

Тема 5

ИНТЕРФЕЙСЫ, ПРОТОКОЛЫ ТЕХНИЧЕСКОЙ ЭКСПЛУАТАЦИИ

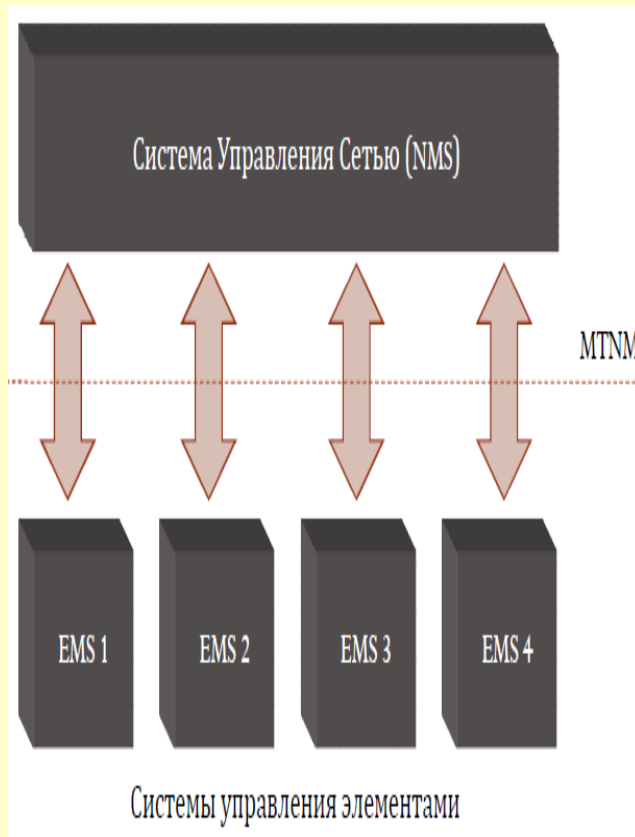
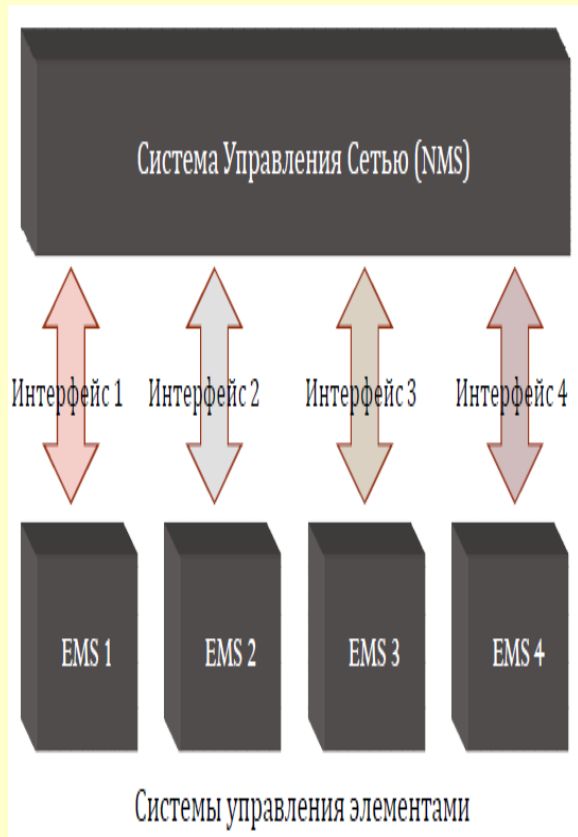
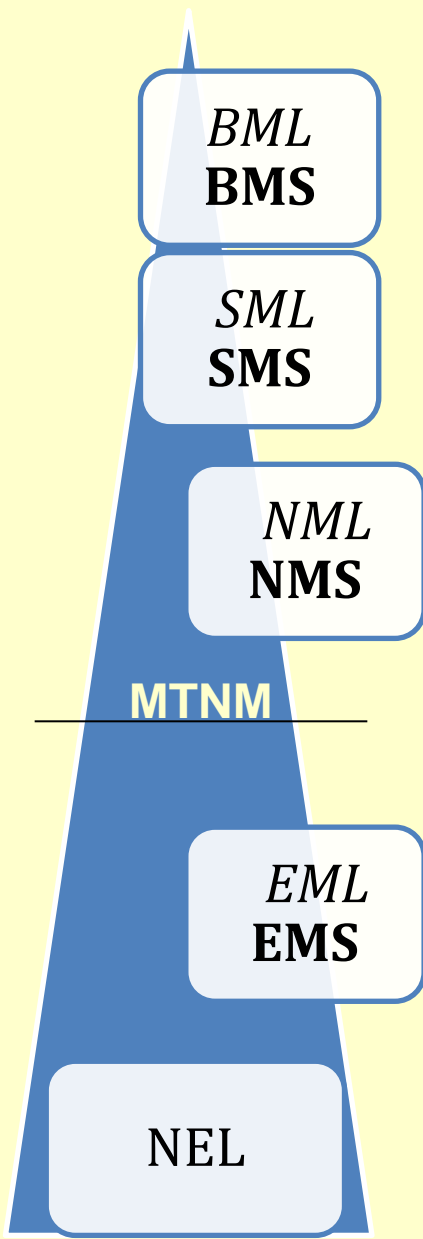
- *Интерфейс управления мультитехнологичными сетями MTNM*
- *Интерфейс интеграции компонентов систем эксплуатационного управления OSS-MTOSI*
- *Middleware - связующее программное обеспечение*
- *Протоколы эксплуатационного управления оборудованием сети*
 - ❖ *CORBA, SOAP*
 - ❖ *TR-069*
 - ❖ *Протокол SNMP*

Интерфейс управления мультитехнологичными сетями **MTNM** (*Multi-Technology Network Management*)

- ❖ Служит для обмена информацией (точка-точка) **между** системой управления сетью **NMS** и системами управления элементами **EMS** в многоуровневой логической архитектуре управления TMN. Решает проблему унифицированного управления гетерогенными сетями
- ❖ Проработан для технологий PDH, SDH/SONET, WDM, ATM, FR, Ethernet, DSL.
- ❖ Поддерживает функции эксплуатационного управления FCAPS TMN и процессы eTOM.
- ❖ Реализует «северный» интерфейс в EMS ведущих вендоров.
- ❖ Разработан TM Форумом и принят МСЭ-Т (Рек. М.3170.х 2007 г.).
- ❖ Использует CORBA (клиент-сервер архитектура).

