Network Engineering Department

Internet Notes

6. Services

Collaborators:
Sr. Lluís Casals Ibáñez
Dr. David Rincón Rivera
Sra. Immaculada Ruiz Vela
Dr. Rafael Vidal Ferré

Dr. Daniel Guasch Murillo

January 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Politècnica Superior d'Enginyeria
de Vilanova i la Geltrú

6.1. Features application level

Basic concepts

What is meant by "**application level**"?

- Programs that allow access to **communications services** offered over TCP / IP networks
  - E-mail
  - File transfer
  - Remote terminal
  - Video transmission in real time ...
- **Very different** applications ⇒ **very specific** services ⇒ varied technical requirements.
  - Mail can suffer high delays, but streaming video in real time, no.
  - The video can support packet loss, no mail.

6.1. Features application level

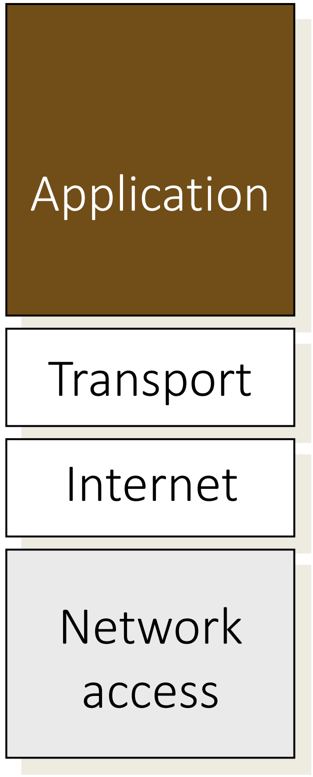
Basic concepts

Distinction between **service and application**

- **Service:** abstract concept
- **application:** piece of software that provides the service

Applications to model TCP / IP

- It is the layer closest to the user
- Uses the services provided by the transport layer
  - Through programming interface, sockets
- It is responsible for opening working sessions



6.1. Features application level

Classification of telematics services

Connection oriented

- A logic circuit communication is established, before starting
- Similar to a phone call concept: call before speaking
- Example: file transfer
- Before transferring, you open a connection to the server

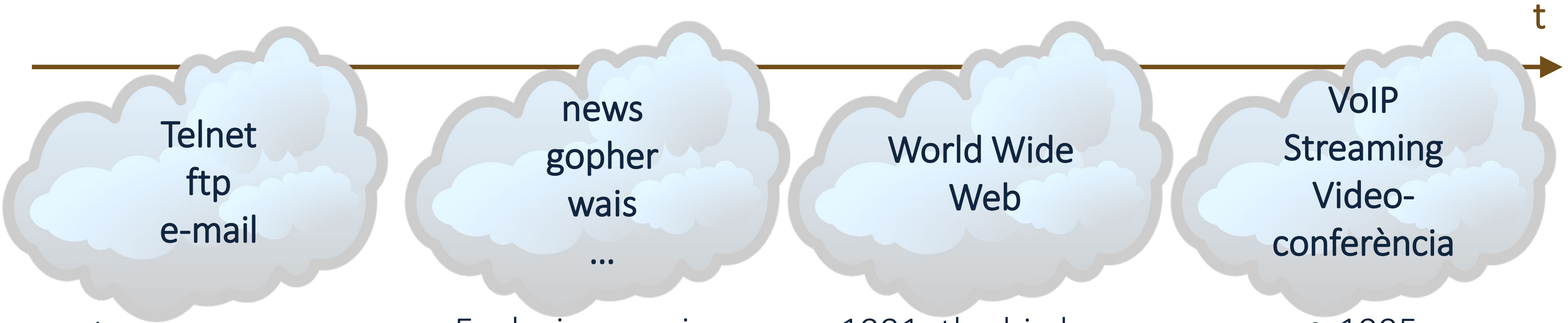
Connectionless

- It is not necessary to establish any prior circuit; only the information is sent
- Similar to a telegram concept is sent offline
- Example: email

6.1. Features application level

Evolution of internet services

- Emulació de terminal: TELNET (Terminal Networking)
- Transferència de fitxers: FTP (File Transfer Protocol)
- Correu electrònic: SMTP (Simple Mail Transfer Protocol)
- World Wide Web: HTTP (HyperText Transfer Protocol)
- Localització de dominis: DNS (Domain Name System)
- Administració de xarxa: SNMP (Simple Network Management Protocol)
- Notícies electròniques: NNTP (News Network Transfer Protocol)



Classic services:
1969-1973

Explosion services
1980-90

1991: the big hop
Not only text
Attractive
Easy to use

~ 1995
real time
traditional circuit
network services

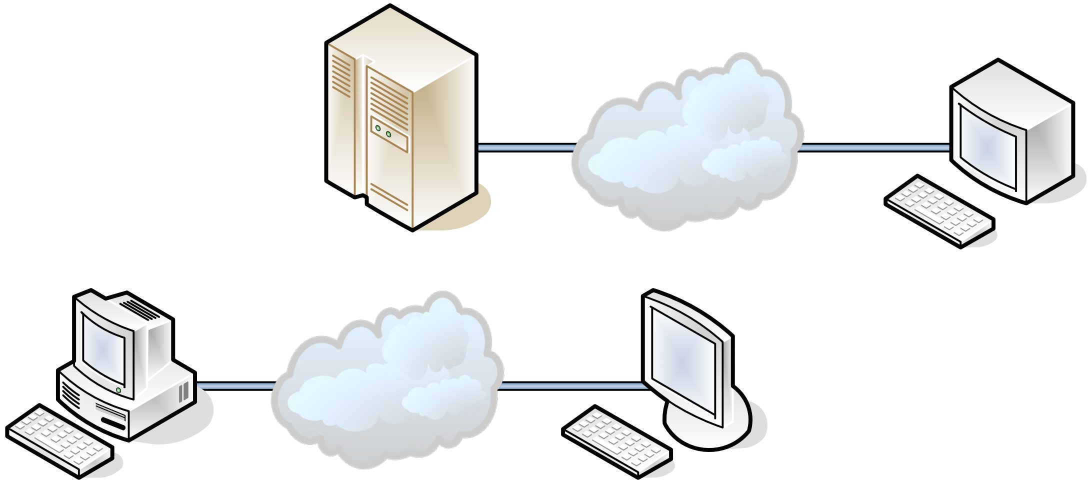
Application architecture

Application architecture

- Model following applications that offer a certain service
- References are built as software parts and how they interact between them

Two classical models

- Client-server
- *Peer-to-peer*



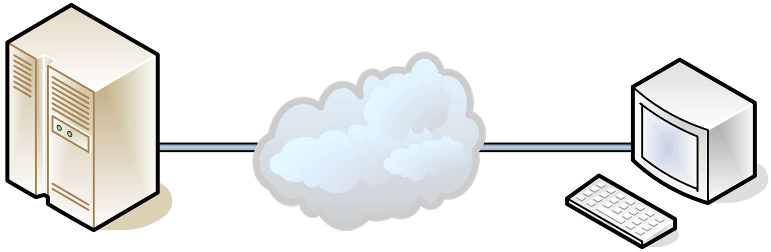
6.1. Features application level

Client - server architecture

The TCP / IP services often follow this model

Server

- Machine dedicated permanently to provide services to other Internet hosts
- Runs programs that listen for connections (daemons)
- You can serve multiple clients simultaneously

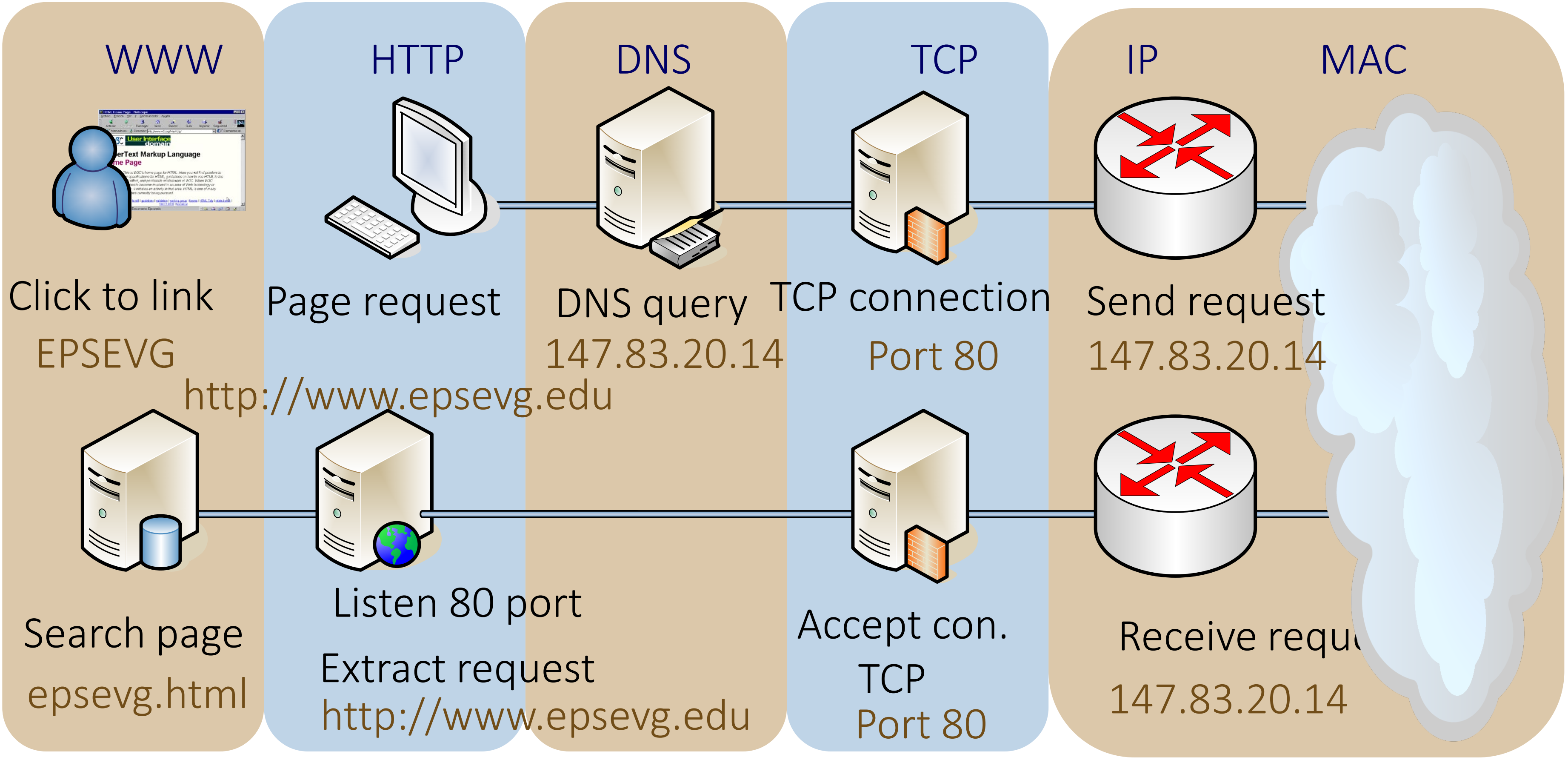


Client

- User machine, "simple"
- It connects to the server to request a service
- In principle, only manages a connection for each service

6.1. Features application level

Example client - server architecture

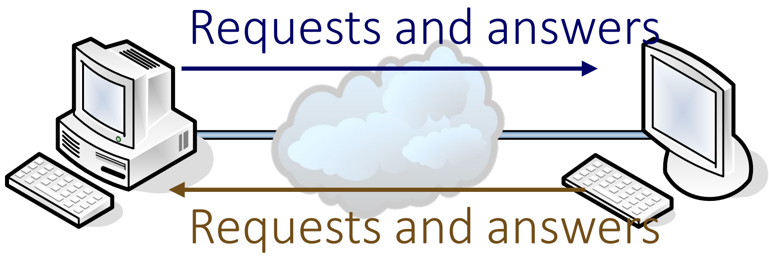


6.1. Features application level

Architecture peer - to - peer

All machines are equally important

- No distinction between client and server
- All can access the service (client) or offer (server)



- example: file sharing (Napster, Morpheus, etc)

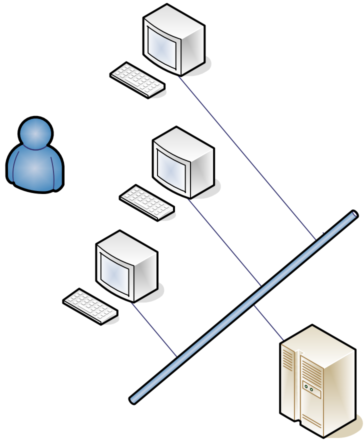
6.2. telnet

Terminal Networking: telnet

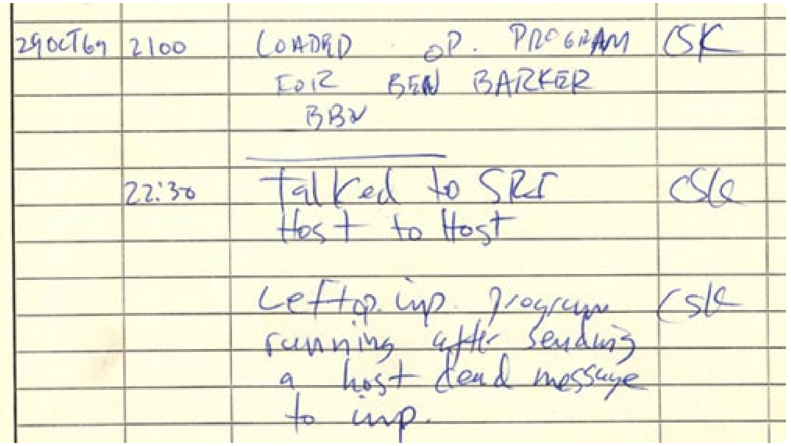
Telnet allows a local user to open a terminal session on a remote machine

- The user works as if next to the remote machine.

It is the oldest Internet application: 1969



- Internet initial objective: sharing and remote access to supercomputers.
  - Anecdote: 3 different terminals at the office of the head of ARPA



29 OCT 69	2100	LOADED OP. PROGRAM FOR BEN BARKER BBN	CSK
	22:30	talked to SRI Host to Host	CSK
		Left op. imp program running after sending a host dead message to imp.	CSK

6.2. telnet

Basic characteristics

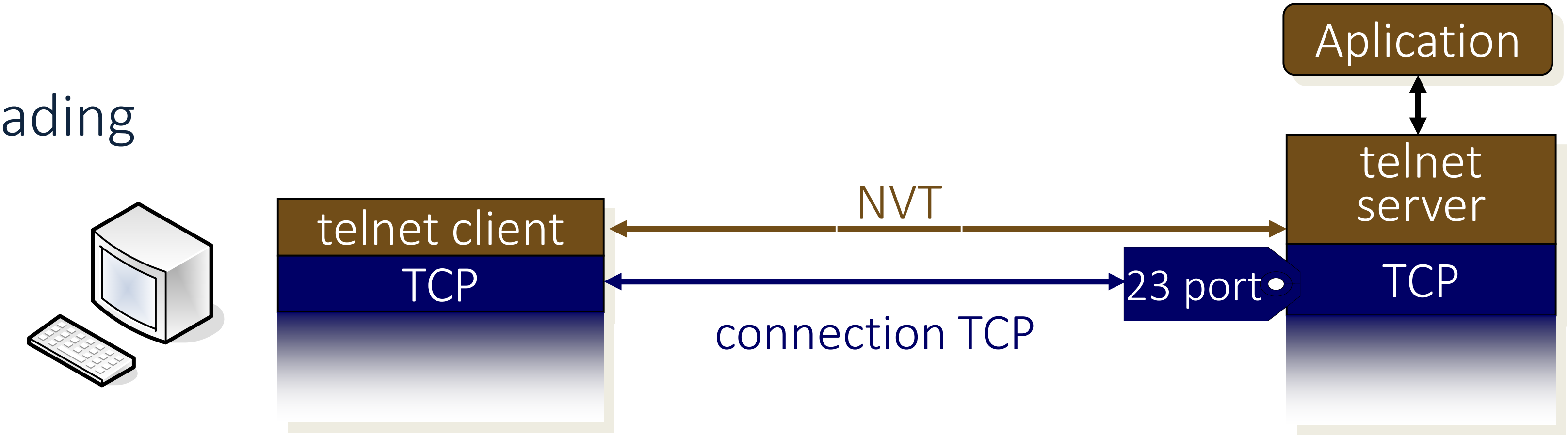
Defined in RFC 854

Telnet uses TCP transport service

- Two symmetrical communications port 23

Basic services:

- Defines a network virtual terminal (Network Virtual Terminal NVT) that provides a standard interface to remote systems
- Allows options trading



NVT, Network Virtual Terminal

Virtual Terminal Network

- Simulates a screen and a keyboard
- Designed as half duplex protocol and exchange line by line
  - only one host transmits simultaneously. After sending a line, the client expects to receive data from the server. The server sends the data and then a go ahead, indicating that the client can transmit
- Later option character-to-character

7-bit ASCII filled to 8 characters with an initial 0

- Each line ends with a combination CR and LF (ASCII)
- The characters that begin with 1 are orders

Normally only it used for a short period of time to negotiate options a terminal emulator.

- Terminal Type: ASCII, IBM 3270, VT100

NVT is also used FTP, SMTP, Finger ...

6.2. telnet

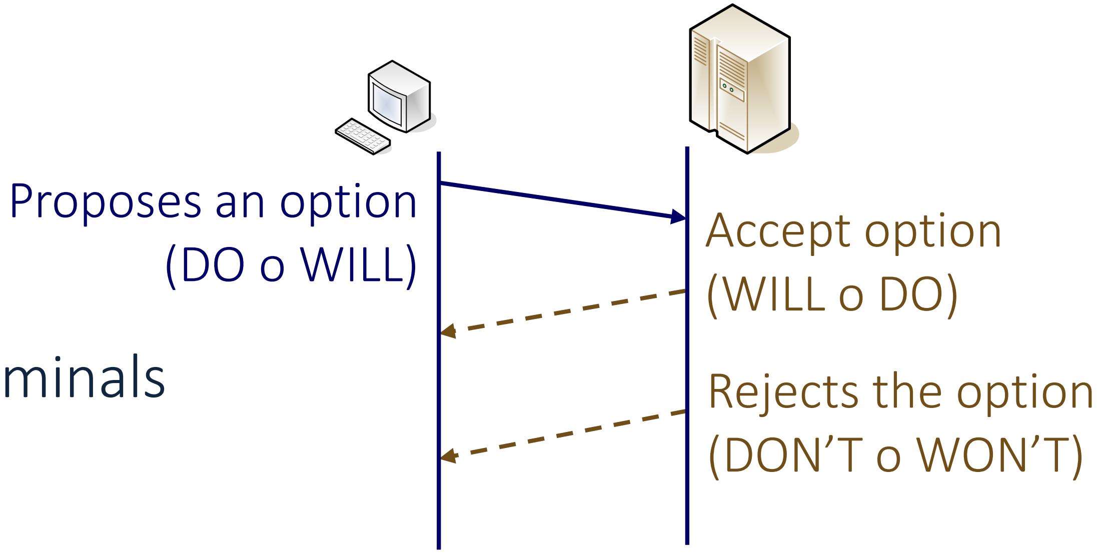
Establishing communication

Authentication :

- login + password

Initial options trading

- To go beyond the NVT, adapting to the terminals
- Options:
  - Eco local or remote
  - Check the status of the opposite end (status)
  - 7/8 bits per character
  - Exchange of information on the terminal
    - type, speed, CR or CR + LF ...

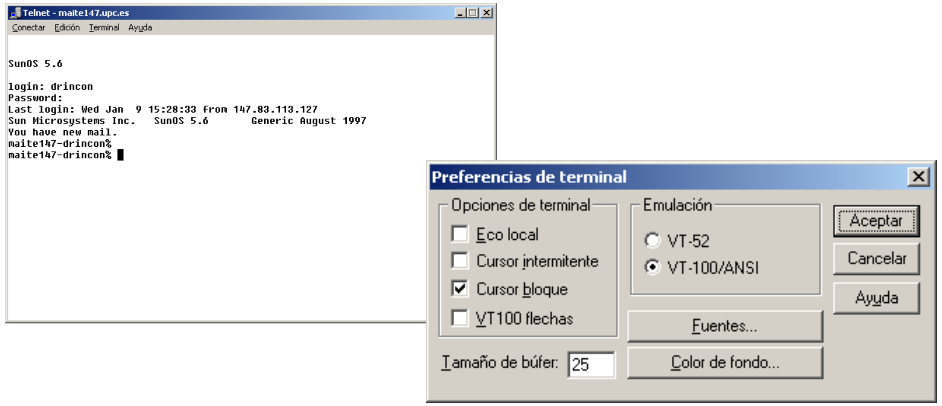


- Flow control
- Edit line or character mode
- Encryption
- Authentication

6.2. telnet

Telnet client

Integrated to (Unix, Windows ...) operating system



```
SunOS 5.6

login: drincon
Password:
Last login: Wed Jan  9 15:28:33 from 147.83.113.127
Sun Microsystems Inc.   SunOS 5.6       Generic August 1997
You have new mail.
maite147-drincon%
maite147-drincon% █
```

Preferencias de terminal

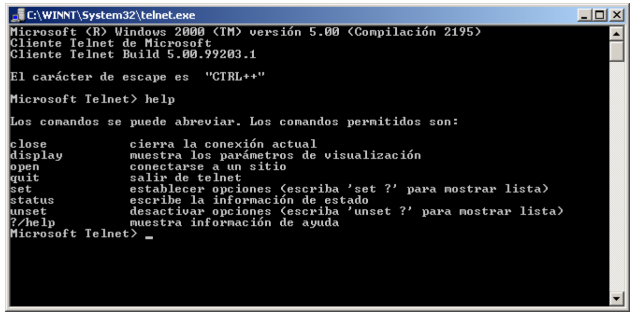
Opciones de terminal
- ☐ Eco local
- ☐ Cursor intermitente
- ☑ Cursor bloque
- ☐ VT100 flechas

Emulación
- ○ VT-52
- ● VT-100/ANSI

Tamaño de búfer: 25

Syntax: telnet[host][port]

- We can telnet to other ports that are not 23
- Possibility to request help



```
Microsoft (R) Windows 2000 (TM) versión 5.00 (Compilación 2195)
Cliente Telnet de Microsoft
Cliente Telnet Build 5.00.99203.1

El carácter de escape es  "CTRL++"

Microsoft Telnet> help

Los comandos se puede abreviar. Los comandos permitidos son:

close           cierra la conexión actual
display         muestra los parámetros de visualización
open            conectarse a un sitio
quit            salir de telnet
set             establecer opciones (escriba 'set ?' para mostrar lista)
status          escribe la información de estado
unset           desactivar opciones (escriba 'unset ?' para mostrar lista)
?/help          muestra información de ayuda
Microsoft Telnet> _
```

6.2. telnet

Telnet client

Basic orders

`close` closes the current connection.

`logout` It makes the user disconnects and closes the connection.

`open` opens a connection to a host.

`mode` indicates whether the transmission is made character by character (character) or a character line (line) generated is sent when an EOL (end of line).

`quit` exits the program.

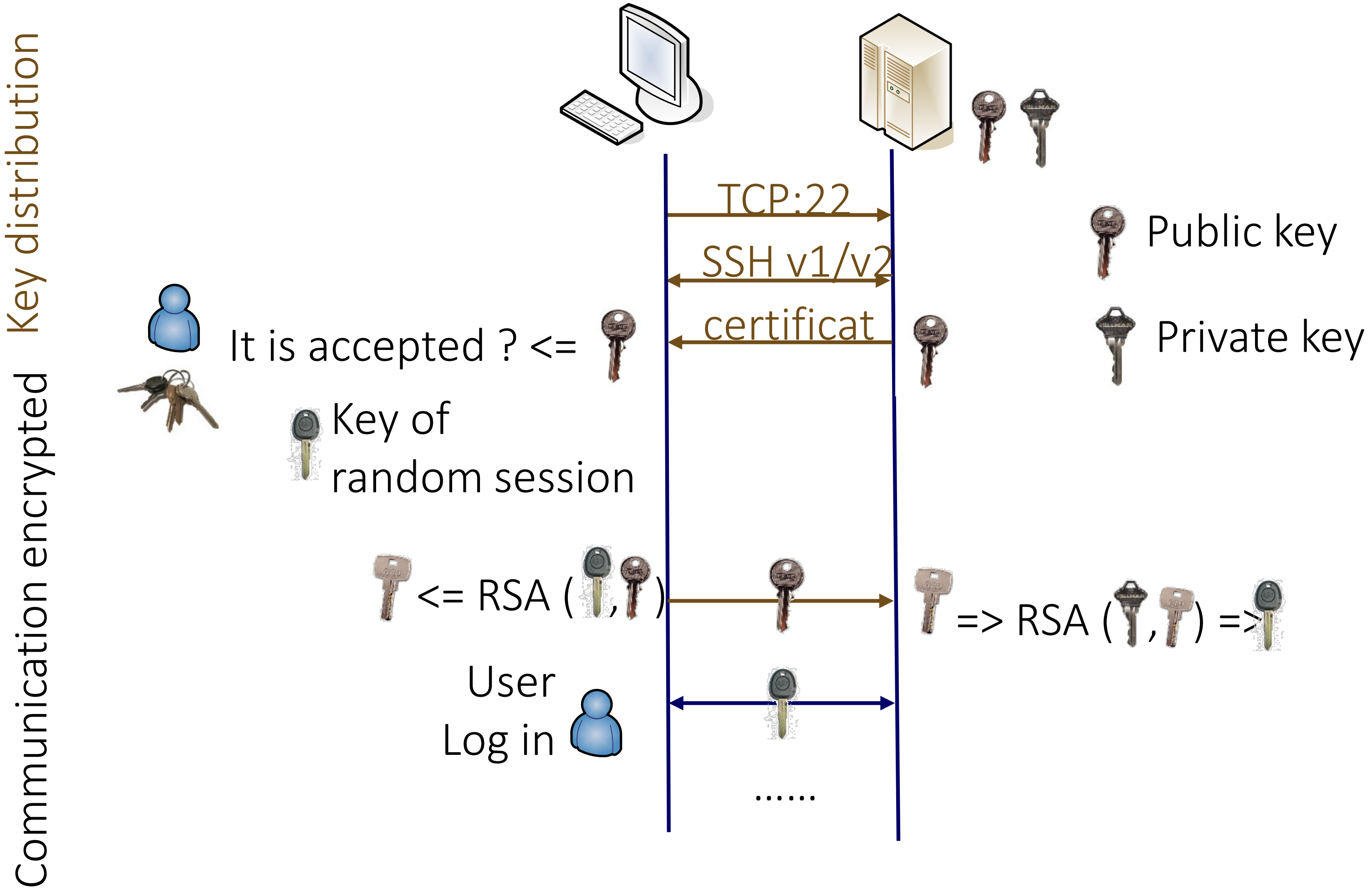
`set` active operating parameters (echo, escape, erase, kill, of, quit)

`status` information displays connection status, mode, echo and the escape character.

control character (CTRL], CTRL): Exits the mode of execution of orders. You can return to the terminal return.

6.2. telnet

SSH (Secure Shell) secure improvement and telnet



File transfer: FTP

One of the basic Internet applications

FTP (*File Transfer Protocol*)

- Sharing files between remote systems
- Ability to send, receive, delete, and manage files and directories

Various protocols

- FTP: *File Transfer Protocol*. RFC 959
- SFTP: *Simple File Transfer Protocol*.
- TFTP: *Trivial File Transfer Protocol*. RFC 1350

6.3. FTP

FTP Features

Assumes that you have a reliable end to end service (TCP)

- Two TCP connections
  - Control port 21: NVT session
  - Data port 20
- Control connection:
  - Command dialog (client) and response codes (server)
  - Begins at the time the client connects to the server

client (y) ↔ (21) Server

- Data Connection opens only to copy files or make lists
  - The customer reservations x port and tells the server (command port) then connects

Client (x) ↔ *(20) Servidor*

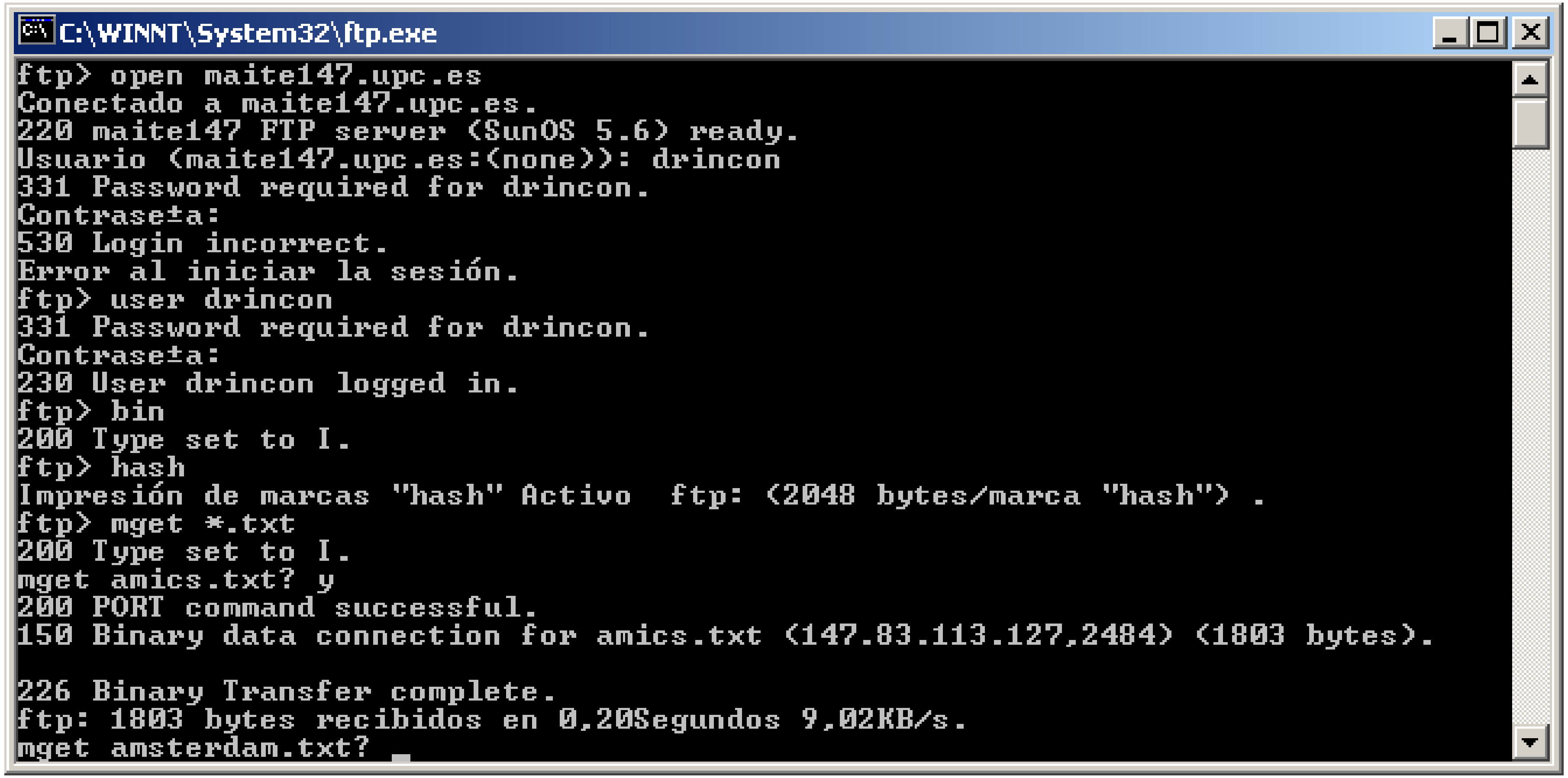
FTP Features

- Interactive access the person or machine
  - Response codes to enable 3-digit remote control
  - The first indicates the type of operation
    - 2xx : Success
    - 1xx : action has started
    - 3xx : a compromise has been achieved successfully
    - 4xx : transient error
    - 5xx : permanent error

FTP Features

- Information Format
  - ASCII to ASCII text files
    - Special characters are interpreted as changing line
  - Binary: for other files
    - the entire file is treated as a bit stream
  - It is very important to use the right way !!
    - Example: A Word file is NOT a text file
- Naming
  - local file → remote file
- Authentication:
  - login + password

