Computer architectures

What does the user see?
Services...

The user's perspective

- From the hardware you see the terminal
  - Screen (display)
  - Keyboard
  - Pointing device
- The most important “sights” are abstract things
  - Operator (user) interface
  - Processes (tasks, threads): running programs
  - Devices, files on their symbolic names
  - Users: their names, account numbers, email addresses, ownership and access categories
  - Nodes: computers, systems

User in front of the terminal

- Using the input devices
  - controls the machine, the running program(s) with a commands language;
- See what appears on the display,
  - interprets the response language elements.
- When the machine is “controlled”, a UI (User Interface e.g. command interpreter) process is running, which
  - controls through the OS and its services, by “requesting” the OS to execute commands!
  - Meanwhile, it keeps in mind, “sees” (deals with) the previously mentioned abstract “things” (command language, processes, devices and files, other users and accesses, other hosts, etc.)

The User Interface

- Mainly there are two types of UI
  - Command interpreter (shell)
  - Graphical interface
- Known user interfaces: cmd.exe , sh , DCL , Powershell , Win GUI , X desktops, etc.
- Interactive and batch use
- Are
  - commands language elements,
  - respond language elements,
- these must be known.

What do we see on graphic interfaces?

- **Tools**: icons...
- **Files**: icons, depending on their content. Actions with them: selection, dragging, attribute query, etc. (Double click: the associated application may start...)
- **Folders**: folder icons. Paths: the branches on a patterned wooden structure.
- **Processes**: windows, icons...
- **Hosts:** icon or drop-down list item. Sometimes we go back to the command-line interface...
- **Users**: icons or names...
- We can also see: menus, trays, etc. ...

The processes

- **Process: a program (not containing parallel structures), while running**
- **Program versus process**
- **The process context: ... identification information: pid, status information, etc.**
- **The user interface is also process(es)**
- **Why do you have to deal with processes?**

  “Shut down”, synchronizing, communicating...
- **What do we “see” from the processes?**
  - **Their ID, icon or window ...**
  - **And they also have a user interfaces ...**

Tools

- We can refer to devices (peripherals) with symbolic names
- Symbolic names can be used in commands
- There is a working (default) device (in Unix it is not necessary to refer to it)
- There is command to change the default device
- We can create file system on block-oriented devices
- Character-oriented devices can also be handled

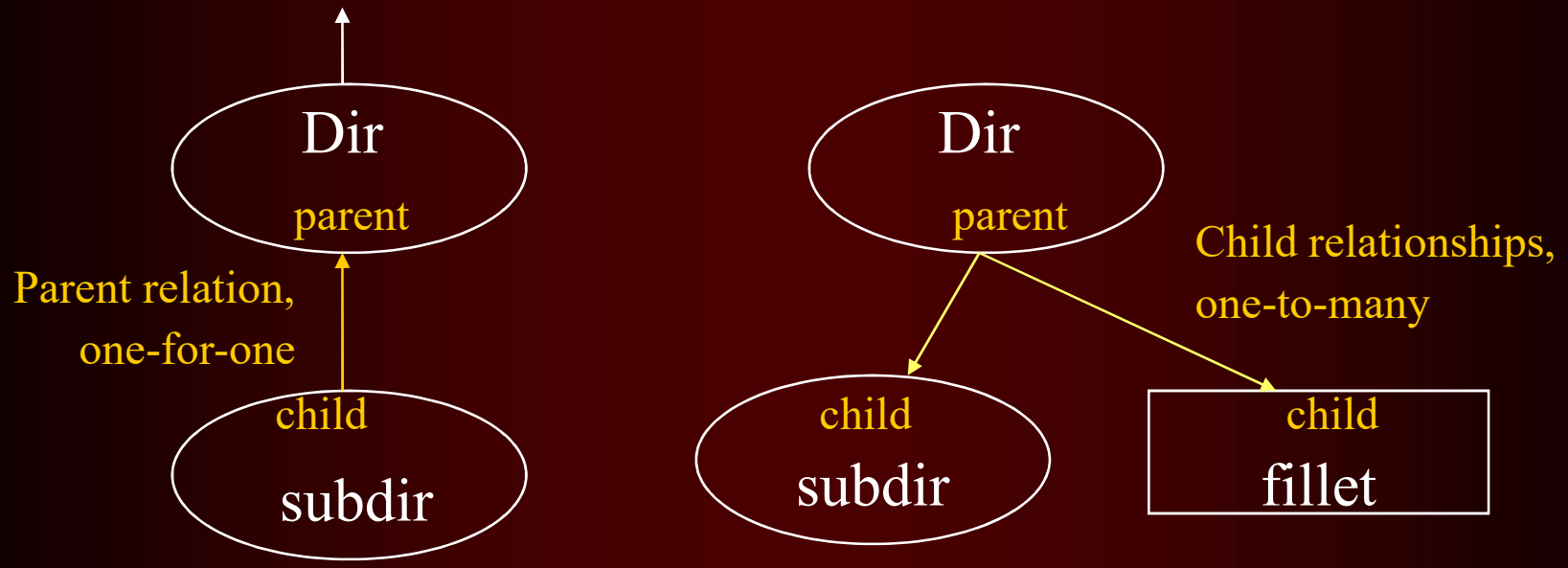
Files

- **File: data elements related in some way, with a name, on a structured device**
- **There may be naming conventions and restrictions**
- **Names can be referred in commands**
- **The data element: byte, word, field, record**
- **According to their content can be: text, document, binary data (image, sound, object program, executable program, etc.) File attributes.**

The directory

- Our idea so far: there is a file pool, containing files with their names. It should be arranged! For example, to collect and manage groups of files together.
- Directory: a file that contains entries about other files. It has a name, with conventions.
- Library? (Also includes “their child”.)
- In modern operating systems, all files - except one - are registered in a directory
- This gives a “parent-child” relationship

Parent directory



- **Parent directory: the parent of a directory.**
- **It has a symbolic name, The name has an OS shell dependency. This name helps to define the relative path.**