MTNM – унифицированный интерфейс между NMS и EMS

МНОГОУРОВНЕВАЯ АРХИТЕКТУРА ТМН

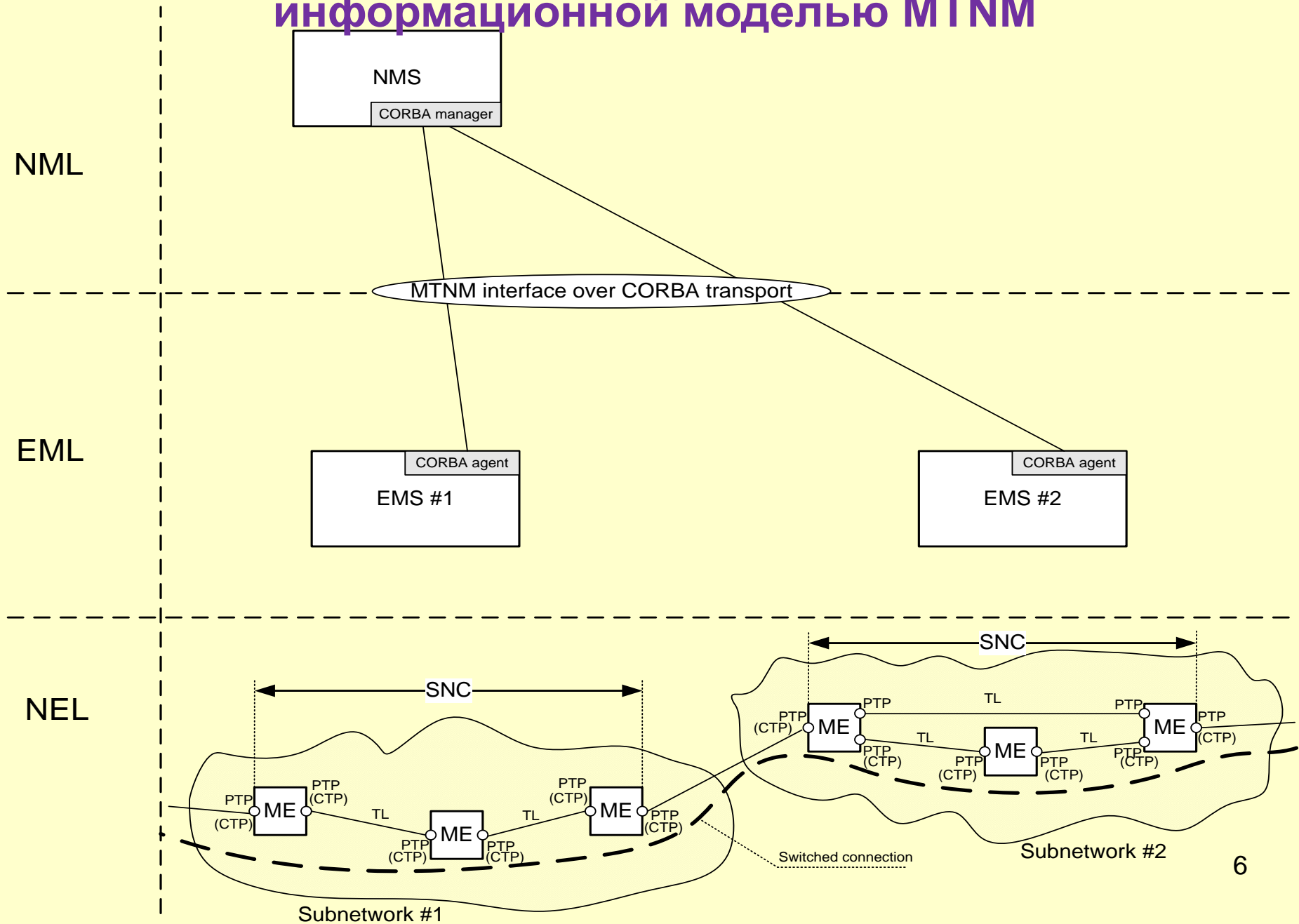


Информационная модель MTNM для транспортной сети

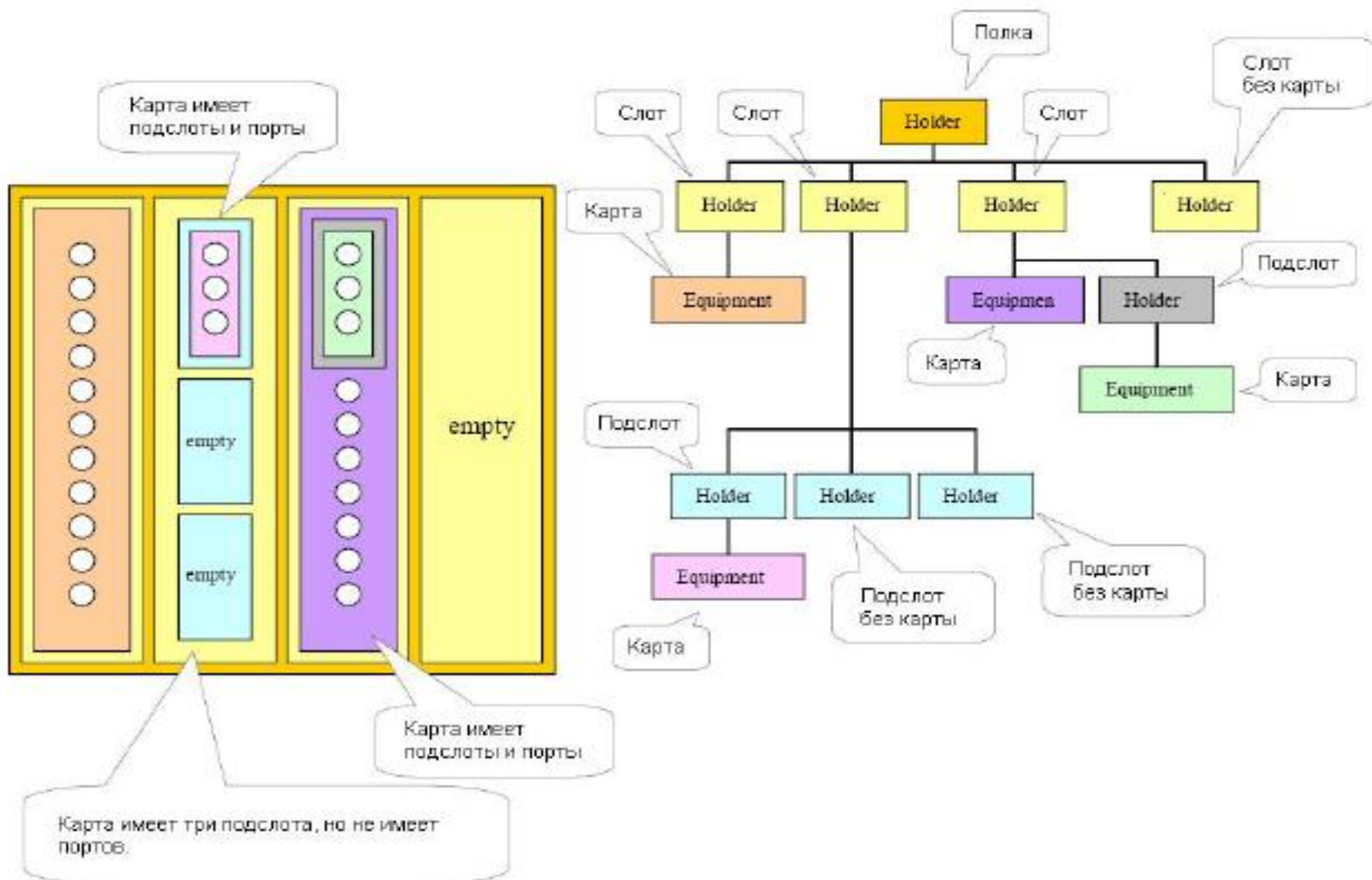
Класс	Описание	Пример атрибутов	Общие атрибуты
EMS	Представляет абстракцию EMS, управляющей подсетью	<ul style="list-style-type: none"> Версия ПО Тип EMS 	<ul style="list-style-type: none"> Имя Пользовательская метка Сокращенное имя EMS Владелец Дополнительная информация
Subnetwork	Подсеть. Содержит управляемые элементы (Managed Element) и физические соединения между ними (Topological Link). Через подсеть возможно установление сквозного соединения — Subnetwork Connection	<ul style="list-style-type: none"> Тип подсети Поддерживаемые скорости 	
Managed Element (ME)	Управляемый элемент. Является отображением сетевого элемента (NE) для EMS	<ul style="list-style-type: none"> Местонахождение Версия ПО Название продукта Производитель Поддерживаемые скорости ПД 	
Topological Link (TL)	Элемент топологии сети — физическая связь между портами сетевых элементов	<ul style="list-style-type: none"> Направление Конечные точки Скорость ПД Аварийная сигнализация 	
Termination Point (TP)	Сетевое окончание. Логическая абстракция, представляет собой конечную точку Topological Link или Subnetwork Connection	<ul style="list-style-type: none"> Направление Наличие интерфейса мониторинга 	

Класс	Описание	Пример атрибутов	Общие атрибуты
Connection Termination Point (CTP)	Окончание логического соединения (SNC)	<ul style="list-style-type: none"> • Состояние соединения • Режим отображения 	<ul style="list-style-type: none"> •Имя •Пользовательская метка •Сокращенное имя EMS •Владелец •Дополнительная информация
Subnetwork connection (SNC)	Транспортный маршрут данных через подсеть. Формируется как ассоциирование портов на границе подсети.	<ul style="list-style-type: none"> • Направление • Конечные точки • Скорость ПД 	
Physical Termination Point (PTP)	Физическое окончание. Отображает окончание Topological Link. Фактически, является абстракцией порта сетевого элемента	<ul style="list-style-type: none"> • Направление • Наличие интерфейса мониторинга 	
Equipment Holder	Описание стативов, стоек, слотов для оборудования	<ul style="list-style-type: none"> • Тип • Аварийная сигнализация • Установленное оборудование • Допустимое оборудование • Состояние • Местонахождение • Производитель 	

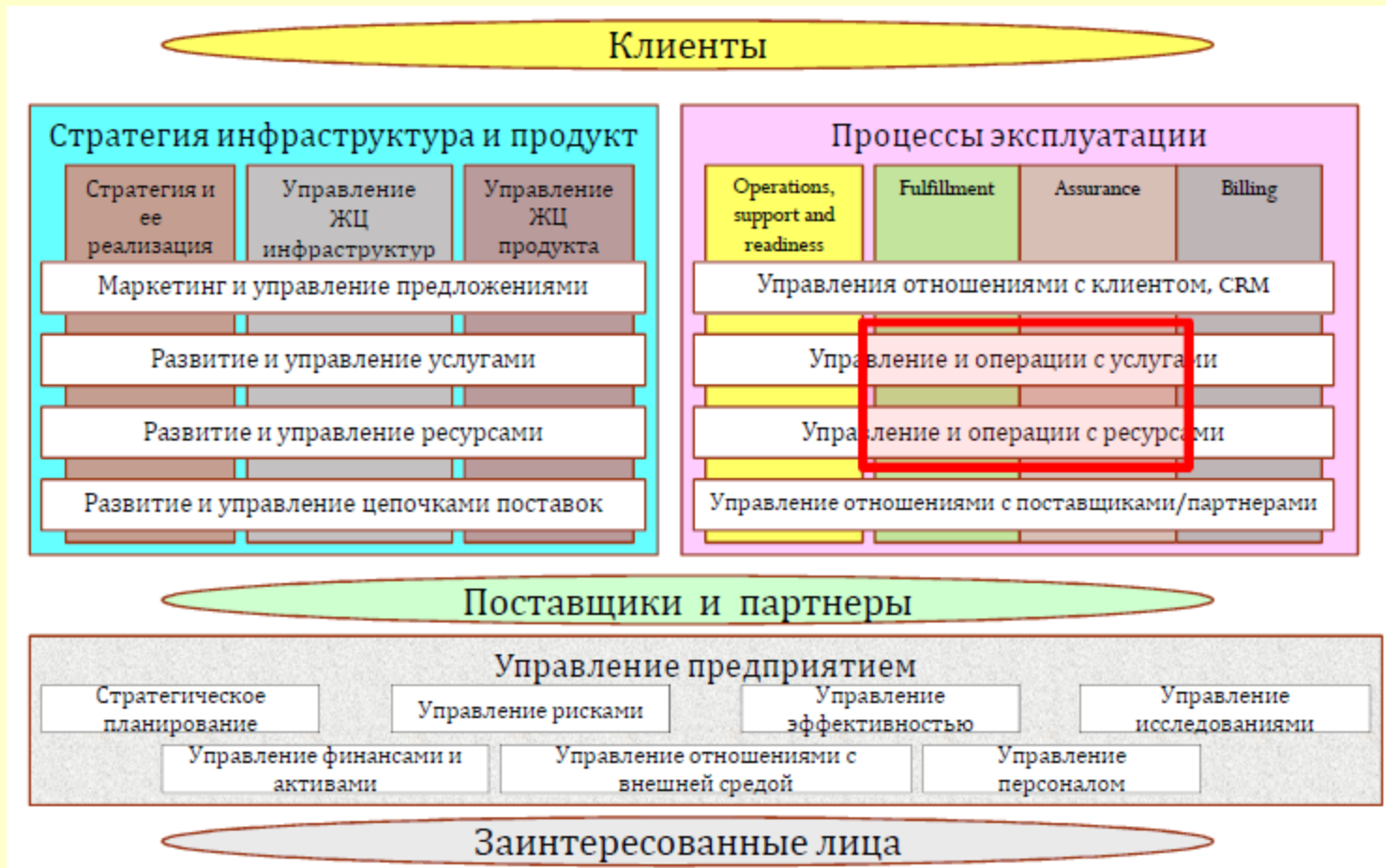
Представление сети в соответствии с информационной моделью MTNM



Отображение структуры оборудования (equipment) с помощью MTNM



Соответствие MTNM процессам eTOM



Процессы передающие информацию через MTNM-интерфейс

- **Конфигурация и активация услуги** (*Service Configuration & Activation*)
- **Управление проблемами при предоставлении услуги** (*Service Problem Management*)
- **Предоставление ресурса** (*Resource Provisioning*)
- **Управление неисправностями ресурса** (*Resource Trouble Management*)
- **Управление рабочими характеристиками ресурсов** (*Resource Performance Management*)
- **Сбор и распределение информации от ресурса** (*Resource Data Collection & Processing*)