Example FTP session with Windows 2000

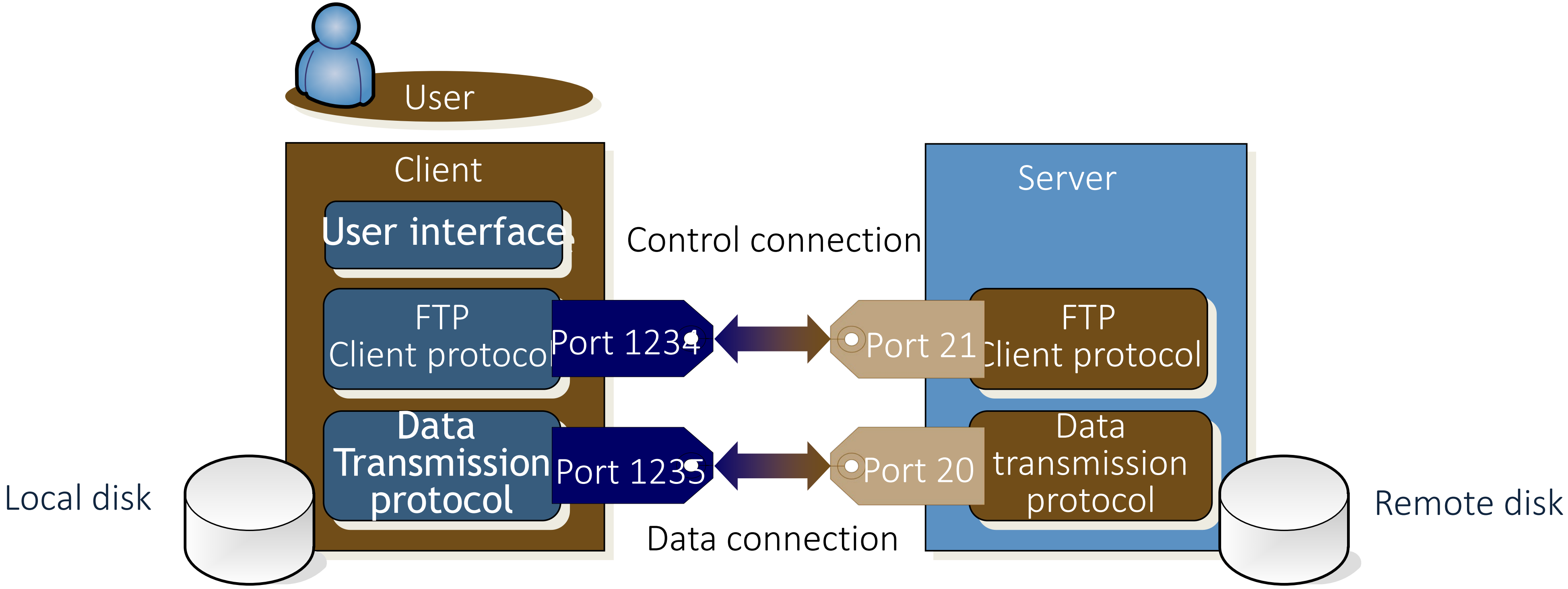


```
ftp> open maite147.upc.es
Conectado a maite147.upc.es.
220 maite147 FTP server (SunOS 5.6) ready.
Usuario (maite147.upc.es:(none)): drincon
331 Password required for drincon.
Contrase±a:
530 Login incorrect.
Error al iniciar la sesión.
ftp> user drincon
331 Password required for drincon.
Contrase±a:
230 User drincon logged in.
ftp> bin
200 Type set to I.
ftp> hash
Impresión de marcas "hash" Activo  ftp: (2048 bytes/marca "hash") .
ftp> mget *.txt
200 Type set to I.
mget amics.txt? y
200 PORT command successful.
150 Binary data connection for amics.txt (147.83.113.127,2484) (1803 bytes).

226 Binary Transfer complete.
ftp: 1803 bytes recibidos en 0,20Segundos 9,02KB/s.
mget amsterdam.txt? _
```

FTP model

- FTP daemon inetd
- For each data transfer a new connection is created
- The protocol data connection is a particularization of telnet



User commands in FTP client

Control functions:

- Identify the type of transfer: **ascii, binary**
- Confirm each file transferred: **prompt**
- Exit: **quit, bye**
- Help: **help**
- Open/ close connection: **open, close**

Data transfer functions:

- Copy files between hosts: **get, put, mget, mput**
- Add a local file to a remote file: **append**

Control functions files:

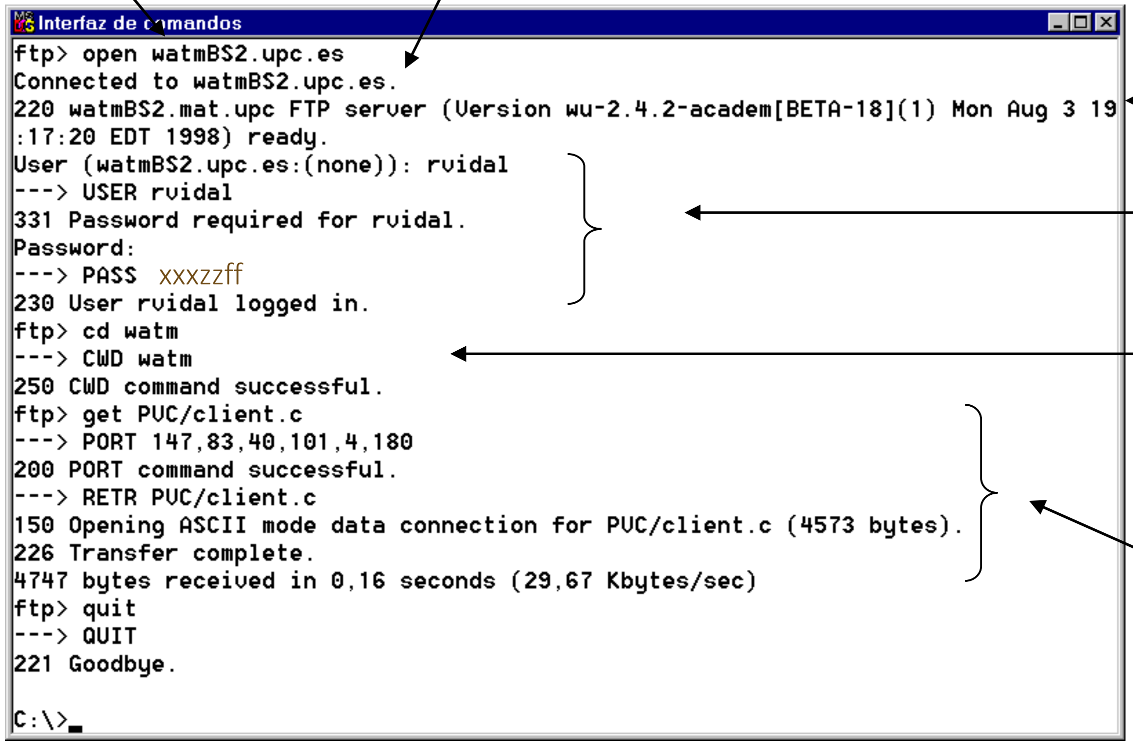
- List a directory : **ls, dir**
- Print the current directory : **pwd**
- Change directory : **cd, lcd**
- Delete / rename a file : **delete, rename**
- Show the connection status : **status**

6.3. FTP

Example FTP control dialog

We open a session

client indicates successful connection



```
ftp> open watmBS2.upc.es
Connected to watmBS2.upc.es.
220 watmBS2.mat.upc FTP server (Version wu-2.4.2-academ[BETA-18](1) Mon Aug 3 19
:17:20 EDT 1998) ready.
User (watmBS2.upc.es:(none)): rvidal
---> USER rvidal
331 Password required for rvidal.
Password:
---> PASS  xxxzzff
230 User rvidal logged in.
ftp> cd watm
---> CWD watm
250 CWD command successful.
ftp> get PVC/client.c
---> PORT 147,83,40,101,4,180
200 PORT command successful.
---> RETR PVC/client.c
150 Opening ASCII mode data connection for PVC/client.c (4573 bytes).
226 Transfer complete.
4747 bytes received in 0,16 seconds (29,67 Kbytes/sec)
ftp> quit
---> QUIT
221 Goodbye.

C:\>_
```

message server

Login and Password

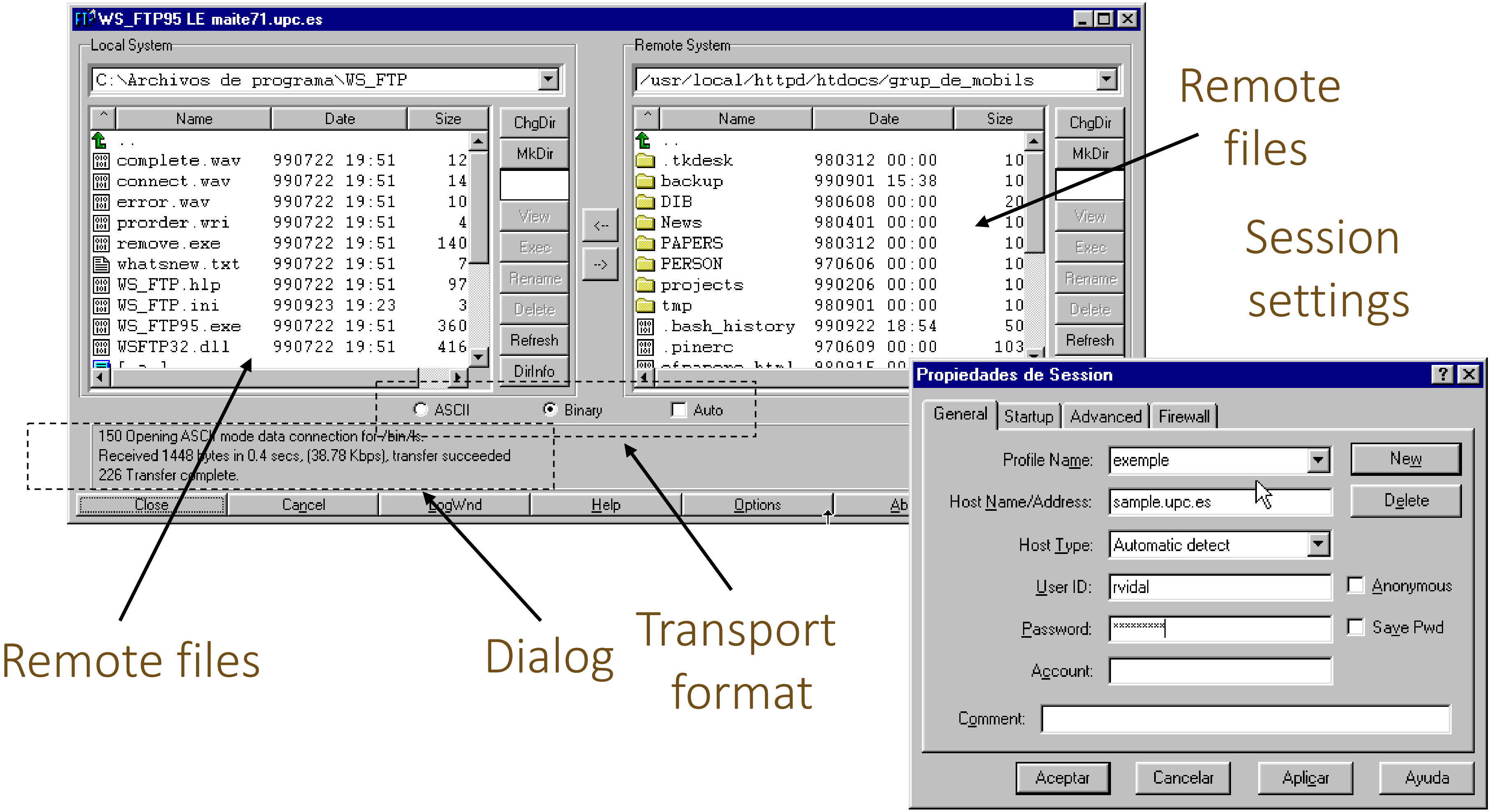
change directory

new connection to copy file

Copiem un fitxer

6.3. FTP

Example FTP interface



WS_FTP95 LE maite71.upc.es

Local System: C:\Archivos de programa\WS_FTP

```
Name            Date          Size
..
complete.wav    990722 19:51    12
connect.wav     990722 19:51    14
error.wav       990722 19:51    10
prorder.wri     990722 19:51     4
remove.exe      990722 19:51   140
whatsnew.txt    990722 19:51     7
WS_FTP.hlp      990722 19:51    97
WS_FTP.ini      990923 19:23     3
WS_FTP95.exe    990722 19:51   360
WSFTP32.dll     990722 19:51   416
```

Remote System: /usr/local/httpd/htdocs/grup_de_mobils

```
Name            Date          Size
..
.tkdesk         980312 00:00    10
backup          990901 15:38    10
DIB             980608 00:00    20
News            980401 00:00    10
PAPERS          980312 00:00    10
PERSON          970606 00:00    10
projects        990206 00:00    10
tmp             980901 00:00    10
.bash_history   990922 18:54    50
.pinerc         970609 00:00   103
```

○ ASCII ● Binary □ Auto

```
150 Opening ASCII mode data connection for /bin/ls.
Received 1448 bytes in 0.4 secs, (38.78 Kbps), transfer succeeded
226 Transfer complete.
```

Propiedades de Session

General | Startup | Advanced | Firewall

Profile Name: exemple
Host Name/Address: sample.upc.es
Host Type: Automatic detect
User ID: rvidal □ Anonymous
Password: ********* □ Save Pwd
Account:
Comment:

Remote files

Session settings

Remote files

Dialog

Transport format

6.3. FTP

Anonymous FTP

- Some servers allow anonymous access
- Identification by login anonymous, password = e-mail

```
C:\TMP>ftp ftp.upc.es
Connected to diable.upc.es.
220-    O O O
220-    O O O           Servei d'FTP de la UPC
220-    O O O
220-    U P C
220-
220 diable.upc.es FTP server () ready.
User (diable.upc.es:(none)): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230---------------------------------------------------
230-
230-   Benvingut al servei d'FTP anonim de la UPC.
230- Bienvenido al servicio de FTP anonimo de la UPC.
230-  Welcome to the anonymous FTP service on the UPC.
230-
230-     T'has connectat des de 'maite30.upc.es'.
230-
230- Si tens problemes amb el sistema pots consultar a
230-                 ftpmanager@upc.es
230-
230- Usuaris connectats: 2 de 15 permesos
230-
230 User ftp logged in.  Access restrictions apply.
ftp>
```

TFTP: Trivial File Transfer Protocol

FTP is difficult to implement and offers more of what you want to use on occasion.

On certain occasions it is better to use TFTP

- Small and easy to use protocol
- TFTP only allows file transfer
- TFTP does not require authentication
  - Possible security hole !!
- No wait server type (only 1 transfer at once)
- Most errors cause one end of the connection :
  - can not satisfy the request:
    - You can not find the file, access denied, there is no user
  - Incorrectly built package
  - Loss of the device during the conversation

TFTP: Trivial File Transfer Protocol

Uses UDP as a transport mechanism. **Unreliable**

- Maximum size: 512 bytes
- each packet is confirmed

Used for diskless stations (RARP, BOOTP)

Typical for downloading images on computers (routers, etc.)

Different transfer modes

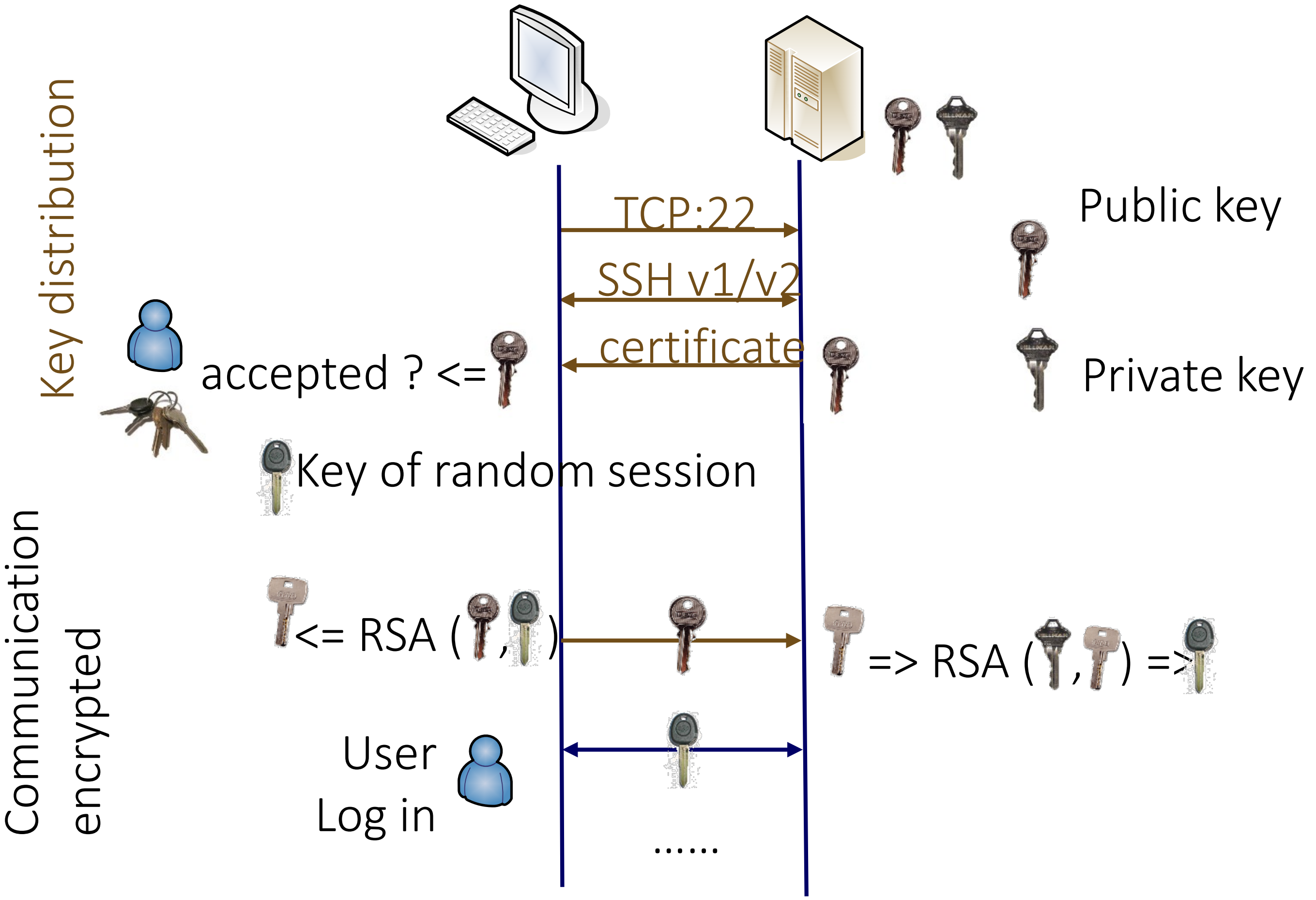
- netascii: text
- octet: binary format
- email: Data sent to a user rather than a file

User interface: *tftp host*

- put, get, sdadus, ascii, binario

6.3. FTP

SFTP (Secure FTP) improves and secures the ftp



Email Basics

Email: probably the "star" network service

Evolution of the first systems of communication between users of the same computer

- Order mail from Unix
- It allows sending text messages left in the mailbox (mailbox) user receiver

1971: Ray Tomlinson wrote the first mail transfer protocol between Unix hosts

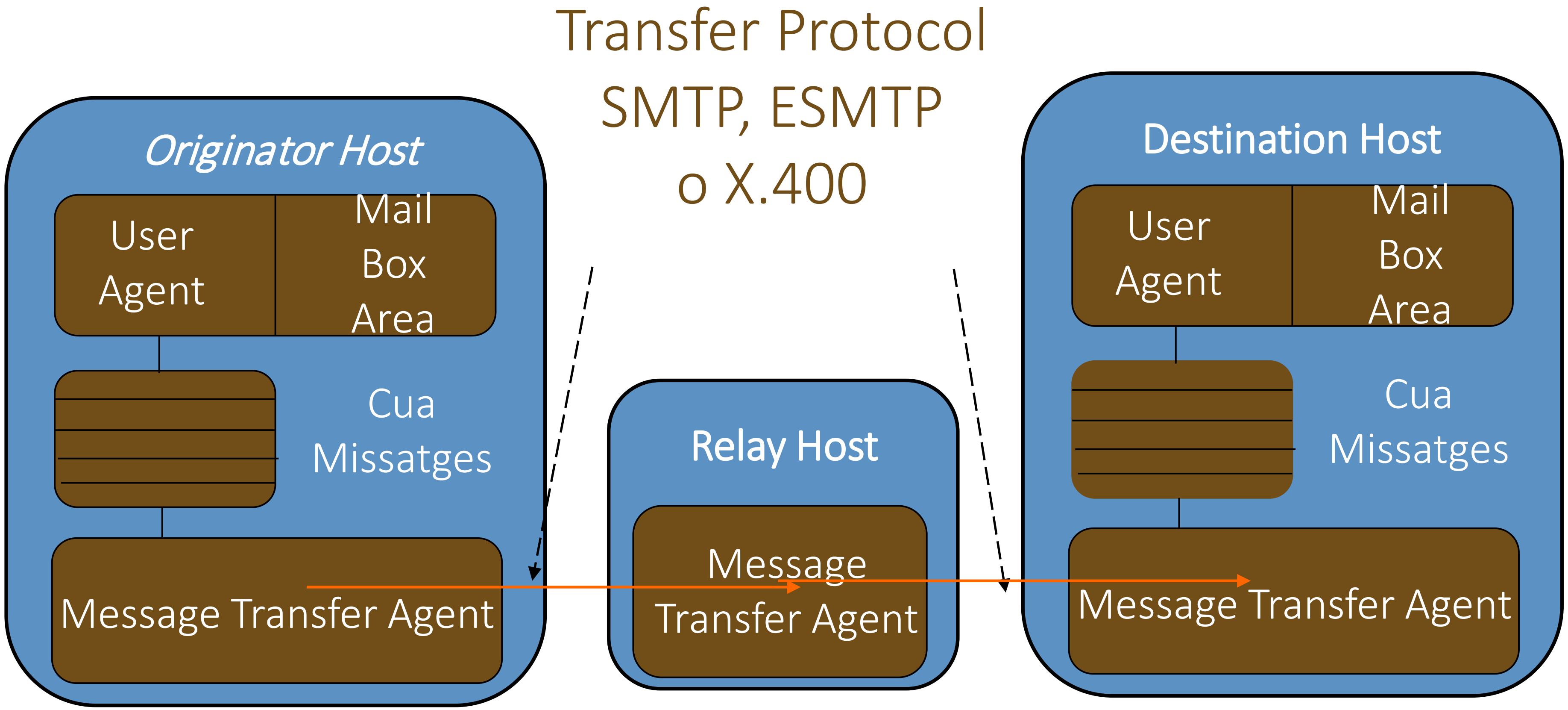
- Address = username + hostname
- Separated by @

6.4. Email

Elements of a system Email

- User Agent (UA) or client e-mail: which allows us to check, reply and edit mail
- Message: two parts, envelope and content, with a particular format
- Mailbox: where received messages are saved or to send
- Message Transfer Agent (MTA): elements responsible for transporting mail network
- Store-and-forward : message transfer technique. The message goes through MTAs that are forwarded until it reaches the destination mailbox

Elements of a system Email



Email Protocols

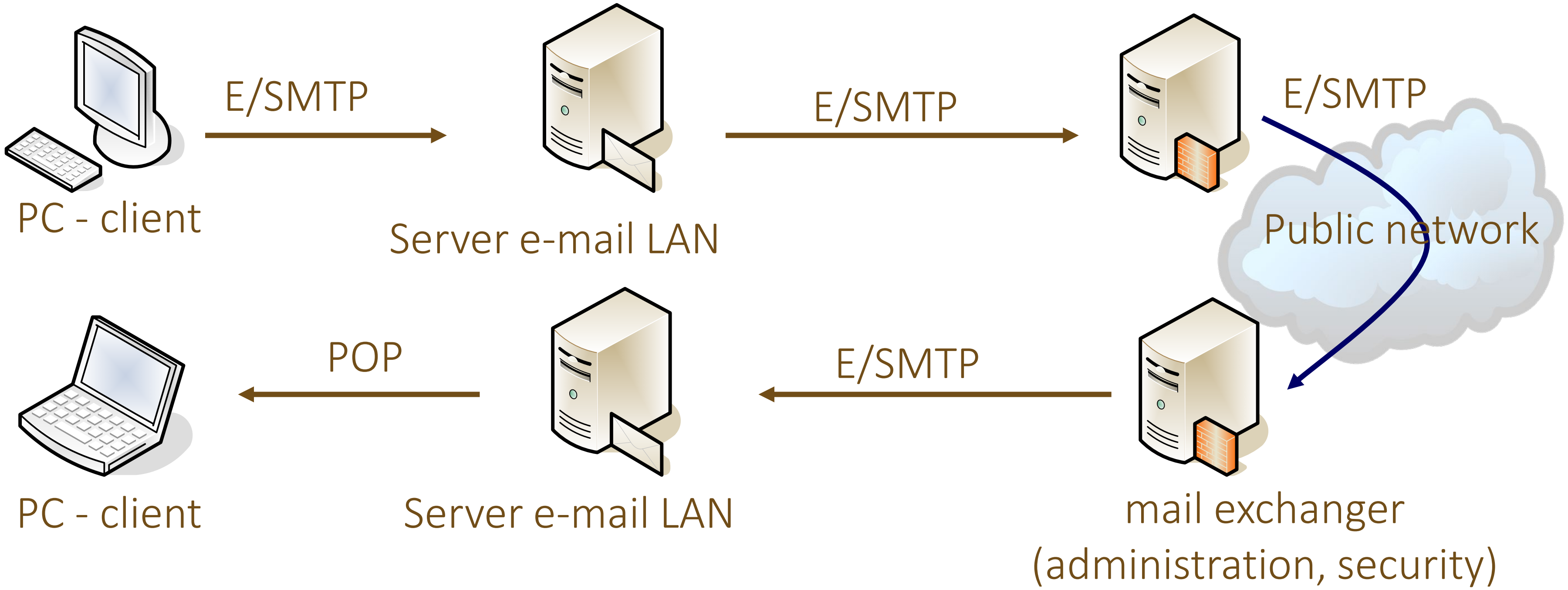
- Standard for the format of ARPA Internet Messages (RFC 822)
  - Describes the format of messages
- SMTP (Simple Mail Transfer Protocol):
  - Classic server Mail Transfer Protocol (RFC 821)
  - Simple file transfer (7 bits / character) using NVT in a telnet session
- ESMTP (Extended SMTP):
  - Transferring complex data (8 bits per character)
- MIME (Multipurpose Internet Mail Extensions):
  - Multimedia extensions for attachments
- POP3 (Post Office Protocolo):
  - Mail access from a personal computer (not a host)
- IMAP4 (Internet Access Protocolo):
  - Evolution of POP

6.4. Email

Model Store & Forward

Advantages model Store and Forward

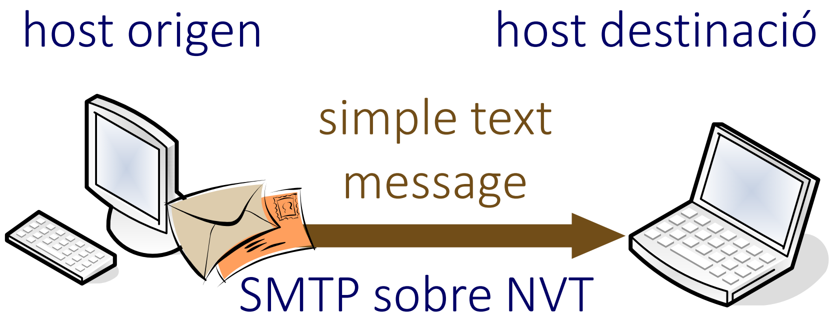
- Using Email safely
- Saving block sending email at specified times
- Translation formats

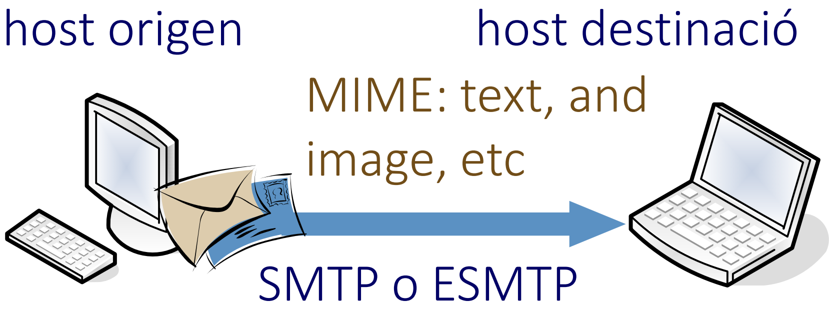


6.4. Correu electrònic

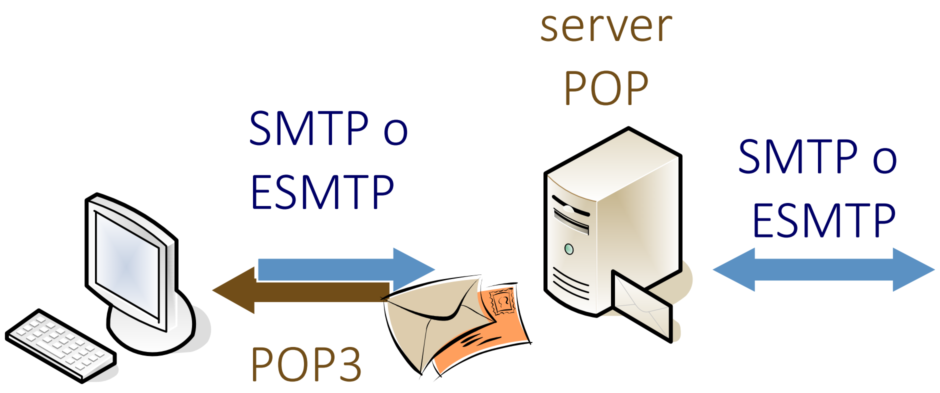
Transfers post

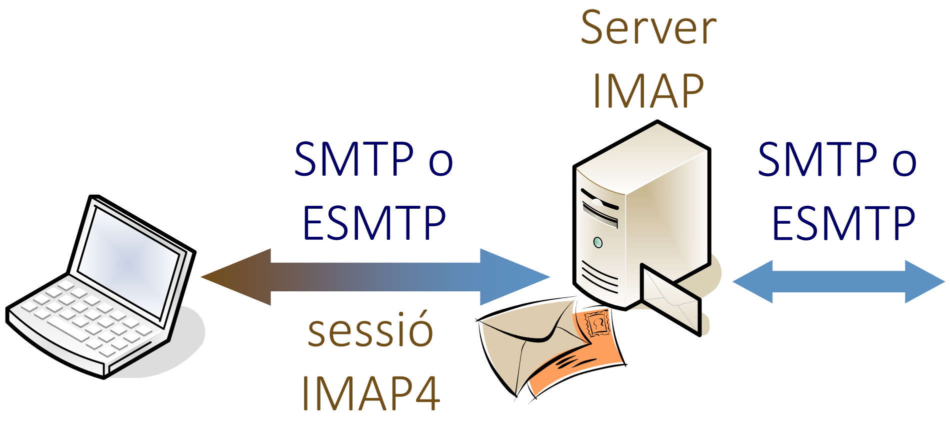
Transfer host-host





Host-terminal transfer





Main relay - mail servers in the UPC

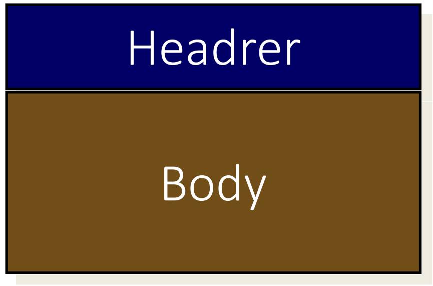
```
C:\>nslookup
Server predetermined:  dns1.red.retevision.es
Address:  62.81.16.129

> set type=mx
> upc.es
Server :  dns1.red.retevision.es
Address:  62.81.16.129

Answer no authoritative:
upc.es  MX preference = 10, mail exchanger = dukas.upc.es
upc.es  MX preference = 20, mail exchanger = moneo.upc.es
upc.es  MX preference = 30, mail exchanger = mail.rediris.es

upc.es  nameserver = euler.upc.es
upc.es  nameserver = backus.upc.es
dukas.upc.es    internet address = 147.83.2.62
moneo.upc.es    internet address = 147.83.2.91
mail.rediris.es internet address = 130.206.1.11
euler.upc.es    internet address = 147.83.2.10
backus.upc.es   internet address = 147.83.2.3
> exit
```

Parts of an email



Header

Received
Date *
From *
To *
cc
bcc (blind cc)
Message-Id
Reply-To

keywords
Subject
Comment
Encrypted
References (previous message identifiers)
In-Reply-To
Sender (if not the originator of the message)

White line

Body

Example of a mail header

Received: from diable.upc.es [147.83.98.7]by mat.upc.es (8.7.6/8.7.3) with ESMTP id VAA13564 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:55:56 GMT

Received: from mail.tiip.edu[192.208.46.30] by diable.upc.es (8.8.6/8.8.6) with ESMTP id VAA01230 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:56:36 +0100 (MET)

Received: by mail.tiip.edu with SMTP (Microsoft Exchange Server Internet Mail Connector Version 4.0.996.35) id <01BCE480.75FDDED0@mail.tiip.edu>; Wed, 29 Oct 1997 15:36:44 -0500

Date: Wed, 29 Oct 1997 15:36:39 +0200 (MET DST)

From: David Rincon <drincon@tiip.edu>

To: Rafael Vidal <rvidal@mat.upc.es>

Subject: Re: Microsoft i IPv6

In-Reply-To: <393BD954.82FF42E7@mat.upc.es>

Message-ID: <Pine.GSO.4.10.10006051848230.20185-100000@tn-nit.tiip.edu>

MIME-Version: 1.0

Content-Transfer-Encoding: QUOTED-PRINTABLE

Content-Type: TEXT/PLAIN; charset=ISO-8859-1

Status: U

X-UIDL: 726e36d36e1699eef6b8bb936c02c013

6.4. Email

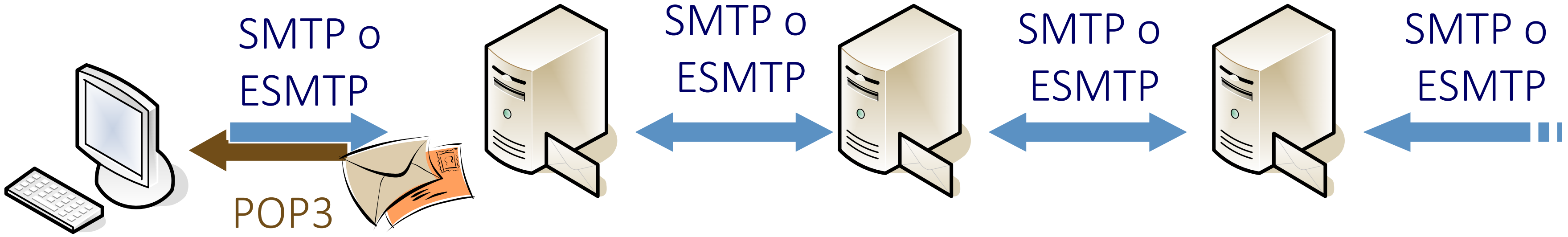
SMTP: Simple Mail Transfer Protocol

- RFC 821
- Defines a mechanism mail transfer between 2 machines
  - origin machine
    - it can be a client or another host (mail relay)
  - destination machine
    - It must be a server; It can not be a customer
    - for each transfer, a session is established
    - A session may comprise various e
    - TCP protocol, port 25

Time recording, step identifier and SMTP

Emails have a record of intermediate hosts and timestamps (time step).