Root directory, file system

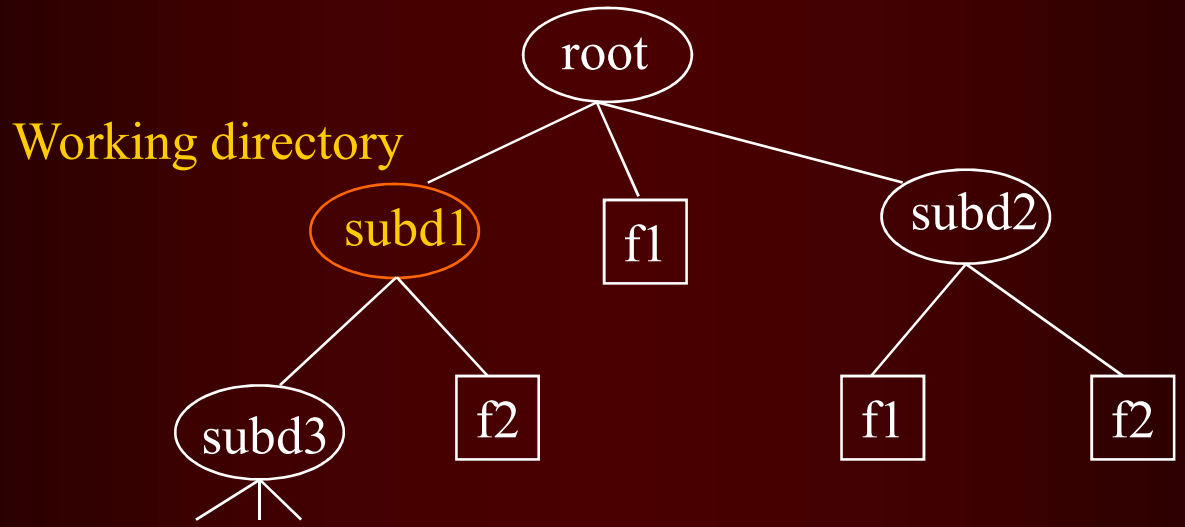
- The extension of the "parent-child" relation gives a hierarchical tree structure
- Root directory: a distinguished directory of the device. Not registered. Its content is in a prominent place. It is the starting point of the hierarchical tree structure.
- Its symbolic name is OS dependent
- File system: a hierarchical structure implemented on a block-oriented device, in which
  - files can be identified, their attributes and blocks are accessible,
  - the block occupancy of the device is managed.

Notes, path

- **Path: a list of directory names in parent-child relationship (the end of the list can also be a file name), which identifies a directory or a file starting from one of the directories**
  - **The list separator is OS shell dependent**
- **It can start**
  - **from the root directory (absolute path),**
  - **from the default (working) directory (relative path).**
- **Default directory (working directory): noted and highlighted by the OS. The starting point of the relative path: support quick search, it is not needed to refer to it explicitly.**
  - **It has a symbolic name, it depends on the OS shell.**

File system

- **Hierarchical structure on a block-oriented device**



Users

- There are other users (in fact: groups)
- Their identifiers are known for communication
  - their names,
  - their email address,
  - their website address etc.
- There are also ownership categories
  - xy owns this and this ...
  - this group is the group owner of this ...
  - There is no ownership relationship ...
- Access categories (rdwx)

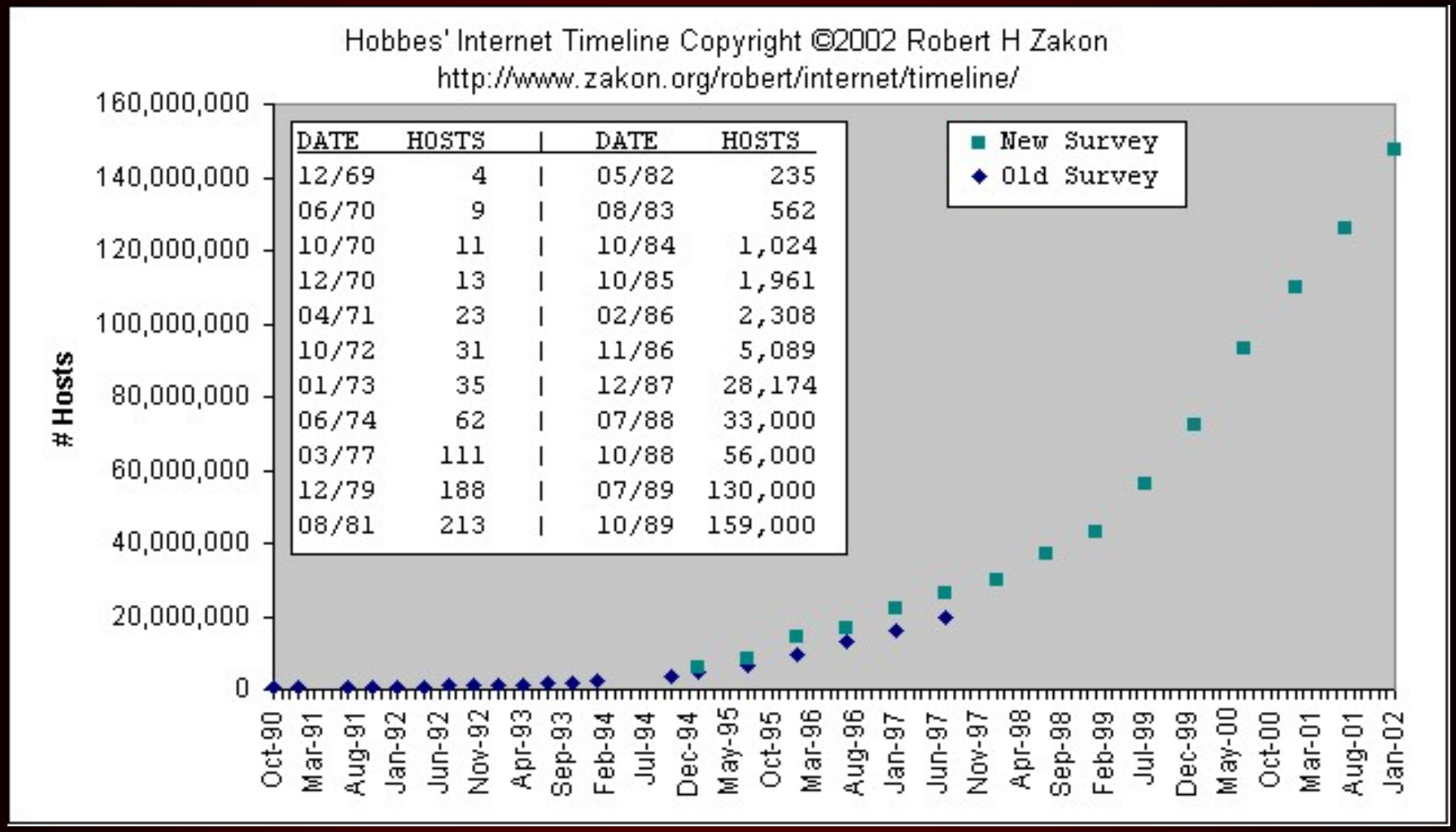
Networks: computer systems

- **Network classes: GAN , WAN , MAN, LAN, PAN**
- **The reasons of "networking".**
  - **Resource sharing, and concentration**
  - **Computer communication. Today it is almost the biggest driving force.**
- **Nodes**
  - **switches (network devices),**
  - **hosts (end devices).**
- **Data transmission media**