Мультитехнологичный интерфейс эксплуатационных систем **MTOSI** (*Multi-Technology Operations System Interface*)

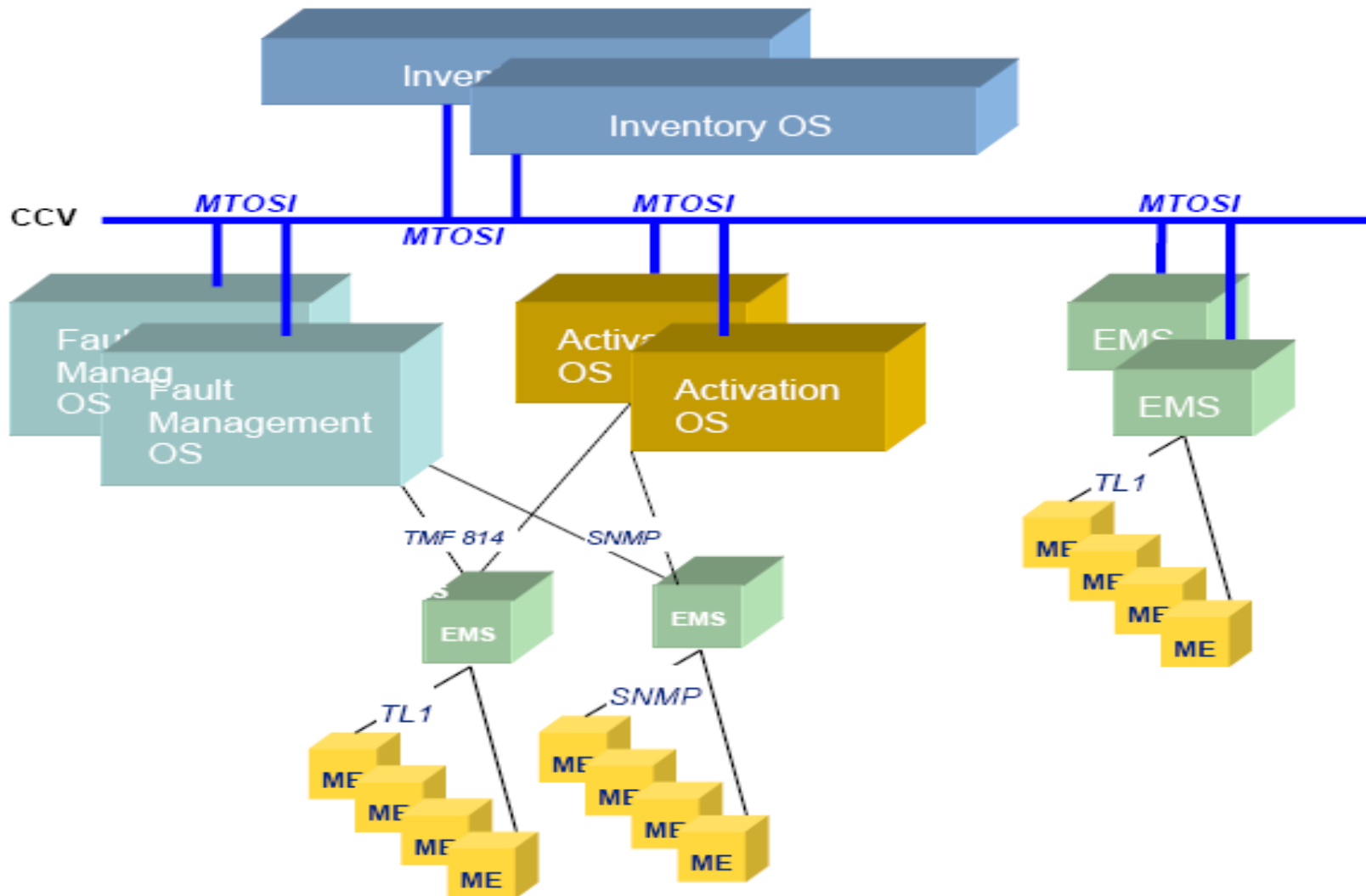
- *Стандартизован TMF для реализации интерфейсов взаимодействия между подсистемами/компонентами **OSS**.*
- В основу принципа коммуникации положена сервисная шина предприятия ESB (*enterprise service bus*). ESB связующее программное обеспечение, обеспечивающее централизованный и унифицированный событийно-ориентированный обмен сообщениями между различными информационными системами на принципах SOA. В общем виде это CCV-общая коммуникационная среда (**Common Communication Vehicle** - is the common **middleware** required to implement MTOSI. **CCV** is a middleware abstraction that allows MTOSI interfaces to be bound to different middleware technologies as needed).
- MTOSI – *расширение MTNM* с точки зрения области применения интерфейса не только между уровнем управления сетевыми элементами и уровнем управления сетью, а и уровнем услуг. MTNM интерфейс больше ориентирован на CORBA- архитектуру взаимодействия, которая используется для распределенных вычислительных систем, а **MTOSI**, **основанный на XML, SOAP**, в качестве транспорта сообщений использует шину ESB.

Сравнение MTNM и MTOSI

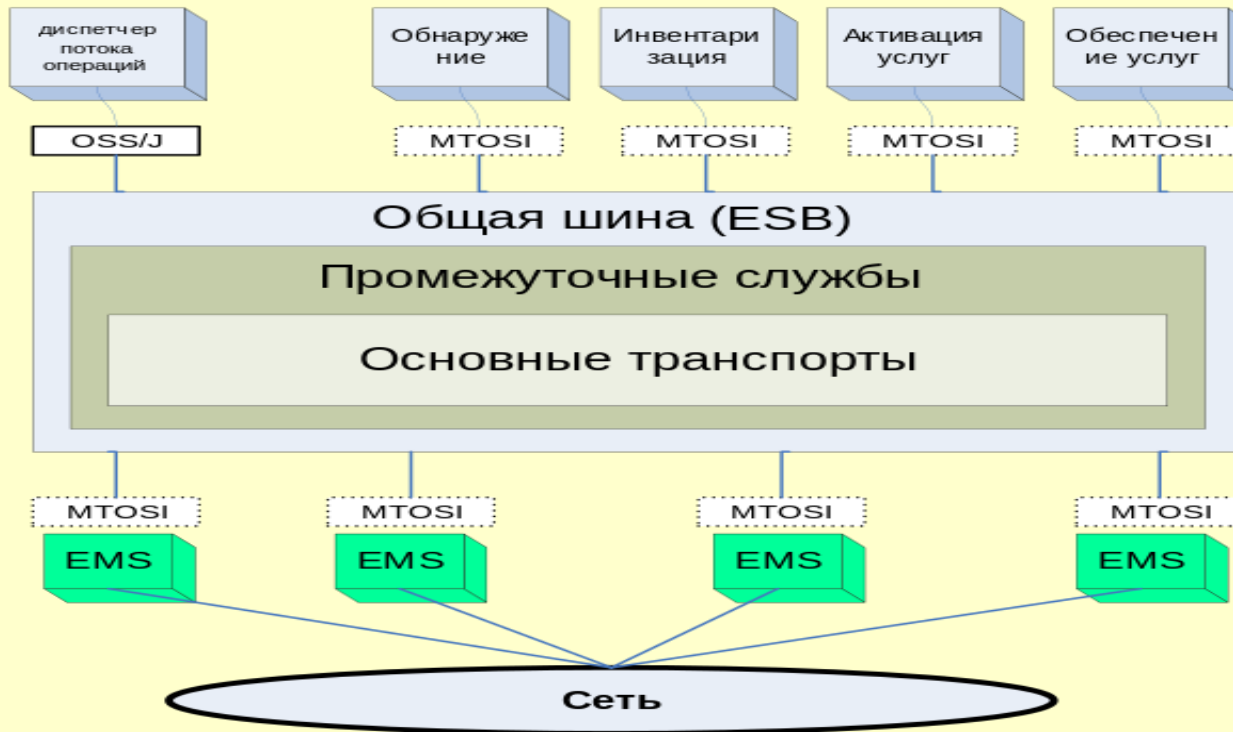
	MTNM	MTOSI
Информационная модель	TMF 608	
Взаимодействующие системы	EMS<->NMS	OS (EMS/MNS/SMS)
Режим общения	Точка-точка	Общая шина (CCV)
Транспортная технология	CORBA (клиент-сервер)	JMS (Java Message Service), HTTP, SOAP
Дополнения		Дополнены механизмы групповых операций с оборудованием, механизмы уведомлений об изменениях сети и ошибках

Место интерфейса MTOSI в системе управления оператором

(Уч. Пос. АГФ ЭУИ РИС. 6.6.)



Системы разных производителей взаимодействуют с помощью MTOSI



Middleware в инфраструктуре оператора связи

Интерфейсы northbound (к IT) и southbound (к сети)



Middleware термин, означающий слой или комплекс технологического программного обеспечения для взаимодействия между различными приложениями, системами, компонентами.

