам в Mac OS X;
- **ICA** - закрытый протокол для сервера приложений;

- **LPP** - Протокол представлений;
- **NCP** - Протокол организации обмена между рабочей станцией и файловым сервером;
- **NDR** - Протокол представлений;
- **XDR** - Протокол стандартизации формата данных;
- **X.25 PAD** - Протокол передачи данных с использованием коммутации пакетов;

### ☰ Дополнения

- Уровень представлений обычно представляет собой промежуточный протокол для преобразования информации из соседних уровней. Это позволяет осуществлять обмен между приложениями на разнородных компьютерных системах прозрачным для приложений образом.
- Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом важном уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.
- На представительском уровне передаваемая по сети информация не меняет содержания. С помощью средств, реализованных на данном уровне, протоколы прикладных программ преодолевают синтаксические различия в представляемых данных или же различия в кодах символов, например согласовывая представления данных расширенного двоичного кода обмена информацией EBCDIC используемого мейнфреймом компании IBM с одной стороны и американского кода обмена информацией ASCII с другой.

## 🔗 5. Сеансовый уровень

### ☰ Основные функции

Отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время.

Управляет:

- Созданием/завершением сеанса;
- Правами на передачу данных;
- Поддержанием сеанса в периоды неактивности приложений.

### ☰ Примеры протоколов

- **H.245** - Протокол согласования параметров соединения;
- **ISO-SP** - Сеансовый протокол;
- **ISNS** - Протокол автоматизации открытия, управления и конфигурирования устройств;
- **L2F** - Протокол туннелирования (логическое соединение между двумя конечными точками);
- **L2TP** - Протокол туннелирования;
- **NetBIOS** - Протокол работы в локальных сетях на персональных ЭВМ;
- **PAP** - Протокол проверки подлинности на основе имени пользователя и пароля;
- **PPTP** - Протокол туннелирования;
- **RPC** - Протокол удалённого вызова процедур;
- **RTCP** - Протокол управления передачи в реальном времени;
- **SMPP** - Протокол передачи коротких сообщений;
- **SCP** - Протокол управления сеансами;
- **SDP** - Протокол поддержания потоковых сокетов.

#### ☰ Дополнения

- Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия;
- Сеансы передачи состояются из запросов и ответов, которые осуществляются между приложениями;
- Службы сеансового уровня обычно используются в средах приложений, в которых требуется использование удалённого вызова процедур.

## 🔗 4. Транспортный уровень

#### ☰ Основные функции

- Предоставляет механизм передачи данных;
- Формирует блоки данных, размер которых зависит от протокола. Большие блоки разделяются, а короткие объединяются.

#### ☰ Примеры протоколов

- **CUPD** - Протокол циклических пользовательских пакетов;
- **DCCP** - Протокол контроля перегрузки пакетов;

- **FCP** - Протокол высокоскоростной передачи данных;
- **IL** - Протокол передачи сообщений 9P (сообщения файловой системы OS Plan9);
- **NBF** - Протокол передачи данных сетевого и транспортного уровня, без маршрутизации;
- **NCP** - Протокол обмена между рабочей станцией и файловым сервером;
- **SCTP** - Протокол передачи с управлением потоком;
- **SPX** - Стек протоколов последовательного обмена пакетами;
- **SST** - Протокол передачи данных, аналогичный TCP, но с расширенным управлением потоками;
- **TCP** - Протокол передачи данных;
- **UPD** - Протокол пользовательских пакетов.

#### ☰ Дополнения

- Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

### 🔗 3. Сетевой уровень

#### ☰ Основные функции

- Трансляция логических адресов и имен в физические;
- Определение кратчайших маршрутов;
- Коммутация и маршрутизация;
- Отслеживание неполадок и "заторов" в сети.