- Each time a message passes through an MTA a temporary mark its passage is added, **timestamp**
- The email contains all the way round
- The message identifier puts the first MTA that runs through the message and the message is **unique**



Time recording, step identifier and SMTP

Received: from diable.upc.es [147.83.98.7]by mat.upc.es (8.7.6/8.7.3) with ESMTP id VAA13564 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:55:56 GMT

Received: from mail.tiip.edu[192.208.46.30] by diable.upc.es (8.8.6/8.8.6) with ESMTP id VAA01230 for <rvidal@mat.upc.es>; Wed, 29 Oct 1997 21:56:36 +0100 (MET)

Received: by mail.tiip.edu with SMTP (Microsoft Exchange Server Internet Mail Connector Version 4.0.996.35) id <01BCE480.75FDDED0@mail.tiip.edu>; Wed, 29 Oct 1997 15:36:44 -0500

Date: Wed, 29 Oct 1997 15:36:39 +0200 (MET DST)

From: David Rincon <drincon@tiip.edu>

To: Rafael Vidal <rvidal@mat.upc.es>

Subject: Re: Microsoft i IPv6

In-Reply-To: <393BD954.82FF42E7@mat.upc.es>

Message-ID: <Pine.GSO.4.10.10006051848230.20185-100000@tn-nit.tiip.edu>

MIME-Version: 1.0

Content-Transfer-Encoding: QUOTED-PRINTABLE

Content-Type: TEXT/PLAIN; charset=ISO-8859-1

Status: U

X-UIDL: 726e36d36e1699eef6b8bb936c02c013

Orders and sequence SMTP Protocol

Orders protocol

- HELO, MAIL, RCPT, DATA, QUIT, TURN
- Protocol sequence
  - Source sends an initial packet, the receiver sends host ID
  - Identifying host sends origin (HELO)
  - ESMTP clients and servers are recognized by EHLO HELO instead of
  - Source sends user ID that generates the message (MAIL FROM :)
  - Origin identifies the recipient (RCPT TO :)
  - Source transmits the header data and text (MAIL, DATA)
  - Transmits origin ". <CR> <LF>" (final email)
  - The process is repeated for other posts or ends (QUIT)
  - Possibility of exchanging roles (TURN)

6.4. Email

Example of an SMTP session

```
220 tigger.jvnc.net 6.65c/IDA-1.4.4
Sendmail is ready at Sun, 27 Aug 1995
08:02:55 -0400
HELO MATH-GW.CS.YALE.EDU
250 Hello MATH-GW.CS.YALE.EDU, pleased to meet you
MAIL FROM: <sfeit@pascal.math.yale.edu>
250 <sfeit@pascal.math.yale.edu>
RCPT TO: <feit@tigger.jvnc.net>
250 <feit@tigger.jvnc.net>..Receiver ok
DATA
354 Enter mail, end with "." on a line by itself
Hello!
This is the message content.
The end is marked with a point
.
250 OK
quit
221 tigger.jvnc.net closing connection
Connection closed by foreign host
```

(Server identification)

(Source ID)

(Home email origin)

(Data message)

(final message .<CR><LF>)

(Close connection)

SMTP Protocol Extensions

SMTP and RFC822 were designed for text messages

- As we add other content (videos, pictures)?
- As transport ASCII characters above 127?
- Solution: Modify transport protocol and message format
  - ESMTP, Extended SMTP, RFC 1425 / 1869
    - Clients and servers are recognized by SMTP EHLO HELO instead of
  - MIME, *Multipurpose Internet Mail Extension,* RFC 1521
    - Definition of extensions that support multimedia formats
    - They can be transferred efficiently with ESMTP (and not so much with SMTP)
    - each part of the message is delimited with a mark
    - Facilitates call playback applications

6.4. Email

MIME: Multipurpose Internet Mail Extension

- Supported data type:
  - text
    - plain, richtext (basic format), enriched
  - multipart
    - Mixed (various formats processed sequentially), parallel, digest...
  - application
    - octet-stream (arbitrary), postscript, ...
  - Image
    - BMP, JPEG, PIF, GIF...
  - video
    - mpeg, quicktime, avi, ...
  - audio
    - basic, mpeg,

Example of MIME

```
...
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED;
   BOUNDARY="-559023410-2110444415-1011024789=:19819"
X-Virus-Scaned: by AMaViS perl-11
X-Mozilla-Sdadus2: 00000000

This is the body of the email. There are accents.
   [ Part 2.2, ""  texto/PLAIN (Name: "amics.txt")  41 lines. ]
   [ Not Shown. Use the "V" command to view or save this part. ]

   [ Part 2.3, ""  Application/POSTSCRIPT  1.9MB. ]
   [ Not Shown. Use the "V" command to view or save this part. ]

   [ Part 2.4, ""  Application/OCTET-STREAM (Name: "a.mpg")  939KB. ]
   [ Canot display this part. Pres "V" then "S" to save in a file. ]
```

Post Office Protocol – POP3

Protocol extended to more downloading mail

- Defined in RFC 1939
  - It will being replaced by the IMAP4, RFC 2060
- Uses TCP, port 110
- Available authentication mechanisms
  - USER orders and PAs, the most common: ID and password are passed in clear
  - APOP command, option: more sophisticated, the key is not exposed
  - order AUTH extension (RFC 1734): Use any of the cryptographic authentication options for IMAP4

Server supports multiple connections, each with permissions R / W in your mailbox

Post Office Protocol – POP3

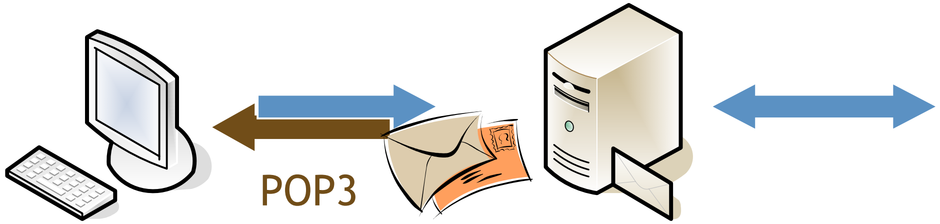
Basic commands:

- Authentication

  USER name, PASS string, APOP name digest, AUTH

- Dialog

  STAT, LIST [msg], RETR msg, DELE msg, NOOP, RSET, QUIT, TOP msg n, UIDL [msg]



Sample dialogue POP3

```
C:                                   (Start TCP connection port 110)
S:                                   (Accepts TCP connection)
S: +OK POP3 server ready
C: USER user-name
S: +OK
C: PAs password
S:+OK user authenticated
C: LIST
S:+OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S:                                   (1 message content, headers, null line and body)
S: .
C: DELE 1
S: +OK message 1 deleted
C:
C: RETR 2
S: +OK 200 octets
S:                                   (2 message content, headers, null line and body)
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK POP3 server signing off
S:                                   (The end of server TCP connection is closed)
C:                                   (The end of client TCP connection is closed)
```

6.4. Email

A mailbox access from multiple clients with POP3

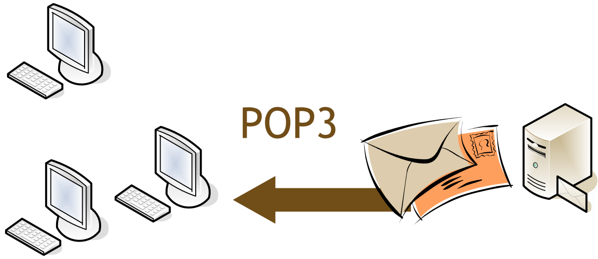
Some clients allow to leave mail server

- Server with support UIDL command: returns unique identifier list messages per mailbox and all sessions

To know if a message has been read or should not have a record to the local mailbox

- POP3 not allowed to do on the server!

It is necessary that the client server delete messages from a certain time (7- 30 days for example)



Sample dialogue and UIDL POP3 APOP

```
C:                                              (Start TCP connection port 110)
S:                                              (Accepts TCP connection)
S: +OK POP3 server ready <1234.697170952@guys.com>
C: APOP asmith c9dcb935ce1d21d02afdebcbd9cb1d5a
S: +OK user authenticated
C: UIDL
S: +OK 2 message (320 octets)
S: 1 XXX001                                    (+ 30 days)
S: 2 XXX079                                     (new message)
S: .
C: RETR 2
S: +OK 200 octets
S:                                              (Message 2 content, headers, null line and body)
S: .
C: DELE 1                               (Client automatically deletes)
S: +OK message 1 deleted
C: QUIT
S: +OK POP3 server signing off
S:                                              (The end of server TCP connection is closed)
C:                                              (The end of client TCP connection is closed)
```

IMAP: Internet Mail Access Protocol

POP problems

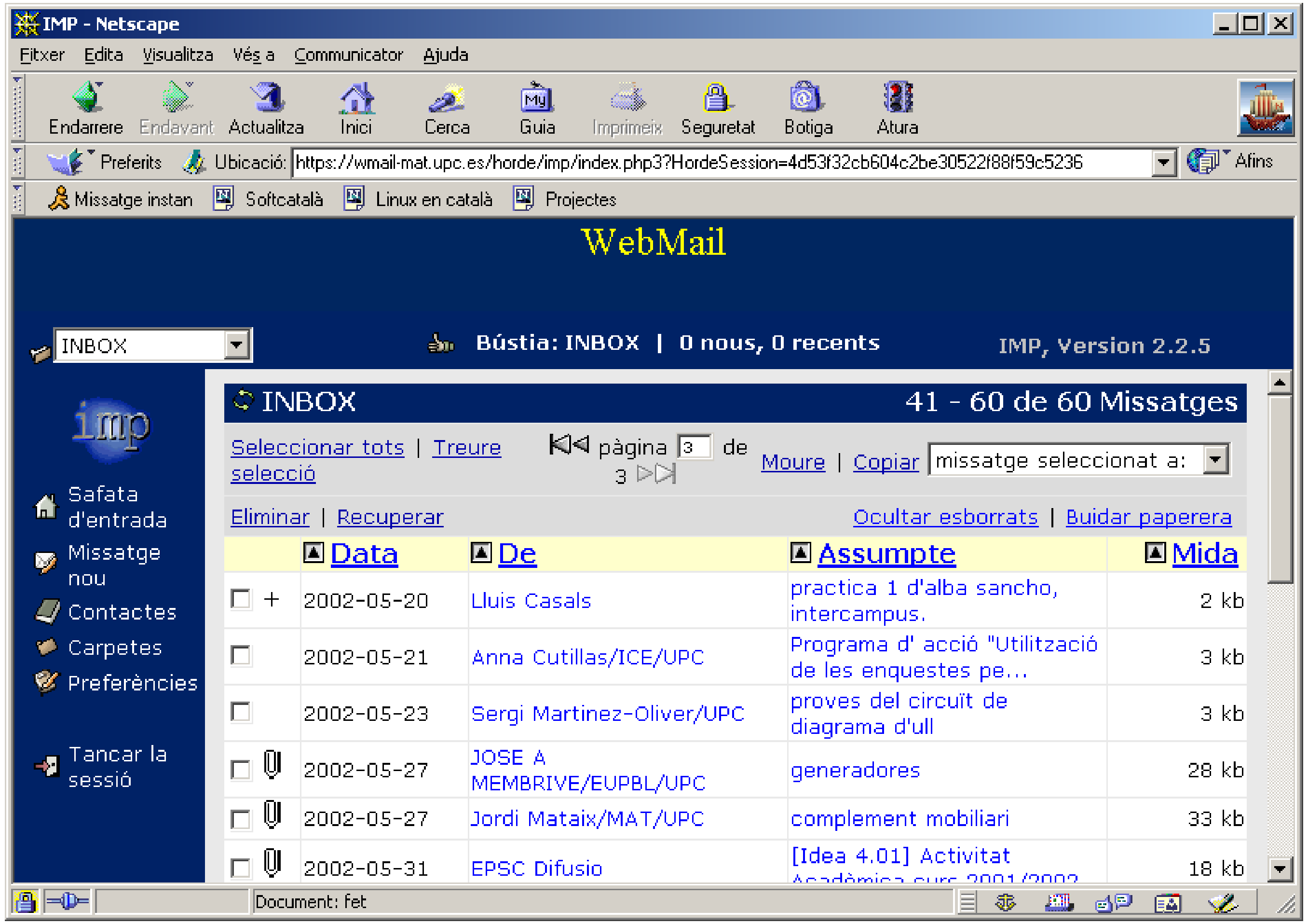
- Mails are "down" to the client (and deleted from the server)
- If you access the same mailbox des from different terminals, messages are distributed
- It is difficult to integrate with new interfaces like WWW

Solution: IMAP

- 1986, Stanford University
- It has not been so far (1995-2000) has been successful
- Current version: IMAP4rev1 (1994) described RFC 2060
- Allows webmail

6.4. Email

IMAP allows webmail



IMP - Netscape
Fitxer Edita Visualitza Vés a Communicator Ajuda
Endarrere Endavant Actualitza Inici Cerca Guia Imprimeix Seguretat Botiga Atura
Preferits
Afins
Missatge instan Softcatalà Linux en català Projectes
WebMail
IMP, Version 2.2.5
imp
Safata d'entrada
Missatge nou
Contactes
Carpetes
Preferències
Tancar la sessió
Seleccionar tots | Treure selecció
pàgina 3 de 3
Moure | Copiar
missatge seleccionat a:
Eliminar | Recuperar
Ocultar esborrats | Buidar paperera
Document: fet

Ubicació: https://wmail-mat.upc.es/horde/imp/index.php3?HordeSession=4d53f32cb604c2be30522f88f59c5236

INBOX

Bústia: INBOX | 0 nous, 0 recents

INBOX 41 - 60 de 60 Missatges

		Data	De	Assumpte	Mida
□	+	2002-05-20	Lluis Casals	practica 1 d'alba sancho, intercampus.	2 kb
□		2002-05-21	Anna Cutillas/ICE/UPC	Programa d' acció "Utilització de les enquestes pe...	3 kb
□		2002-05-23	Sergi Martinez-Oliver/UPC	proves del circuït de diagrama d'ull	3 kb
□	📎	2002-05-27	JOSE A MEMBRIVE/EUPBL/UPC	generadores	28 kb
□	📎	2002-05-27	Jordi Mataix/MAT/UPC	complement mobiliari	33 kb
□	📎	2002-05-31	EPSC Difusio	[Idea 4.01] Activitat Acadèmica curs 2001/2002	18 kb

6.4. Email

IMAP VS POP

POP

- A single server, single inbox 👎
- single terminal: local copy of messages 👎
- Minimum connection: downloaded messages and upload the new 👍

IMAP

- You can work with different mailboxes and servers 👍
- Messages are the server (access from different terminals) 👍
- continuous connection, depending on the connection 👎

News

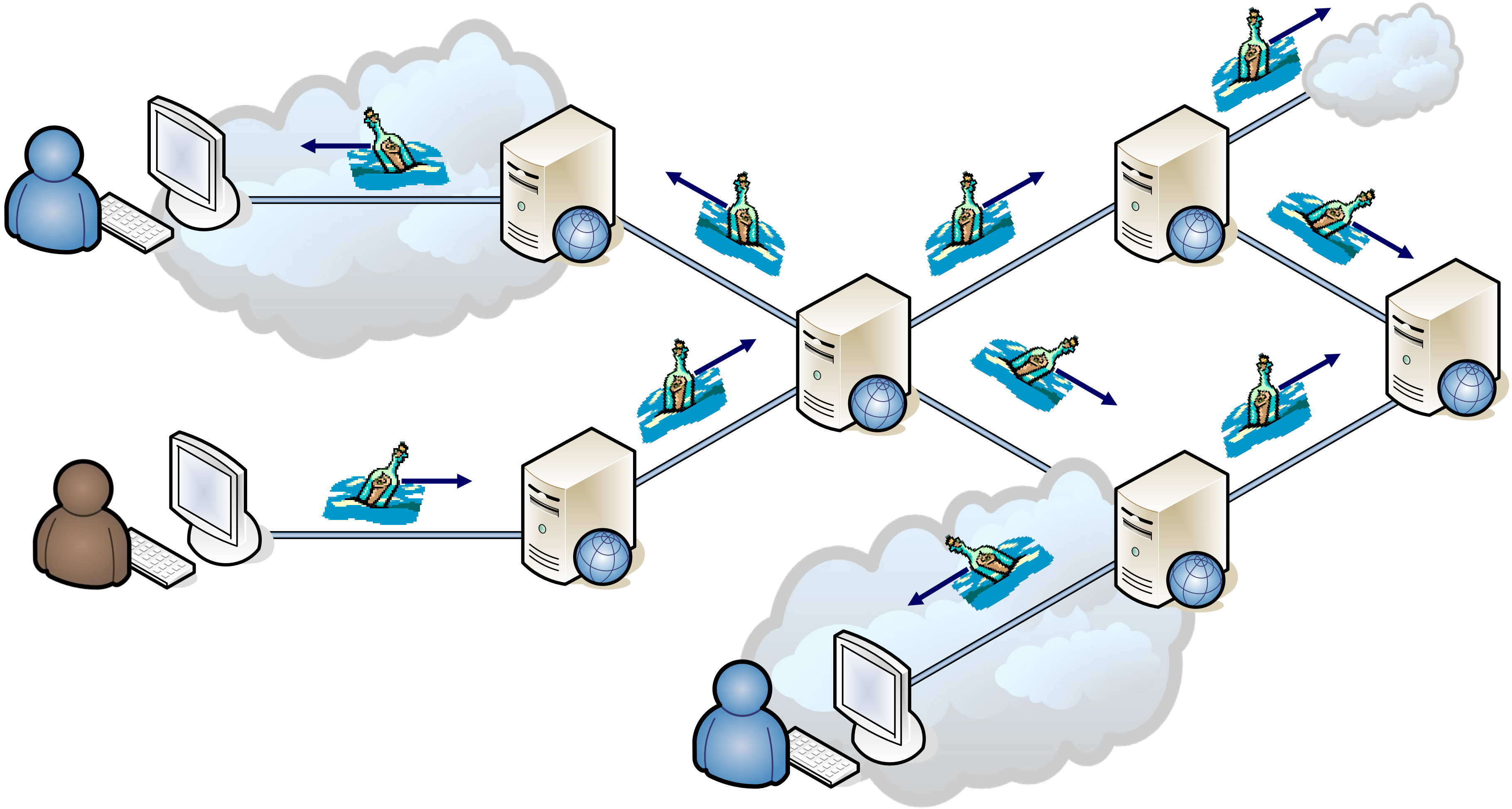
Monographic discussion forums accessible by email

- Protocol nTP (*News Network Transfer Protocol*)
- Example of the distributed nature of the Internet
  - No central node.
  - News servers distribute messages through a process of "dispersion" to neighboring servers
- Organized in hierarchies.
  - Globals:
    - high (alternative topics) and (subjects related to computers), soc (society), news (group management), sci (science), etc.
  - Locals:
    - rediris, es, upc, ieee, etc.
- The names of the groups indicate which is the subject of interest

  comp.os.ms-windows.networking.tcp-ip,upc.tertulia, rediris.anuncios.congresos...

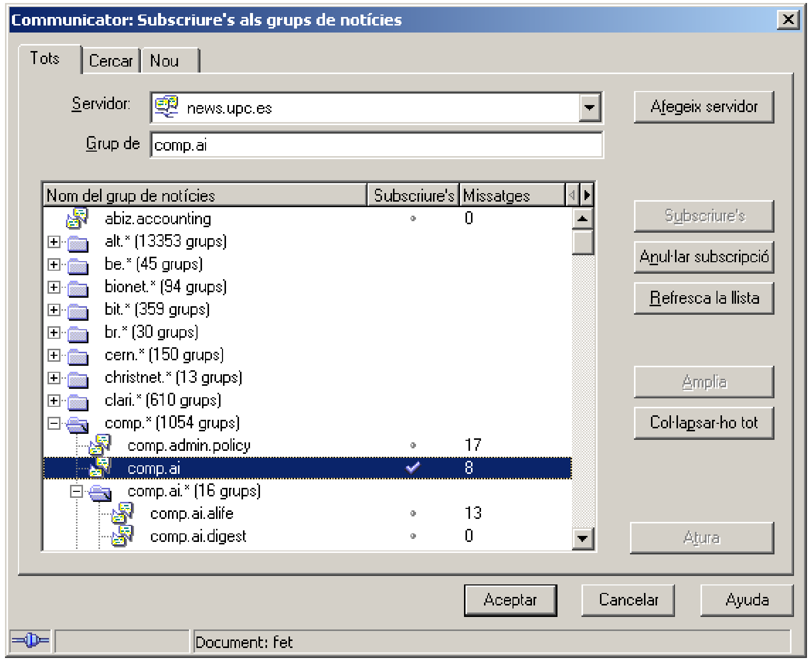
6.5. News

Operating diagram of the news



6.5. News

Example Client News - Netscape



Servidor: news.upc.es

Grup de: comp.ai

Nom del grup de notícies	Subscriure's	Missatges
abiz.accounting	∘	0
alt.* (13353 grups)		
be.* (45 grups)		
bionet.* (94 grups)		
bit.* (359 grups)		
br.* (30 grups)		
cern.* (150 grups)		
christnet.* (13 grups)		
clari.* (610 grups)		
comp.* (1054 grups)		
comp.admin.policy	∘	17
comp.ai	✓	8
comp.ai.* (16 grups)		
comp.ai.alife	∘	13
comp.ai.digest	∘	0

Document: fet

Almost not used

Rs has replaced (XML evolution i www)

6.6. Conclusions

Basic concepts

We must distinguish between service and application

- Service: abstract concept
- application: piece of software that provides the service

There are services-oriented applications and connectionless

There are two basic architectures:

- client - Server
- Peer – to – Peer

Basic services in networks TCP / IP

- DNS: IP obtains correlation - name
- Telnet: allows remote control of a computer
- FTP allows you to transmit files between computers
- Email: send messages between users (POP / IMAP)
- News: create discussion forums on topics

6.7. Web technology

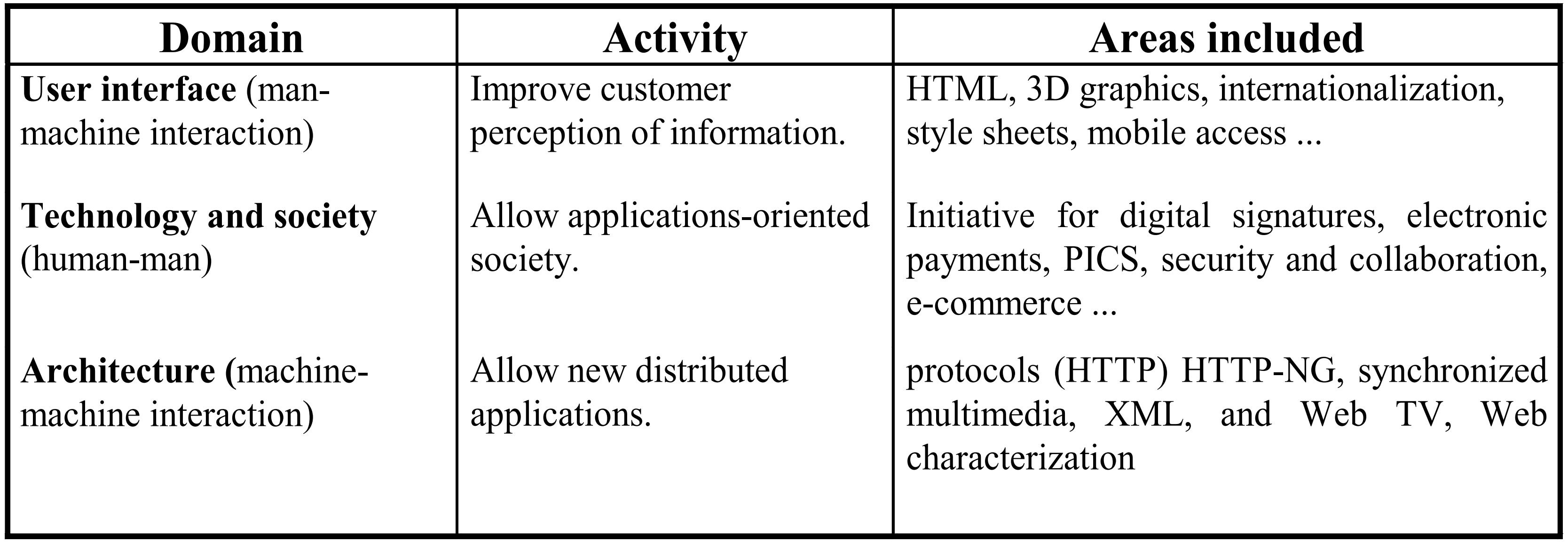
6.7.1. Basic concepts

World Wide Web (WWW)

1989: Tim Berners-Lee creates at CERN as shared global information system

- Objective: Overcoming difficulties of using existing systems (ftp, archie, gopher ...)

Then developed by the W3C (World Wide Web Consortium)



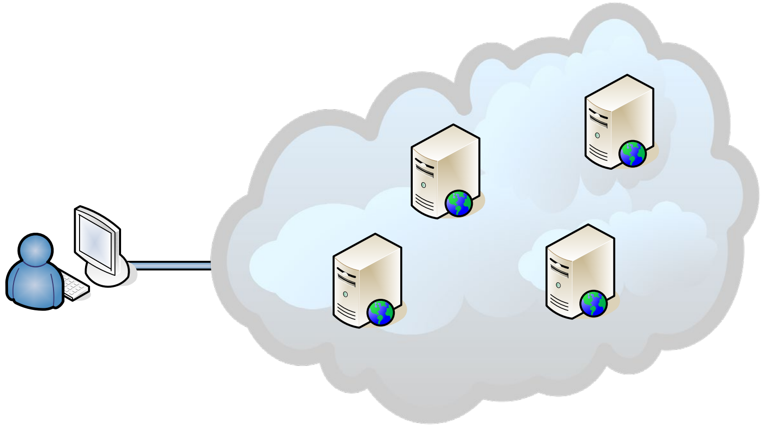
Domain	Activity	Areas included
User interface (man-machine interaction)	Improve customer perception of information.	HTML, 3D graphics, internationalization, style sheets, mobile access ...
Technology and society (human-man)	Allow applications-oriented society.	Initiative for digital signatures, electronic payments, PICS, security and collaboration, e-commerce ...
Architecture (machine-machine interaction)	Allow new distributed applications.	protocols (HTTP) HTTP-NG, synchronized multimedia, XML, and Web TV, Web characterization

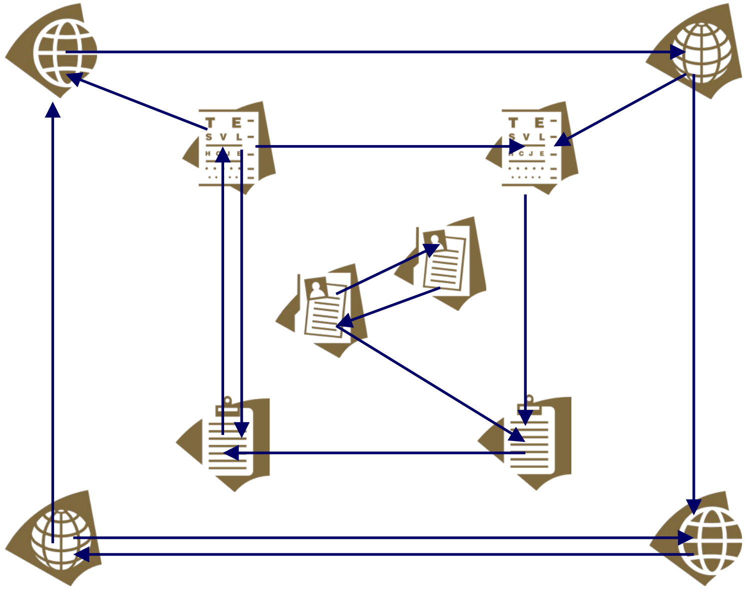
6.7.1. Basic concepts

Work philosophy

Cobweb consists of linked documents without needing to know where they are.