Роль Middleware

1. Активация услуг

- Конфигурация сетевых элементов

2. Работа с неисправностями

- Формирование Trouble Tickets
- Устранение неисправностей путем реконфигурации

3. Сбор информации

- Первичное наполнение БД тех.учета
- Постоянная актуализация данных

4. Мониторинг сетевых ресурсов

- Постоянный сбор информации о состоянии оборудования
- Единая БД мониторинга с возможностью анализа данных

Библиотека стандартных интерфейсов TM Forum

1. **OSS/J (OSS through Java).** *Интерфейсы данной группы предназначены главным образом для обеспечения функций уровня услуг и бизнес-управления, но не управления сетью. Это API для интеграции информационных систем, работающих на платформе Java. Также включает профиль XML, позволяющий использовать интерфейсы OSS/J в средах, отличных от Java.*
2. **MTOSI.** *Интерфейсы для управления сетями и услугами на основе XML.*
3. **OMI (Operations Management Interfaces).** *Интерфейсы управления операционной (эксплуатационной) деятельностью. Для управления сетями и услугами в конвергентной среде. Построены на основе лучших компонентов из интерфейсов TM Forum OSS/J и MTOSI, а также 3GPP. На текущий момент включает интерфейсы для управления решением проблем на уровне услуг, доступа к каталогу ресурсов, обеспечения безопасного доступа в систему, управления неисправностями, управления производительностью и др.*
4. **SMI (SES Management Interface, реализации SOAP и RESTful)** – *интерфейсы управления SES (Software Enabled Service), обеспечивающие доступ к функционалу реализованных сервисов SOA. SMI предоставляет возможности для SES устойчиво взаимодействовать с системами управления на любом этапе цепочки создания стоимости в многопартнерской среде.*

Протоколы эксплуатационного управления оборудованием

- CMIP/Common Management Information Protocol (ITU-T, серия рекомендаций **X.700**)
- CORBA (Common Object Request Broker Architecture) –
Общая архитектура брокера объектных запросов
- TL1/Transaction Language 1
- RMON/Remote MONitoring
- TR-069 CWMP (DSL Forum → Broadband Forum)
- SNMP/Simple Network Management Protocol (IETF)
- CLI

CORBA / *Common Object Request Broker Architecture*

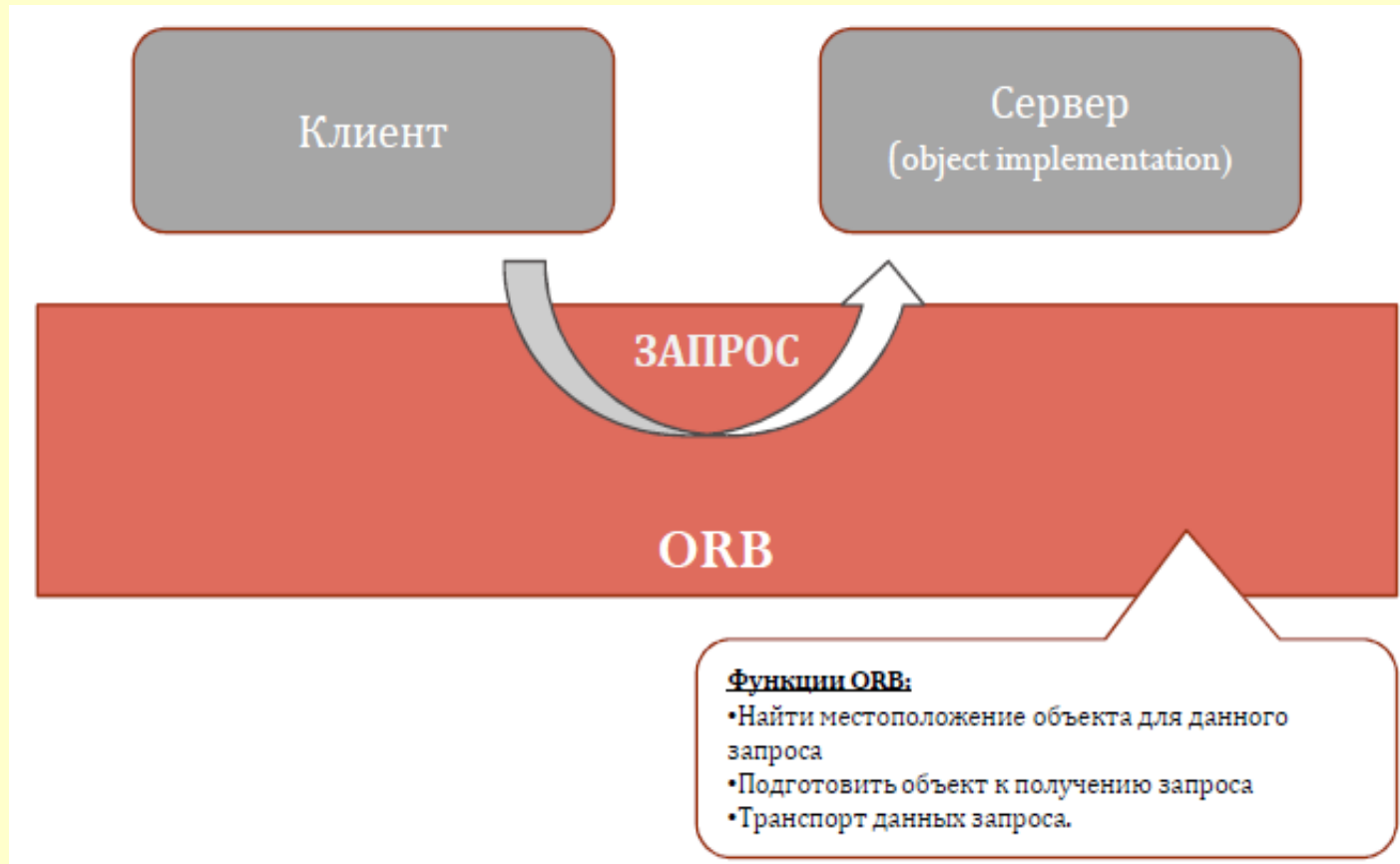
Общая архитектура брокера (посредника) запросов к объектам

- Разработка (с 1991г.) некоммерческой организации OMG (Object Management Group <http://www.omg.org/>). OMG разрабатывает стандарты (в том числе *UML / Unified Modeling Language*) для приложений, взаимодействующих в *условиях разнородной и распределенной среды*.
- Общая архитектура посредника запросов к объектам - структурирована таким образом, чтобы обеспечить интеграцию большого разнообразия объектных систем.
- CORBA – это стандарт для объединения объектов.
- CORBA —это стандарт написания распределённых приложений. Задача CORBA — обеспечить интеграцию изолированных систем, дать возможность программам, написанным на разных языках и работающим на разных узлах сети, взаимодействовать друг с другом так же просто, как если бы они находились в адресном пространстве одного процесса.

Терминология CORBA

- *Объект (Object)*– идентифицируемая сущность, предоставляющая один или несколько сервисов, которые могут быть запрошены клиентом. Объекты формируют *системы объектов (Object System)*.
- Клиенты вызывают сервисы с помощью *запросов (Request)*. Информация в запросе – операция, адресуемый объект, ноль или больше параметров, необязательные комментарии.
- *Интерфейс (Interface)* – описание набора возможных операций, которые клиент может запрашивать от объекта.
- *ORB (Object Request Broker)* – брокер (посредник) запросов к объектам. Это стандартный механизм взаимодействия, с помощью которого объекты могут отправлять друг другу запросы и получать на них ответы в пределах одной машины или в разнородной сети. Он действует по сути, как программная шина-интерфейс для связи приложений с удаленными объектами. Это избавляет программиста от необходимости знать, где находится объект и в каком он состоянии (активен или нет).

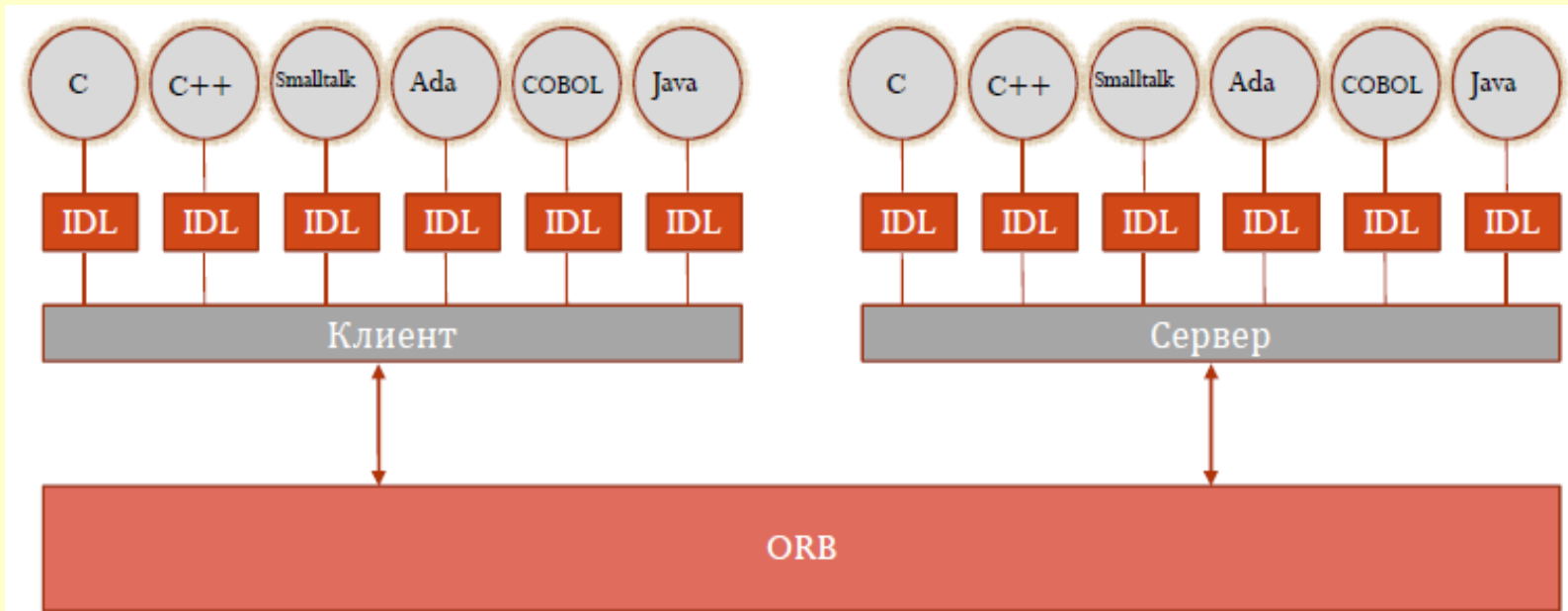
Реализация запроса



CORBA это клиент-серверная технология, которая выполняет функции промежуточного ПО объектной среды. Клиент здесь – удаленное приложение. Взаимодействие между клиентским процессом и сервером происходит с использованием механизма вызова удаленной процедуры ORPC / Object Remote Procedure Call.

OMG IDL / *Interface Definition Language*

Язык определения интерфейсов



➤ **IDL позволяет определить интерфейсы объектов и описать их структуру.**
Не зависит от реализации и является предпосылкой для создания объектов, работающих в гетерогенной среде.

➤ IDL позволяет взаимодействовать клиентским и серверным объектам, написанным на различных языках.

➤ **Взаимодействие объектов, создаваемых в разных языковых средах, обеспечивается с помощью отображения спецификаций IDL в языки программирования C, C++, Java и др. По синтаксису IDL подобен C++.**

Протокол SOAP

SOAP — протокол обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур (RPC), а название было аббревиатурой: *Simple Object Access Protocol* — **простой протокол доступа к объектам**. Сейчас протокол используется для обмена произвольными сообщениями в формате XML, а не только для вызова процедур. Официальная спецификация последней версии 1.2 протокола никак не расшифровывает название SOAP. SOAP является расширением протокола XML-RPC.