#### ☰ Примеры протоколов

**IP / IPv4 / IPv6** - Маршрутизируемый протокол;

**IPX** - Протокол обмена пакетами;

**CLNP** - Протокол передачи данных без установки соединения;

**IPsec** - Стек протоколов для защиты данных;

**RIP** - Простой протокол маршрутизации;

**OSPF** - Протокол динамической маршрутизации.

#### ☰ Дополнения

- Протоколы сетевого уровня маршрутизируют данные от источника к получателю. Работающие на этом уровне устройства (маршрутизаторы) условно называют устройствами третьего уровня (по номеру уровня в модели OSI);
- В пределах семантики иерархического представления модели OSI Сетевой уровень отвечает на запросы обслуживания от Транспортного уровня и направляет запросы обслуживания на Канальный уровень.

## 🔗 2. Канальный уровень

### ☰ Основные функции

- Передача данных узлам, находящимся в том же сегменте локальной сети;
- Обнаружение и исправление ошибок, возникших на физическом уровне.

### ☰ Примеры технологий

- **ARCnet** - Технология локальной вычислительной сети;
- **ATM** - Сетевая высокопроизводительная технология коммутации и мультиплексирования пакетов;
- **CAN** - Стандарт промышленной сети;
- **IEEE 802.3** - Группа стандартов основы семейства технологий пакетной передачи данных Ethernet;
- **EAPS** - Протокол поддержки топологии, исключающей заикливание трафика и её перестроения в случае нарушений в кольцевых сетях Ethernet;
- **FDDI** - Стандарт передачи данных для локальных сетей.

### ☰ Дополнения

- Кадры канального уровня не пересекают границ сетевого сегмента;
- Кадры передаются последовательно с обработкой кадров подтверждения, отсылаемых обратно получателем.

## 🔗 1. Физический уровень

### ☰ Основные функции



- Передача данных через физические среды линий связи, соединяющие сетевые устройства.

#### ☰ Примеры технологий

- **IEEE 802.15** - Bluetooth;
- **IRDA** - Передача данных с использованием инфракрасного диапазона световых волн;
- **EIA RS-232** - Стандарт для асинхронного интерфейса;
- **EIA-422** - ;
- **EIA-423** - Стандарт последовательной передачи данных;
- **RS-449** - Протокол определяющий механические характеристики интерфейса между данными оконечного оборудования и данными оборудования связи.

#### ☰ Дополнения

- Единица измерения, используемая на этом слое — бит, то есть физический уровень осуществляет передачу потока битов по соответствующей физической среде через соответствующий интерфейс.

## Сетевые сервисы

Взаимодействие устройств между собой, а также с другим активным сетевым оборудованием происходит на основе использования сетевых служб, которые обеспечиваются специальными процессами операционной системы:

- "Демонами" в UNIX - подобных ОС;
- Службами в ОС Windows.

Эти процессы создают "слушающий" сокет и "привязывают" его к определённому порту (пассивное открытие соединения), обеспечивая тем самым возможность другим компьютерам обратиться к данной службе.

Клиентская программа или процесс создаёт запрос на открытие сокета с указанием IP-адреса и порта сервера, в результате чего устанавливается соединение, позволяющее взаимодействовать двум компьютерам с использованием соответствующего сетевого протокола прикладного уровня.

#### 📌 Небольшие сноски:

- Демон - программа, запускаемая системой и работающая в фоновом режиме;

- Сокет - программный интерфейс для обеспечения обмена данными между процессами;
- Application Programming Interface (API) - Интерфейс программирования приложения. Описание способов взаимодействия одной программы с другой.
- Порт - целое неотрицательное число, которое идентифицирует назначение сетевых потоков данных в пределах одного устройства.

#### Диапазоны портов

- 0-1023 - Общеизвестные порты

Назначены IANA и в большинстве систем используются исключительно процессами системы или пользователями/программами, имеющими соответствующие привилегии.

- 1024-49151 - Зарегистрированные порты

Номера портов включены в каталог IANA и на большинстве систем могут быть использованы процессами обычных пользователей/программ.

- 49152-65535 - Динамические порты

Предназначены для временного использования, а также в качестве клиентских (используемых для частных служб внутри закрытых сетей).