- only we need to know your ID
- Links: Hypertext





Protocol families

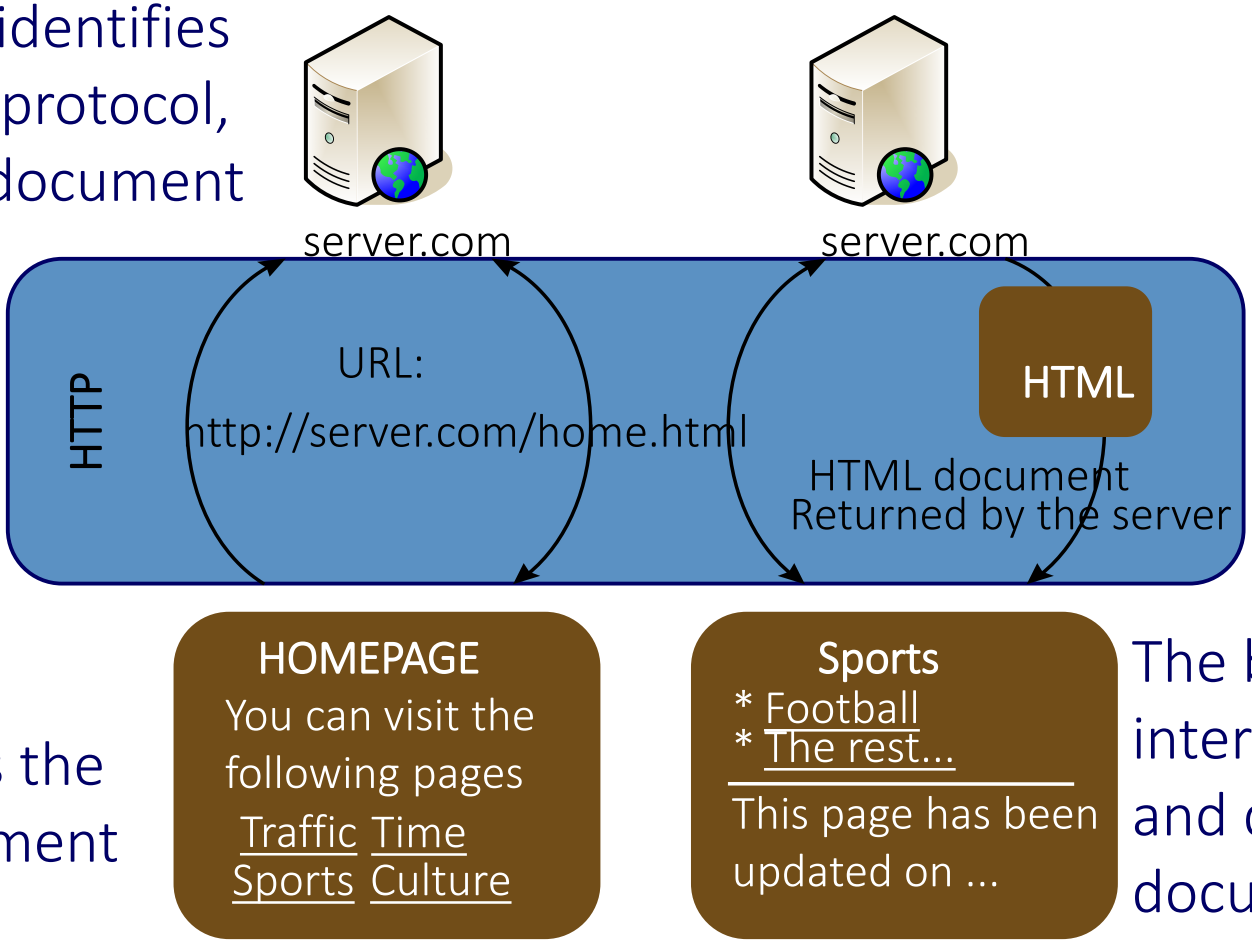
WWW Service Protocols

- **HTML:** *Hypertext Markup Language*
  - Information representation format
- **HTTP:** *Hypertext Transfer Protocol*
  - Information transport protocol
- **URL:** *Uniform Resource Locator*
  - Identifying information

6.7.1. Basic concepts

Basic operation

URL identifies
the protocol,
the server and document



HOMEPAGE

You can visit the following pages

Sports

This page has been updated on ...

HTTP is the protocol that allows clients and servers communicate through the network

The hyperlink contains the URL of the linked document

The browser interprets the HTML and displays the document on the screen

6.7.2. HTML

HyperText Markup Language (HTML)

Born with WWW

Suitable to represent all types of documents

- It has been a real revolution Internet

Standardization : W3C

- Current standard: HTML 4.0.1 (December 1999)

It is evolving into the XML (eXtended Markup Language)

- W3C currently recommended XHTML 1.0 (January 2000)
- XHTML: *the eXtensible Hypertexto Markup Language*

6.7.2. HTML

HTML is an SGML language

It’s a **language**

Use **tags** to format the document (Markup)

The format is given **based on content** not on the appearance of the document (Generalized)

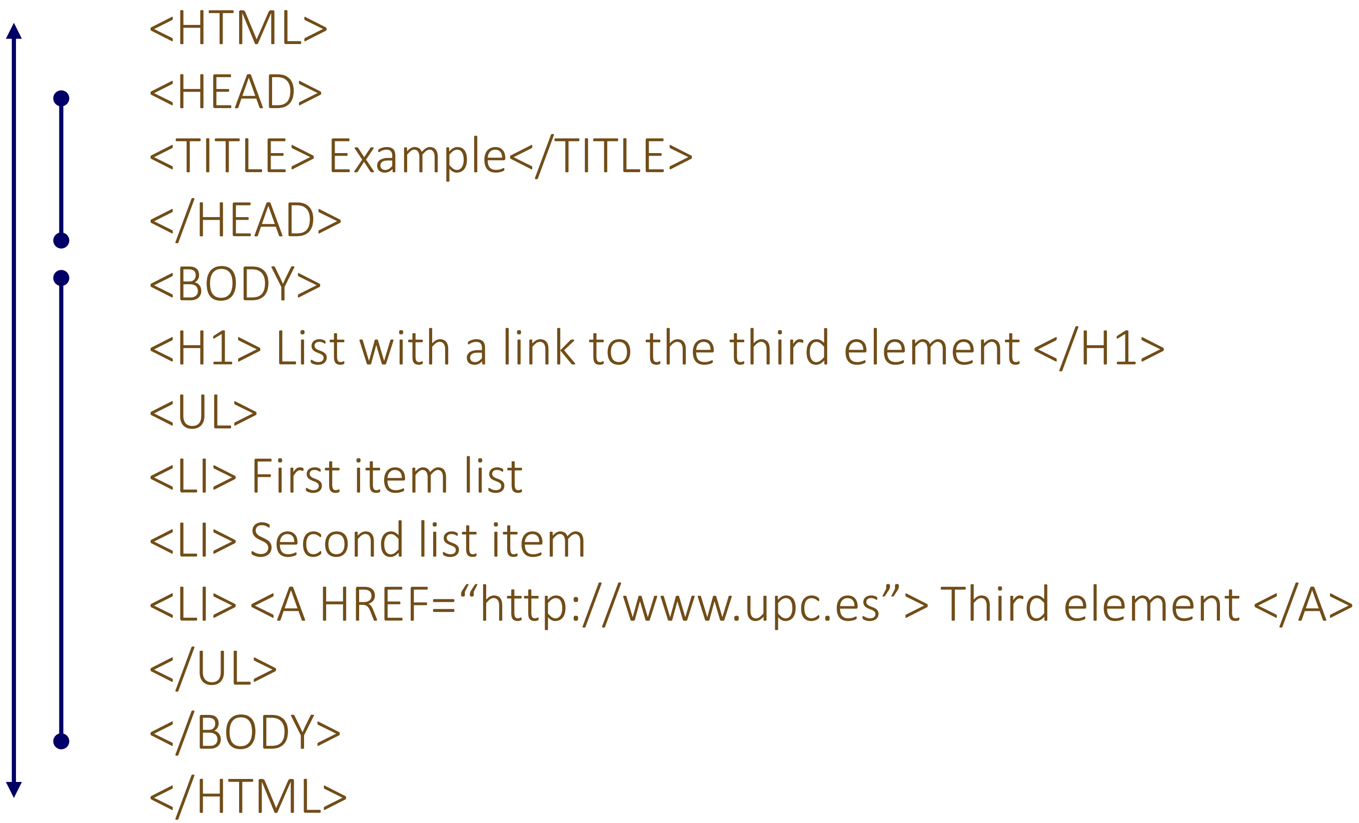
- Flexible application: it adapts to the possibilities of the platform

Widely **accepted** and in the case of HTML, **does not own** (Standard)

- information needed to develop is public

6.7.2. HTML

Example of HTML code



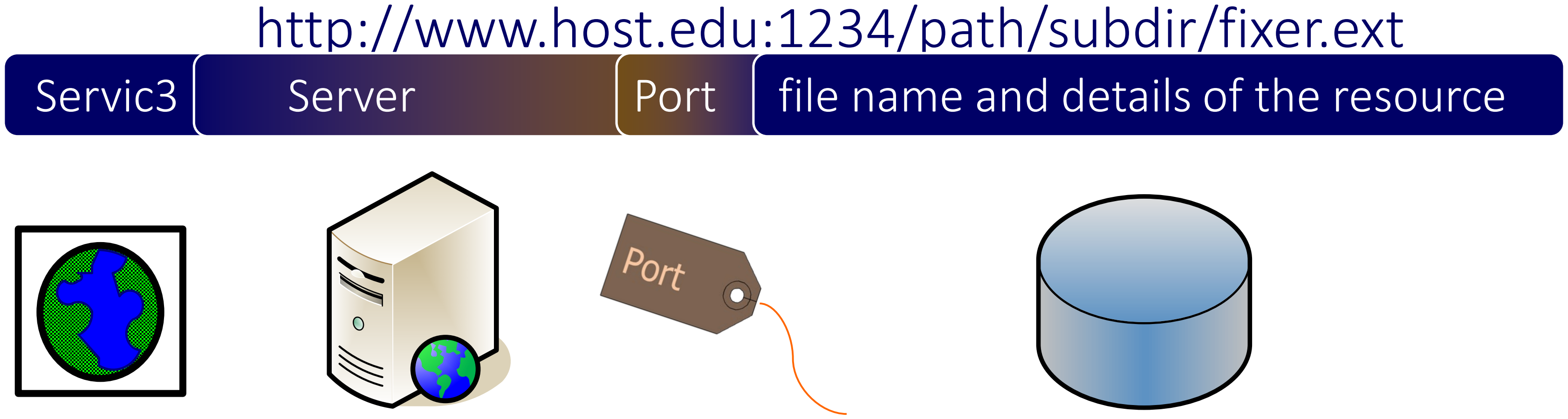
```
<HTML>
<HEAD>
<TITLE> Example</TITLE>
</HEAD>
<BODY>
<H1> List with a link to the third element </H1>
<UL>
<LI> First item list
<LI> Second list item
<LI> <A HREF=“http://www.upc.es”> Third element </A>
</UL>
</BODY>
</HTML>
```

Network Engineering Department

6.7.3. URL

Generic URL structure

- Defined into RFC 1738
- Extension of the concept of filename



http://www.host.edu:1234/path/subdir/fixer.ext

Servic3 | Server | Port | file name and details of the resource

6.7.3. URL

Access to resources through URL

HTTP:	http://www.catacrac.es
FTP:	ftp://soc.alegria.com/
News (NNTP):	news: rec.motorcycles.harley
e-mail (SMTP):	mailto:rvidal@mat.upc.es
Remote files:	file://maite168/c:/docs/capitol3.doc
Local files:	file://c:/docs/capitol3.doc
Partial URLs :	If within the document http://www.upc.es/HomePage.html I want to point to http://www.upc.es/departeaments.html, it is sufficient with departeaments.html

6.7.4. HTTP

HyperText Transfer Protocol (HTTP)

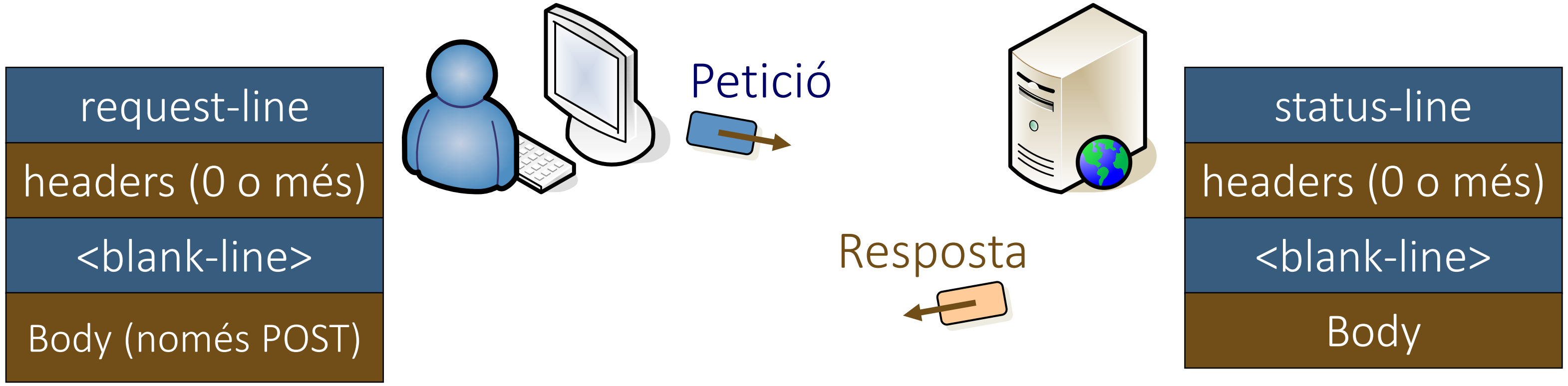
HTTP 1.1 standard, RFC 2616

- It defined an authentication system, RFC 2617
- TCP uses typically port 80
- It requests and responses sent in text format
- Simple dialogue session

Connection	Establishing the connection from the client to the server (TCP / IP,
Petition	The client sends to the server the request message
Answer	The server sends the response message to the client
Closing	The server indicates the end closing the connection

Most Internet traffic corresponds to HTTP !!

Format of HTTP requests and responses



Where we have

- Request-line: *request* *request URI* *HTTP version*
- Status-line: *HTTP version* *3 digit response code* *human readable response phrase*
- Request: GET, HEAD, POST, ...

Description of HTTP commands (request)

GET

- Request that returns the information is identified by request-URI (URL)

HEAD

- Similar to GET. name server returns the headers, but not the content (body) of the specified document.
- It is used to test a hypertext link, accessibility, and recent modification.

PUT

- Asks the customer to accept and store a resource with the request-URI customer demand.

DELETE

- The reverse of the above, to delete a resource

6.7.4. HTTP

Description of HTTP commands (request)

POST

- Request used to send e-mail, news, or forms that can be filled by an interactive user.
- It is the only request sends a content (body) with the request.
- You need a field by the Content-Length header to specify the content length.

OPTIONS

- To ask the server for the capabilities of a particular resource or server in general

TRACE

- Used by application-level debugging

6.7.4. HTTP

Numerical response codes (3 digit response code)

Response	Description
1yz	Informational. Not currently used.
	Success.
200	OK, request succeeded.
201	OK, new resource created (POST command).
202	Request accepted but processing not completed.
204	OK, but no content to return
	Redirection; further action need be taken by user agent.
301	Requested resource has been assigned a new permanent URL.
302	Requested resource resides temporaly under a different URL.
304	Document has not been modified (conditional GET).
	Client error.
400	Bad request.
401	Unauthorized; request requires user authentication.
403	Forbidden for unspecified reason.
404	Not found.
	Server error.
500	Internal server error.
501	Not implemented.
502	Bad gateway; invalid response from gateway or upstream server.
503	Service temporaly unavaible.

6.7.4. HTTP

HTTP 1.0 Headers

Nom del Header	Request	Response	Body
Allow			*
Authorization	*		
Content-Encoding			*
Content-Length			*
Content-Type			*
Date	*	*	
Expires			*
From	*		
If-Modified-Since	*		*
Last-Modified			
Location		*	
MIME-Version	*	*	
Pragma	*	*	
Referer	*		
Server		*	
User-Agent	*		
WWW-Authenticate		*	

Network Engineering Department

6.7.4. HTTP

Example HTTP transaction

Request

GET /foto.gif HTTP/1.0
From: telrvf@maite.upc.es
(Blank line)

Answer

HTTP/1.0 200 OK
Date: Sat, 7 Feb 97 17:49:25 GMT
Server: NCSA/1.3
MIME version: 1.0
Content – type: imatge – gif
Last – modified: Mon, 14 Nov 96 12:04:22 GMT
Content – length: 22700
(blank line)

(The 22700 bytes sent photo.gif in binary format)
Finally, the server closed the TCP connection)

Example telnet access to an HTTP server

- telnet www.upc.es 80
- GET / HTTP/1.0 (intro tow times)

HTTP 1.0 HTTP differences resolved by 1.1

Multiple connections TCP / IP

- TCP / IP is intended to keep bitstreams over an extended period of time.
- Using a TCP / IP connection for each HTTP message (requests take up very few bits and responses in general as well), very bad HTTP interacts with the design of TCP / IP.

Cache

- HTTP 1.0 simply allows the use of the cache, but does not describe how the cache interacts with clients or servers.

Hosts names

- HTTP 1.0 does not allow the same IP address is used for different servers that are on the same machine (host)
- Forces have different entrances to the DNS (Domain Name System)

high demand for IP addresses

Other improvements made by HTTP 1.1

Content negotiation

- A server can have different representations of a resource ⇒ send the most appropriate customer

Authentication without sending the password in clear

- Technique uses shared secret (MD5)

Downloading a given range of bytes

- You can ask for sending a part of a resource

Future development of the HTTP protocol

HTTP 1.1 standardized in June 1999.

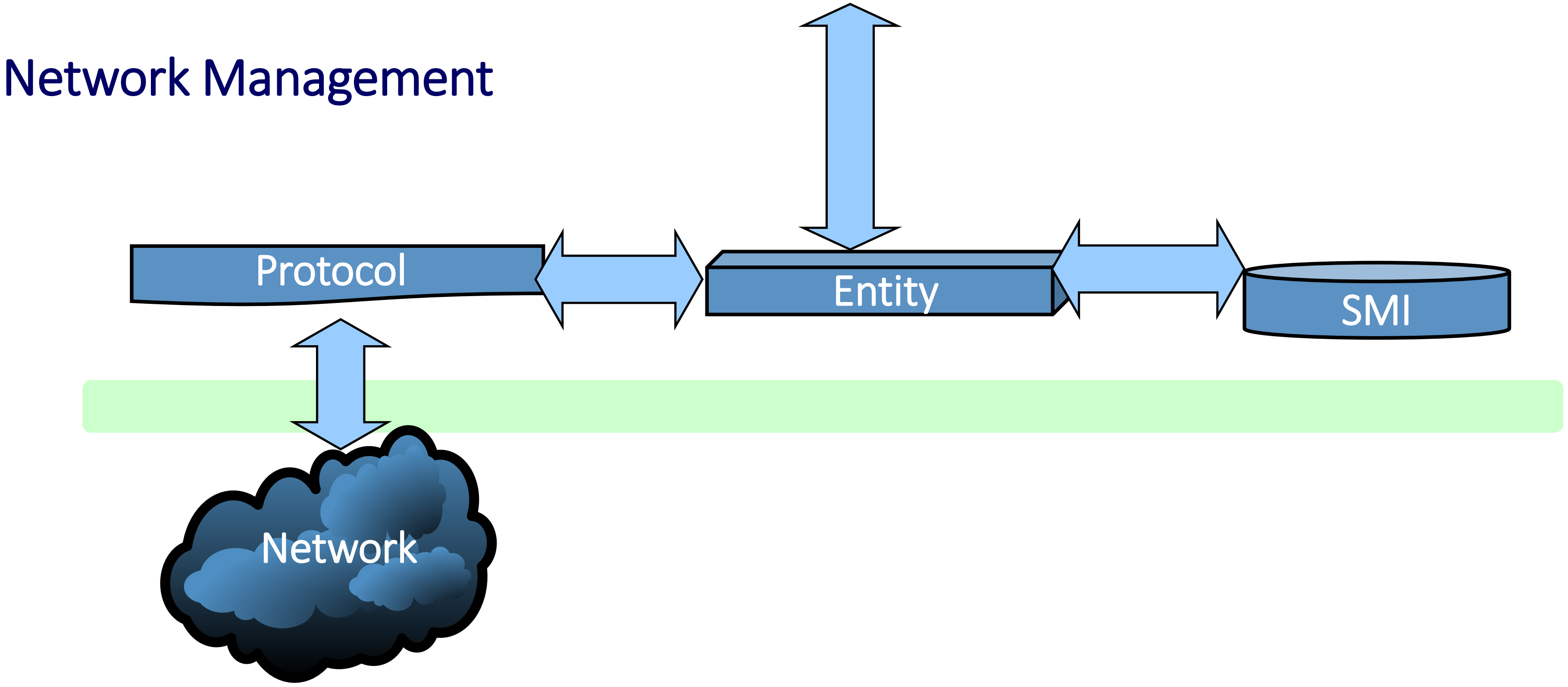
- It is widely supported

HTTP-NG: make a simpler, modular new HTTP, with layer structure

- HTTP 1.1 ms rather complicated and bulky
- MUX: multiplexer different information requests on a TCP connection
- simultaneous presentation

6.8.1. SNMP

SNMP provides a network management environment

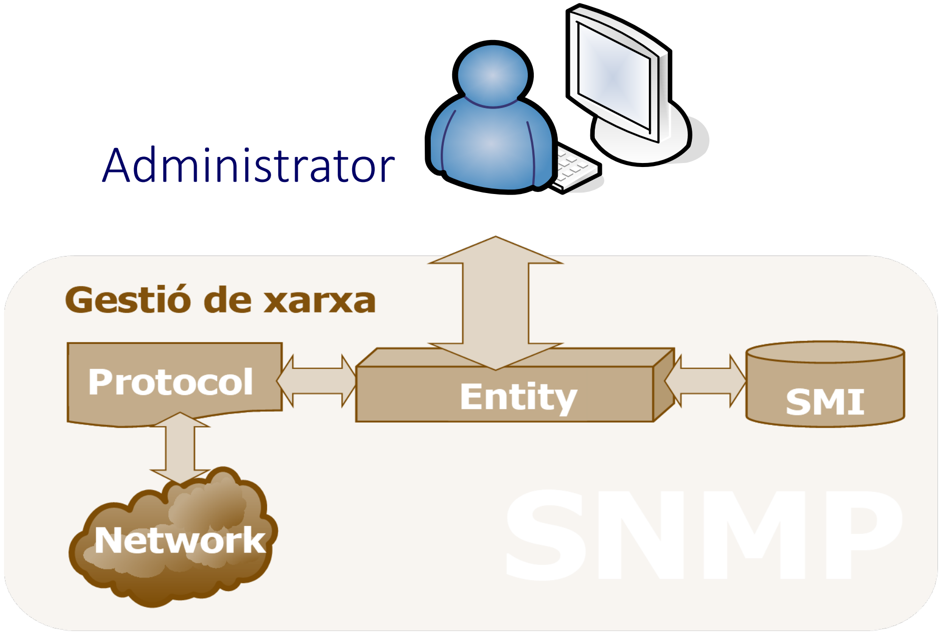


6.8.1. SNMP

Work philosophy

It should be applicable ...

- At the largest scale possible
- With the greater diversity of possible implementation
- With the widest possible layers of protocols
- With the greater diversity of management that can be obtained



6.8.1. SNMP

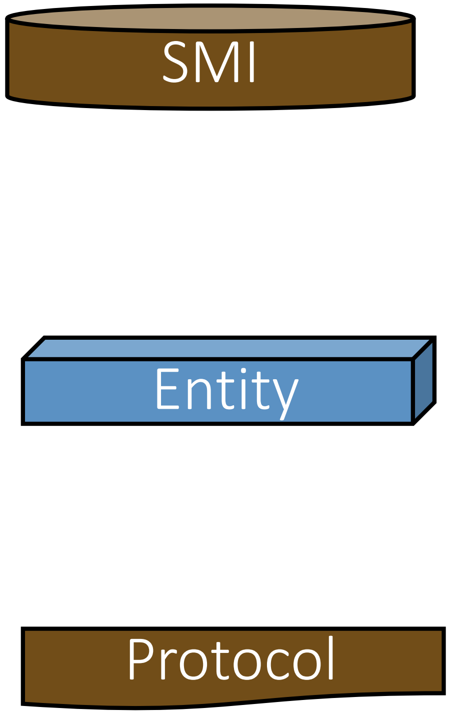
Tasks to be performed

- **Configuration Management:** Configuring the managed network. centralized management.
- **Fault Management**: Detection and, if possible, correction of network failures.
- **Management statistics**: Statistics on network utilization. They allow you to define access policies, pricing, planning, etc.
- **Efficiency management:** Analysis of usage statistics. to discover bottlenecks.
- **Security Management**: Controlling access to managed, generate alarms when intruders are detected devices.

6.8.1. SNMP

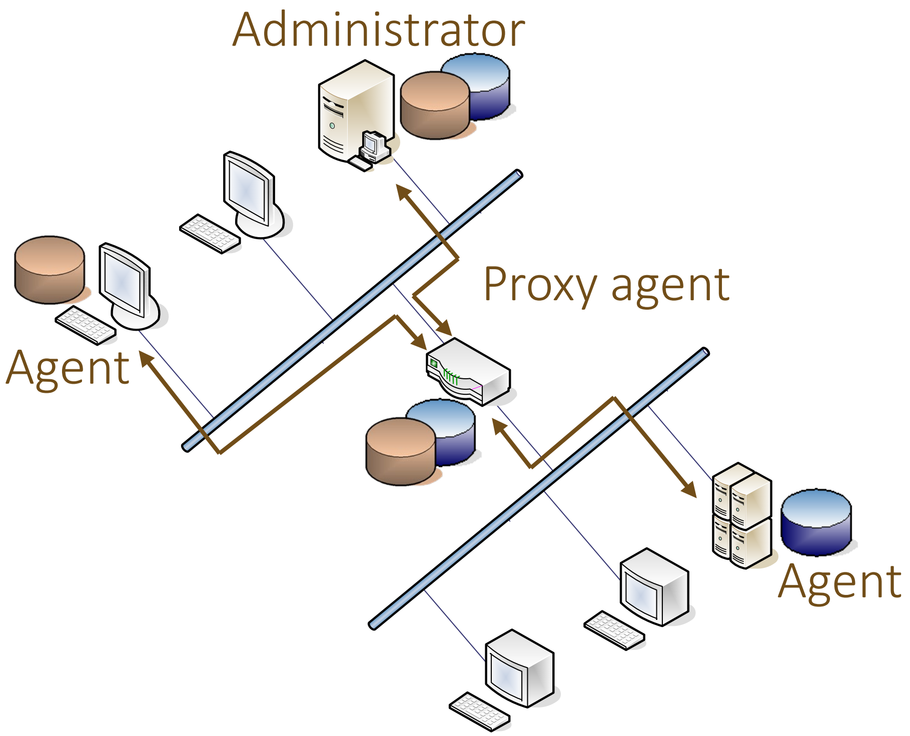
SNMP architecture model

- Database with information on:
  - ✓ Configuration
  - ✓ State
  - ✓ Mistakes
  - ✓ Performance
- Entities managers and administrators:
  - ✓ agents
  - ✓ proxy agents (SNMP v:2.0)
  - ✓ Administrators
  - ✓ Management Information Bases (MIB)
- Communication protocol
  - ✓ Access Transport Layer
  - ✓ Messages



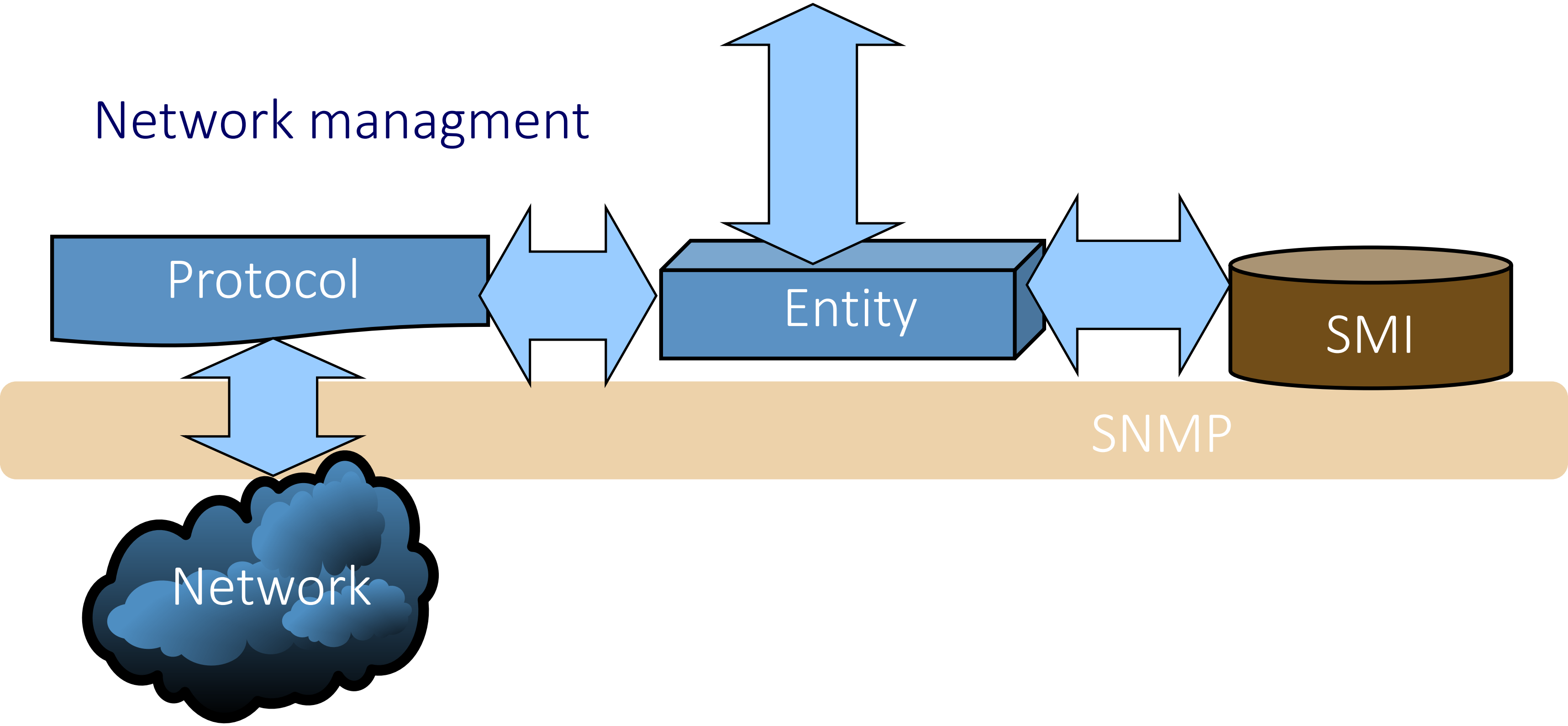
SNMP architecture

- The administrator runs applications, it generates SNMP commands, interrogates agents and presents the data to the user through a graphical interface
- Data is stored in the MIB.
- The SNMP Protocol travels over UDP (ports 161 and 162).



6.8.2. SMI

SMI (Structure of Management Information) It provides the general framework of the MIB



6.8.2. SMI

The characteristics of the SMI are collected in multiple RFCs

RFC 1155: Structure and Identification of Management Informationfor TCP/IP-base Interfaces

RFC 1213: Management Information Base for Netware Management of TCP/IP-base Interfaces: MIB-II

RFC 1643: Definition of Managed Objects for the Ethernet-likeInterface Types

RFC 2021: Remote Network Monitoring Management Information Base 2

.....

6.8.2. SMI

Key points of the structure of management information

Establishes the methodology to create the structure of the MIB, **SMI tree**

Defined as the objects of the MIB, both the syntax and value are created **ASN.1**

Define the methodology to encode the values of MIB objects, **BER**

6.8.2. SMI

The MIBs contain administrable objects

- It contains the logical description of the data network administration (RFC 1155).
- The sets of related variables are grouped in **MIB modules** (listed in RFCs).
- The description of a variable specifies:

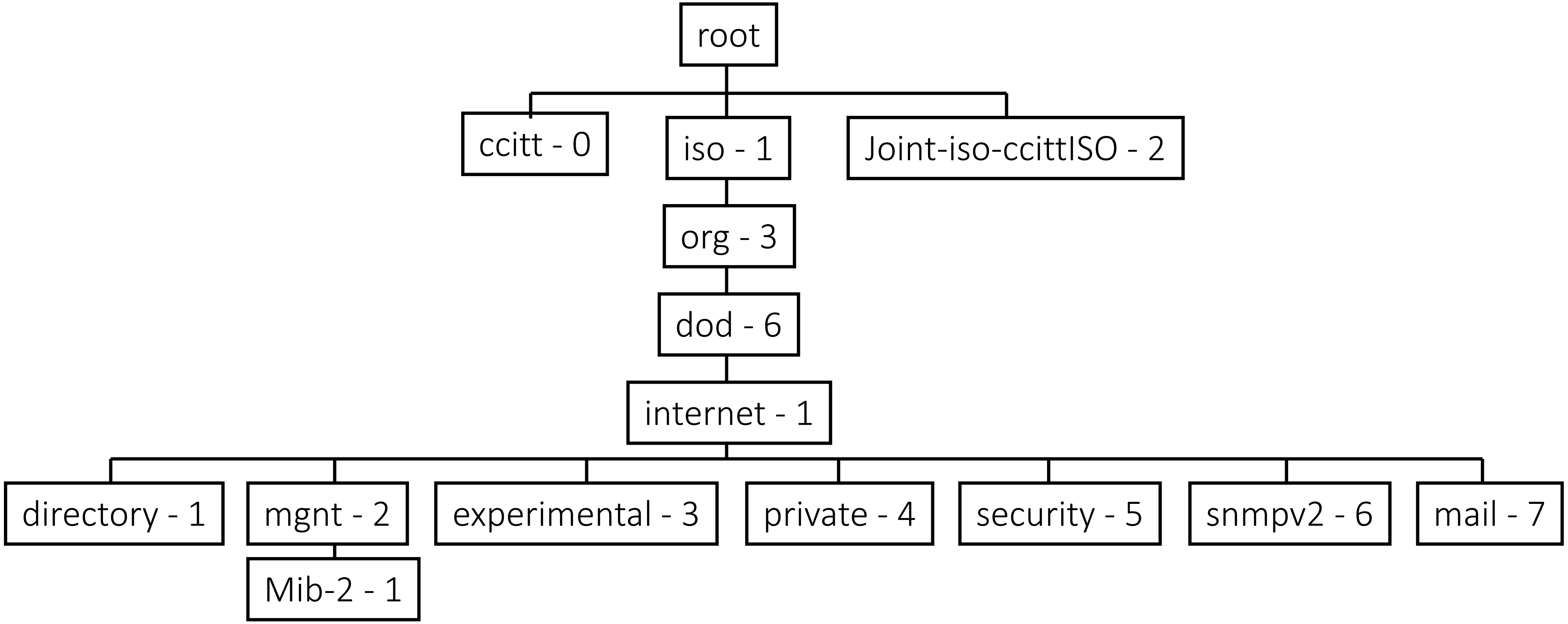
  A definition of what is variable.