- SOAP может использоваться с любым протоколом прикладного уровня: SMTP, FTP, HTTP, HTTPS и др. Однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно.
- Чаще всего SOAP используется поверх HTTP.
- SOAP является одним из стандартов, на которых базируются технологии *веб-сервисов*.

TR-069 CWMP

Протокол дистанционного управления абонентским оборудованием

TR-069 (*technical report 069*) — спецификация, описывающая протокол CWMP (**CPE WAN Management Protocol**). CPE (**customer premises equipment**) —оборудование, установленное в помещении абонента/клиента/пользователя. WAN (**wide area network**) — сеть, охватывающая большие территории.

Стандарт TR-069 опубликован в 2004 году консорциумом «DSL Forum» (переименован в 2008 году в «Broadband Forum»). Ноябрь 2013 года выпуск стандарта версии 1.4.

Цель создания протокола - стандартизация и унификация принципов и методов дистанционного автоматизированного управления оконечным оборудованием клиентов, выпускаемым различными производителями.

CWMP — протокол прикладного уровня семиуровневой модели OSI:

[Физический уровень](#) → [Канальный уровень](#) → [IPv4/IPv6](#) → [TCP](#) → [SSL/TLS](#) → [HTTP](#) → [SOAP](#)

CWMP передаёт данные с использованием протокола SOAP (*simple object access protocol*) — надстройки над HTTP. В протоколе SOAP данные кодируются в формате XML.

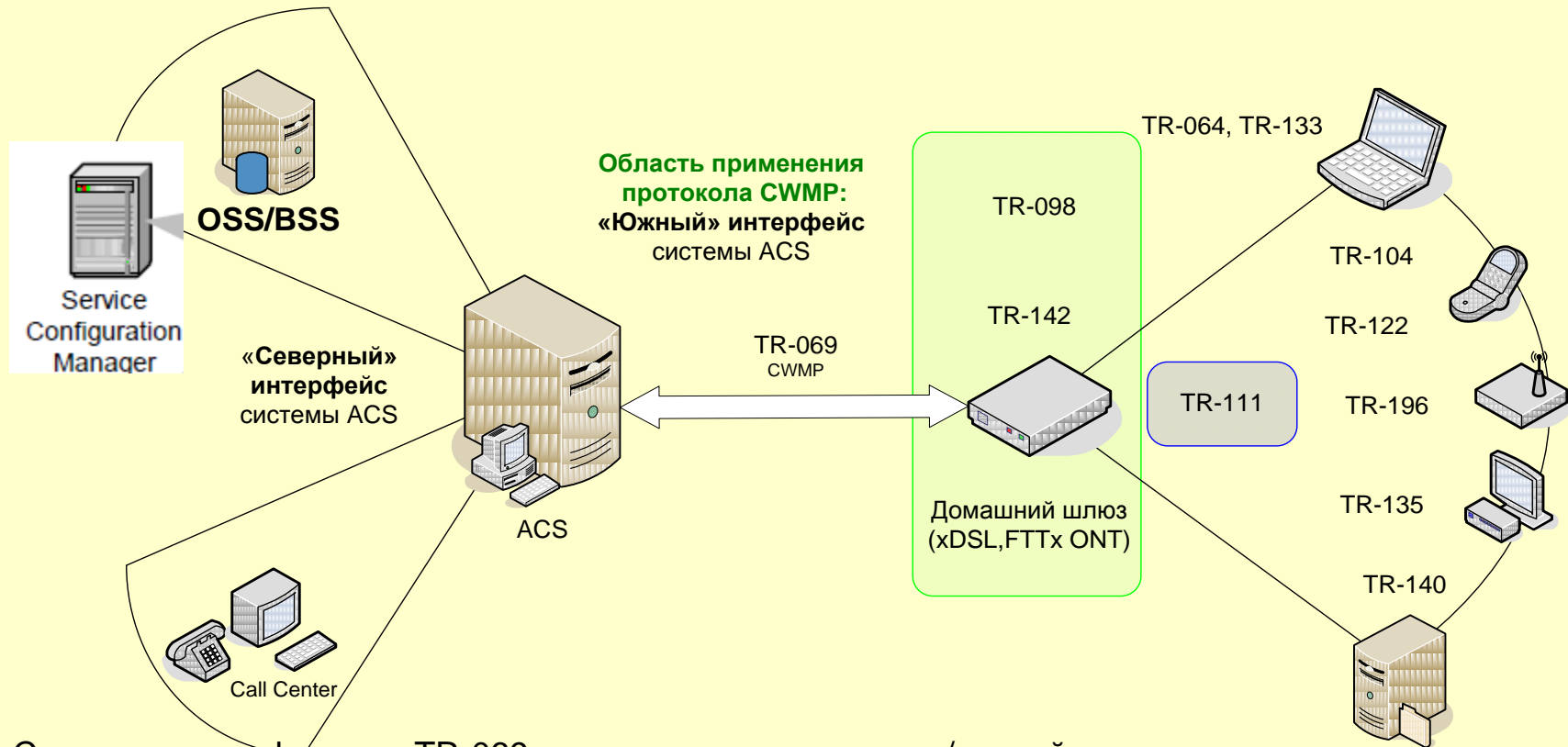
Для защиты соединения используется протокол SSL (Secure Socket Layer) или TLS (Transport Layer Security).

Возможности CWMP:

- начальная настройка устройства при его загрузке и внесение изменений в настройки уже работающего устройства;
- дистанционное (удалённое) обновление ПО/прошивки;
- удалённый доступ к log-файлам и счётчикам производительности;
- удалённая [диагностика](#) состояния устройства.

Прикладной уровень
Уровень представления данных
Сеансовый уровень
Транспортный уровень
Сетевой уровень
Канальный уровень
Физический уровень

Интерфейс систем управления абонентским оборудованием по протоколу TR-069 с OSS/BSS



Согласно спецификации TR-069 на территории оператора/провайдера должен **находиться сервер автоконфигурирования ACS** (*auto configuration server*). **Задачи сервера ACS:**

- организация взаимодействия с оборудованием абонентов/клиентов;
- обработка запросов от устройств;
- подключение дополнительных услуг/сервисов.

Сессия по протоколу CWMP может быть инициирована как со стороны оборудования клиентов CPE, так и со стороны сервера ACS (оператора/провайдера). Чтобы протокол CWMP работал, устройства клиента (modem, gateway, router, IP-phone и др.) должны иметь IP-адрес.

SNMP / *Simple Network Management Protocol*

Простой протокол управления сетью

Стандартный интернет-протокол **для управления** устройствами в IP-сетях на основе архитектур [UDP/TCP](#). К управляемым SNMP устройствам относятся маршрутизаторы, коммутаторы, серверы, рабочие станции, модемные стойки и другие.

Кроме управления устройствами часто SNMP используют **для мониторинга** подключенных к сети устройств.

SNMP определен Инженерным советом интернета (IETF/ *Internet Engineering Task Force*) как компонент семейства стандартов Интернета [TCP/IP](#).

SNMP работает на прикладном уровне TCP/IP (седьмой уровень модели OSI).

Он состоит из набора стандартов для сетевого управления, включая протокол прикладного уровня, схему баз данных и набор объектов данных.

SNMP предоставляет данные для управления в виде переменных, описывающих конфигурацию управляемой системы. Эти переменные могут быть запрошены (а иногда и заданы) управляющими приложениями.

SNMP

Версии :

- SNMPv1 (RFC 1155--1157). 1988г
- SNMPv2 (RFC 1441--1452). 1993г
- SNMPv3 (RFC 3411--3418). 2002г. *С 2004 года IETF признает SNMPv3, определенный в качестве текущей стандартной версии SNMP. IETF отметил SNMPv3 как полный Интернет-стандарт, что является самым высоким уровнем готовности для RFC.* <https://ru.wikipedia.org/wiki/SNMP> . Последнее изменение этой страницы: 14:03, 19 февраля 2016.

В системах управления, построенных на основе протокола SNMP, стандартизуются следующие элементы:

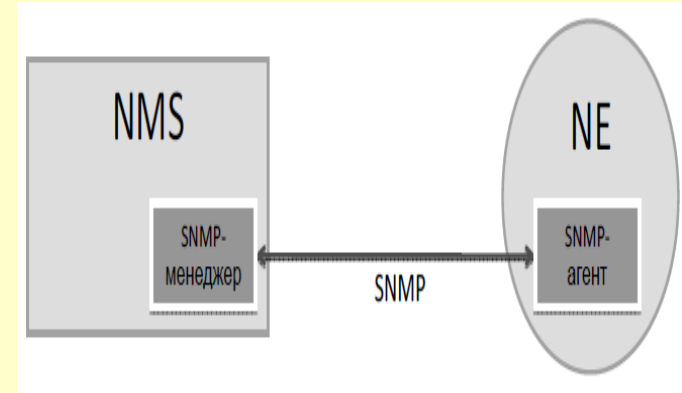
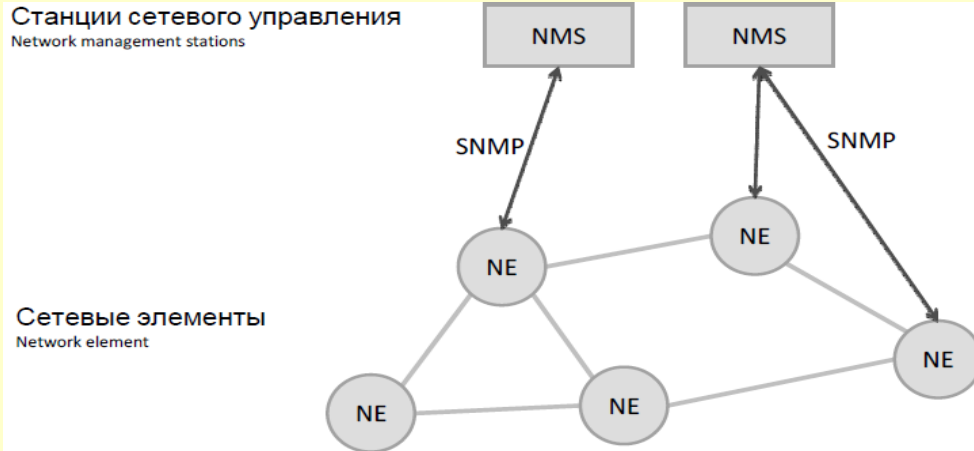
- *протокол взаимодействия агента и менеджера;*
- *язык описания моделей MIB и сообщений SNMP - язык абстрактной синтаксической нотации ASN.1;*
- *несколько конкретных моделей MIB (MIB-I, MIB-II, RMON, RMON 2), имена объектов которых регистрируются в дереве стандартов ISO.*