The Internet, the Net

- "Network of Networks"
- Features:
  - It is constantly growing
  - TCP /IP protocol family
  - Packet switching technology (PST)
  - Unified domain name system
  - Mostly client-server concept services

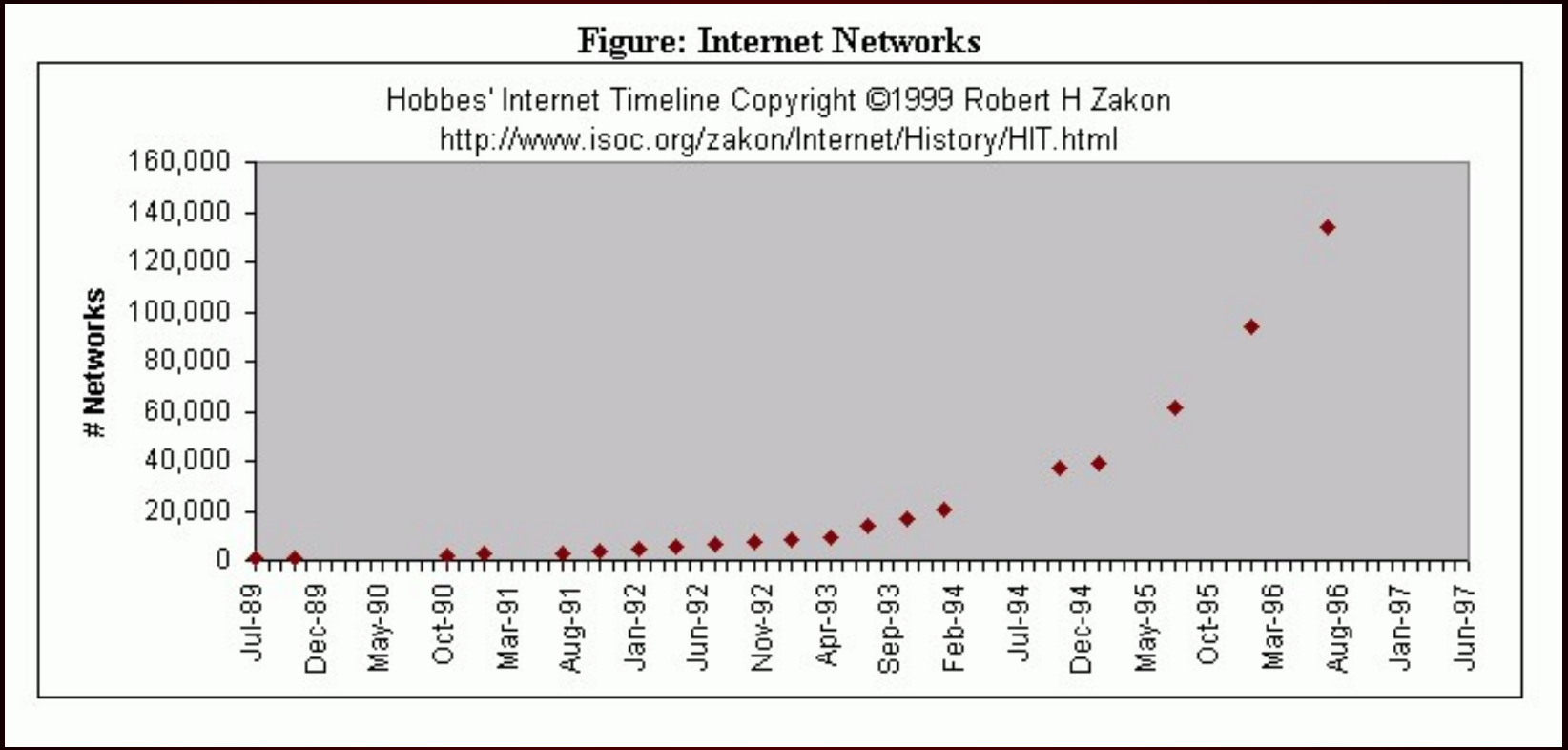
Constantly growing...



DATE	HOSTS	DATE	HOSTS
12/69	4	05/82	235
06/70	9	08/83	562
10/70	11	10/84	1,024
12/70	13	10/85	1,961
04/71	23	02/86	2,308
10/72	31	11/86	5,089
01/73	35	12/87	28,174
06/74	62	07/88	33,000
03/77	111	10/88	56,000
12/79	188	07/89	130,000
08/81	213	10/89	159,000

With permission from the author, from the URL http://www.isoc.org/guest/zakon/Internet/History/HIT.html