  A description of how the value is measured.

  A name for when you read or update the value of the variable to the database.

6.8.3. SMI tree

MIB variables are arranged along a tree structure



6.8.3. SMI tree

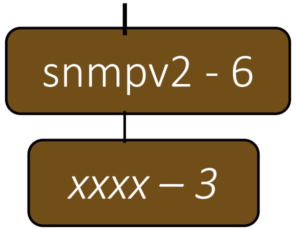
Object names identifiers

- At the end the number of managed system is included
- Every variable can be identified in two ways:

  Object Identifier: starting from the root joining the node numbers

  Object name: start from the root joining the names of nodes

Example:



Object Identifier: ***1.3.6.1.6.3***

name: ***iso.org.dod.internet.snmpv2.xxxx***

6.8.3. SMI tree

Ordination in the MIB tables

- It starts on the left
- This compares to found the first different value
- The element with the highest number in this position is the largest element
- Otherwise, the largest identifier is the largest value

Example:

…………………

1.3.6.1.6.2

1.3.6.1.6.3

1.3.6.1.6.4

1.3.6.1.6.4.1

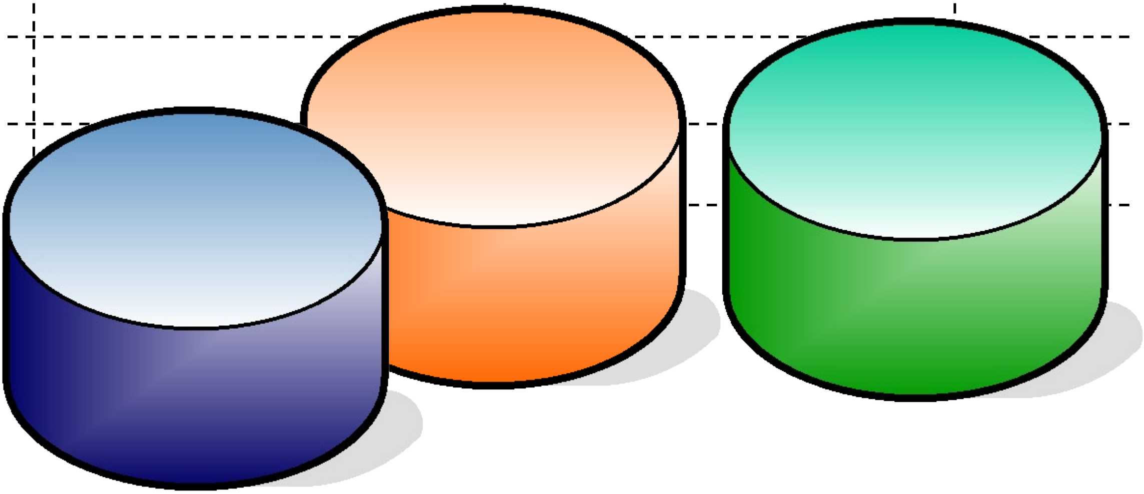
…………………

6.8.4. Codification ASN.1

The data are classified by type and labels

They are grouped by their nature (type, eg .: integer) and use (tag, ex: UPC router)

Tag *Type*	*Universal*	*Application-wide*	*Context-specific*	*Private*
Simple				
Structured				
Tagged				
Other				



6.8.4. Codification ASN.1

UNIVERSAL class data forms the basis of other variables

Basic types
1 - BOOLEAN
2 - INTEGER
3 - BIT STRiNG
4 - OCTET STRinG
9 - REAL
10 - ENUMERATED

Object Type
6 - OBJECT IDENTIFIER
7 - Object descriptor

Reserved
19-5 , 28-...

Type strings
18 - NumericString
19 - PrintableString
27 - GeneralString, ...

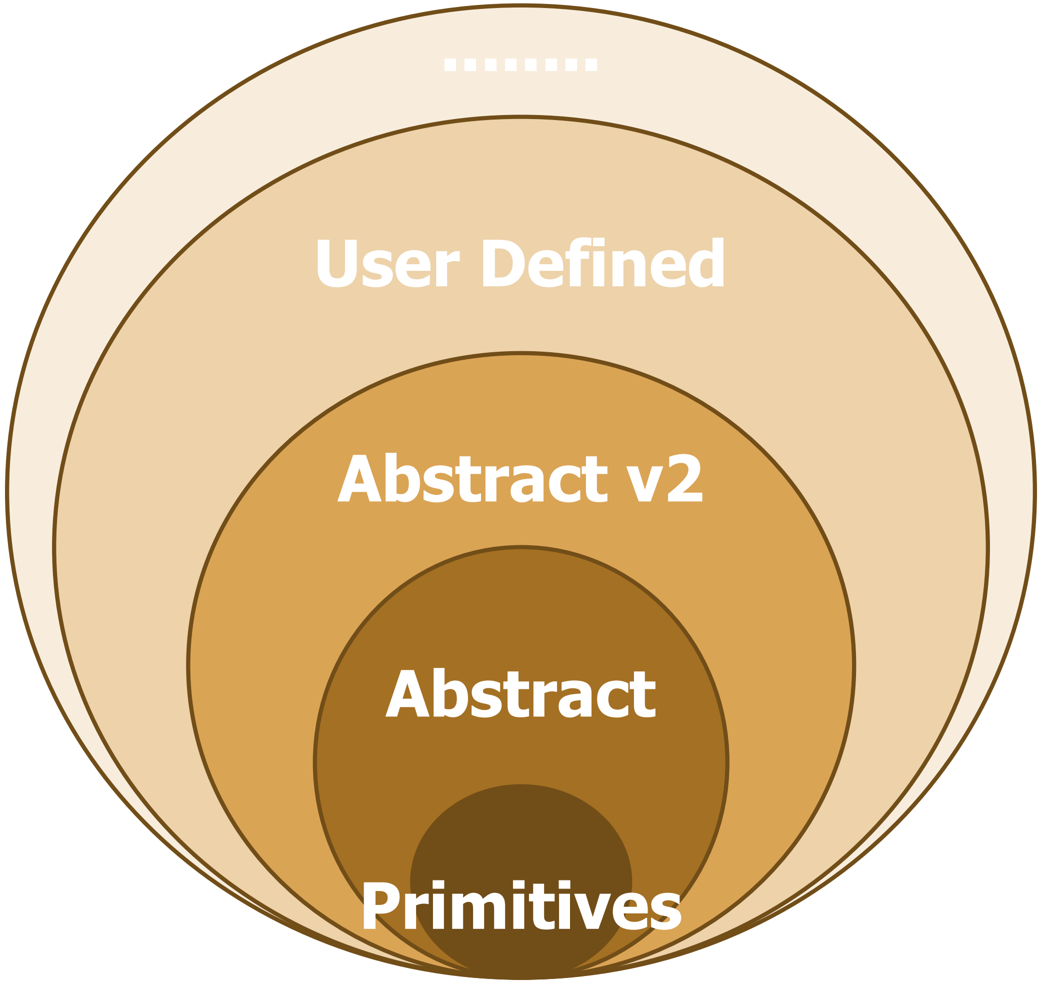
Structured types
16 – SEQUENCE, SEQUENCE OF
17 – SET, SET OF

Various types
5 - NULL
8 - EXTERNAL
23 – UTCTime...

6.8.4. Codification ASN.1

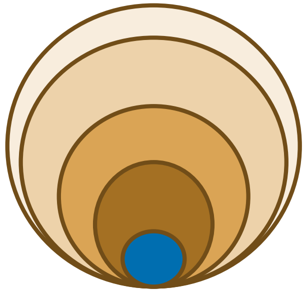
Data Encapsulation

data types are formed from combinations of data already defined in the MIBs



6.8.4. Codification ASN.1

Primitive data

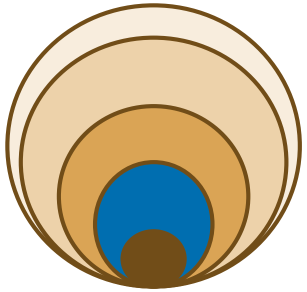


They are the basic data types

- *Integer: 32-bit two's complement. It is used for enumerations*
- *Octet string: string of 0 or more octets. It is used to represent text.*
- *Object Identifier: sequence of integers. It is used to identify objects in the MIBs.*
- *Null*: in white
- *Sequence, sequence-of*: It is used to build tables

6.8.4. Codification ASN.1

Abstract data (SNMP v1)

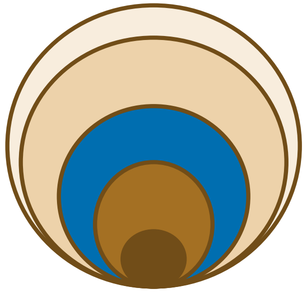


Add the first degree of abstraction

- **NetworkAdres**: network address protocol dependent
- **IpAddresCounter**: 4-byte OCTET STRING
- **Counter**: 32-bit unsigned with overflow
- **Gauge**: 32-bit unsigned with saturation
- **TimeTicks**: Timer 32-bit (hundredths of seconds). Maximum 497 days
- **Opaque**: any data format OCTET STRING

6.8.4. Codification ASN.1

Abstract data (SNMP v2)

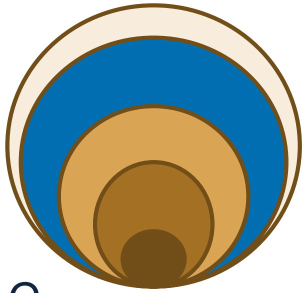


They are a refinement of the above

- *Integer32:* as INTEGER
- *Counter32*: as Counter
- *Counter64*: as Counter but with 64 bits
- *Gauge32*: as Gauge
- *Unsigned32*: as Gauge

6.8.4. Codification ASN.1

Textual conventions



The textual conventions extend the descriptions of the types of managed objects contained in them MIBs

- *Define new types*
- *Represent existing types*
- *Represent the values of the types*
- *Encode the values of existing types*

6.8.5. BER rules

The Basic Encoding Rules, BER, specify how to encode any value defined ASN.1

A value can be encoded in various ways

Use the type OCTET STRING

Follow the structure: type - length - value

There are three methods

- Primitive, definite-length encoding
- Built, definite-length encoding
- Built, indefinite-length encoding

6.8.5. BER rules

BER encoding format

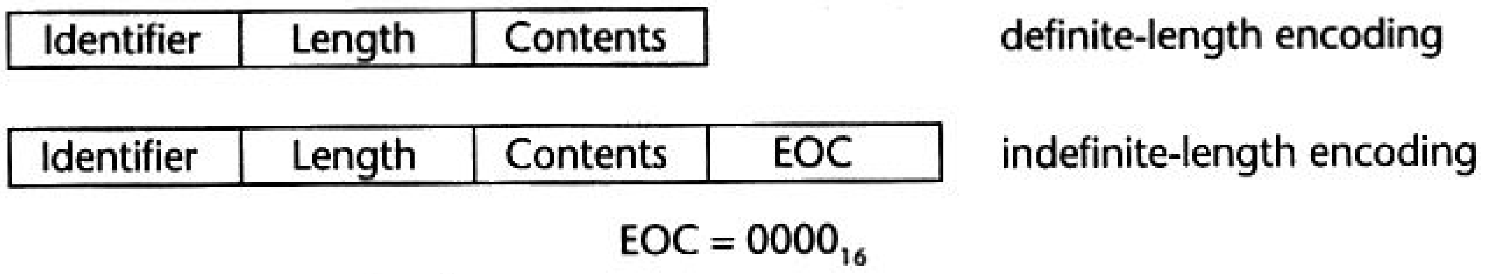
The objects are encoded with four fields

Identifier: managed object reference refers

Length: indicates the number of bytes used

Content: value of the object

Final content: marks the end of the value in the case of variable length

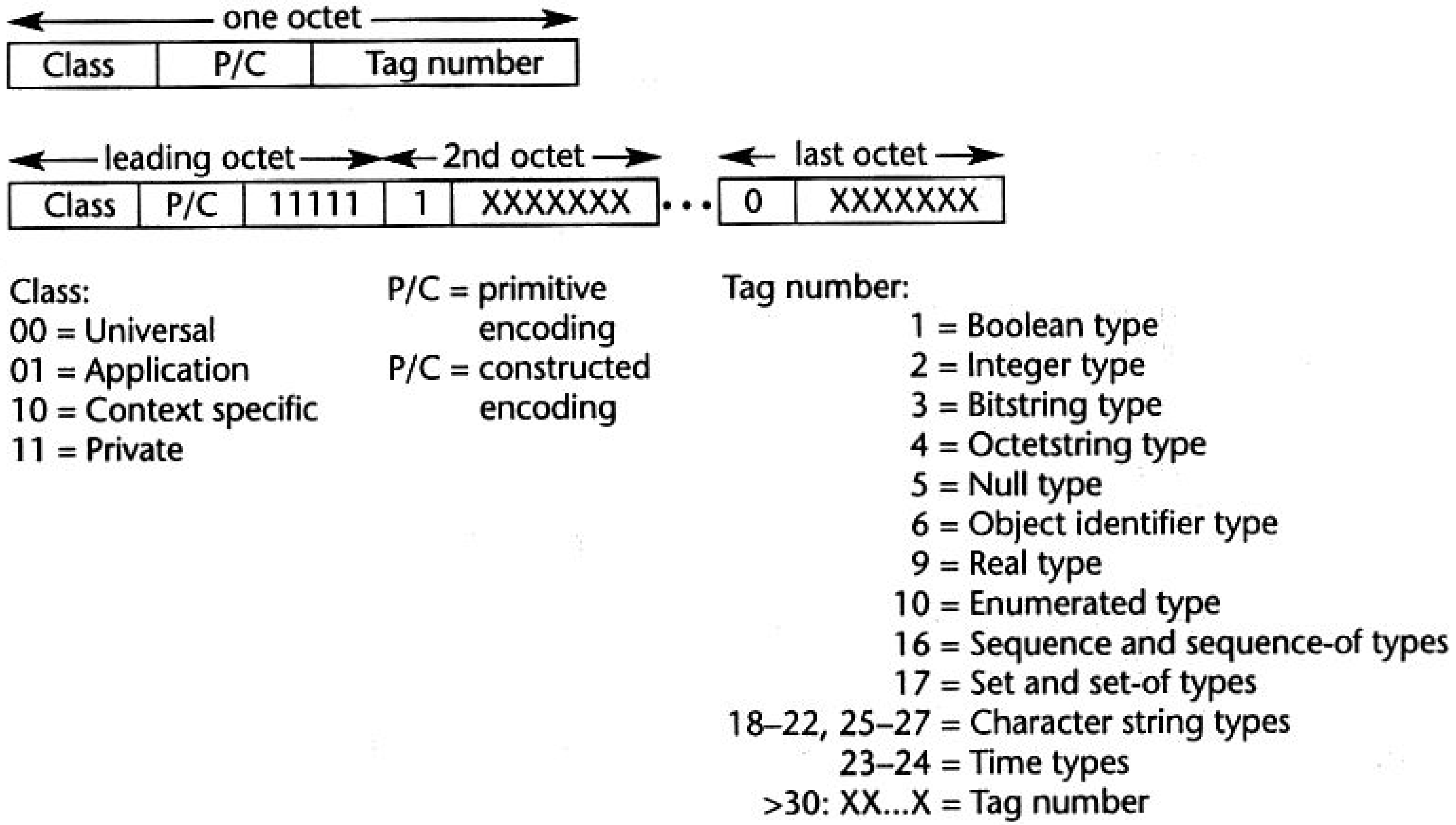


Identifier	Length	Contents		definite-length encoding
Identifier	Length	Contents	EOC	indefinite-length encoding

EOC = 0000_{16}

6.8.5. BER rules

Format identifier field of the BER



Tags <31

one octet

Class	P/C	Tag number

Tags ≥31

leading octet — 2nd octet — last octet

Class	P/C	11111	1	XXXXXXX	• • •	0	XXXXXXX

Class:
00 = Universal
01 = Application
10 = Context specific
11 = Private

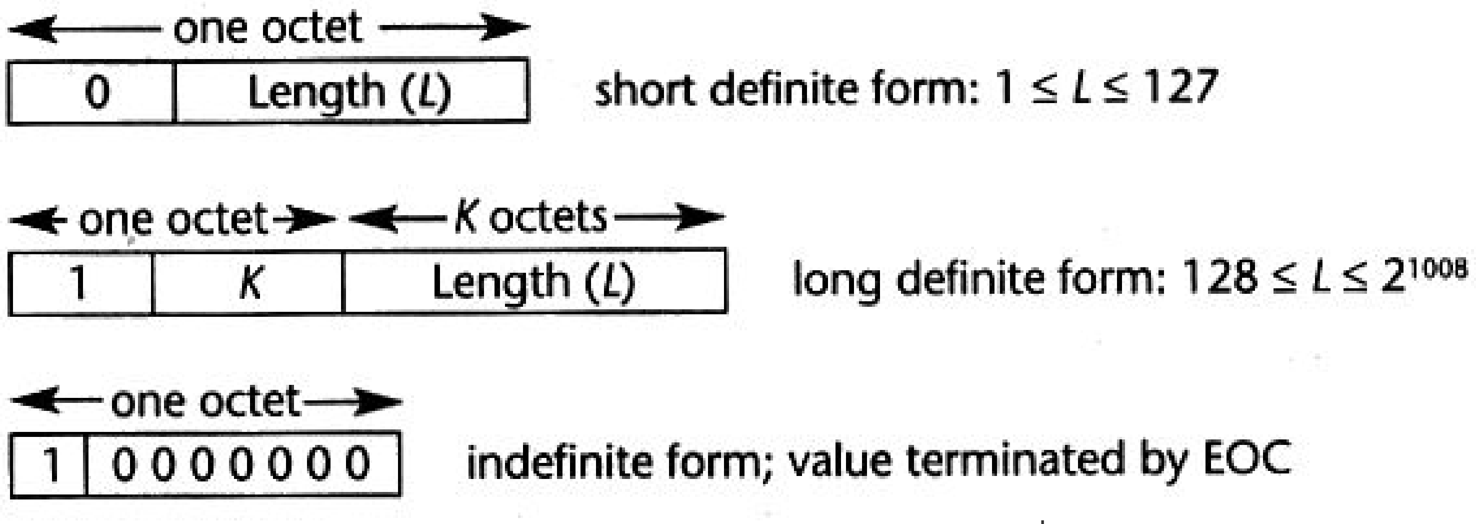
P/C = primitive encoding
P/C = constructed encoding

Tag number:

1	Boolean type
2	Integer type
3	Bitstring type
4	Octetstring type
5	Null type
6	Object identifier type
9	Real type
10	Enumerated type
16	Sequence and sequence-of types
17	Set and set-of types
18–22, 25–27	Character string types
23–24	Time types
>30: XX...X	Tag number

6.8.5. BER rules

Field format length BER



short definite form: $1 \leq L \leq 127$

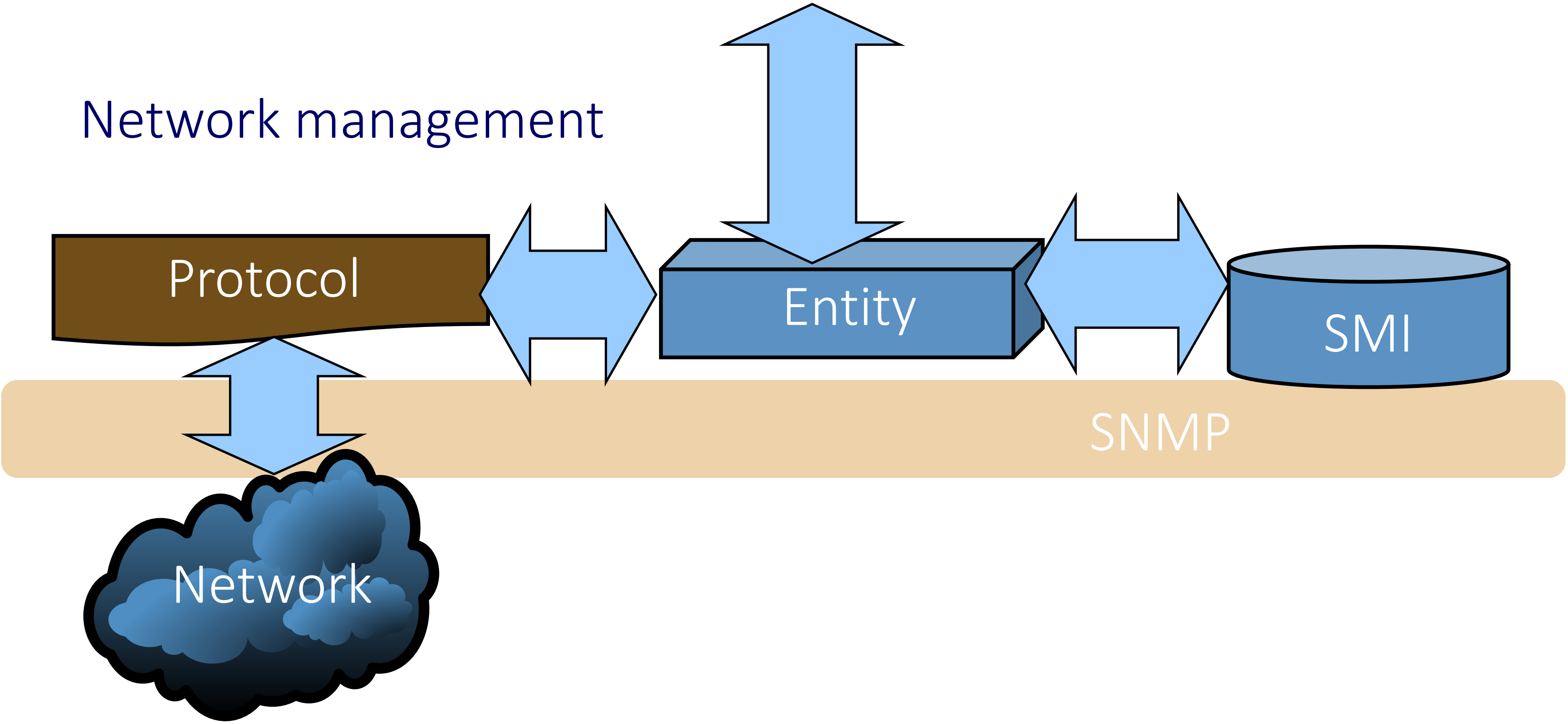
long definite form: $128 \leq L \leq 2^{1008}$

indefinite form; value terminated by EOC

El campo EOC es formado por dos bytes a cero

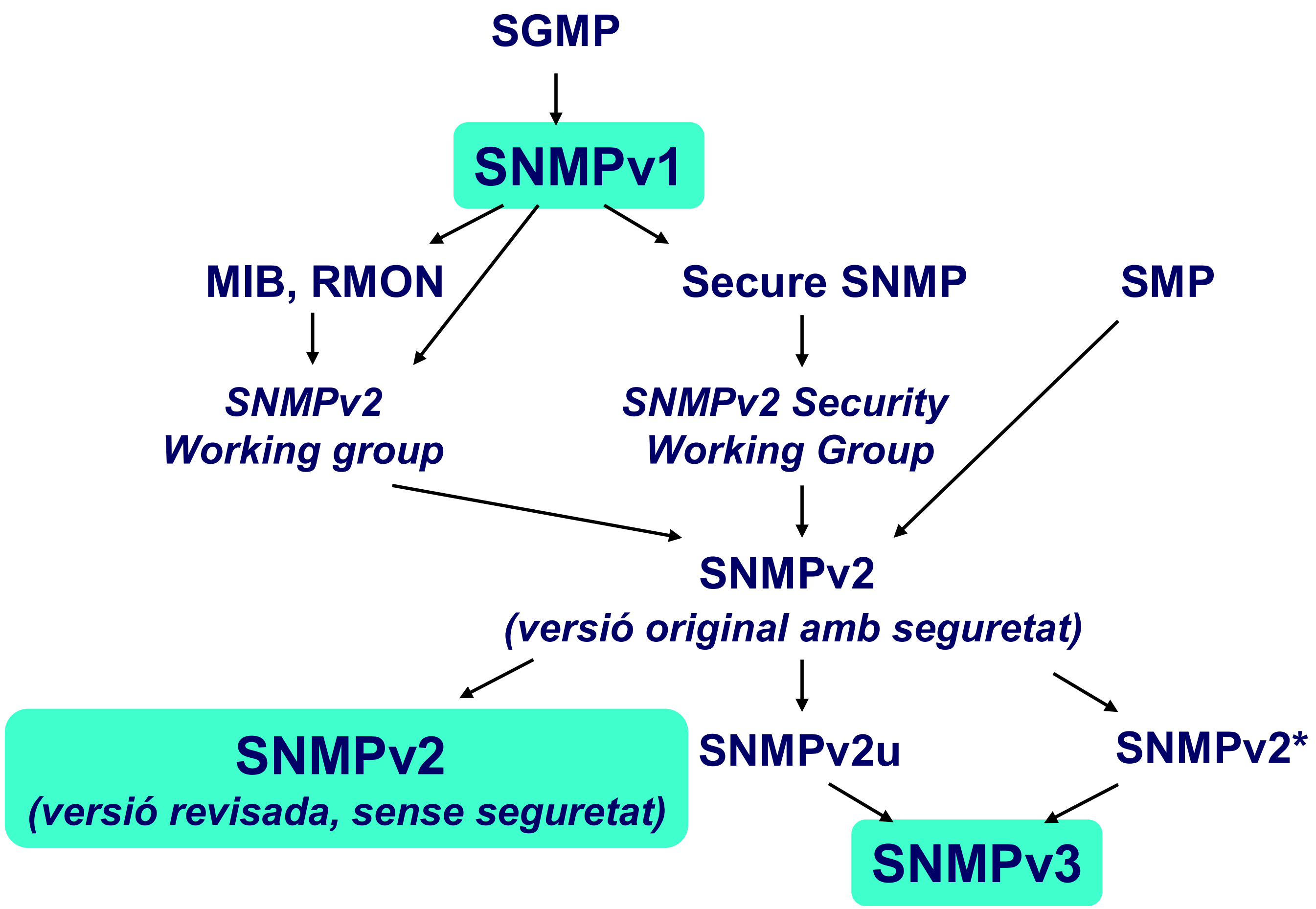
6.8.6. SNMP Protocol

Through the network management protocol entities communicate over the network



6.8.6. SNMP Protocol

Evolution of SNMP Protocol



6.8.7. SNMP Protocol v1

SNMP version 1 see on 5 RFCs collection

RFC 1155: Structure and Identification of Management Information for TCP/IP-base Interfaces

RFC 1157: A Simple Network Management Protocol (SNMP)

RFC 1212: Concise MIB definitions

RFC 1213: Management Information Base for Netware Management of TCP/IP-base Interfaces: MIB-II

RFC 1643: definition of Managed Objects for the Ethernet-likeInterface Types

6.8.7. SNMP Protocol v1

SNMP needs transport-level services to send messages

Transport protocols supported

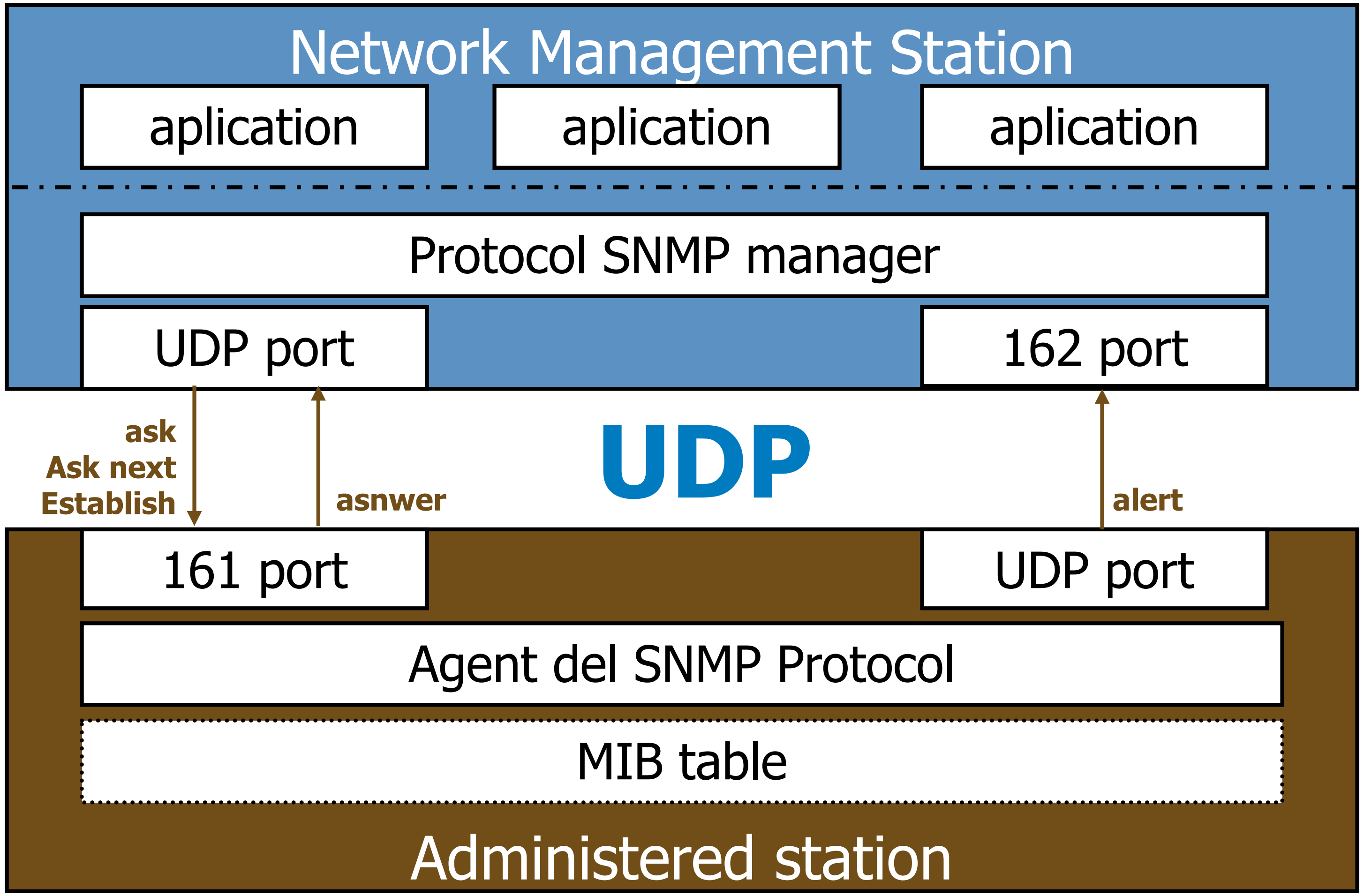
- **UDP**: User Datagram Protocol architecture TCP/IP
- **CLTS**: ConnectionLes transport Services architecture OSI

It is designed to run on transport protocols connection-oriented

- The administrator can keep an open connection to anticipate the agent.
- The agent can close a connection if you need resources

6.8.7. SNMP Protocol v1

Access Transport Layer



6.8.7. SNMP Protocol v1

UDP does not guarantee delivery of messages

If a message is lost ...