Архитектура протокола SNMP

При управлении сетью и мониторинге с использованием протокола SNMP рассматриваются следующие основные три компонента:

- **Элементы сети (NE)** - управляемые устройства, которые могут относиться к любому виду устройств: маршрутизаторы, серверы доступа, коммутаторы, концентраторы, IP-телефоны, IP-видеокамеры, компьютеры-хосты, принтеры и т. п.
Элемент сети содержит **SNMP агента** - ПО, запущенное на сетевом элементе. Агент обладает локальным знанием управляющей информации и переводит эту информацию в специфичную для SNMP форму или из неё (медиация данных).
- **Система (станции) сетевого управления (Network Management System / NMS).**
NMS обеспечивают основную часть обработки данных, необходимых для сетевого управления. В любой управляемой сети может быть одна и более NMS.
NMS содержит **SNMP менеджера** - ПО, устанавливаемое на один или более административных компьютеров (где функционируют программные средства, называемые *менеджерами*), которые выполняют отслеживание или управление группой хостов или устройств в сети. *Менеджеры* SNMP обрабатывают данные о конфигурации и функционировании управляемых систем и преобразуют их во внутренний формат, удобный для поддержания протокола SNMP. Протокол также разрешает активные задачи управления, например, изменение и применение новой конфигурации через дистанционное изменение этих переменных.
- **База управляющей информации (Management information base / MIB).**
SNMP MIB – компонент, который обеспечивает структурированность данных, которыми обмениваются агенты и менеджеры. По сути - это некая база данных в виде текстовых файлов. Доступные через SNMP переменные организованы в иерархии. Эти иерархии, как и другие метаданные (например, тип и описание переменной), описываются с помощью MIB.

Архитектура протокола SNMP



Для повышения масштабируемости и административной управляемости вводится понятие **прокси-агента**, который может переправлять операции протокола SNMP, а также понятие менеджера промежуточного уровня, который скрывает несущественные подробности управляющей информации от систем управления сетями верхнего уровня, интегрируя получаемые от агентов данные. Это позволяет создавать многоуровневые системы управления, соответствующие архитектурному стилю «многоуровневый клиент-сервер».

Более детальная классификация компонентов по ролям:

Менеджер

Менеджер промежуточного уровня
Система управления сетями

Агент

Минимальный агент
Прокси-агент
Менеджер промежуточного уровня

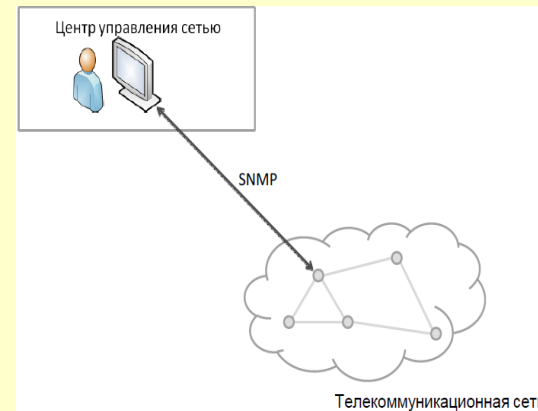
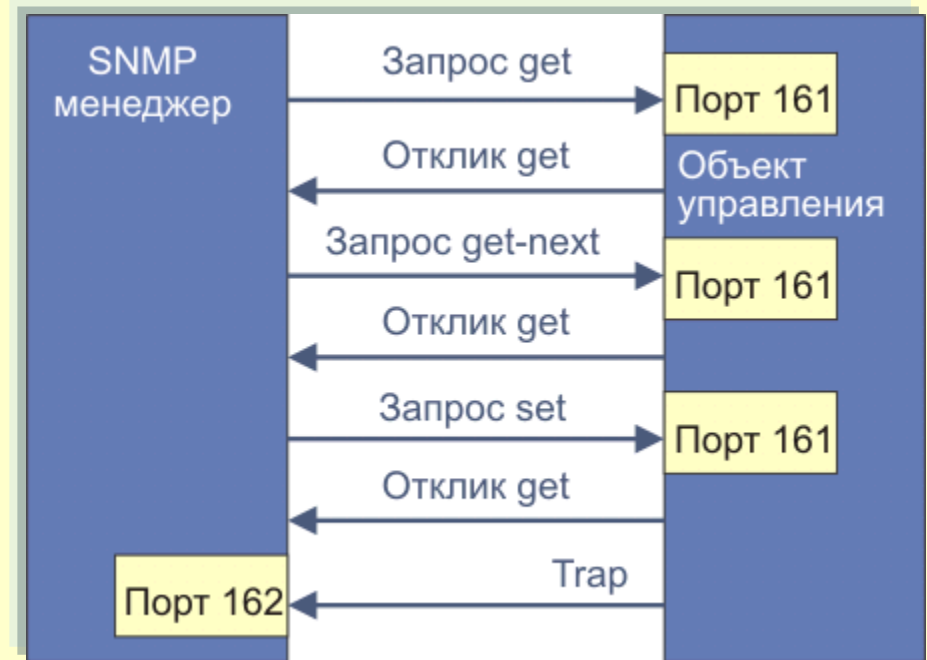
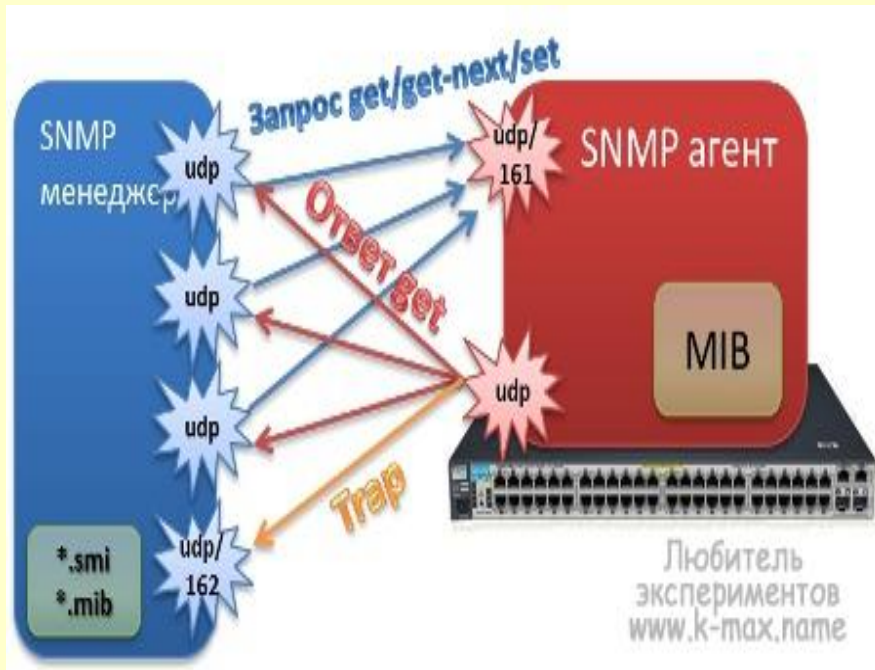


Схема взаимодействия SNMP-агент - SNMP-менеджер



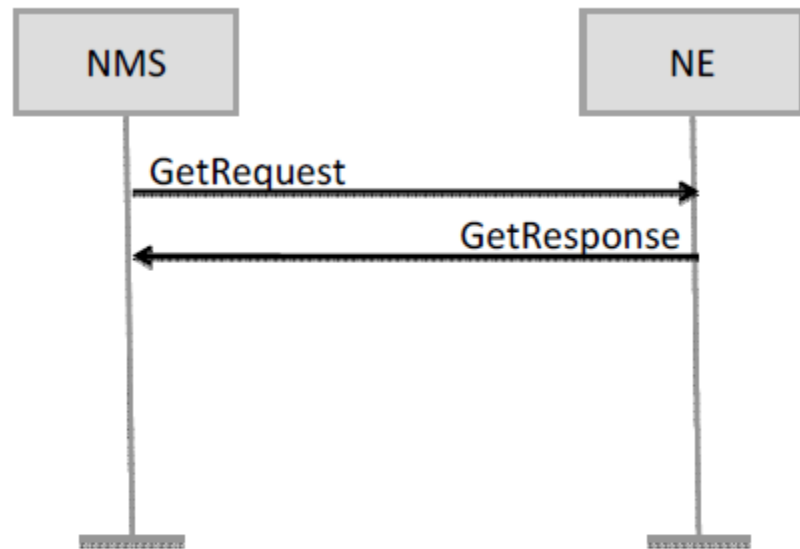
SNMP менеджер отправляет запросы агенту на порт `udp/161` (если конфигурационно в агенте не задан другой порт) с произвольного порта. В запросе SNMP менеджера указывается порт и адрес источника. Далее агент принимает пакет и обрабатывает (если выполняются определенные условия). В процессе обработки формируется ответ, который отправляется на порт взятый из исходного запроса. Ответ отправляется с `udp/161` порта. Можно сказать, что SNMP агент таким образом предоставляет доступ SNMP менеджеру к данным, хранящимся в базе MIB. При этом, в момент отправки, SNMP менеджер вставляет в PDU некий ID (RequestID), а агент в ответном PDU вставляет данный ID без изменения, для того чтобы менеджер различал пакеты от разных агентов. SNMP агент может быть настроен на посылку Trap уведомлений, которую он выполняет с доступного порта на `udp/162` порт SNMP менеджера.