#### Небольшие сноски:

- Internet Assigned Numbers Authority (IANA) - Администрация адресного пространства Интернет. Управляет пространствами IP-адресов, доменов верхнего уровня, а также регистрирует типы данных MIME и параметры прочих протоколов интернета.
- Порты 0-49151 не должны использоваться без регистрации IANA. Процедура регистрации описана в разделе 19.9 RFC 4340
- Request for Comments (RFC) - документ содержащий технические спецификации и стандарты, широко применяемые во всемирной сети.

## Примеры сервисов

### Сервис печати

Позволяет отправлять документы на удаленные принтеры.

## ☰ LDP

Протокол распределения меток.  
7 уровень.

LDP сессия функционирует поверх TCP/IP (порт 646).

LSR1 (коммутирующий метки маршрутизатор) и LSR2 при обмене сообщениями hello узнают транспортные адреса друг друга. Если транспортный адрес LSR1 больше чем транспортный адрес LSR2, то LSR1 становится "активным" соседом, а LSR2 "пассивным", иначе, наоборот. Далее, LDP сессия устанавливается по следующему сценарию:

1. "Активный" сосед устанавливает TCP/IP сессию на порт 646.
2. "Активный" сосед посылает сообщение Init, включающее в себя свои параметры LDP сессии.
3. "Пассивный" сосед проверяет параметры LDP сессии в сообщении Init на предмет совместимости с локальными настройками LDP.
4. "Пассивный" сосед отвечает сообщением Init, включающее в себя свои параметры LDP сессии.
5. "Активный" сосед проверяет параметры LDP сессии в сообщении Init на предмет совместимости с локальными настройками LDP
6. Сессия установлена.

Если на каком-то этапе происходит что-то не предвиденное (приходит не тот тип пакета, ожидаемое сообщение не приходит вообще, или не совпадают параметры LDP сессии в сообщении Init и т.п.), то сессия считается не установленной. LSR, обнаруживший ошибку посылает сообщение Shutdown или Reject своему соседу.

## ☰ IPP

Протокол межсетевой печати.  
7 уровень.

Является перегруженной версией HTTP.

Тело HTTP сообщения имеет Content-type: application/ipp и содержит следующие поля:

- version-number (2 байта) — старший и младший номер версии протокола, текущее значение 0x0101;
- operation-id или status-code (2 байта) — код операции (запрос) или код состояния (ответ);
- request-id (4 байта) — уникальный номер для сопоставления запросов и ответов;
- attribute-group — группа атрибутов, поле может включаться ноль или более раз:

- begin-attribute-group-tag (1 байт) — указывает начало группы атрибутов и идентифицирует её тип;
- attribute — ноль или более полей атрибутов:
  - attribute-with-one-value — атрибут с одним значением или первое значение атрибута, имеющего несколько значений:
    - value-tag (1 байт) — синтаксис атрибута (целое значение, перечисляемое, строка и т.д.);
    - name-length (2 байта) — длина имени атрибута в байтах;
    - name — имя атрибута;
    - value-length (2 байта) — длина значения атрибута в байтах;
    - value — значение атрибута;
  - additional-value — ноль или более полей значений атрибута:
    - value-tag (1 байт) — синтаксис атрибута;
    - name-length (2 байта) — 0x0000;
    - value-length (2 байта) — длина значения атрибута в байтах;
    - value — значение атрибута;
- end-of-attributes-tag (1 байт) — 0x03, указывает конец полей атрибутов;
- data — данные, необходимые для операции.

## ☰ SMB

Протокол удаленного доступа к файлам, принтерам и другим сетевым ресурсам.

7 уровень.

8	16	24	32 bits
IDF			
Command	RCLS	Reserved	ERR
ERR (cont)	REB/FLG	Reserved	
Reserved			
Reserved			
Reserved			
Tree ID		Process ID	
User ID		Multiplex ID	
WCT	VWV		
BCC		BUF	
SMB header structure			

Основные элементы структуры заголовка SMB:

- **IDF (Identifier)** — Идентификатор протокола: 0xff, 0x53('S'), 0x4d('M'), 0x42('B').