GetRequest or GetNextRequest ⇨ messages are repeated

SetRequest ⇨ first checks whether the operation was a GetRequest and if necessary the message is repeated SetRequest

Trap ⇨ **You can not know** ⇨ the administrator must periodically contact the agents (**polling**)

6.8.7. SNMP Protocol v1

Being a distributed system need to group computers belonging to environment management: SNMP communities

The concept of community can offer services:

Authentication: *the agent can limit access to authorized communities MIB names*

Access policies: *the agent can assign different privileges to each administrator*

Proxy: an agent can act as proxy agents of another community

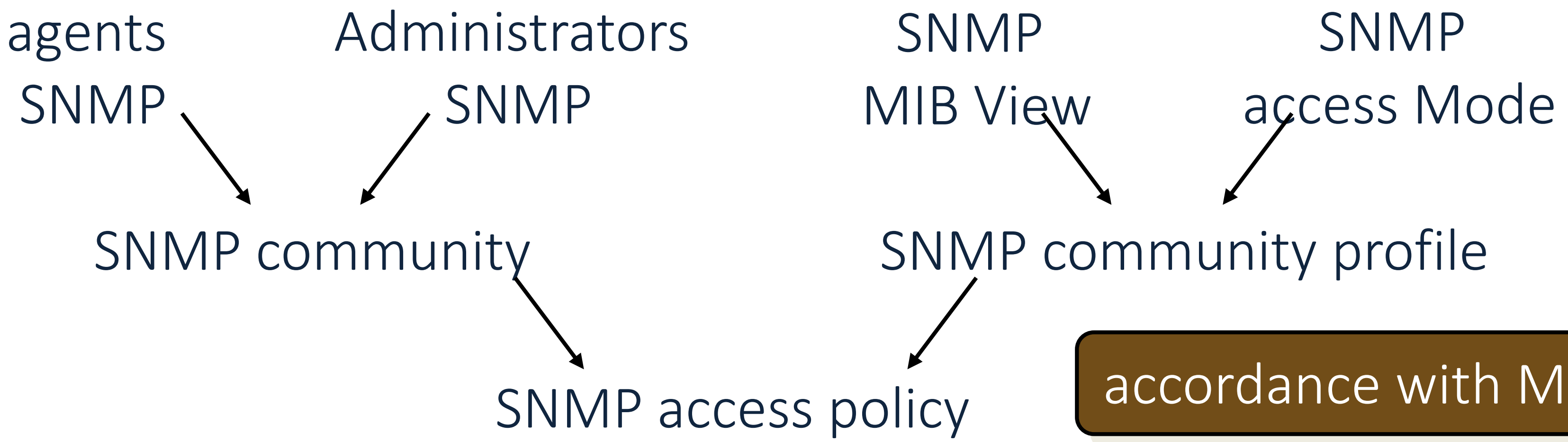
6.8.7. SNMP Protocol v1

Through policies access privileges on objects are managed communities

It takes practice by two concepts:

Views: selections of objects of MIBs

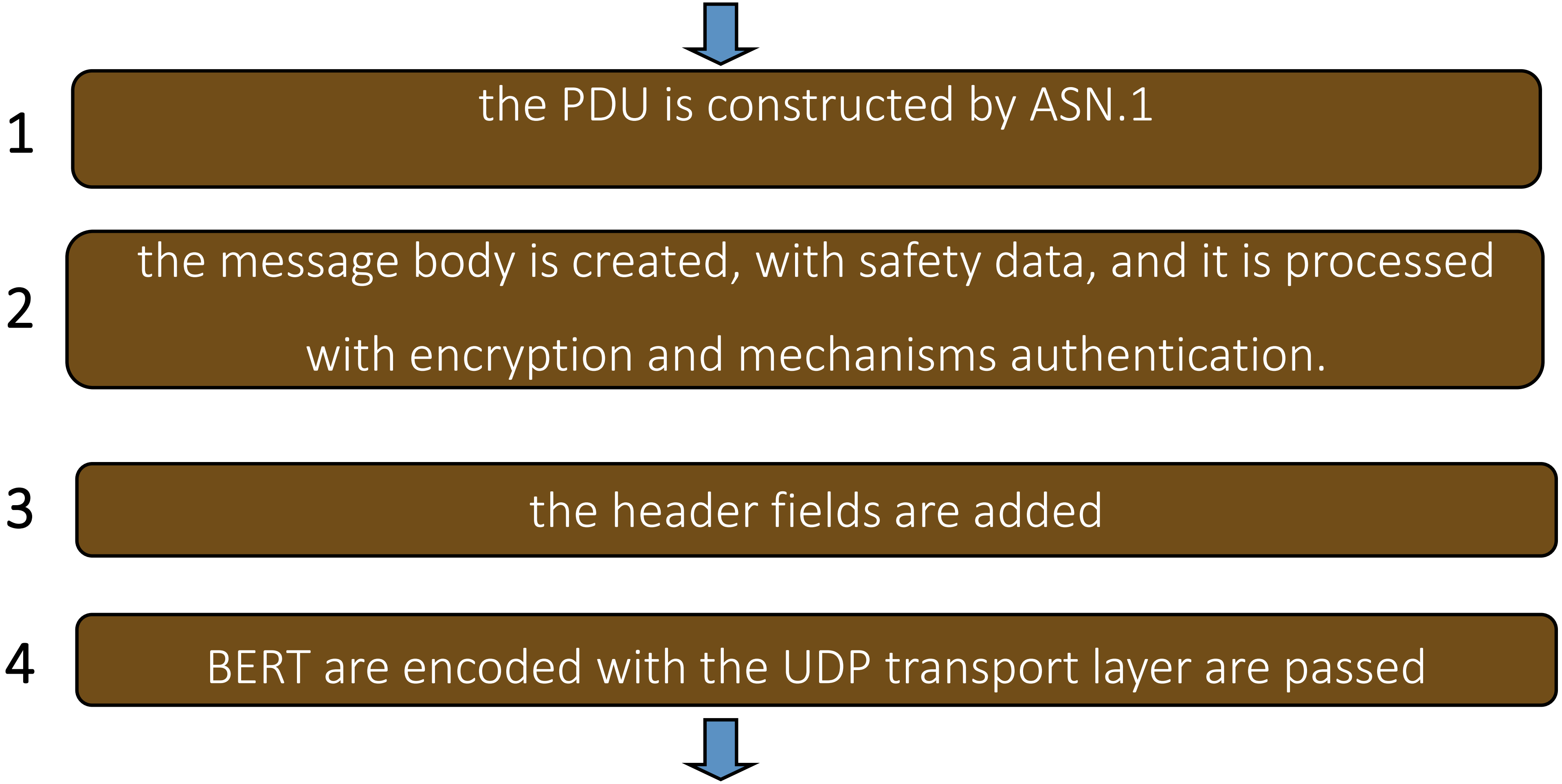
Access Mode: privileges (READ-ONLY, READ-WRITE)



accordance with MIB access !!

6.8.7. SNMP Protocol v1

Transmission sequence SNMP messages



1 the PDU is constructed by ASN.1

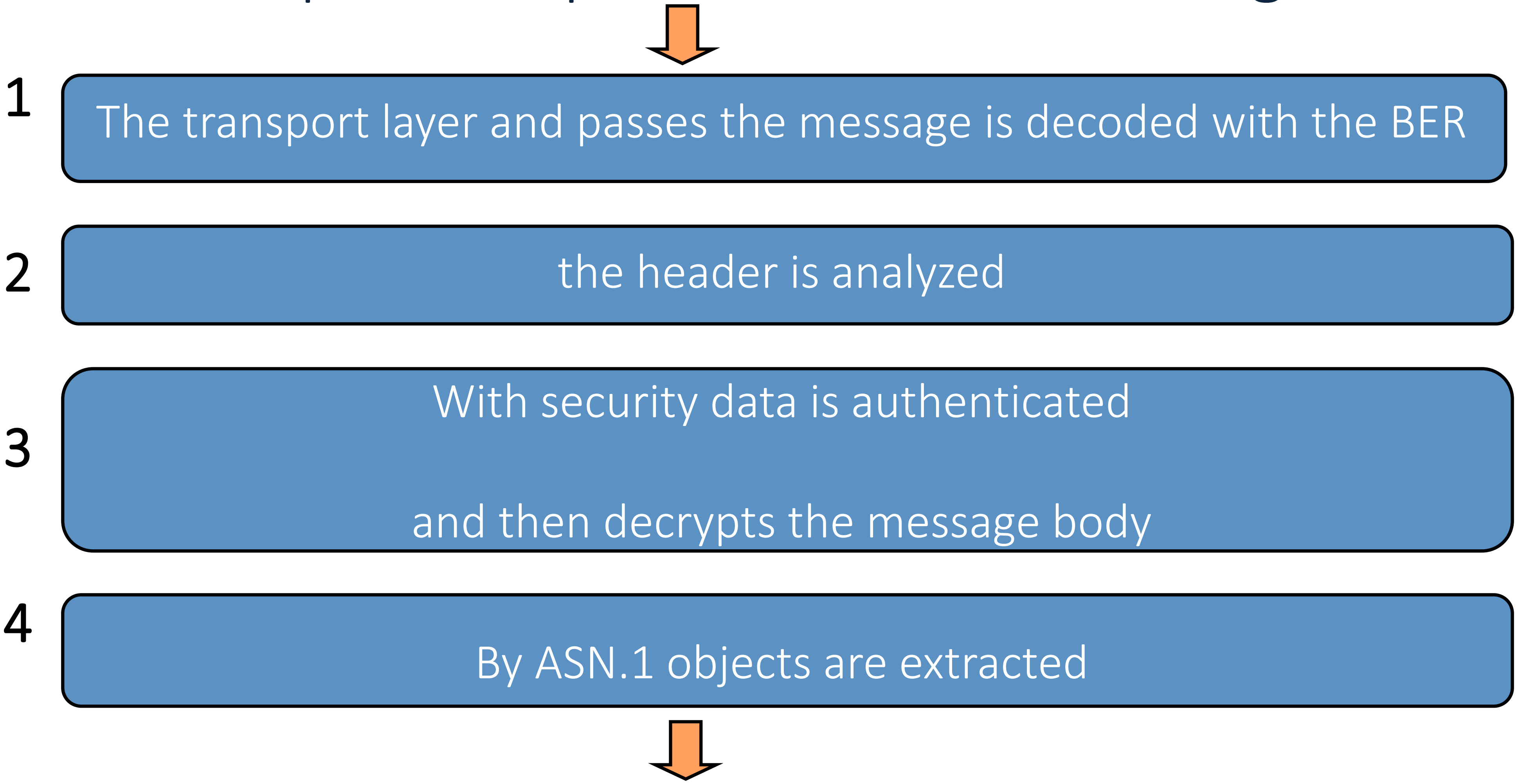
2 the message body is created, with safety data, and it is processed with encryption and mechanisms authentication.

3 the header fields are added

4 BERT are encoded with the UDP transport layer are passed

6.8.7. SNMP Protocol v1

Reception sequence of SNMP messages



1 The transport layer and passes the message is decoded with the BER

2 the header is analyzed

3 With security data is authenticated

and then decrypts the message body

4 By ASN.1 objects are extracted

6.8.7. SNMP Protocol v1

SNMP v1 type messages

The SNMP specifies 5 types of messages:

Get Request: the manager asks the value of variables MIB agent

Get next Request: it does not take the name of the variable

Set Request: the manager asks for changes in values of variables of the MIB agent

Get Response: agent answers the manager to the above commands. It contains the original message but the response

6.8.7. SNMP Protocol v1

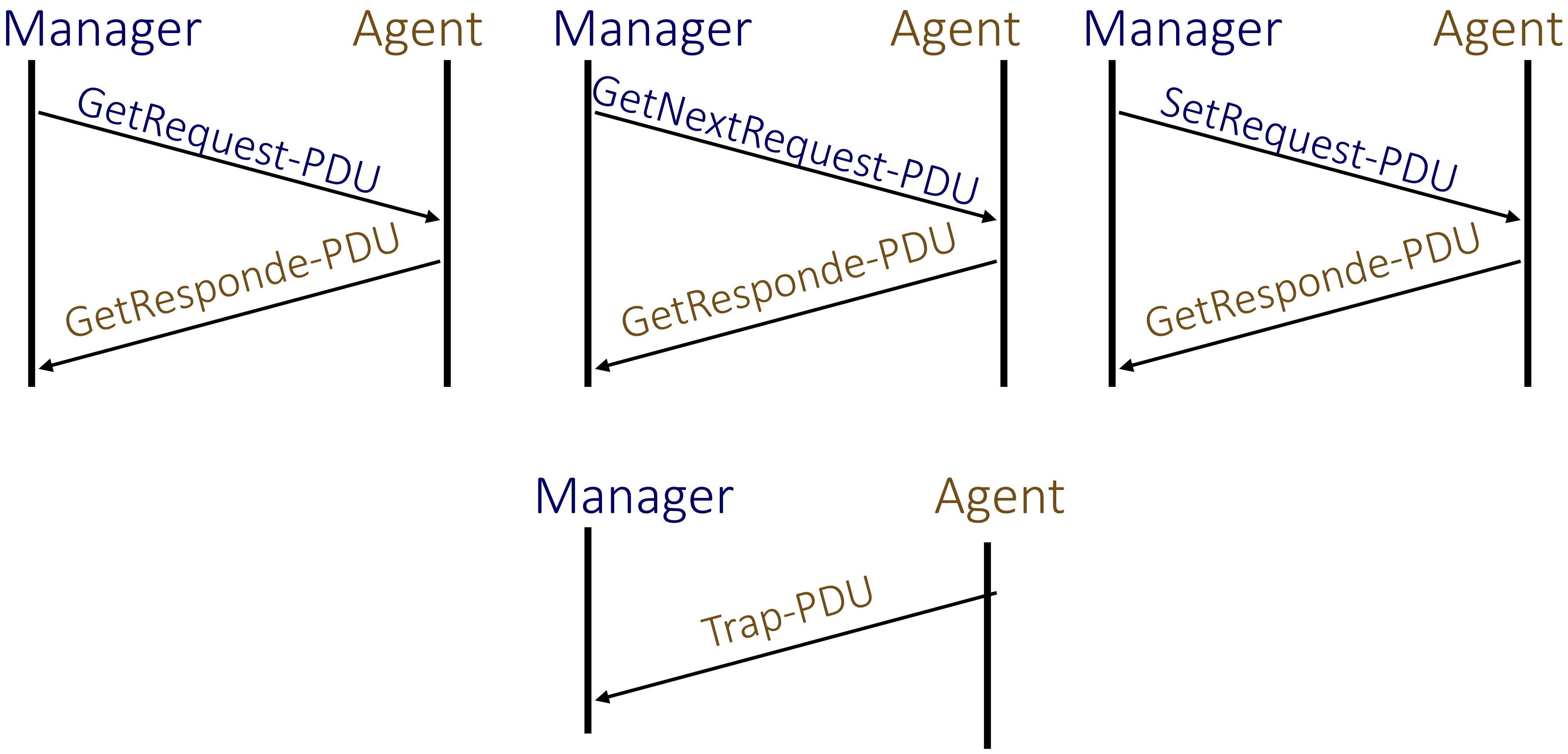
SNMP v1 type messages

Trap: unsolicited responses. alarms :

- **coldStart:** the agent was initialized by itself
- **warmStart:** the agent has restarted by itself
- **linkDown:** the interface has been disabled
- **linkUp:** interface is activated
- **authentificationFailure:** message received from an agent that does not belong to the community
- **egpNeighborLos:** an EGP is disabled (your IP address is sent)
- **enterpriseSpecific:** specific alarm code

6.8.7. SNMP Protocol v1

Flowchart messages



Manager Agent

GetRequest-PDU

GetResponde-PDU

Manager Agent

GetNextRequest-PDU

GetResponde-PDU

Manager Agent

SetRequest-PDU

GetResponde-PDU

Manager Agent

Trap-PDU

6.8.7. SNMP Protocol v1

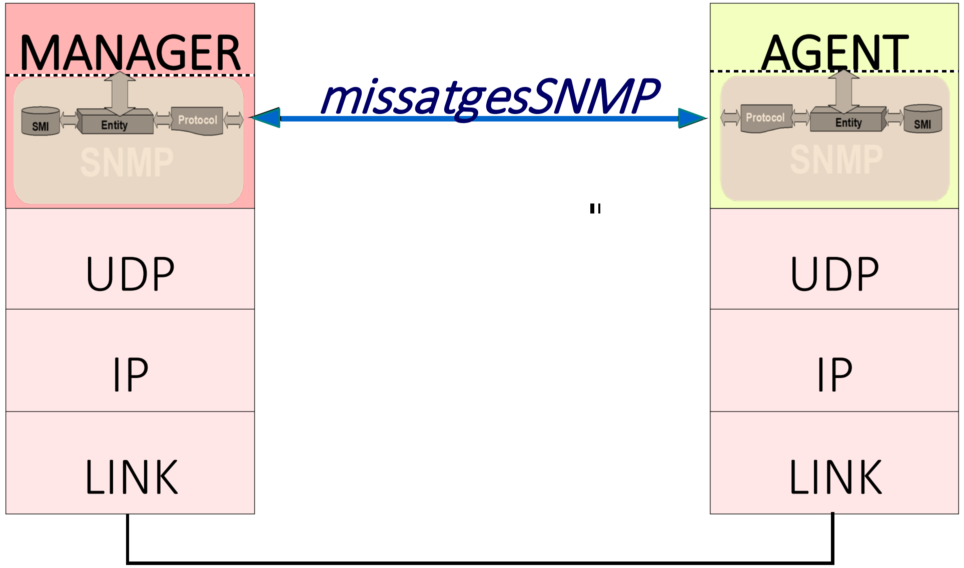
SNMP messages fields v1

Protocol Version: 0 by v1, v2 1 and 3 by v3

Community name: it is used as a password

Message ID: indicates the type of message

Request ID: it is used to relate requests and responses



6.8.7. SNMP Protocol v1

SMNP messages fields v1

Status error: 0 for requests. A answers a different value of 0 indicates an error

- noError (0): no error
- tooBig (1): the result does not fit in the response PDU
- noSuchName (2): object does not exist
- badValue (3): incorrect object value (response to a set)
- readOnly (4): reading object names (response to a set)
- genError (5): Error Cause Unknown

Index error: 0 to requests. In their responses they indicate the variable that has caused problems

6.8.7. SNMP Protocol v1

Fields SMP v1 Posts

Object Identifier: *type of the object that generated the error (sysObjectID)*

Agent address: *address of the object that generated the alarm*

Alarm ID: *alarm type indicator*

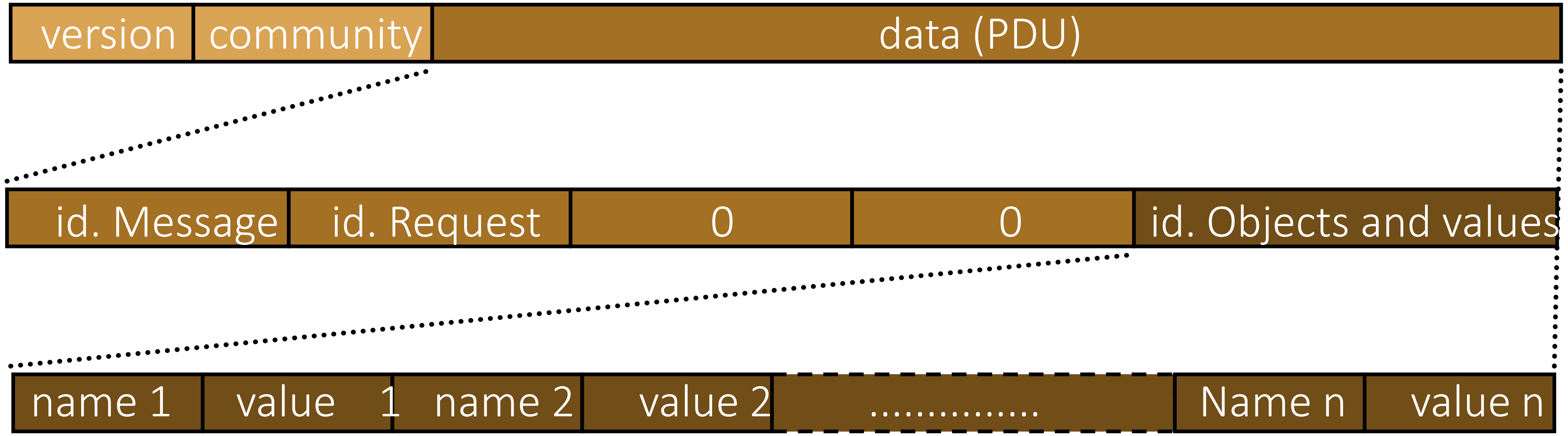
ID alarm code: *alarm code*

Timestamp: *time elapsed between the last reset of device and the current alarm*

ID objects and values: variables and orders / answered values. There are so many variables are requested as / respond

6.8.7. SNMP Protocol v1

Messages Get Request, Get Next Request y Set Request



version	community	data (PDU)

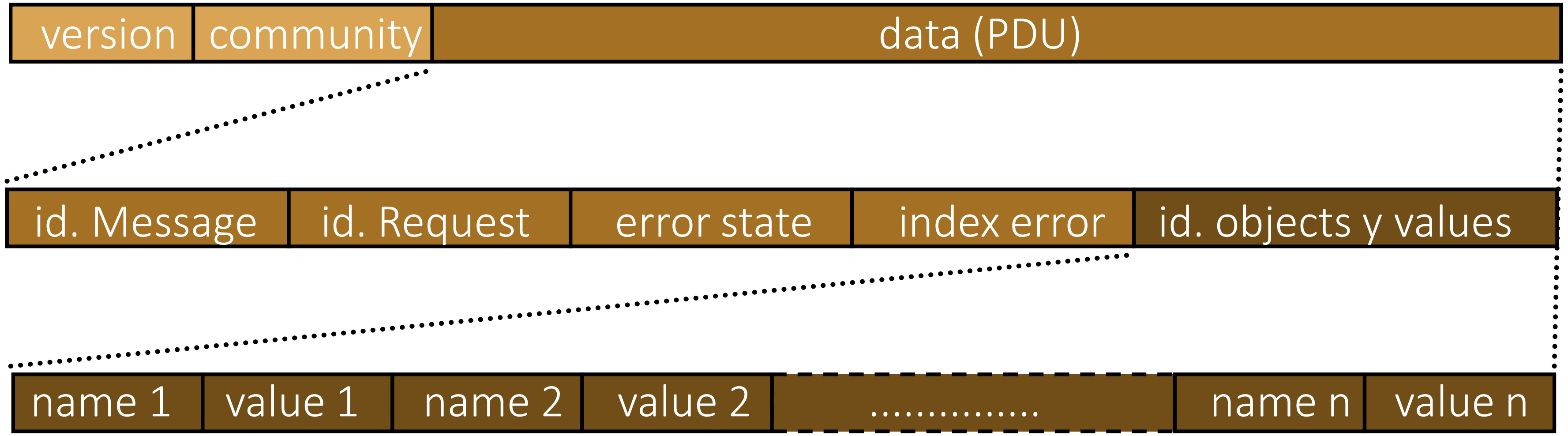
id. Message	id. Request	0	0	id. Objects and values

name 1	value 1	name 2	value 2	....	Name n	value n

PDU= Protocol Data Unit

6.8.7. SNMP Protocol v1

Messages Get Response



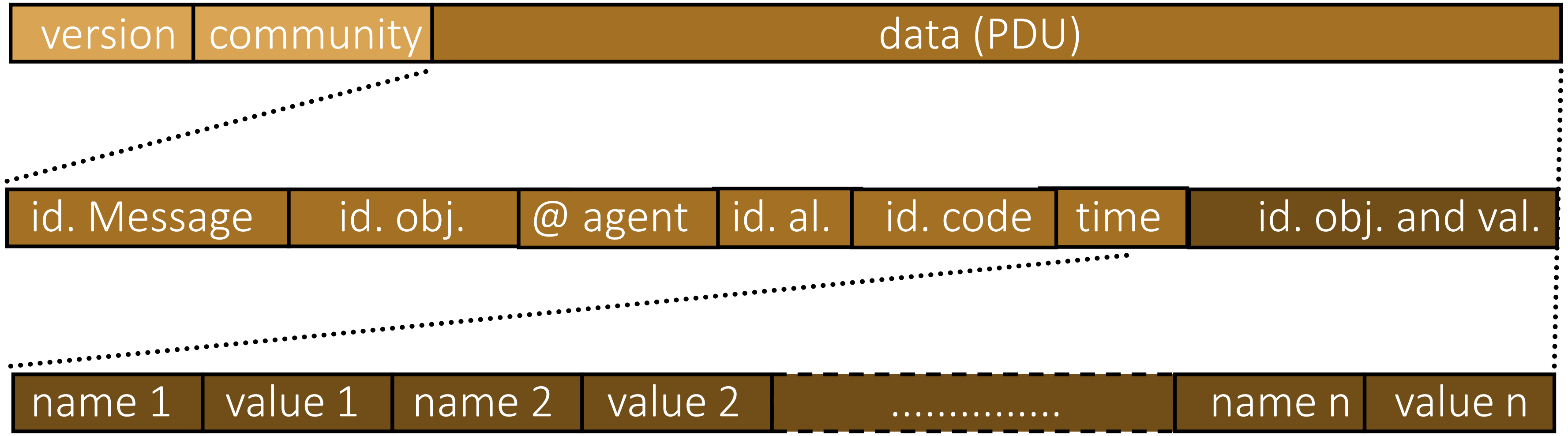
version | community | data (PDU)

id. Message | id. Request | error state | index error | id. objects y values

name 1 | value 1 | name 2 | value 2 | .... | name n | value n

6.8.7. Protocol SNMP v1

Messages Trap



version	community	data (PDU)

id. Message	id. obj.	@ agent	id. al.	id. code	time	id. obj. and val.

name 1	value 1	name 2	value 2	....	name n	value n

6.8.8. SNMP Protocol v2

SNMP version 2 is collected in 8 RFCs

RFC 1901: Introduction to Community-based SNMPv2

RFC 1902: Structure of Mangement Information for SNMPv2

RFC 1903: Textual Conventions for SNMPv2

RFC 1904: Conformance Statements for SNMPv2

RFC 1905: Protocol Operations for SNMPv2

RFC 1906: Transport Mappings for SNMPv2

RFC 1907: Management Information Base for SNMPv2

RFC 1908: Coexistence between Version 1 and Version 2 of Internet-Standard Network Management Framework

6.8.8. SNMP Protocol v2

Version 2 includes improvements SNMP v1 respect in 4 large blocks

Scope: grasps the concept of SMP framework of the SMP (Simple Management Protocol)

Size, speed and efficiency: new messages are incorporated

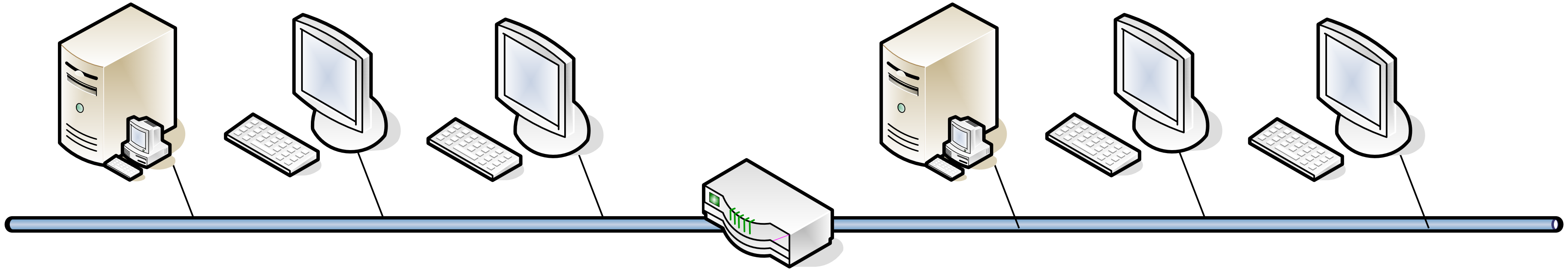
Security and privacy: security features Secure SNMP are incorporated into the original version, which are drawn to the revised version

Development and compatibility: based on the SNP is increased compatibility with new communications architectures OSI

6.8.8. SNMP Protocol v2

In the scope of 3 types of access it is proposed to MIBs

- *Manager-agent request-response*: As in the v1, the administrator can obtain and modify the agent MIBs
- *Manager-Manager request-response*: new area where two administrators can share an agent MIBs
- *Agent-Manager unconfirmed*: As in the v1, the agent can send alerts to the administrator



6.8.8. SNMP Protocol v2

Development in the field of transport architectures increase

SNMPv2 can use transport services protocols:

UDP: User Datagram Protocol

CLNS: OSI ConnectionLes_Mode Network Service

CONS: OSI Connection-Oriented Network Service

IPX: Novell Internetwork Packet Exchange

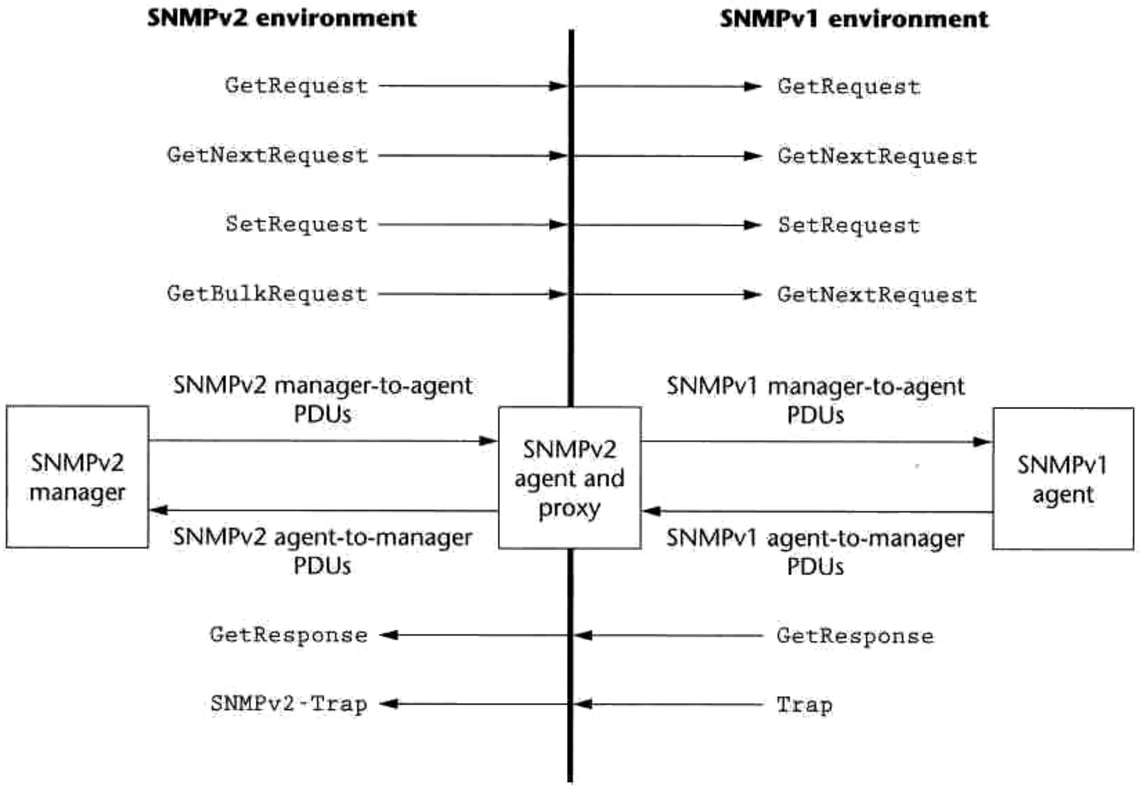
DDP (Appletalk): Apple Network

6.8.8. SNMP Protocol v2

In the field of compatibility is defined as can coexist version v1 and v2

There are 2 alternatives:

- Include a **proxy** that link the two **communities**
- *Two readers entities* message