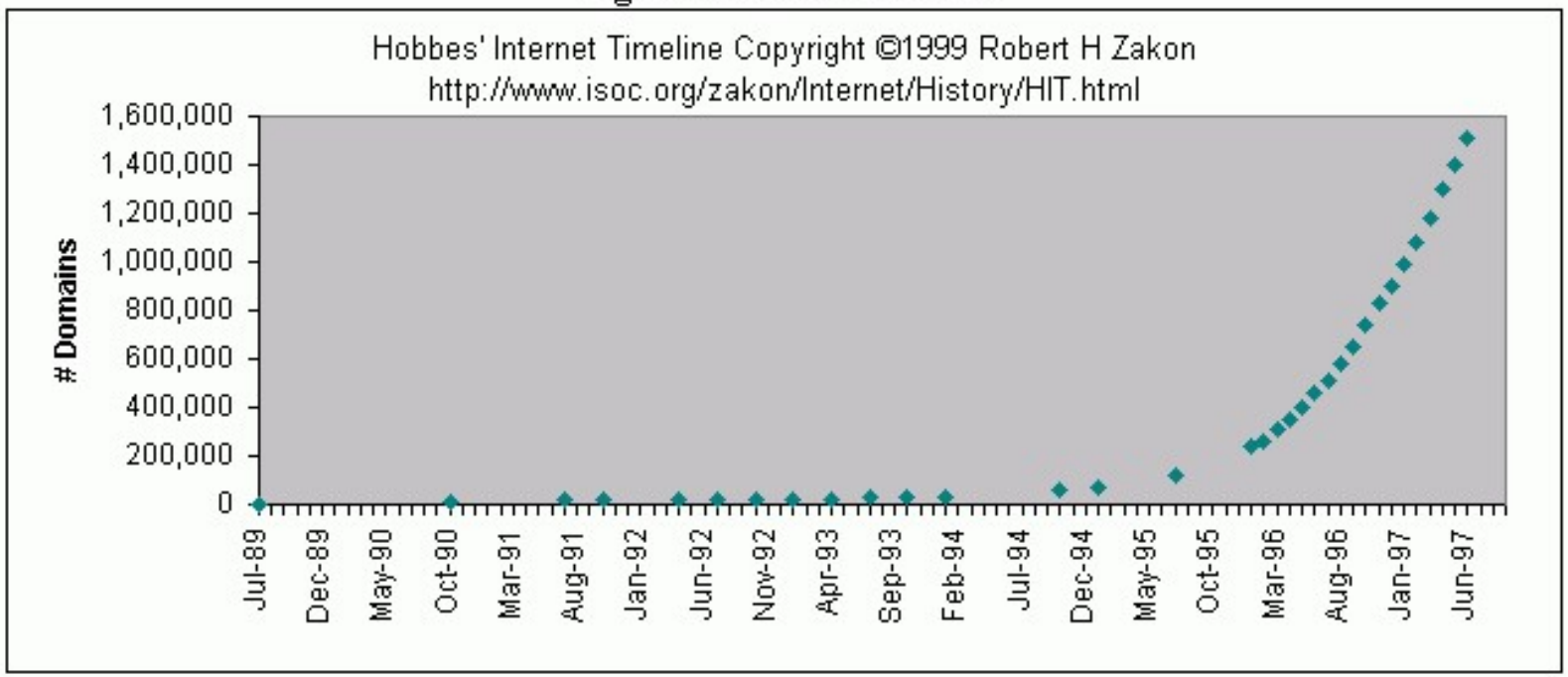
Constantly growing...



With permission from the author, from the URL http://www.isoc.org/guest/zakon/Internet/History/HIT.html

Constantly growing...

Figure: Internet Domains



With permission from the author, from the URL http://www.isoc.org/guest/zakon/Internet/History/HIT.html

It uses the TCP/IP protocol family

- Application layer
  - SMTP/MIME, POP3
  - TELNET , SSH
  - FTP
  - HTTP ... etc.
- Transport layer
  - TCP , UDP
- Network layer
  - IP
- Data link and physical layer
  - Ethernet, Token Ring, SLIP, PPP, WiFi, etc.

Packet Switching Technology (PST)

- Packet Switching Technology
- There are no dedicated lines between machines, but
- the data are in packets (packets),
- the packages are containing the address of the sender and the destination,
- routers control the packets, finally
- they check the addresses, forwards, and the package arrives...

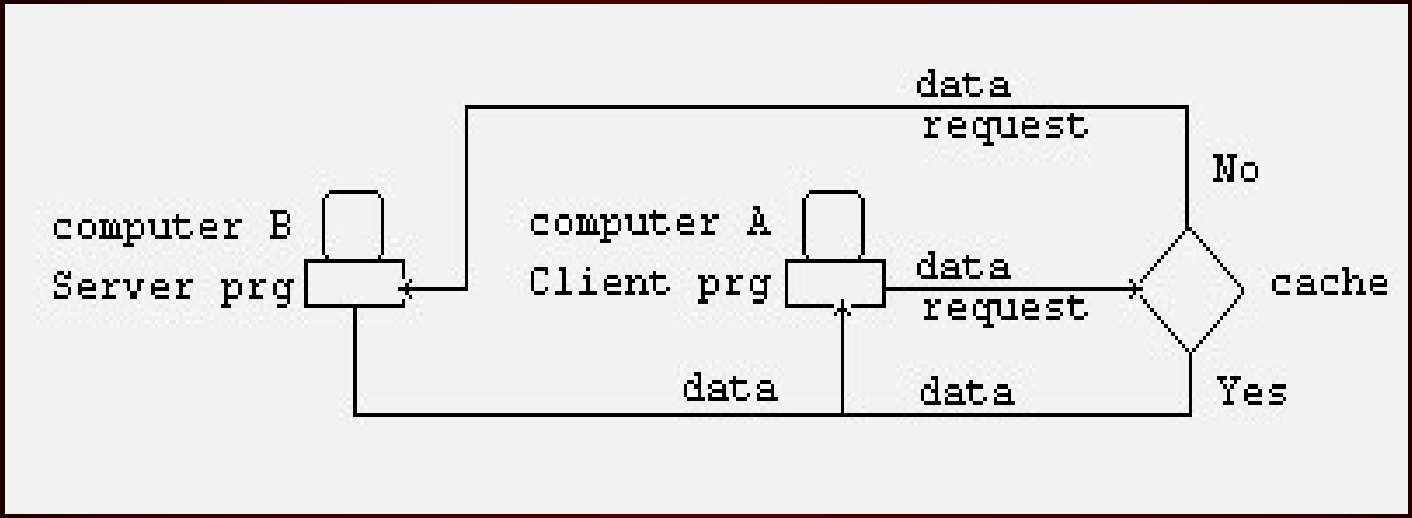
Unified Domain Name System

- The IP address of the machines is unique
- We prefer names, but
- there may be name repetitions, naming conventions are required ...
- They divided the world into provinces, domain names
- Domains have subdomains and nodes ...
- A name of the namespace is indexed

  jerry.iit.uni-miskolc.hu
- Name service: shared database

Client-server concept

- The client A's request is served by the server B with its response
- Clients can usually use a transitional cache



The Internet story

http://www.isoc.org/guest/zakon/Internet/History/HIT.html

- **In the 60s with the support of DARPA (Defense Advanced Research Project Agency).**
- **Dual purpose:**
  - **packet switching,**
  - **no network centers...**
- **1969: ARPANet**
- **Standardization: RFCs**
  - **1974: TCP/IP (RFC793/791)**
- **Tendency to cooperate**

The story continues

- 1983: the term Internet, MILNET-ARPANET, begins to be used
- 1984: the first name provider
- 1992: ISOC (Internet Society) is founded (organizing place for future vision, standards, organizing place for groups), Internet, the number of nodes exceeds 1 million
- 1992: WWW is born
- 1993: Internet Network Information Center (NIC) is established (registration, maintenance of standards)
- (Since then: commercialized)