- **Command** — команда протокола.
- **RCLS** — код класса ошибки.
- **ERR** — код ошибки.
- **Tree ID (TID)** — идентификатор соединения с сетевым ресурсом.
- **Process ID (PID)** — идентификатор клиентского процесса фактического соединения.
- **User ID (UID)** — идентификатор пользователя; используется сервером для проверки прав доступа пользователя.
- **Multiplex ID (MID)** — идентификатор группы пользователя; используется сервером для проверки прав доступа группы пользователя.
- **WCT** — количество параметров, следующих за заголовком.
- **BCC** — количество байт данных, следующих за параметрами.

## ✓ Сервис СУБД

Позволяет работать с базами данных, хранящимися на удаленных серверах.

### ☰ ODBC

Интерфейс доступа к базам данных.

В начале 1990 г. существовало несколько поставщиков баз данных, каждый из которых имел собственный интерфейс. Если приложению было необходимо общаться с несколькими источниками данных, для взаимодействия с каждой из баз данных было необходимо написать свой код. Для решения возникшей проблемы Microsoft и ряд других компаний создали стандартный интерфейс для получения и отправки источникам данных различных типов. Этот интерфейс был назван Open Database Connectivity, или открытый механизм взаимодействия с базами данных. С помощью ODBC прикладные программисты могли разрабатывать приложения для использования одного интерфейса доступа к данным, не беспокоясь о тонкостях взаимодействия с несколькими источниками. Это достигается благодаря тому, что поставщики различных баз данных создают драйверы, реализующие конкретное наполнение стандартных функций из ODBC API с учётом особенностей их продукта. Microsoft усовершенствовала ODBC для разработчиков приложений. Истинный интерфейс ODBC является обычным процедурным API. Вместо создания простой оболочки процедурного API разработчики Microsoft создали набор абстрактных классов, представляющих логические сущности в базе данных.

### ☰ JDBC

Платформенно независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД.

JDBC API содержит два основных типа интерфейсов: первый – для разработчиков приложений и второй (более низкого уровня) – для разработчиков драйверов.

#### ✓ Файл - Серверная СУБД

Класс по способу доступа к БД, при котором данные располагаются централизованно на файл-сервере.

СУБД располагается на каждом клиентском компьютере.

Доступ СУБД к данным осуществляется через локальную сеть.

Синхронизация чтений и обновлений осуществляется посредством файловых блокировок.

Пример: Microsoft Access, Paradox, dBase.

#### ✓ Клиент - Серверная СУБД

Класс по способу доступа к БД, при котором данные располагаются централизованно на файл-сервере.

СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме.

Все клиентские запросы на обработку данных обрабатываются централизованно.

Пример: MySQL, PostgreSQL, MS SQL Server.

#### ✓ Трехуровневая архитектура

Архитектурная модель программного комплекса, предполагающая наличие в нем трёх типов компонентов:

- Клиентских приложений - не должен иметь прямых связей с БД, быть нагруженным основной бизнес-логикой и хранить состояние приложения. Здесь может быть механизм авторизации, алгоритмы шифрования, проверка значений на допустимость и соответствие формату, несложные операции с данными;
- Серверов приложений - большая часть бизнес-логики. Серверы приложений проектируются таким образом, чтобы добавление к ним дополнительных экземпляров обеспечивало горизонтальное масштабирование производительности программного комплекса и не требовало внесения изменений в программный код приложения;

- Серверов баз данных - хранение данных.

## ☑ Сервис сообщений

Позволяет обмениваться текстовыми, голосовыми или видео сообщениями.

### ☰ SMTP

Простой протокол передачи почты.  
Уровень 7.