Сообщения (Команды) SNMP

Команда SNMP	Тип PDU	Назначение
GetRequest	0	Запрос к агенту от менеджера, используемый для получения значения указанной переменной
GetNextRequest	1	Запрос к агенту от менеджера, используемый для получения следующего в иерархии значения переменной (следующий логический идентификатор на дереве MIB)
SetRequest	2	Запрос к агенту присвоить переменной соответствующее значение. Используется для описания действия, которое должно быть выполнено
GetReponse,	3	Ответ от агента к менеджеру на GETrequest, GETnextrequest SETrequest, GETBulkRequest , InformRequest , возвращающий запрошенные значения переменных. Содержит информацию о состоянии , коды ошибок и др.
Trap	4	Прерывание (ловушка). Одностороннее уведомление от SNMP агента сетевого объекта менеджеру о каком-либо событии
GetBulkRequest	5	Запрос к агенту на пересылки больших объемов данных, например, таблиц
InformRequest	6	Уведомление между менеджерами. Менеджер обращает внимание партнера на определенную информацию в MIB. <i>Может использоваться, например для обмена информацией о MIB. В ответ менеджер формирует аналогичный пакет в подтверждение того, что исходные данные получены.</i>
SNMPv3-Trap	7	Отклик на событие (расширение по отношению v1 и v2).
Report	8	Отчет

Типы сообщений (PDU)

- GetRequest
- GetResponse (Response)
- SetRequest
- Trap
- GetNextRequest
- GetBulkRequest
- InformRequest

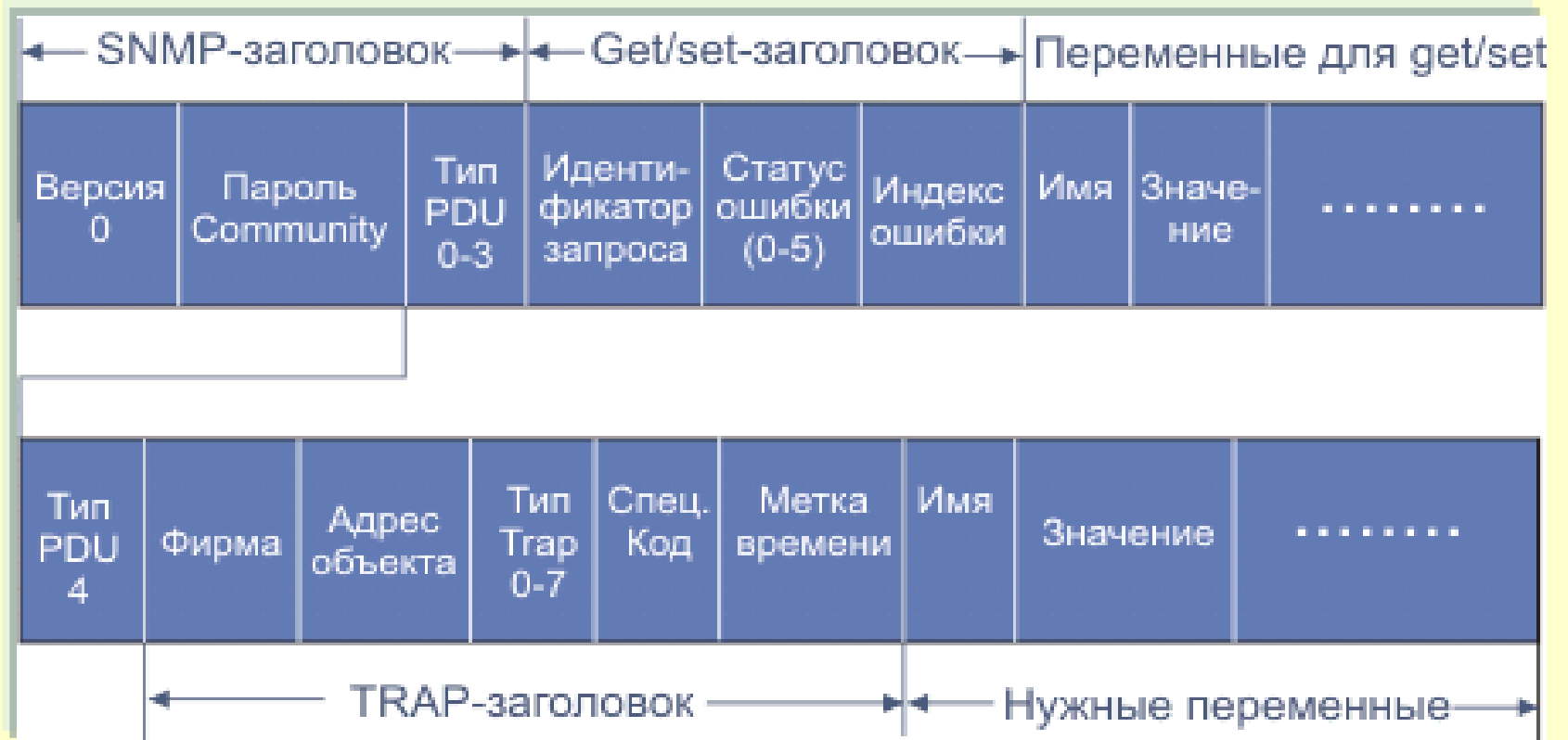


Структура сообщения

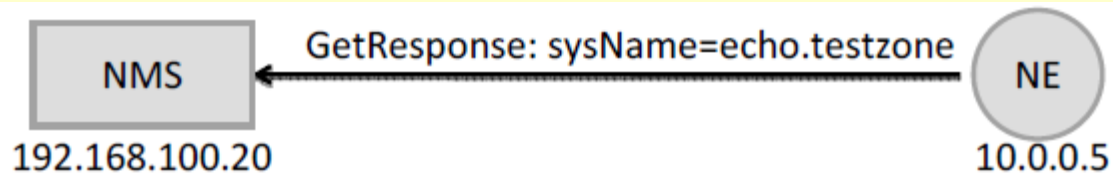
- Версия протокола (version)
- Имя группы (community name)
- Тип сообщения (PDU)

- Идентификатор запроса (RequestID)
- Статус и индекс ошибки
- *Список переменных -- имя:значение*

Формат SNMP-сообщений, вкладываемых в UDP-дейтограммы



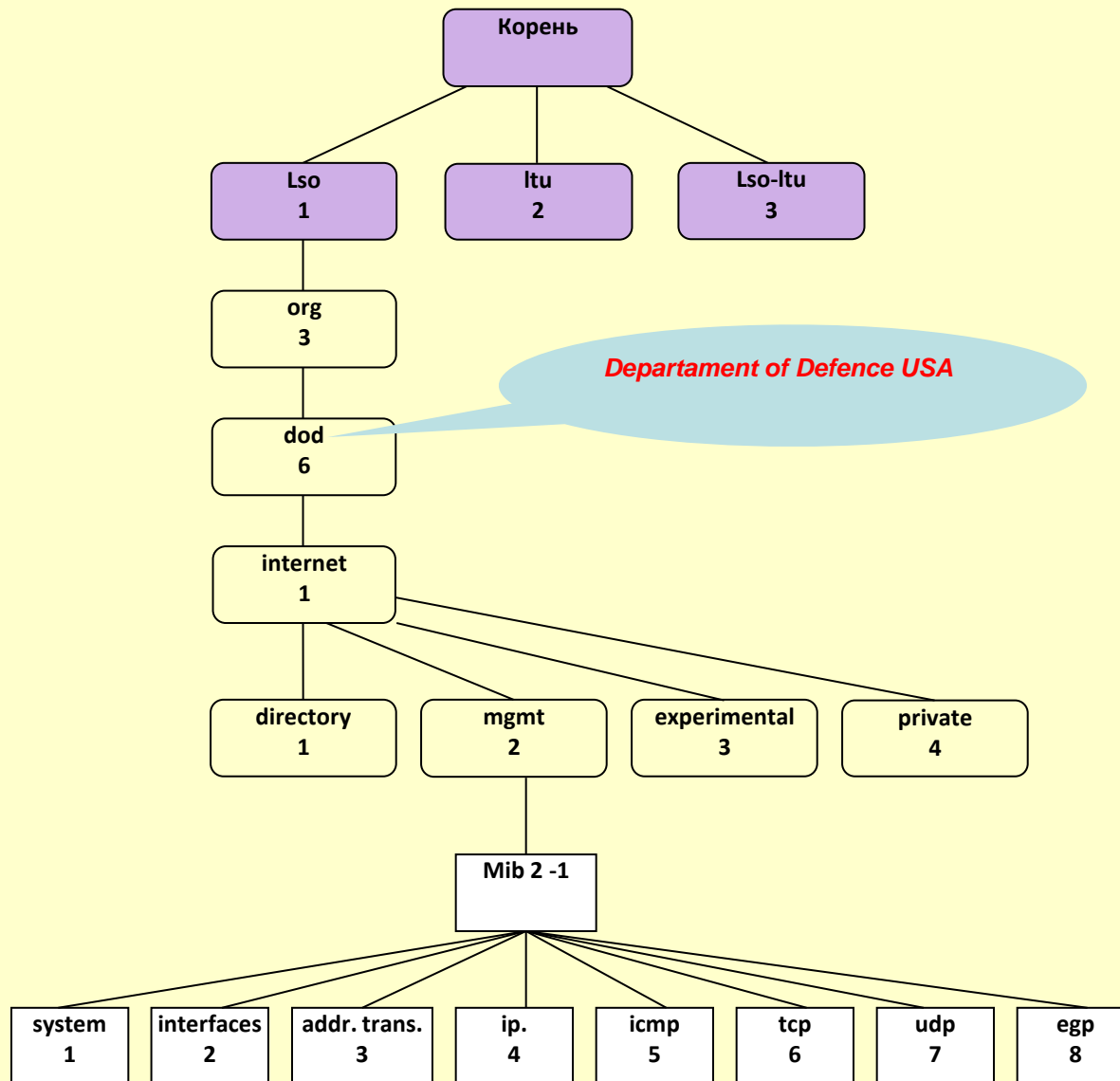
Пример сообщения



No. -	Time	Source	Destination	Protocol	Info
7	11.170000	192.168.100.20	10.0.0.5	SNMP	get-request SNMPv2-MIB::sysName.0
8	11.179000	10.0.0.5	192.168.100.20	SNMP	get-response SNMPv2-MIB::sysName.0

[-] Frame 8 (96 bytes on wire, 96 bytes captured)
[-] Ethernet II, Src: 02:00:00:00:00:0b (02:00:00:00:00:0b), Dst: ca:03:06:44:00:1c (ca:03:06:44:00:1c)
[-] Internet Protocol, Src: 10.0.0.5 (10.0.0.5), Dst: 192.168.100.20 (192.168.100.20)
[-] User Datagram Protocol, Src Port: snmp (161), Dst Port: 30487 (30487)
[-] Simple Network Management Protocol
version: version-1 (0)
community: public
[-] data: get-response (2)
[-] get-response
request-id: 19485
error-status: noError (0)
error-index: 0
[-] variable-bindings: 1 item
[-] SNMPv2-MIB::sysName.0 (1.3.6.1.2.1.1.5.0): echo.testzone
[-] Object Name: 1.3.6.1.2.1.1.5.0 (SNMPv2-MIB::sysName.0)
SNMPv2-MIB::sysName: echo.testzone

База управляющей информации МІВ



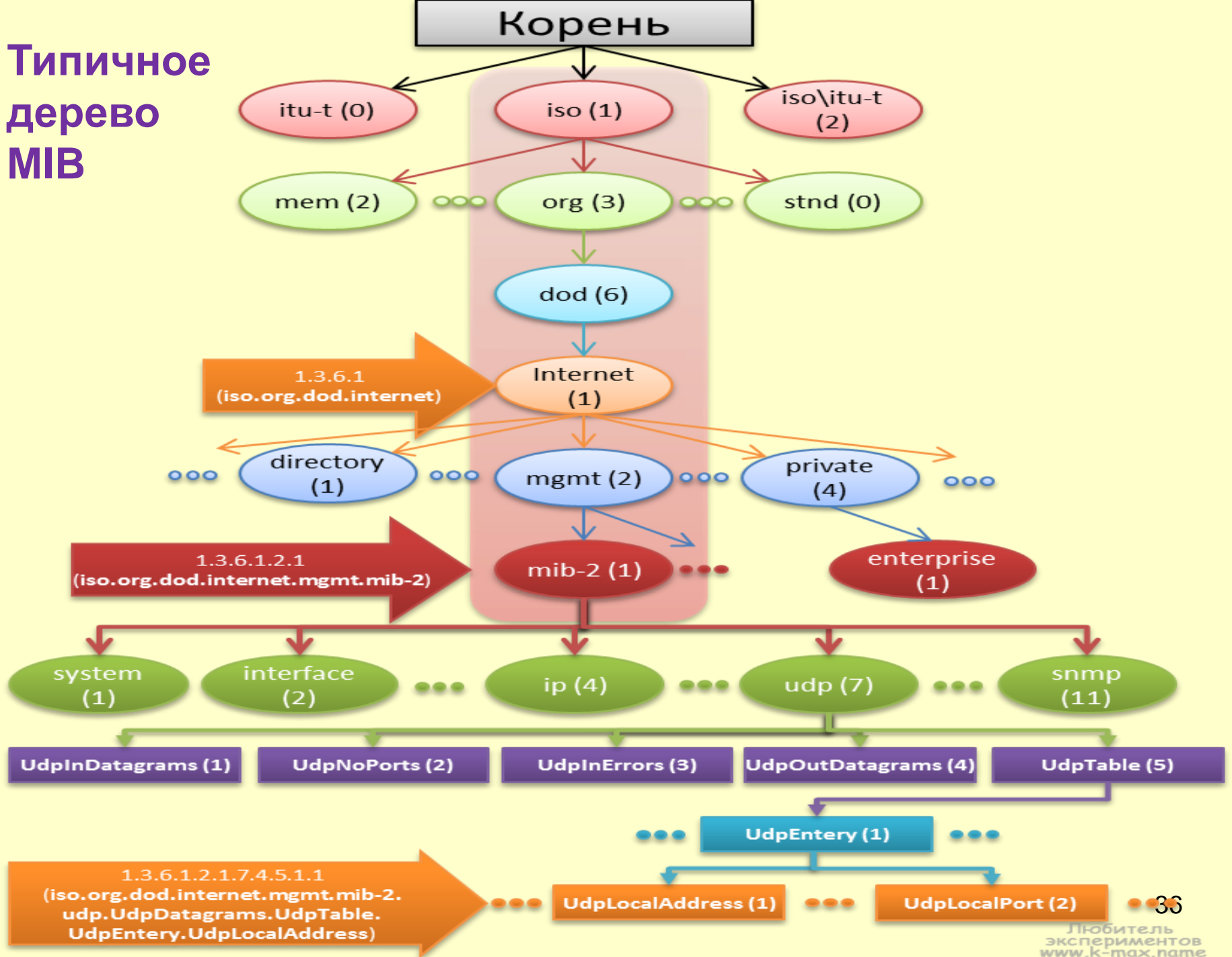
Для именования переменных базы МІВ и однозначного определения их форматов используется дополнительная спецификация, называемая **SMI** - Structure of Management Information .

Например, ветка 1.3.6.1.2.1.1 (мнемонический эквивалент: *iso.org.dod.internet.mgmt.mib-2.system*) описывает общую информацию о системе.

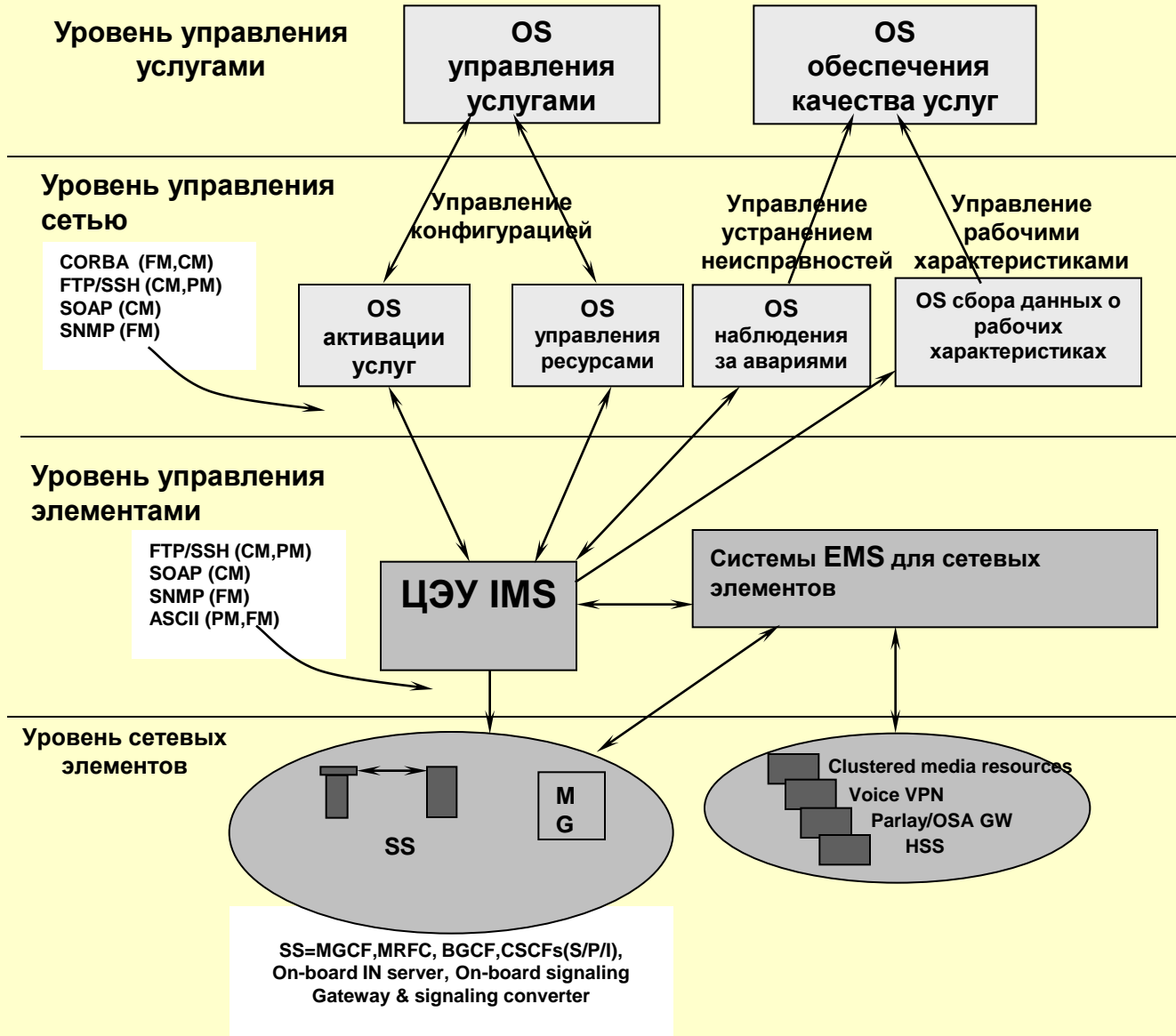
Опишем некоторые переменные из этой ветки: sysDescr (1.3.6.1.2.1.1.1) — краткое описание системы; sysUpTime (1.3.6.1.2.1.1.3) — время с момента последнего перезапуска; sysName (1.3.6.1.2.1.1.5) — имя системы.

Переменные и сведения об их типе определены также в МІВ. А сами типы переменных — в SMI.

Типичное дерево MIB



Структура и протоколы эксплуатационного управления IMS в иерархии уровней управления TMN



Согласно многоуровневой иерархической модели управления TMN для ЭУ IMS используются следующие протоколы

на уровне управления сетью :

- **CORBA SOAP FTP/SSH SNMP.**

на уровне управления элементами:

- **SOAP FTP/SSH ASCII SNMP.**

По классификации FCAPS данные протоколы выполняют следующие функции:

- **CORBA – используется для управления устранения неисправностей и управления конфигурацией (FM,CM);**
- **SOAP - для управления конфигурацией (CM);**
- **SNMP - для управления устранения неисправностями (FM);**
- **FTP/SSH - для управления конфигурацией и управления рабочими характеристиками (CM,PM);**
- **ASCII - для управления рабочими характеристиками и управления устранения неисправностей (PM,FM).**

Контрольные вопросы

1. Какая организация разработала протокол CORBA?
2. Дайте характеристику протокола SOAP.
3. Что такое TR-069 CWMP?
4. Область применения протокола CWMP и роль сервера автоконфигурирования ACS.
5. На какие уровне модели OSI работает протокол?
6. Какие основные три компоненты предусмотрены в архитектуре SNMP?
7. Возможности и интерфейсы протокола TR-069 CWMP.
8. Перечислите назначение основных команд (PDU - protocol data unit) SNMP.
9. Что такое MIB?
10. Какие протоколы используются в многоуровневой модели управления TMN при эксплуатационном управлении IMS?

Источники

1. Гребешков А.Ю. **Управление сетями электросвязи по стандарту TMN:** Учеб. Пособие.- М.: Радио и связь, 2004. 155 с. <http://aes.psuti.ru/wp-content/uploads/2010/03/GrebeshkovAU-TMN.pdf>
2. Интернет источники
3. Олифер В.Г., Олифер Н.А. Компьютерные сети. 1 – 5 – ое Издание.
4. Основы SNMP, 2-е издание Автор: Шмидт К., Мауро Д. 520 с. Январь 2012