The Hungarian story

- It was initially merged with the IIF, later with the NIIF
- 1985-90: Tibor Vámos' bold initiative: a national research packet-switched network. Embargo. Successful!
- 1990-95: Internet technologies, we were receptive
  - 1993-95: HBONE
  - 1996: Profit oriented companies are appearing.
  - ... today it is one of the most modern domestic networks, 10 Gbit/s
  - HBONE+ - Dark Fiber

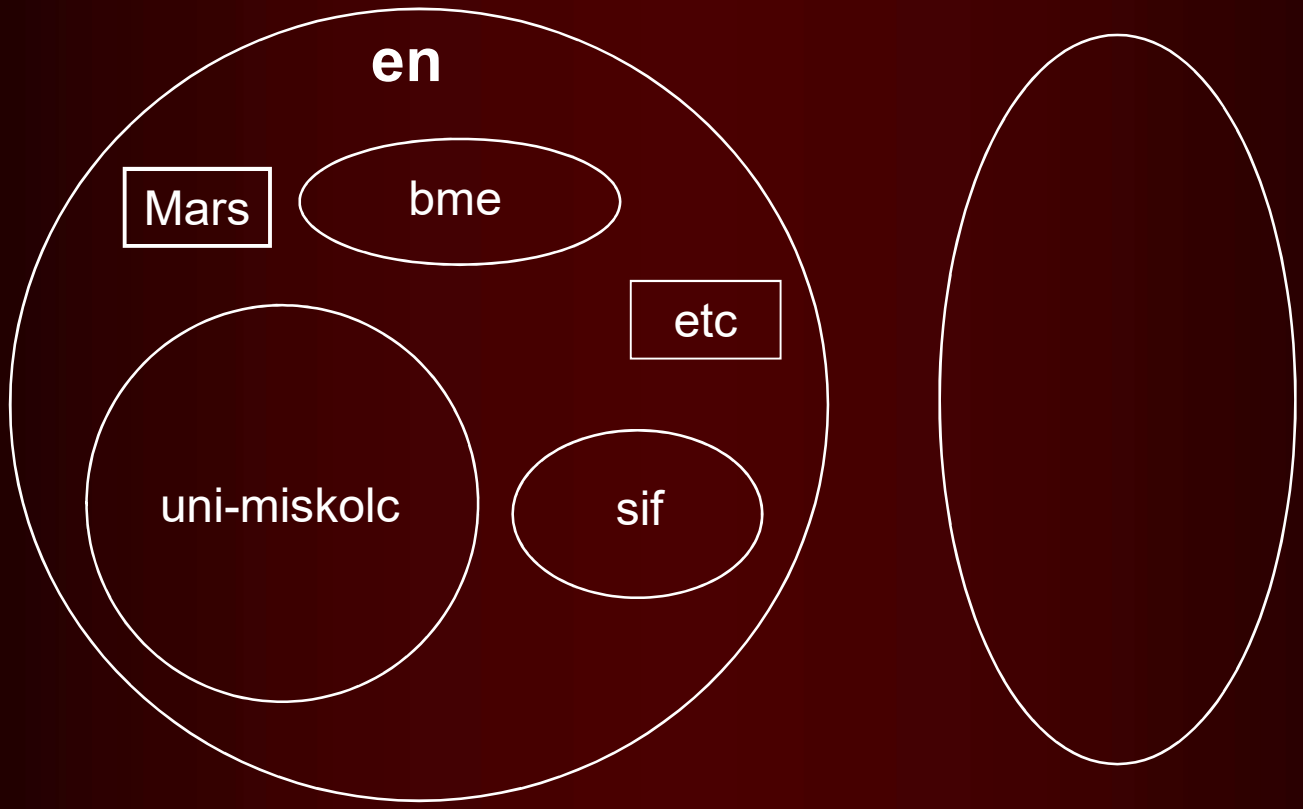
Nodes: hosts

- Hosts: identified systems.
- They provide services. They provide a user interfaces, I use them (local/remote machine use) (but there are other services too!).
- Two things are required
  - A connection must be established,
  - a session must be established.
- Sometimes these are diminished and eliminated.

The domain

- A named “area”.
  Geographical and other cohesive forces.
- Within a domain there can be:
  - nodes with their unique name, address,
  - subdomains with their names.
- The hierarchy is visible:
  parent-child relationship like a file system!
- The top domain concept

An example for the domain



The syntax of names, the namespace

- **node-name.subdomain-name.top-domain-name**
- node-name.top-domain-name
- subdomain-name.top-domain-name

You can see the hierarchical namespace!

- Thought: the name indexes the DNS database!
- The DNS database provides information:
  - structured information about the domain,
  - host IP address, HW characteristics, routing information, etc.

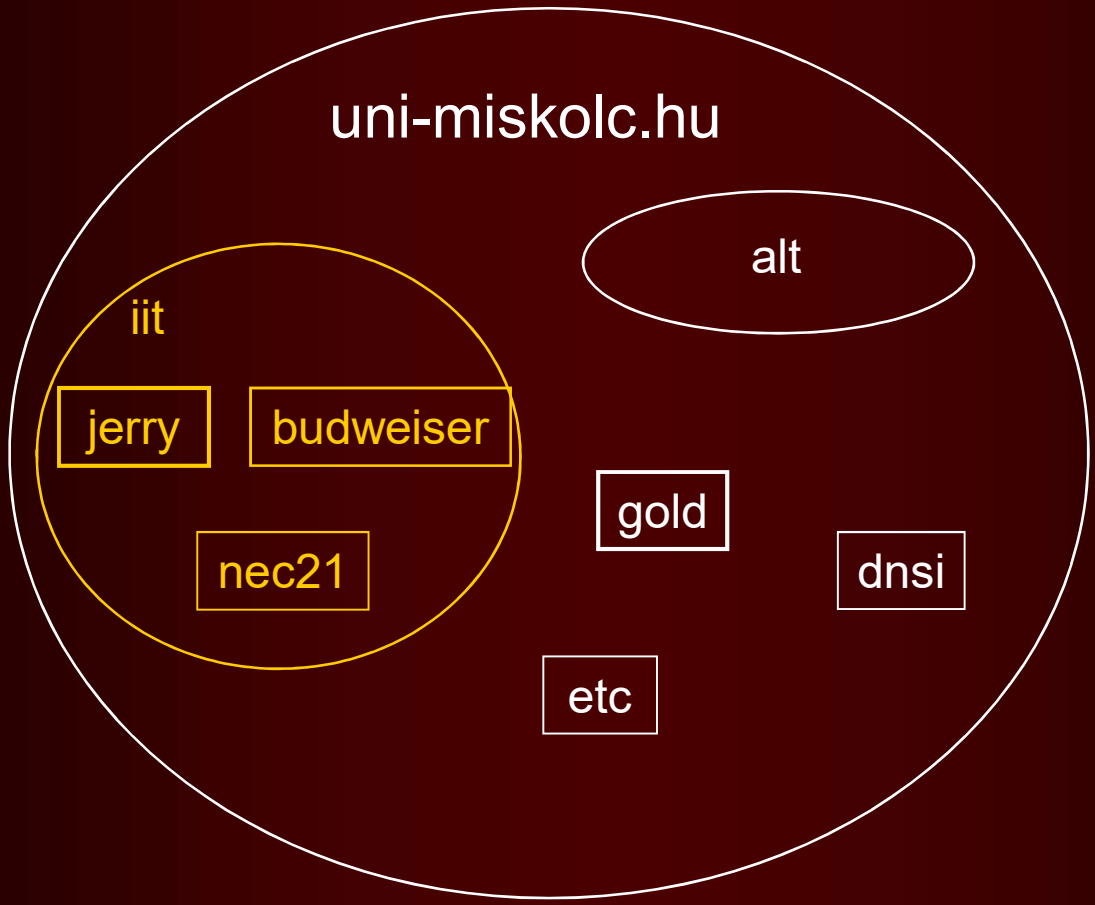
Providers of the DNS database

- The information in the DNS database is distributed and decentralized.
- The principle of delegation: an organization assumes responsibility for the information in the zone (almost like a province) delegated to it.
- This organization is obliged to operate a name server
- The service provider knows the names and addresses of the machines belonging to its zone, it also knows the name servers of the delegated domains!

The resolvers

- **The information is requested by the so-called resolver.**
- **Actually RTL** (Run Time Library) **routines.**
  E.g.: gethostbyname(IP)
- **They are built into the clients (e.g. ssh , ftp, etc.).**
- **They know their name server.**
- **They try to resolve the request according to the given search order. Eg: first in the local table, then in the NIS, then in the DNS.**

An imagined example of name resolution



I'm looking for something from whatever

- The resolver of the client running on "*whatever*" cannot find "*something*" in the *"local/NIS/LDAP"* database, then
  - first it addresses the **name server** (*dns.uni-miskolc.hu*).
  - the DNS must know all names in its zone. So it resolves.
  - If necessary, *dns* inquires "downstream" from delegated name providers!
  - or if necessary, inquires a so-called **top name server** , (root server) and further name servers down from there.

Small tasks

- Check the nslookup utility!

> nslookup some-name

> nslookup ip address

For example :

> nslookup zeus.iit.uni-miskolc.hu

> nslookup 193.6.5.33

For example:

> nslookup mail.iit.uni-miskolc.hu

Name: hera.iit.uni-miskolc.hu

Addresses: 2001:738:6001:500::4

193.6.5.4

Aliases: mail.iit.uni-miskolc.hu

Establishing the connection

- The most important information for this is the **host identifier** (address, name) and **service identifier** (port address and service protocol).
- The service identifier is often “wired” into the process that initiates the connection, it does not need to be entered.
- Its purpose: to establish a connection in order to enable the use of the service by creating a session. (Start a connection management process on the host, which provides the connection).

Available services

- **What *iit* provides…**

See on the department's website!

- **What Uni Miskolc provides...**

Our laboratories

- in 24-hour mode
  - Lab 101 : 28 machines Windows / Unix
  - Lab 102 : 16 machine Windows / Unix
  - 103 . laboratory : 40 machines Windows / Unix

- closed labs (105 robots).
- Follow the rules of use!

Account number in iit range

- Everyone gets it automatically (ldap directory)
  - Login name + (initial) password
  - For computer use (login, ssh), electronic correspondence (e-mail), ftp
  - Storage: quota username
    - Reasons for filling: cache in browsers; graphical interface settings; spam; disk usage rate: du –sch *
  - Rules of use!
    - http://www.iit.uni-miskolc.hu/laborok-felszereltseg/hasznalati-szabalyzat.html
- System administrator: mail://root@iit.uni-miskolc.hu
- MS Windows: via the website winadmin.iit.uni-miskolc.hu

Linux

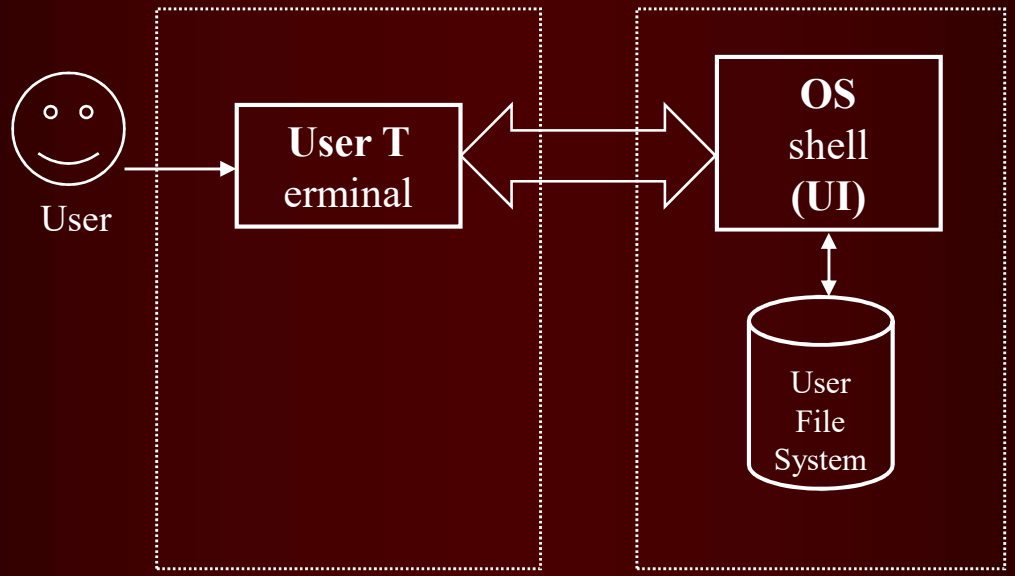
- Logins
  - Ctrl + Alt + F7 name password (on graphical interface)
  - Recommended interfaces:
    - WindowMaker (fast and puritanical)
    - Blackbox/Fluxbox (fast and convenient)
    - Gnome (convenient, resource intensive)
    - KDE (like MS Windows, resource intensive)
  - Ctrl + Alt + F1 name password
    (on command line interface – virtual terminal)

Remote login

- with ssh or putty client within the iit domain
  - There is no serious limitation. Learn machine names
  - *name* .iit.uni-miskolc.hu …
  - As usual, the bash shell starts, the /home/ *gr* / *username* directory is the login directory (~/, or $HOME/)
- From outside the iit domain only the jerry is reachable

> ssh *username*@jerry.iit.uni-miskolc.hu

Remote login model



E-mail

- All the accounts has an e-mail address

  username@iit.uni-miskolc.hu

- Multiple mail clients in iit domain
  - mail, mutt, pine, mozilla-thunderbird, etc.
  - Webmail: https://webmail.iit.uni-miskolc.hu
- Remotely available POP3 provider (port 110, 995 encrypted)

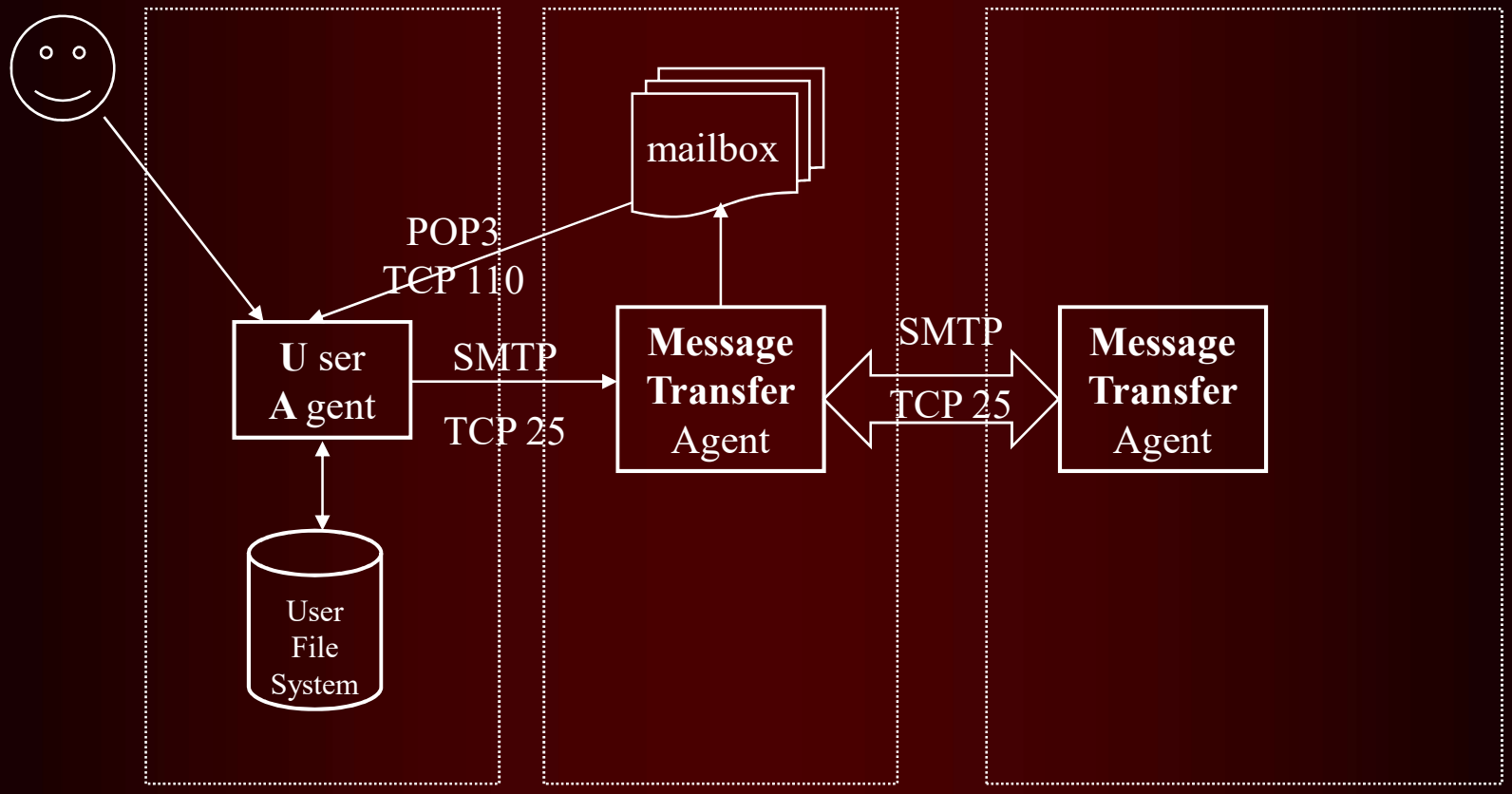
pop3.iit.uni-miskolc.hu

- Within the iit domain SMTP service available (port 25, not encrypted)
- Outside the iit domain SMTP service available(port 465, encrypted, password protected)

  smtp.iit.uni-miskolc.hu

Mail from the iit domain can be forwarded automatically (~/.procmailrc or ~/.forward)

The model of e-mail



E-mail - SMTP
(Simple Mail Transfer Protocol - Basic)

S: 220 smtp.server.com Simple Mail Transfer Service Ready
C: HELLO client.example.com
S: 250 Hello client.example.com
C: MAIL FROM : < mail @ samlogic.com >
S: 250 OK
C: RCPT TO : < john @ mail.com >
S: 250 OK
C: DATA
S: 354 Send message contents ; end with < CRLF >.< CRLF >
C: *<The message data (body text, subject , e-mail header , attachments etc)>*
C: .
S: 250 OK, message accepted for delivery : queued as 12345
C: QUIT
S: 221 Bye

E-mail - SMTP
(Simple Mail Transfer Protocol - Extended)

S: 220 smtp.server.com Simple Mail Transfer Service Ready
C: EHLO client.example.com
S: 250-smtp.server.com Hello client.example.com
S: 250-SIZE 1000000
S: 250 AUTH LOGIN PLAIN CRAM-MD5
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: adlxdkej
S: 334 UGFzc3dvcmQ6
C: lkujsefxlj
S: 235 2.7.0 Authentication successful
…

E-mail list service

- Student groups can request the service
  - List members receive mail sent to the list
  - You can join or leave the list
  - We need a list administrator, 1-2 people
  - It must be requested and approved.
- Other fields of activity corresponding to the university's activities may also require a list...
  - But they may not get it.

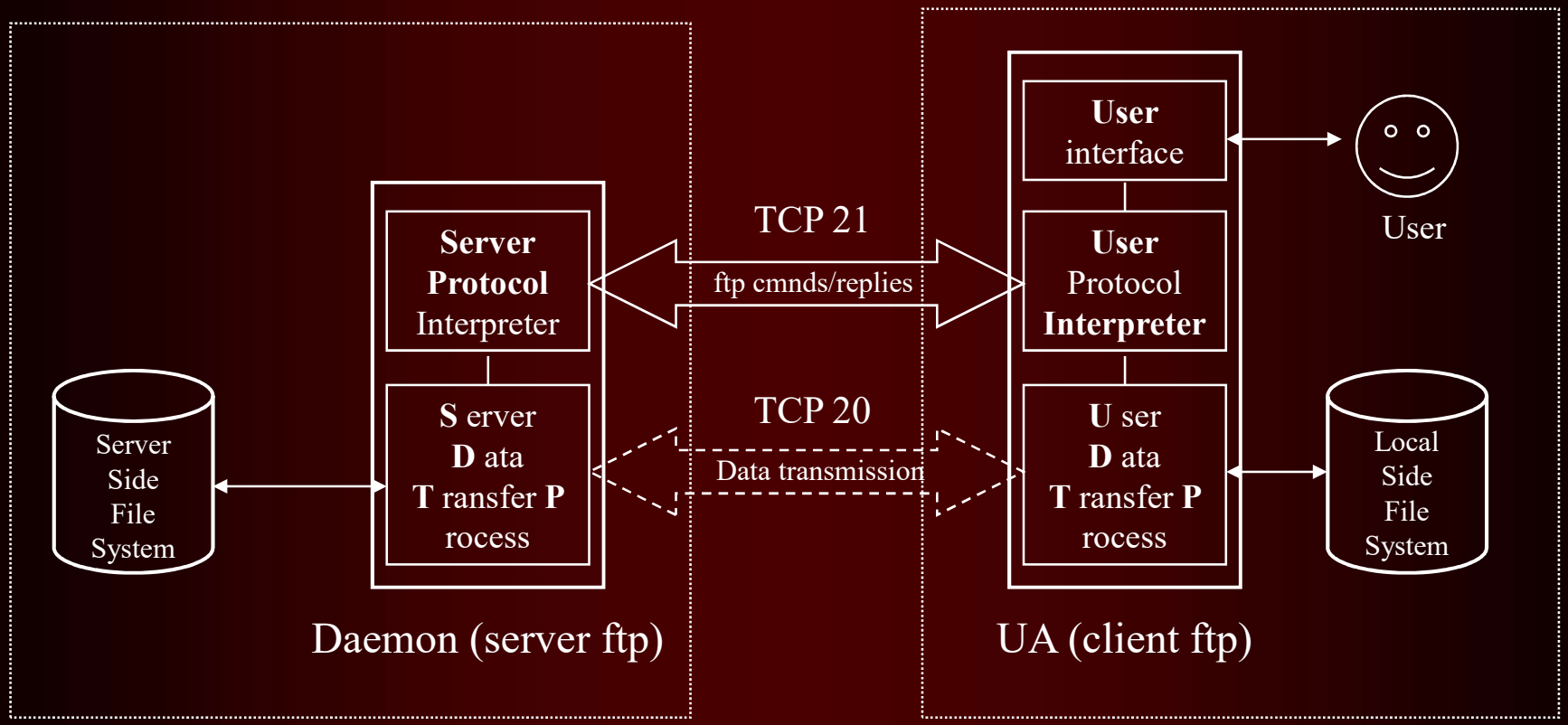
FTP servers

- With the help of the service, we can upload and download files from our own storage at iit
- ftp://ftp.iit.uni-miskolc.hu on port 21
- Many clients can be used
  - Unix-Linux: ftp, mc, etc.
  - Windows: Total Commander, IE, etc.

- Debian Mirror (unlimited, also outside iit)

http://debian.uni-miskolc.hu/debian

The ftp model



Web service

- Many different browsers (clients) are available
  - firefox (Mozilla), Netscape, galeon, lynx
- Each of our users can have their own WEB page:
  http://users.iit.uni-miskolc.hu/~username
  ~/public_html directory can be created in it
    index.html
    CGI programs, PHP too
- Department's website: http://www.iit.uni-miskolc.hu

Office suite

- OpenOffice.org: good compatibility with MS Office.
  Launch: soffice
  - Relatively resource demanding
  - Table manager, draftsman and presentation maker,
  - It can export to pdf format.
- A faster but less compatible document editor

  abiword
- Some text editors
  - nano, pico: they are simple
  - joe: simple, but different from the previous ones
  - mcedit: similar to DOS edit, simple
  - vi, vim: even with a simple, stuttering connection

Other useful tools

- **PDF document readers**
  - **acroread (accurate, more resource intensive)**
  - **xpdf (easier)**
- **File manager**
  - **mc (Midnight Commander)**

Development environments

- GNU Compiler Collection

  gcc

  g++
- kdevelop: for graphical development (uses gcc)
- anjuta: also
- javac, java, netbeans, eclipse: the latter is graphical, but very demanding on resources

WLAN service

- 4 Access Points on the first floor of the IIS building:
  - At the door of room 100 in the corridor
  - At the door of lab 103 in the corridor
  - In office 107
    - 802.11/b,g,n; DHCP automatic IP address and NAT
- Network name (SSID) : IITAP
- With the connection settings :
  - Network authentication: WPA2-Enterprise
  - Encryption: AES or TKIP
  - EAP type: EAP-TTLS
  - Password verification type: PAP
- The IIT's LDAP accounts can be used.

Other (transparent) services

- **NFS** (nfs provider: hera.iit.uni-miskolc.hu)
  - This provides the ~/ (HOME) directories
- **DNS** (hera.iit.uni-miskolc; dns.uni-miskolc.hu)
  - 193.6.5.33 , 193.6.10.1
  - nslookup, host clients can use this
- **LDAP** (hera.iit.uni-miskolc.hu)
  - This is the central registry system for managing account numbers
  - ldapsearch and finger client can use this

Services of ME

- All university citizens can request an account number (and thus a mailing address) in the domain uni-miskolc.hu
  - For remote machine use on the gold.uni-miskolc.hu machine
    - With all storage areas
    - ksh shell
  - Own website here too
  - In the same place, pine, mutt, etc. correspondence
  - Correspondence here https://webmail.uni-miskolc.hu
- It is operated by ME ISzK
  - www.uni-miskolc.hu WEB provider,
  - firewall, virus filtering for mail...

Summary

- What does the user see?
  - User interface
  - Processes (tasks, threads): running programs
  - Devices, files by their symbolic names
  - Users: their names, account numbers, e-mail addresses, ownership and access categories
  - Nodes: computers, systems, their services
- What services can they access?

Computer architectures

What does the user see?

Services ...

End