SMTP-операция состоит из трёх последовательностей команда/ответ (см. пример ниже). Описание последовательностей:

- **MAIL FROM** — устанавливает обратный адрес (то есть Return-Path, 5321.From, mfrom). Это адрес для возвращённых писем.
- **RCPT TO** — устанавливает получателя данного сообщения. Эта команда может быть дана несколько раз, по одной на каждого получателя. Эти адреса также являются частью оболочки.
- **DATA** — для отправки текста сообщения. Это само содержимое письма, в противоположность его оболочке. Он состоит из заголовка сообщения и тела сообщения, разделённых пустой строкой. DATA, по сути, является группой команд, а сервер отвечает дважды: первый раз на саму команду DATA, для уведомления о готовности принять текст; и второй раз после конца последовательности данных, чтобы принять или отклонить всё письмо.

### ☰ POP3

Протокол сбора сообщений.  
7 уровень.

В протоколе POP3 предусмотрено 3 состояния сеанса:  
Авторизация

Клиент проходит процедуру Аутентификации.

Транзакция

Клиент получает информацию о состоянии почтового ящика, принимает и удаляет почту.

Обновление

Сервер удаляет выбранные письма и закрывает соединение.

### ☰ IMAP

Протокол сбора сообщений.

7 уровень

Соединение IMAP 4.1 подразумевает установление связи между клиентом и сервером. Клиент посылает серверу команды, сервер клиенту - данные и уведомления о статусе выполнения запроса. Все сообщения, как клиента, так и сервера, имеют форму строк, завершающихся специальной последовательностью.

## ☑ Файловый сервис

Позволяет получать доступ к файлам, хранящимися на удалённых серверах.

### ☰ FTP

Протокол передачи файлов.

7 уровень.

FTP использует множественное (как минимум - двойное подключение). Поток управления и поток передачи данных. По первому передаются команды и ответы на них, а по второму передаются запрошенные данные.

## ☑ Сервис приложений

Позволяет использовать различные программы, запускаемые на удалённых серверах.

### ☰ RPC

Класс технологий, позволяющих программам вызывать функции или процедуры в другом адресном пространстве.

5 уровень.

Протоколы обеспечивающие RPC:

- **SOAP** - Протокол обмена структурированными сообщениями в распределенной вычислительной среде. Является одним из стандартов, на которых базируются технологии веб-служб;



- **XML-RPC** - Протокол вызова удалённых процедур, использующий XML для кодирования своих сообщений и HTTP в качестве транспортного механизма. Является прародителем SOAP;

## ☰ REST

Не является протоколом.

Набор правил того, как программисту организовать написание кода серверного приложения, чтобы все системы легко обменивались данными и приложение можно было масштабировать.

## ✎ Модели обслуживания

### ✓ IaaS

Инфраструктура как сервис.

Модель обслуживания в облачных вычислениях, по которой потребителям предоставляются по подписке фундаментальные информационно-технологические ресурсы - виртуальные серверы с заданной вычислительной мощностью, операционной системой и доступом к сети. Потребитель как правило приобретает время, умноженное на характеристики сервера.

### ✓ SaaS

ПО как сервис.

Модель обслуживания в облачных вычислениях, при которой подписчикам предоставляется готовое прикладное программное обеспечение, полностью обслуживаемое провайдером. Поставщик в этой модели самостоятельно управляет приложением, предоставляя заказчикам доступ к функциям с клиентских устройств, как правило через мобильное приложение или веб-браузер.

В модели SaaS:

- приложение приспособлено для удаленного использования;
- одним приложением пользуется несколько клиентов;
- оплата взимается либо в виде ежемесячной абонентской платы, либо на основе объёма операций;
- техническая поддержка приложения включена в оплату;
- модернизация и обновление приложения происходит оперативно и прозрачно для клиентов

Как и во всех формах облачных вычислений, заказчики платят не за владение программным обеспечением как таковым, а за его аренду.

### ✓ PaaS

Платформа как сервис.

Модель обслуживания в облачных вычислениях, по которой потребитель получает доступ к использованию информационно-технологических платформ:

- ОС
- СУБД
- Средствам разработки и тестирования

В этой модели вся информационно-технологическая инфраструктура, включая вычислительные сети, серверы, системы хранения, целиком управляется провайдером, им же определяется набор доступных видов платформ и набор управляемых параметров платформ.

Потребителю предоставляется возможность использовать платформы, создавать их виртуальные экземпляры, устанавливать, разрабатывать, тестировать, эксплуатировать на них прикладное ПО, при этом динамически изменяя количество потребляемых вычислительных ресурсов.

Провайдер облачной платформы может взимать плату с потребителей в зависимости от уровня потребления, тарификация возможна по времени работы приложений потребителя, по объёму обрабатываемых данных и количеству транзакций над ними, по сетевому трафику.

Кратко:

- IaaS - сервера, на которых можно разместить ПО. VDS/VPS.
- SaaS - готовое ПО, которое предоставляется по подписке в виде веб-сервиса. Например Microsoft 365.
- PaaS - платформа на которой вы можете разрабатывать ПО, не используя ресурсы вашего ПК. Например Google Collab.

② **Виртуализация на уровне ядра - Все виртуальные машины используют общее модифицированное ядро сервера. При этом для пользователя каждая виртуальная машина выглядит как отдельный сервер.**

Достоинства данного вида виртуализации высокая скорость работы виртуальных машин, малые издержки на процесс виртуализации, экономия ресурсов за счет загрузки одного, общего для всех ядра ОС.

Основным недостатком является невозможность запуска виртуальной машины с ОС отличной от базовой.

### ☰ Дополнения

Существуют также и другие виды виртуализации, например аппаратная виртуализация.

## Стандарты

Стандарты издаются различными обществами:

- ISO (International Organization for Standardization) - международная организация по стандартизации.
- IEEE (Institute of Electrical and Electronics Engineers) - институт инженеров электротехники и электроники. Некоммерческая ассоциация из США. Крупнейшая в мире ассоциация технических специалистов. Издаёт широко применяемые в мире стандарты.

Примеры стандартов:

- IEEE 802.3 - основа семейства технологий пакетной передачи данных Ethernet. Имеет десятки модификаций.
- IEEE 802.11 - набор стандартов беспроводной связи с использованием радиоволн и видимого света.
- ISOC (Internet Society) - Общество интернета. Имеет права на все документы RFC и прикладывает много усилий на их распространение.
- IANA (Internet Assigned Numbers Authority) - Администрация адресного пространства Интернет. Управляет пространствами IP-адресов, доменов верхнего уровня, а также регистрирует типы данных MIME и параметры прочих протоколов интернета. Имеет отделение RIR (Regional Internet Registry) - занимается распределением адресного пространства IP в пределах географического региона.

В каждой организации свой порядок принятия стандартов. Например в ISO существует 6 стадий:

1. Стадия предложения - Новое предложение направляется на голосование членам соответствующего ТК или ПК, чтобы определить необходимость включения соответствующего пункта в программу работы..
2. Подготовительная стадия - Могут приниматься различные редакции рабочих черновиков, до тех пор, пока рабочая группа не решит, что ею разработано наилучшее техническое решение рассматриваемой проблемы.

3. Стадия комитета - Могут готовиться разные редакции черновика комитета до тех пор, пока не будет достигнут консенсус по техническому содержанию текста. Как только консенсус достигнут, текст окончательно редактируется для представления в качестве черновика международного стандарта.
4. Стадия вопросов - Черновик международного стандарта (ЧМС) распространяется среди всех членов ИСО Главным Секретариатом ИСО для голосования и замечаний в течение пяти месяцев. Он одобряется для представления в качестве окончательной редакции черновика международного стандарта (ОЧМС), если две трети У-членов ТК/ПК высказываются «за», и не более одной четверти от общего количества голосов поданы «против». Если критерии одобрения не выполнены, текст возвращается в исходный комитет для дальнейшего изучения, и исправленный документ снова публикуется для голосования и замечаний в качестве черновика международного стандарта.
5. Стадия одобрения - Окончательная редакция черновика международного стандарта распространяется среди всех членов ИСО Главным Секретариатом ИСО для итогового голосования за/против в течение двух месяцев. Если в этот период поступают технические замечания, на этой стадии они уже не рассматриваются, но регистрируются для анализа в ходе будущего пересмотра данного международного стандарта. Текст одобряется для представления в качестве окончательной редакции черновика международного стандарта, если две трети высказываются «за», и не более одной четверти от общего количества голосов поданы «против». Если эти критерии одобрения не выполнены, стандарт возвращается в исходный комитет для пересмотра с учётом технических причин, представленных в поддержку голосов «против».
6. Стадия публикации - Итоговый текст отсылается в Главный Секретариат ИСО, который публикует данный международный стандарт.

Стадия	Подраздел стадии			90 Подразделы стадии принятия решения			
	00 Регистрация	20 Начало основной работы	60 Завершение основной работы	92 Повторение предыдущей фазы	93 Повторение текущей фазы	98 Прекращение	99 Продолжение
00 Предварительная стадия	00.00 Предложение нового проекта получено	00.20 Предложение нового проекта рецензируется	00.60 Завершение рецензирования			00.98 Предложение нового проекта отвергнуто	00.99 Одобрение голосования за предложение нового проекта
10 Стадия предложения	10.00 Предложение нового проекта зарегистрировано	10.20 Начато голосование за новый проект	10.60 Завершение голосования	10.92 Предложение возвращено подающему на уточнение формулировки		10.98 Новый проект отвергнут	10.99 Новый проект одобрен
20 Подготовительная стадия	20.00 Новый проект зарегистрирован в рабочей программе Технического комитета/Подкомитета	20.20 Начато изучение Рабочего черновика	20.60 Завершение периода подачи замечаний			20.98 Проект удалён	20.99 Рабочий черновик одобрен для регистрации в качестве Черновика комитета
30 Стадия комитета	30.00 Черновик комитета зарегистрирован	30.20 Начато изучение/голосование за Черновик комитета	30.60 Завершение периода голосования/подачи замечаний	30.92 Черновик комитета возвращён в рабочую группу		30.98 Проект удалён	30.99 Черновик комитета одобрен для регистрации в качестве Черновика международного стандарта
40 Стадия вопросов	40.00 Черновик международного стандарта зарегистрирован	40.20 Начато голосование за Черновик международного стандарта: 5 месяцев	40.60 Завершение голосования	40.92 Полный отчёт опубликован: Черновик международного стандарта возвращён в Технический комитет или Подкомитет	40.93 Полный отчёт опубликован: итоги нового голосования за Черновик международного стандарта	40.98 Проект удалён	40.99 Полный отчёт опубликован: Черновик международного стандарта одобрен для регистрации в качестве Окончательной редакции черновика международного стандарта
50 Стадия одобрения	50.00 ОЧМС зарегистрирован для официального одобрения	50.20 Начато голосование за Окончательную редакцию черновика международного стандарта: 2 месяца. Окончательная редакция передана в секретариат	50.60 Завершение голосования. Окончательная редакция возвращается в секретариат	50.92 Окончательная редакция черновика международного стандарта возвращена в Технический комитет или Подкомитет		50.98 Проект удалён	50.99 Окончательная редакция одобрена для публикации
60 Стадия публикации	60.00 Международный стандарт готовится к публикации		60.60 Международный стандарт опубликован				
90 Стадия пересмотра		90.20 Международный стандарт в стадии планового рецензирования	90.60 Завершение рецензирования	90.92 Международный стандарт требуется исправить	90.93 Международный стандарт подтверждён		90.99 Отзыв международного стандарта, предложенного Техническим комитетом или Подкомитетом
95 Стадия отзыва		95.20 Начато голосование за отзыв	95.60 Завершение голосования	95.92 Решено не отзываться международный стандарт			95.99 Отзыв международного стандарта

**Остальное (команды)**

- ping - Отправляет пакеты по указанному IP, необходим, чтобы проверить связь с указанным адресом.
- ipconfig - Показывает информацию о настройке протокола IP для Windows.
- ifconfig - Показывает настройки сетевых интерфейсов в LINUX.
- tracert - Показывает путь, который проходит пакет до указанного адреса, а также задержку по этому пути (Windows).
- traceroute - Аналог tracert для Linux.