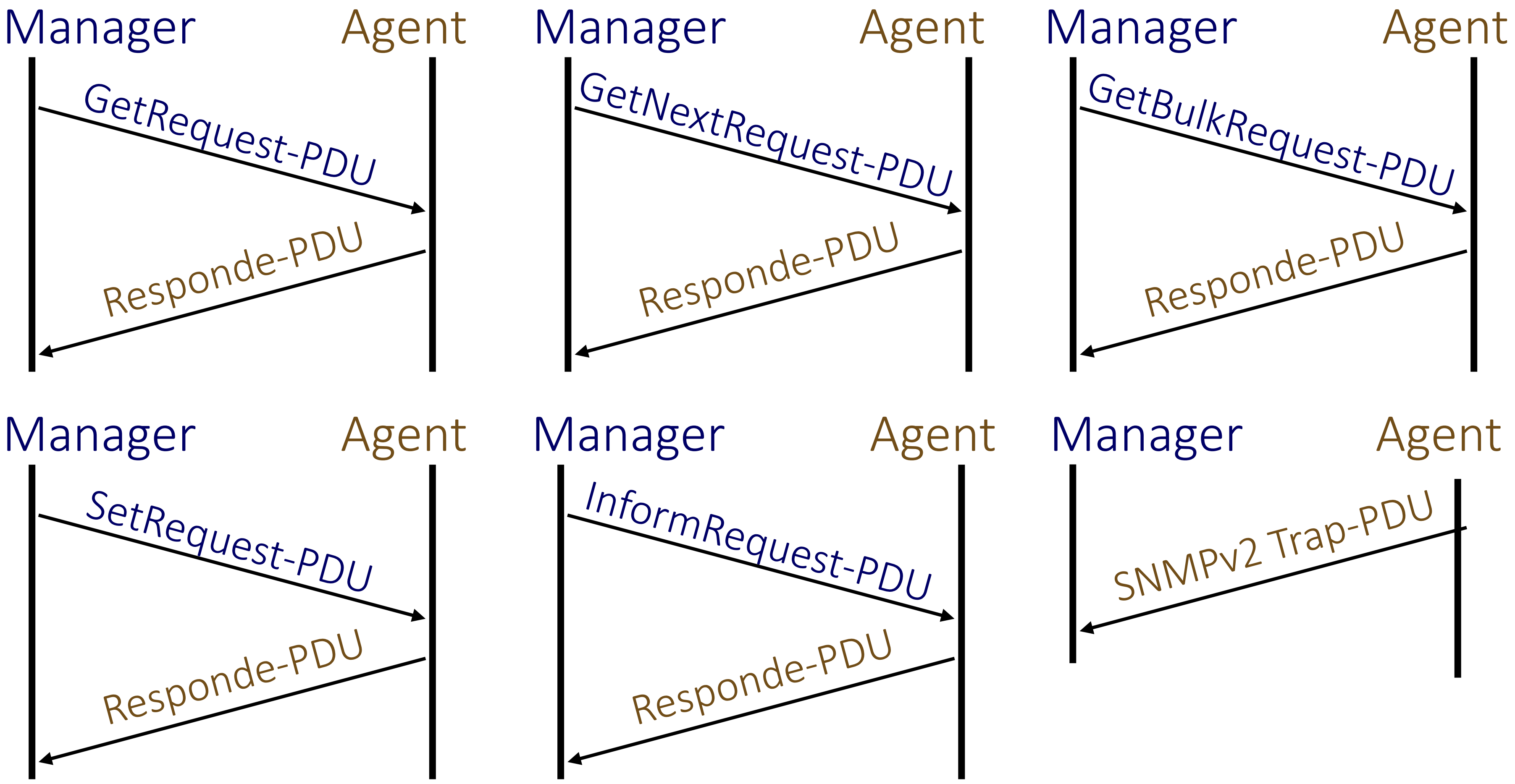
Message type SNMP v2

The SNMP specify 7 types of messages:

- **Get Request:** the value of the MIB variables are asked
- **Get Next Request:** You do not need the name of the variable
- **Get Next Bulk**: as get next request, but allowed to ask more than one variable to see and indicate repetitions
- **Set Request:** changes called for in the MIB variables
- **Response:** answers to the above commands
- **Inform Request**: information between managers
- **Trap:** alarms, follows the NOTIFICATION-TYPE macro
- **Report**: undefined

6.8.8. SNMP Protocol v2

Flowchart messages



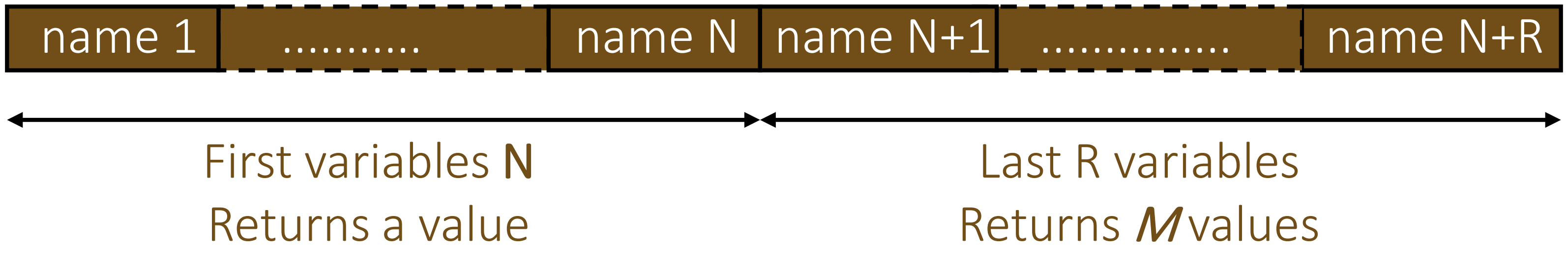
6.8.8. SNMP Protocol v2

Message fields SMNP v2

In the fields described in version 1 must be added the following two:

No repeat: number of variables without repeating (N)

Max iter: maximum number of repetitions (M)



6.8.8. SNMP Protocol v2

Message fields SNMP v2

The field error can take the following values:

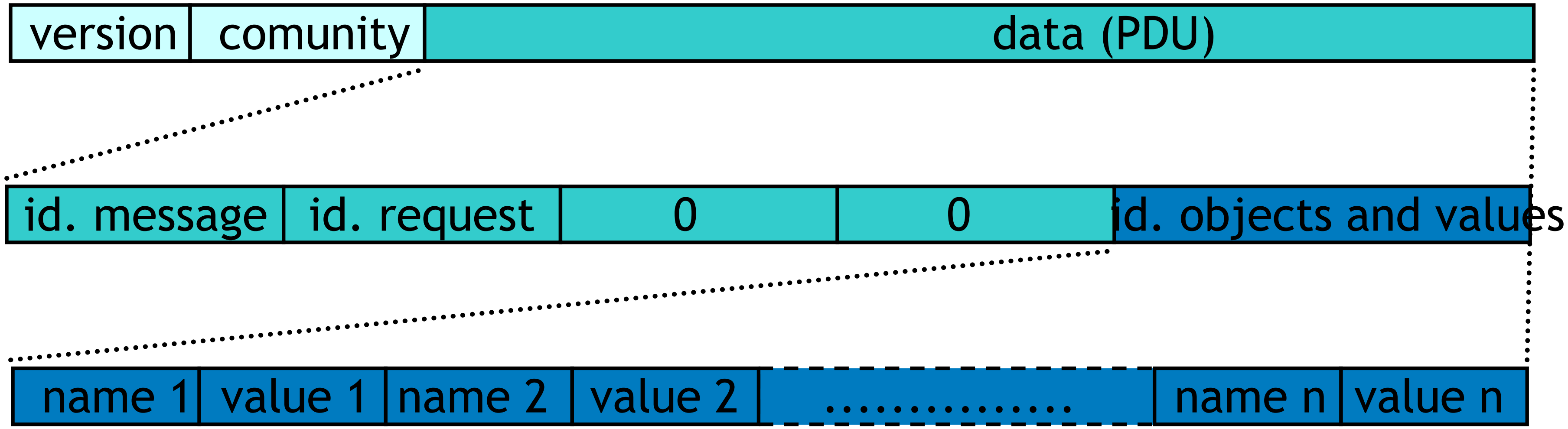
	(1)	(2)	(3)
noError(0)	X	X	X
tooBig(1)	X	X	X
noSuchName(2)			
badValue(3)			
readOnly(4)			
genError(5)	X	X	X
noAccess(6)		X	
wrongType(7)		X	
wrongLength(8)		X	
wrongEncoding(9)		X	

	(1)	(2)	(3)
wrongValue(10)		x	
noCreation(11)		X	
InconsistentValue(l2)		X	
resourceUnavailable(13)		X	
commitFailed(14)		x	
undoFailed(15)		X	
authorizationError(16)	X		
notWritable(17)		X	
inconsistentName (18)		X	

(1) GetRequest, GetNextRequest, GetBulkRequest (2) SetRequest (3) InformRequest

6.8.8. SNMP Protocol v2

Messages Get Request, Get Next Request, Set Request, Trap y Inform Request



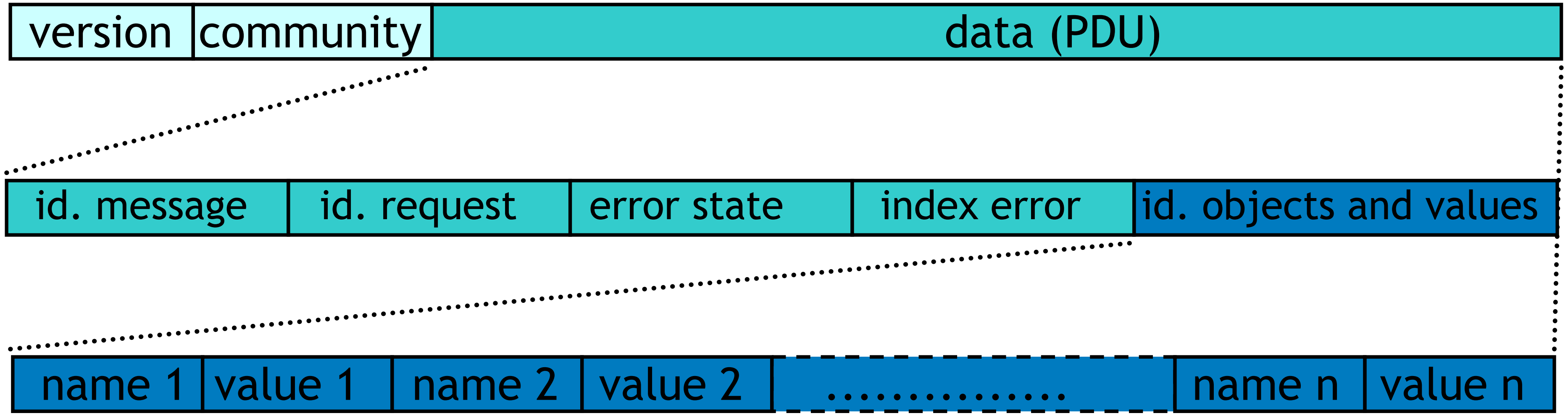
version	comunity	data (PDU)

id. message	id. request	0	0	id. objects and values

name 1	value 1	name 2	value 2	…………….	name n	value n

6.8.8. SNMP Protocol v2

Messages Response

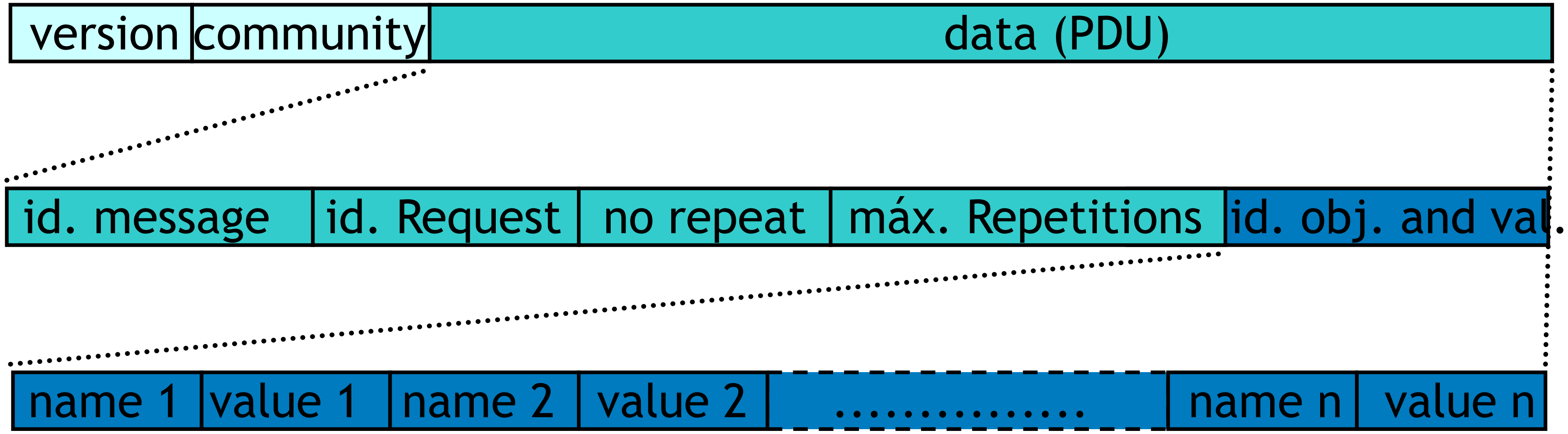


version | community | data (PDU)

id. message | id. request | error state | index error | id. objects and values

name 1 | value 1 | name 2 | value 2 | .... | name n | value n

Messages Get Bulk Request



6.8.9. SNMP Protocol v3

SNMP version 3 is collected in 6 RFCs

RFC 2271: An Architecture for Describing SNMP Management Frameworks

RFC 2272: Message Processing and Dispatching for SNMP

RFC 2273: SNMPv3 Applications

RFC 2274: Use-based Security Model for SNMPv3

RFC 2275: View Acces Control Model (VACM) for SNMP

Internet Draft: Introduction to Version 3 of the Internet Network Management Framework

6.8.9. SNMP Protocol v3

Version 3 incorporates improvements in 3 large blocks

Security:

- USM: User-Based Security Model
- VACM View-Based access Model

New structure of messages

- There are the same messages in SNMPv2
- Encapsulation is applied to implement safety skills

MIBs are expanded

- new objects and structures are defined

6.8.9. SNMP Protocol v3

The User-based Security Model provides authentication services i encrypted

Protects to:

- Impersonation
- Amendments to information
- Alteration of the message flow (reordering, duplicate, ...)
- Information revealed

Don’t protect to:

- Denial of Service
- Analysis of message traffic

6.8.9. SNMP Protocol v3

The View-Based Access Control Model provides access control services to the MIBs (RFC2275)

It has two main functions

- Determine what type of access has a remote entity to a local object.
- Determine the access control policy by the Agent and enabled for remote configuration

It is based on 5 elements

- Groups: groups of associations <securityModel, securityName>
- Security level: rights to a group depends on the security level of the message
- Context: subset of objects from a local MIB
- Views of the MIBs: subtree of the local SMI
- Access policy: priority applies a set of access privileges

6.8.9. SNMP Protocol v3

Fields SNMP v3 messages

In the fields described in versions 1 and 2 must be added:

Version: *indicates the version of SNMP version 3 worth 3.*

Message ID: *is used to relate requests and responses between two entities [0, 231-1]*

Max. Size: *indicates the maximum number of bytes that the sender supports the messages [484, 231-1]*

Attributes: *octet string where the three least significant bits mean:*

- ***Report:*** to '1' force PDU sending a report to the sender
- ***Private:*** to '1' the message is encrypted
- ***Autor:*** to '1' the message carries Authentication (P = 0 i A = 1 not allowed)

6.8.9. SNMP Protocol v3

Safety parameters: contains the security parameters USM

Safety parameters: contains the security parameters USM

- **Id authentication engine:** engine parameter snmp authentication EngineID involved in communication.
- **Id boot authentication:** snmpEngineBoots authentication engine parameter involved in communication.
- **Id authentication time:** snmpEngineTime authentication engine parameter involved in communication.
- **User name:** Indicates from whom the message.
- **Authentication parameters:** message authentication code HMAC
- **Privacy parameter:** initial value for the algorithm DES CBC

6.8.9. SNMP Protocol v3

Fields SNMP v3 messages

Security model: indicates the security methods applied [0, 231-1]. They are reserved 1 (SNMPv1), 2 (SNMPv2), 3 (USM)

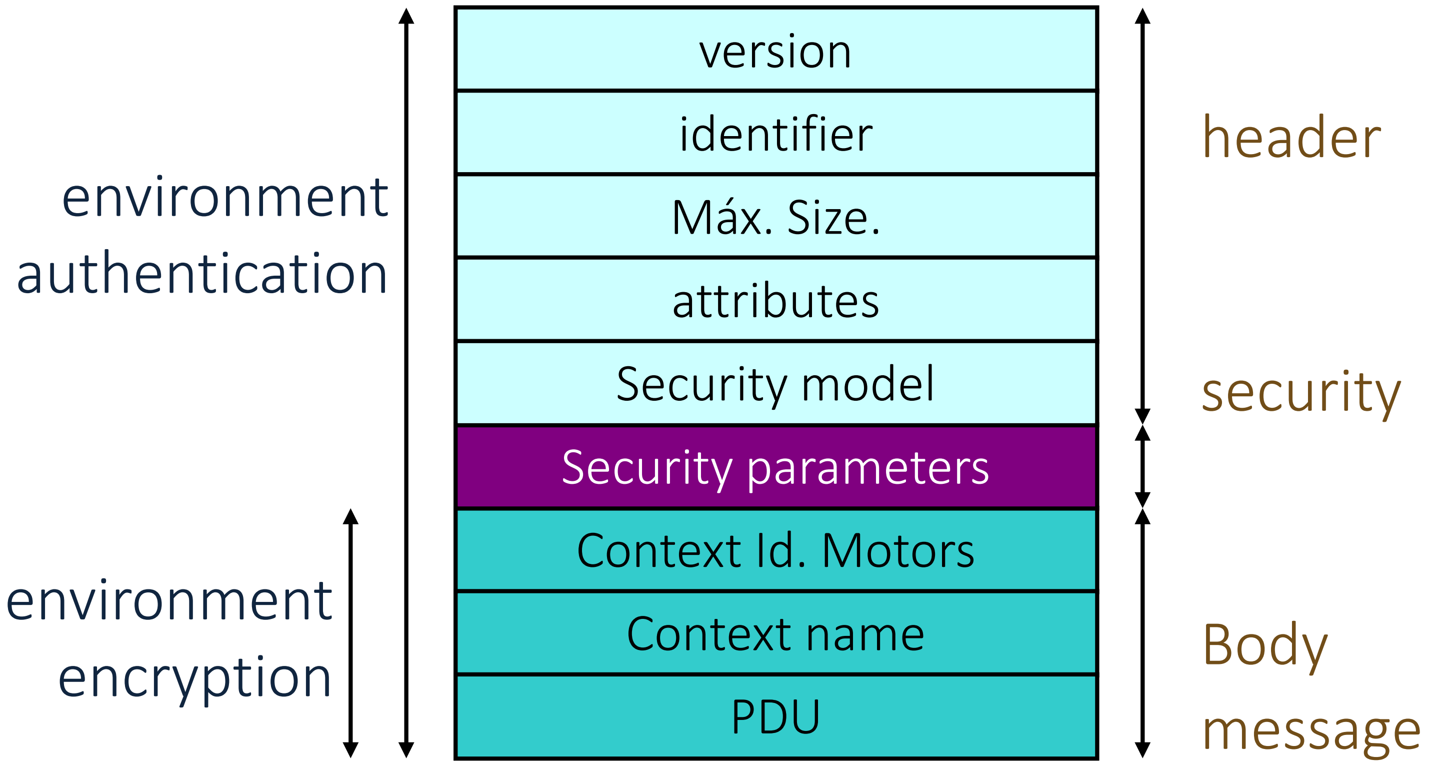
Context Motor Id.: Identifier SNMP entity that generates

Context name: context identifier

PDU: SNMP data Version 2!

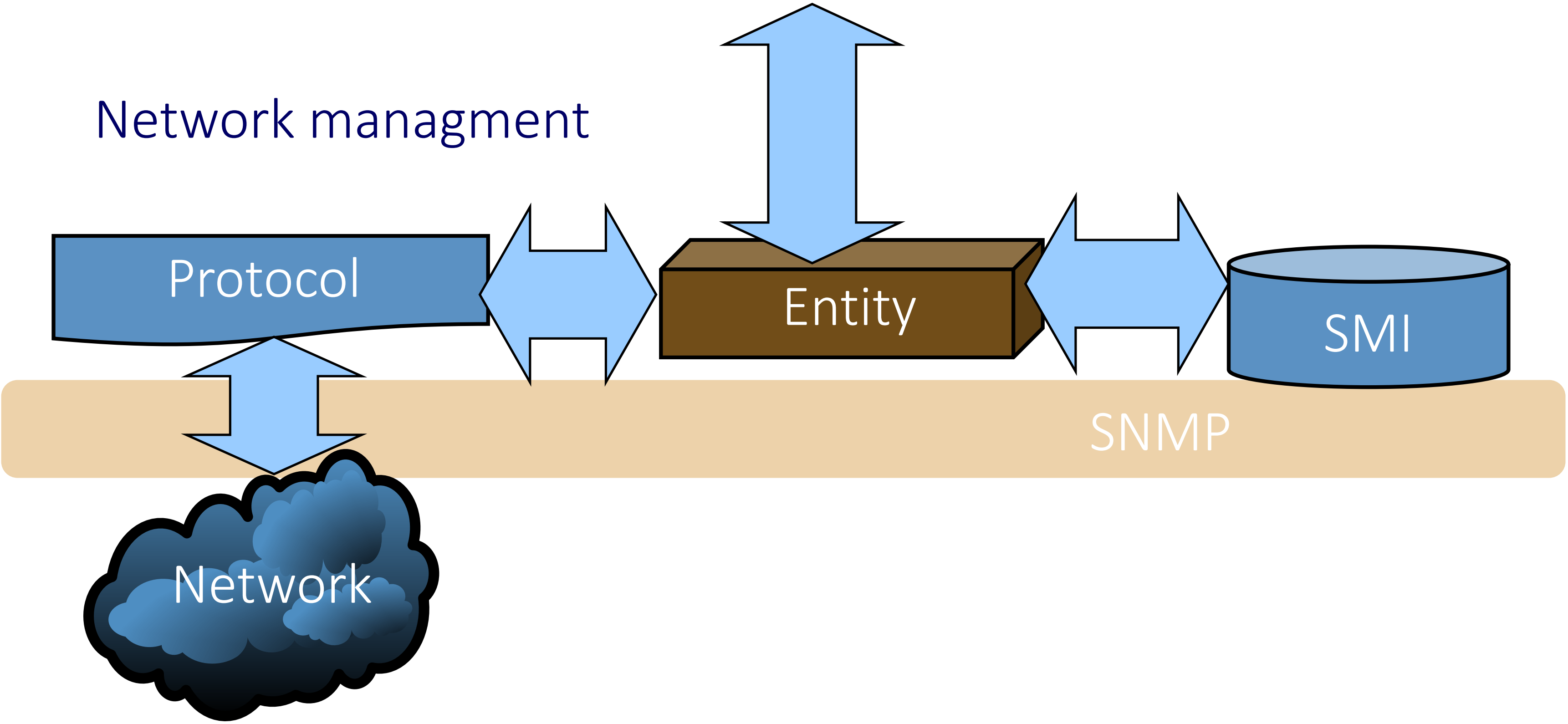
6.8.9. SNMP Protocol v3

SNMP v3 messages



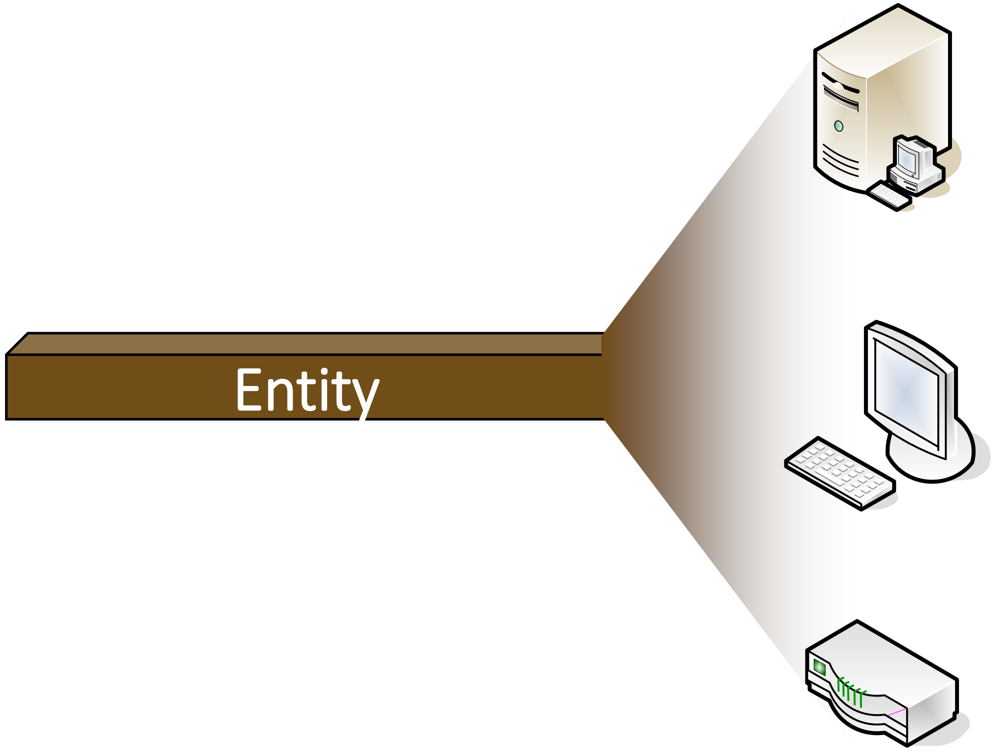
6.8.10. SNMP Entities

SNMP provides a network management environment



6.8.10. SNMP Entities

In the SNMP architecture 3 types of entities are defined

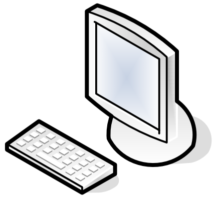


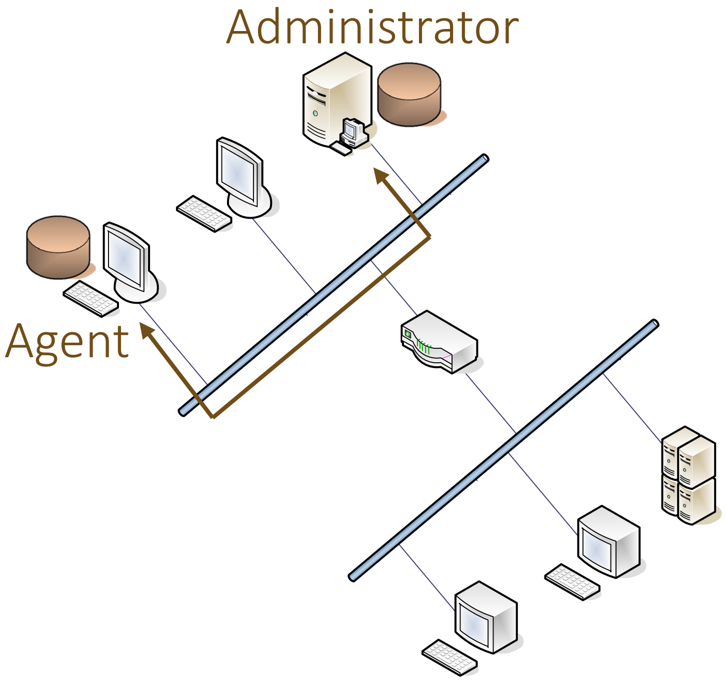
6.8.10. SNMP Entities

Agents

- Software installed on all managed systems

- Tasks
  - ✓Respond to requests for administrators
  - ✓Upgrades
  - ✓Report problems



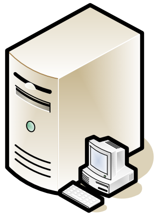


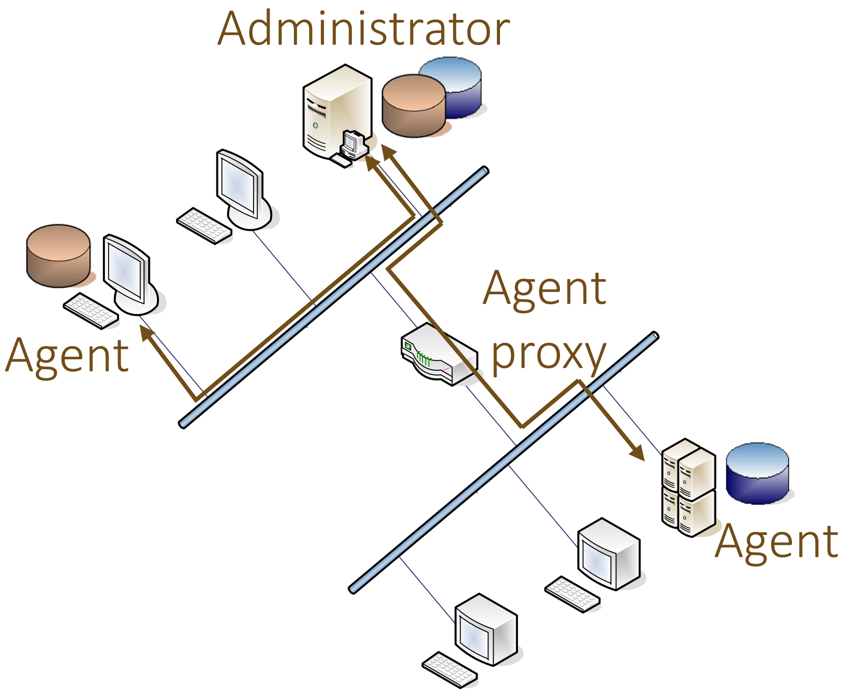
6.8.10. SNMP Entities

Administrators

- •Software installed on all systems that manage network

- Tasks
  - ✓Send and receive SNMP messages to the managed systems
  - ✓Manage network





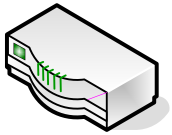
Network Engineering Department

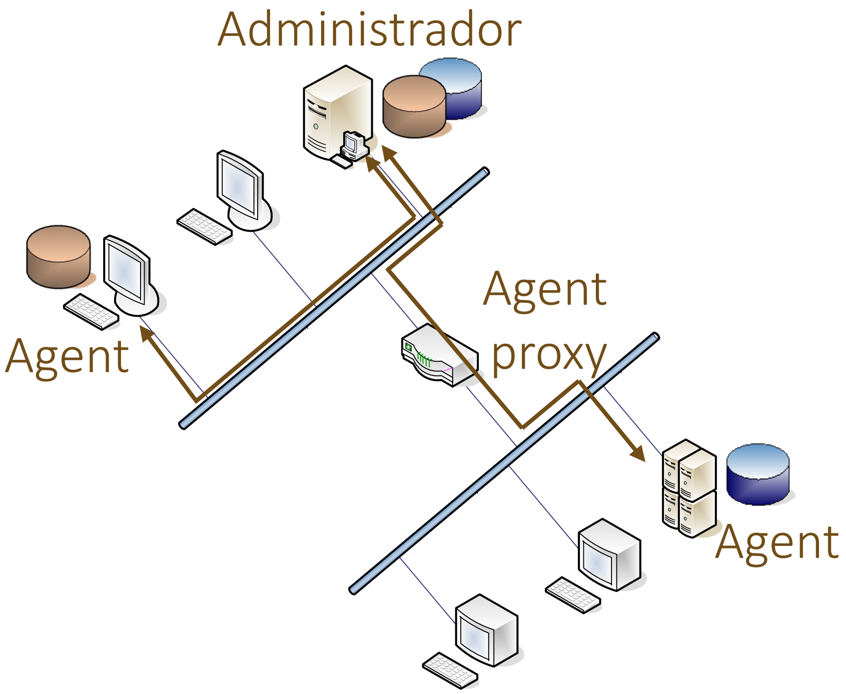
6.8.10. SNMP Entities

Proxy agents

- Software installed on a computer network management

- Tasks
  - ✓Redirect messages but does not interpret SNMP objects
  - ✓Translates between different SNMP versions (v2)
  - ✓Translates between SNMP and other network protocols (v2)

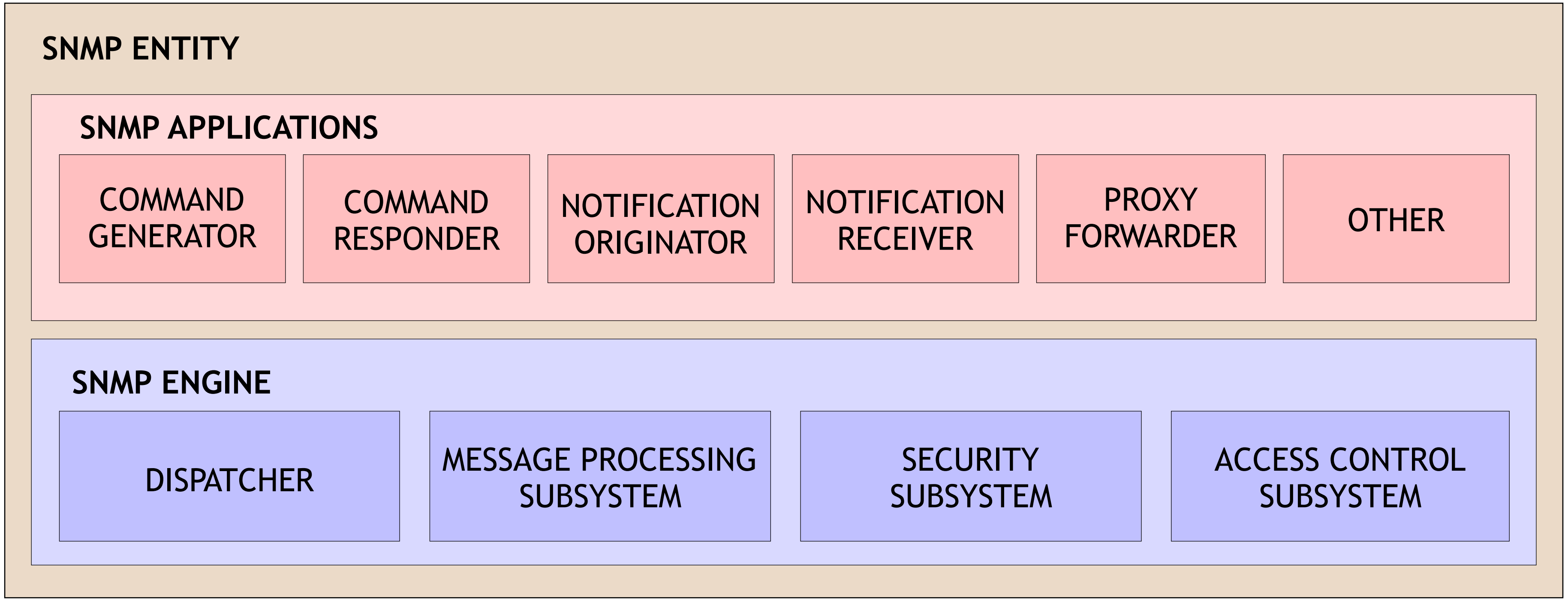




6.8.10. SNMP Entities

General scheme of an SNMP entity

Based in SNMPv3



6.8.10. SNMP Entities

Based in SNMPv3

Motor scheme entity

Dispatcher: Manager of the messages, is responsible for distributing messages both internally and network

Message Processing Subsystem: It is responsible for interpreting messages

Security Subsystem: It is responsible for providing security in the message flow

access Control Subsystem: It is responsible for granting access privileges to objects

6.8.10. SNMP Entities

Scheme applications entity

Based in SNMPv3

Command Generator: creates the request messages and interprets Reply

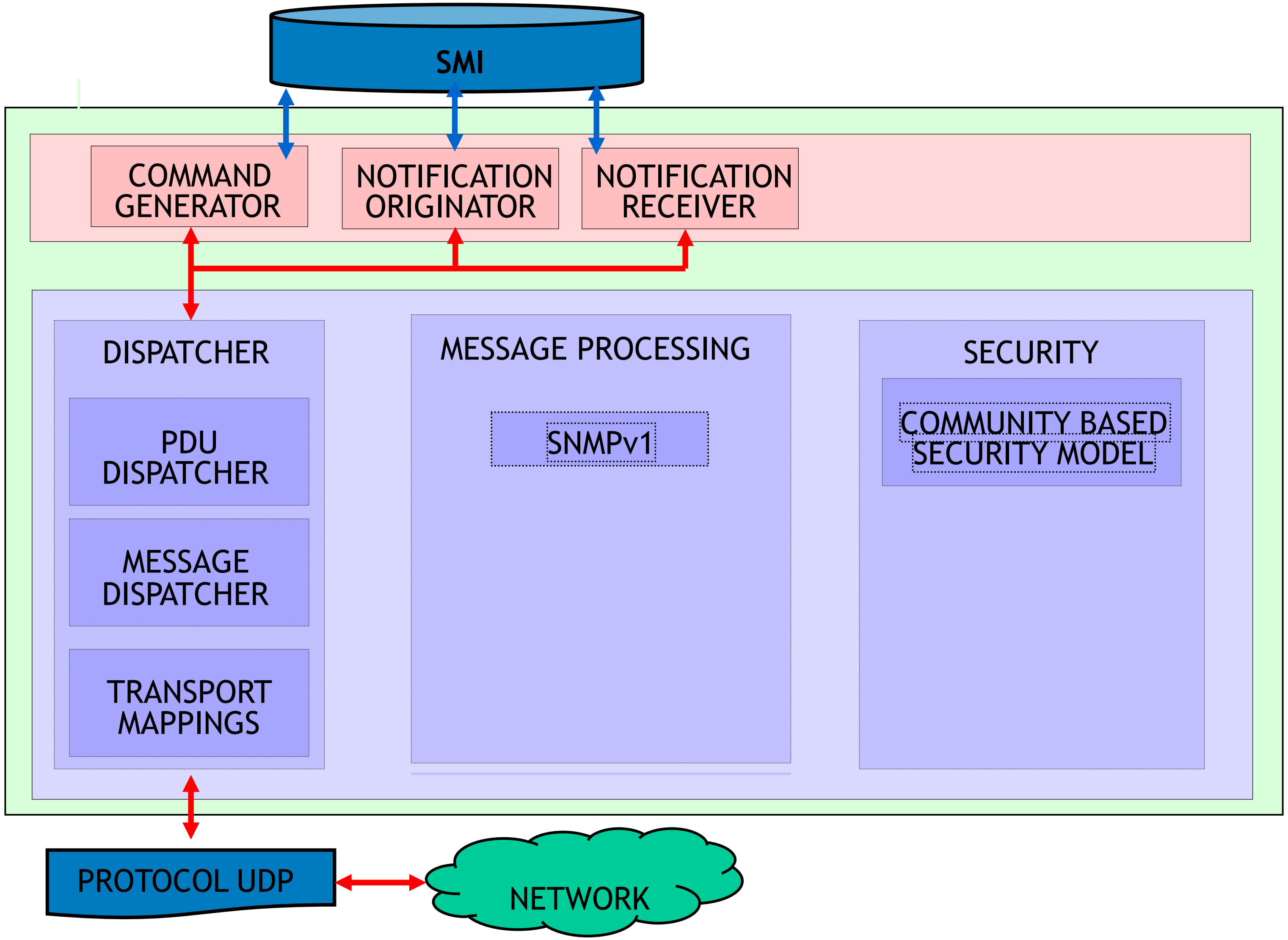
Command Responder: interprets the request message and generates respond

Notification Originator: It monitors the network and generates messages and inform trap

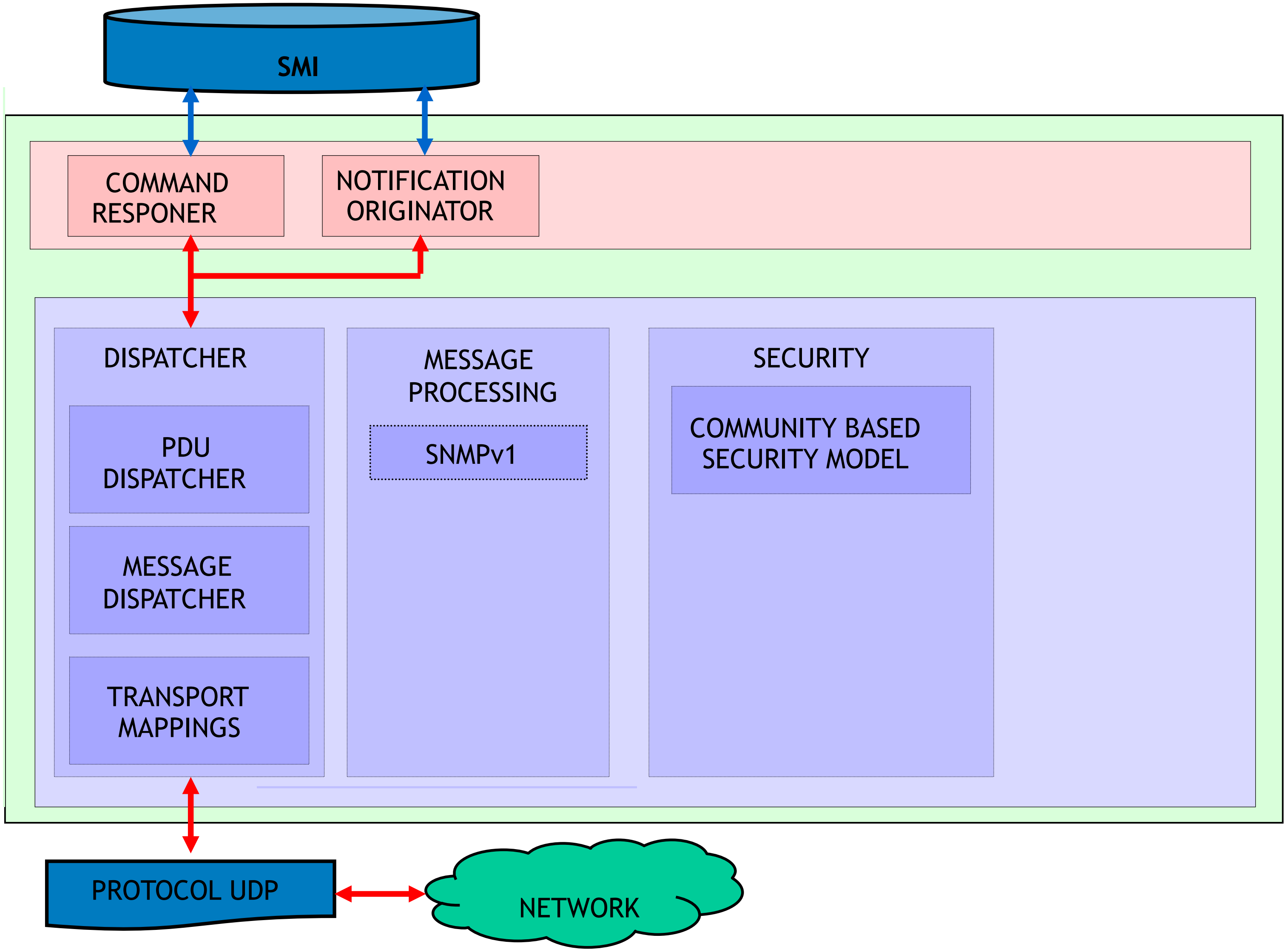
Notification Receiver: interprets messages and inform trap. For reports generated the corresponding response

Proxy Forwarder: redirect SNMP messages

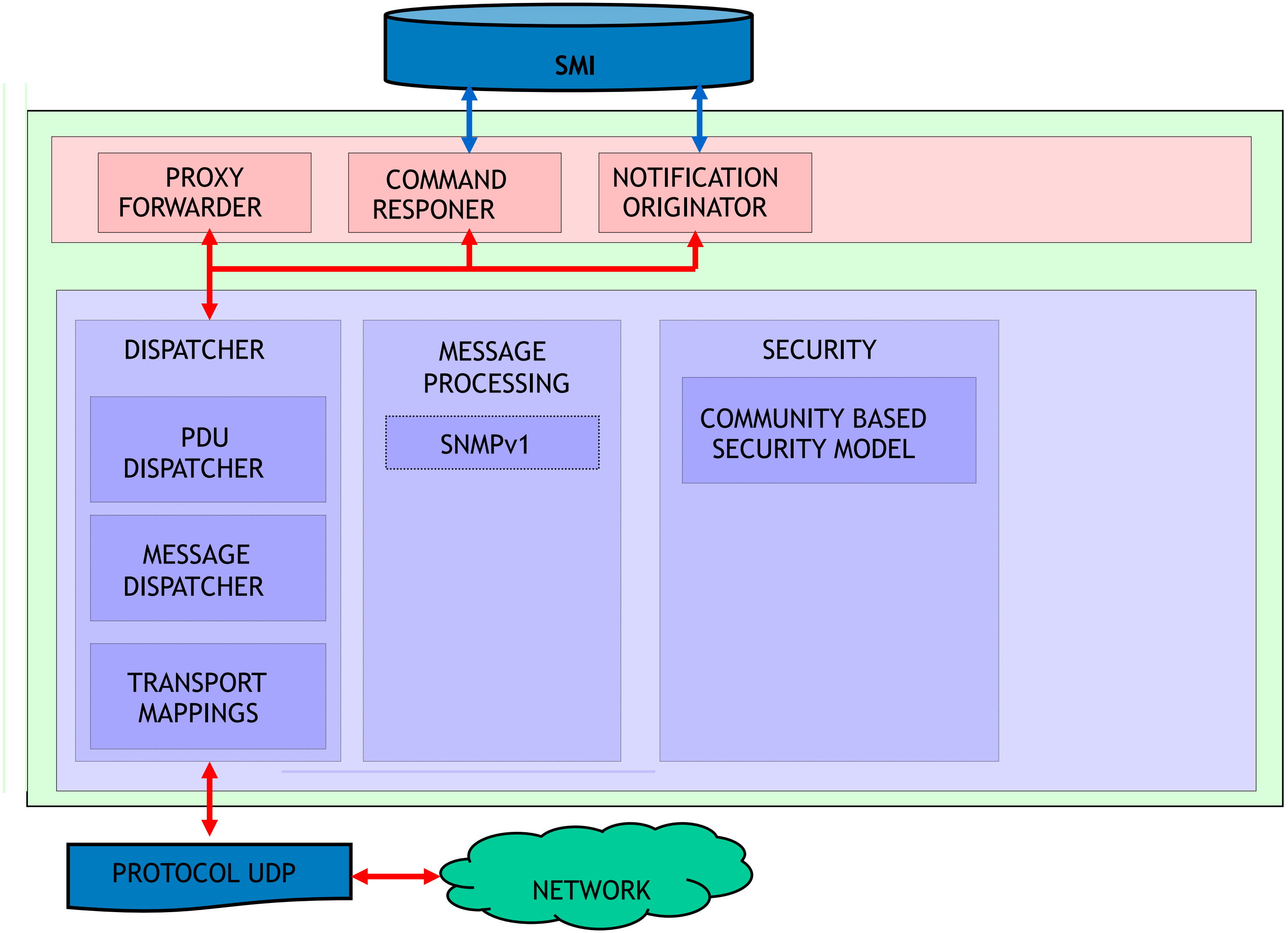
Entity manager



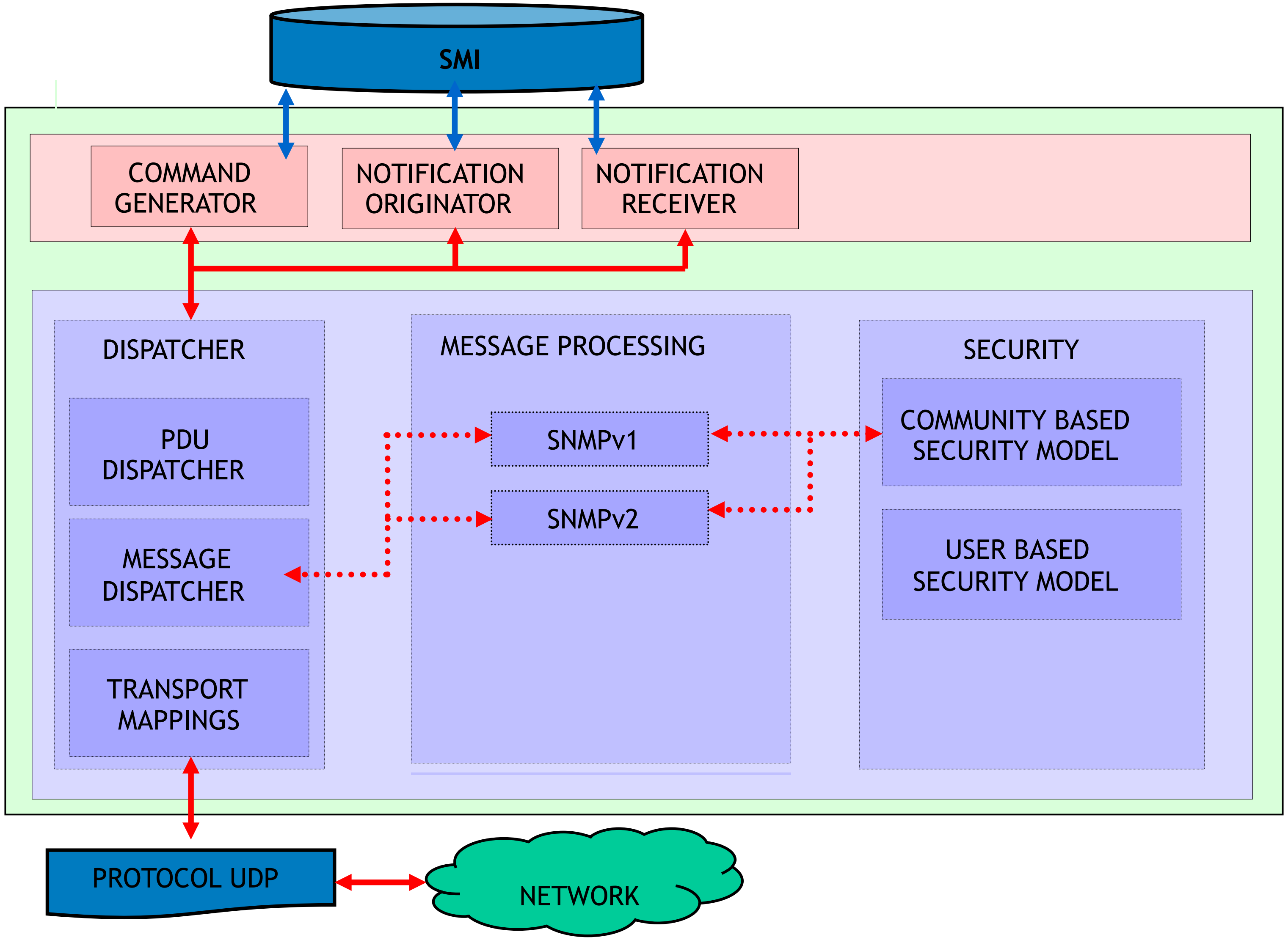
Entity agent



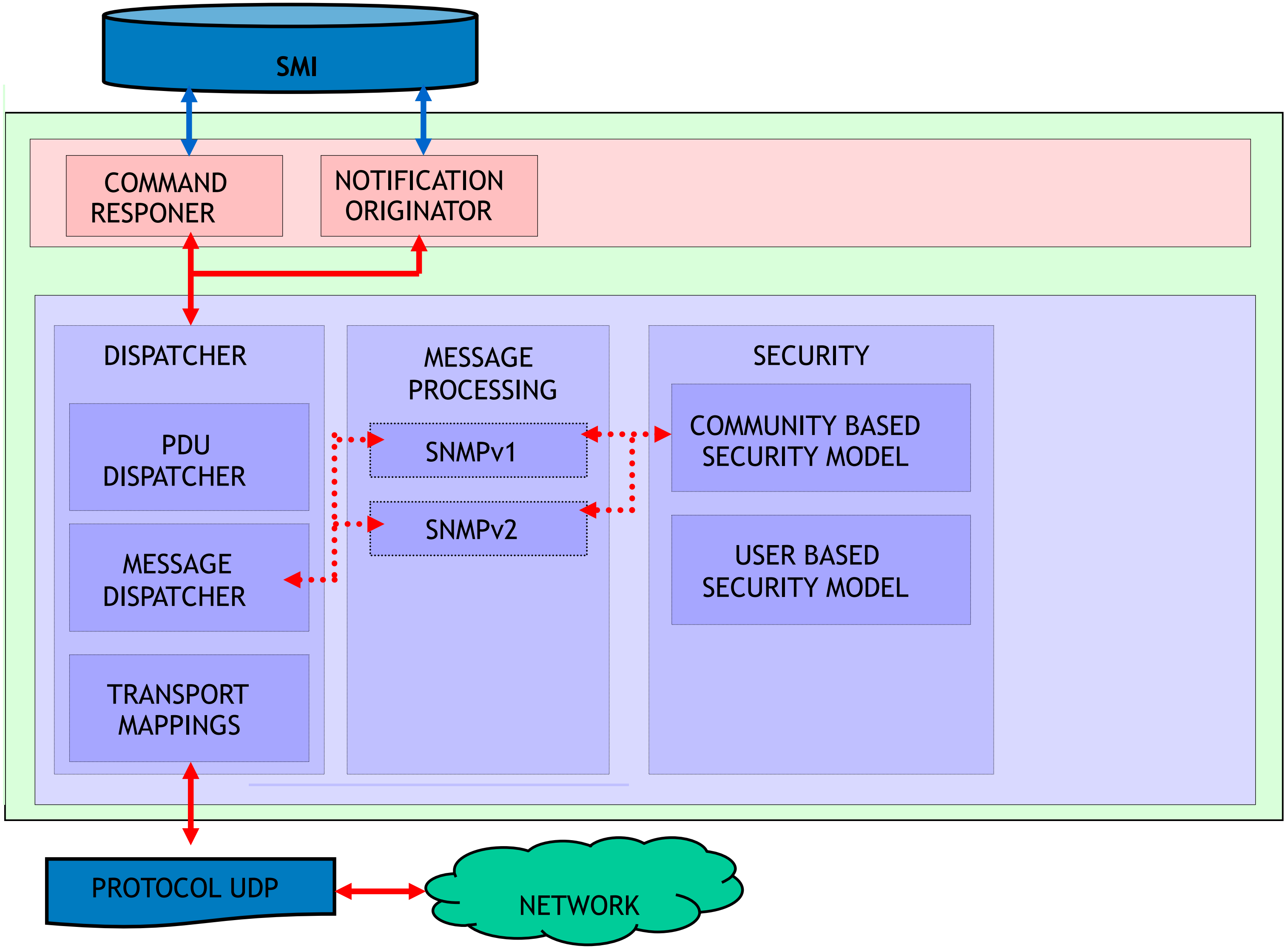
Entity proxy



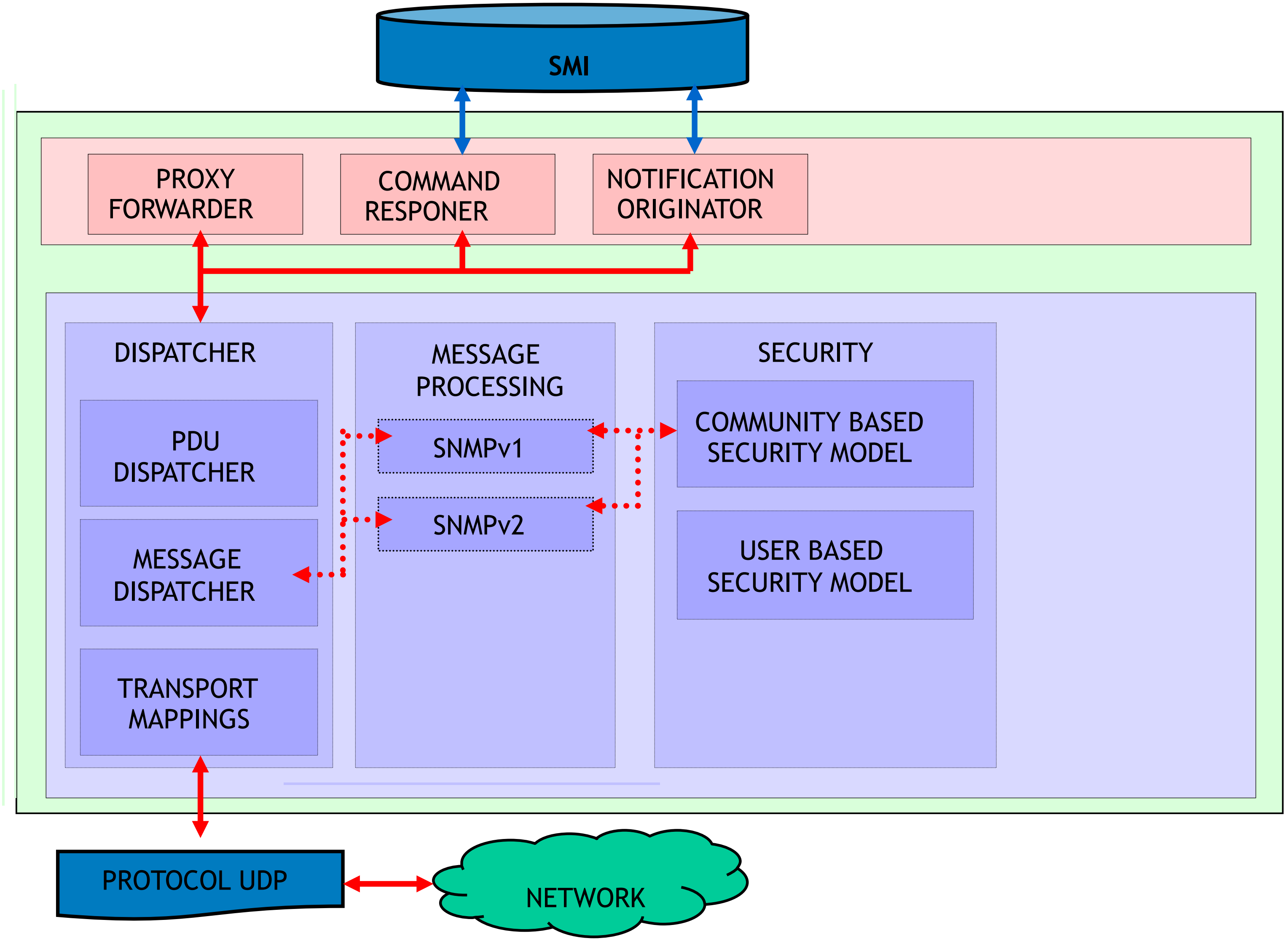
Entity administrator



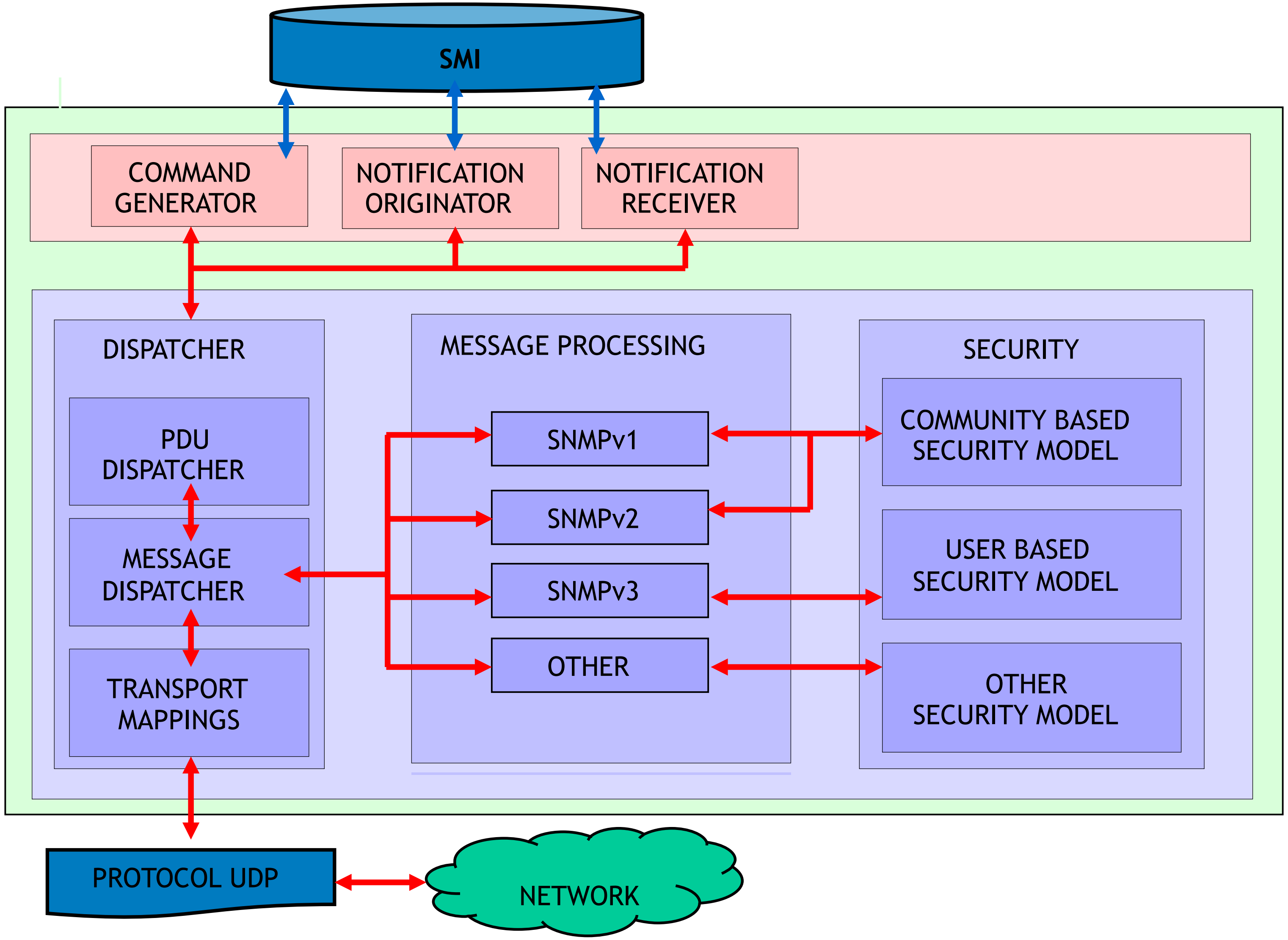
Entity Agent



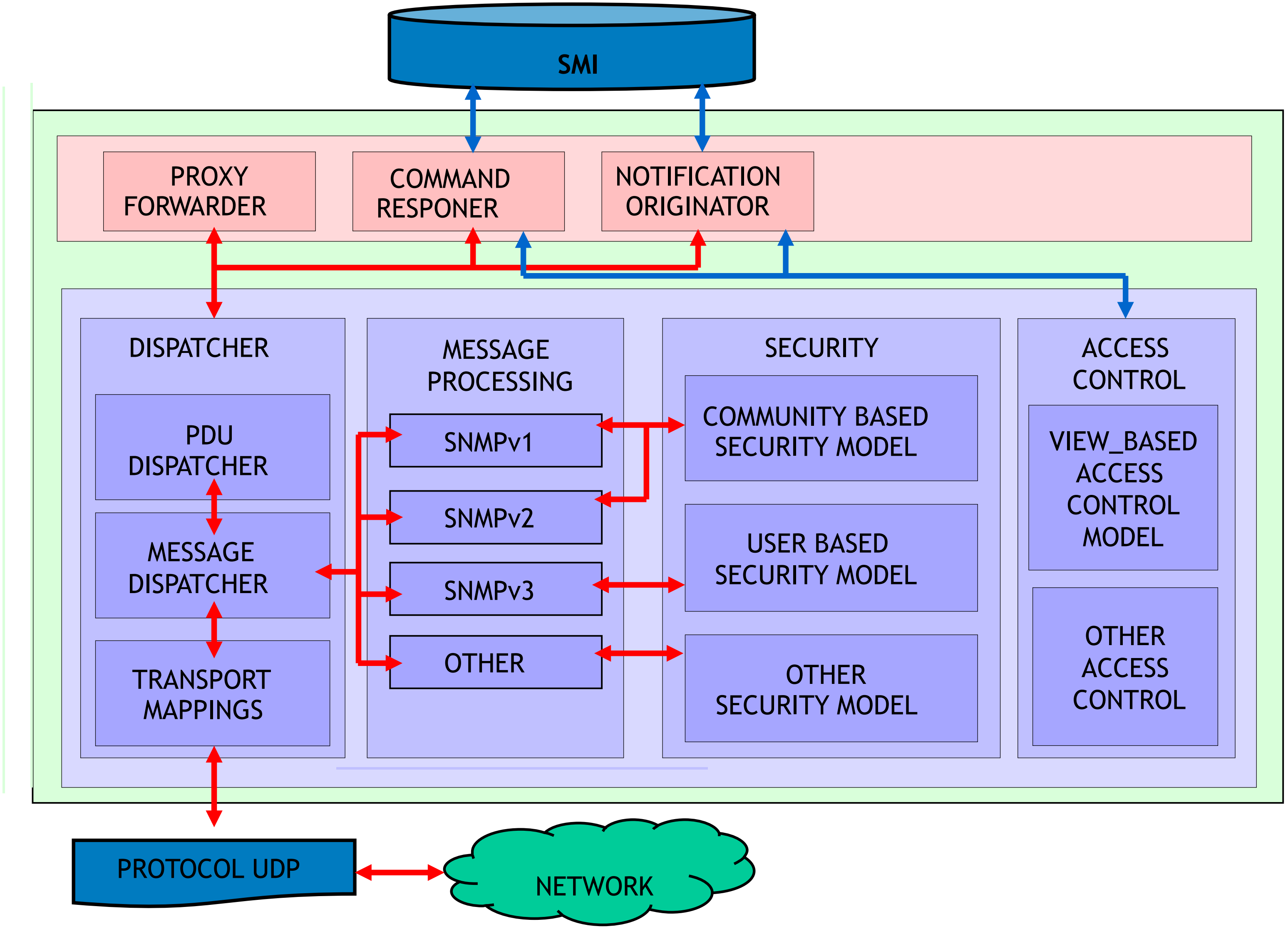
Entity proxy



Entity administrator



Entity Agent



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH