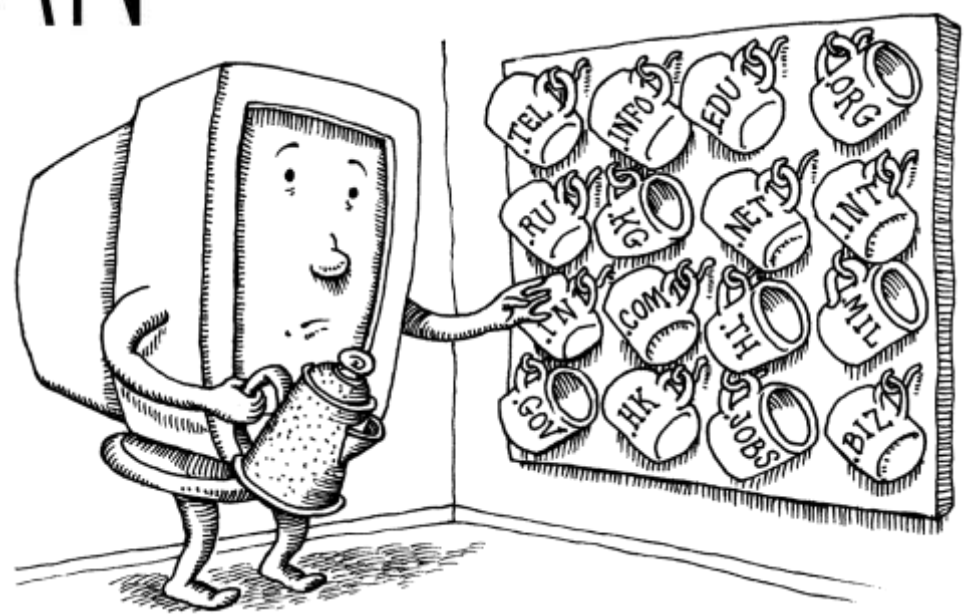
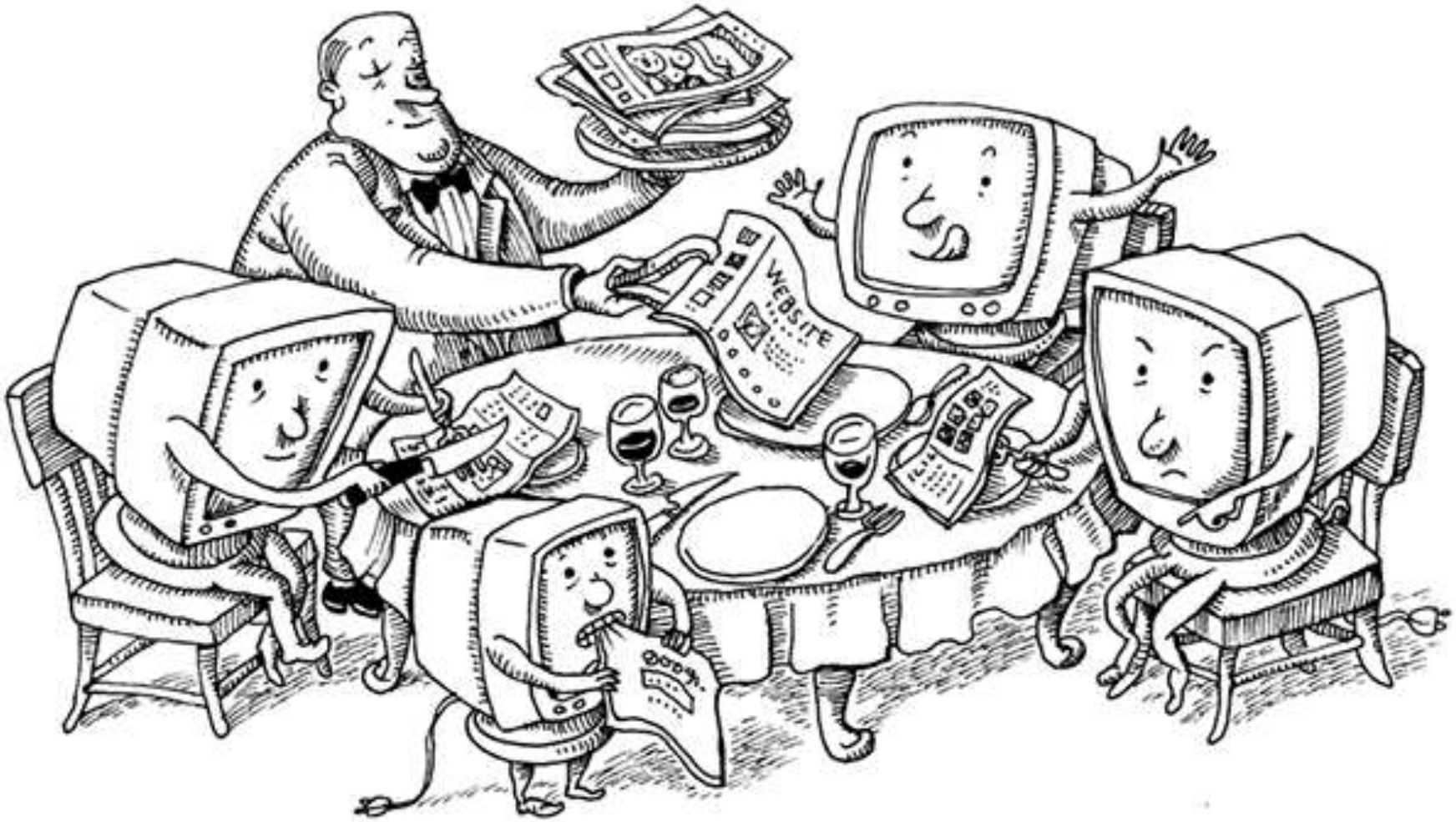
WEB SERVER

HÖGSKOLAN VÄST

TROLLHÄTTAN

HÖGSKOLAN VÄST





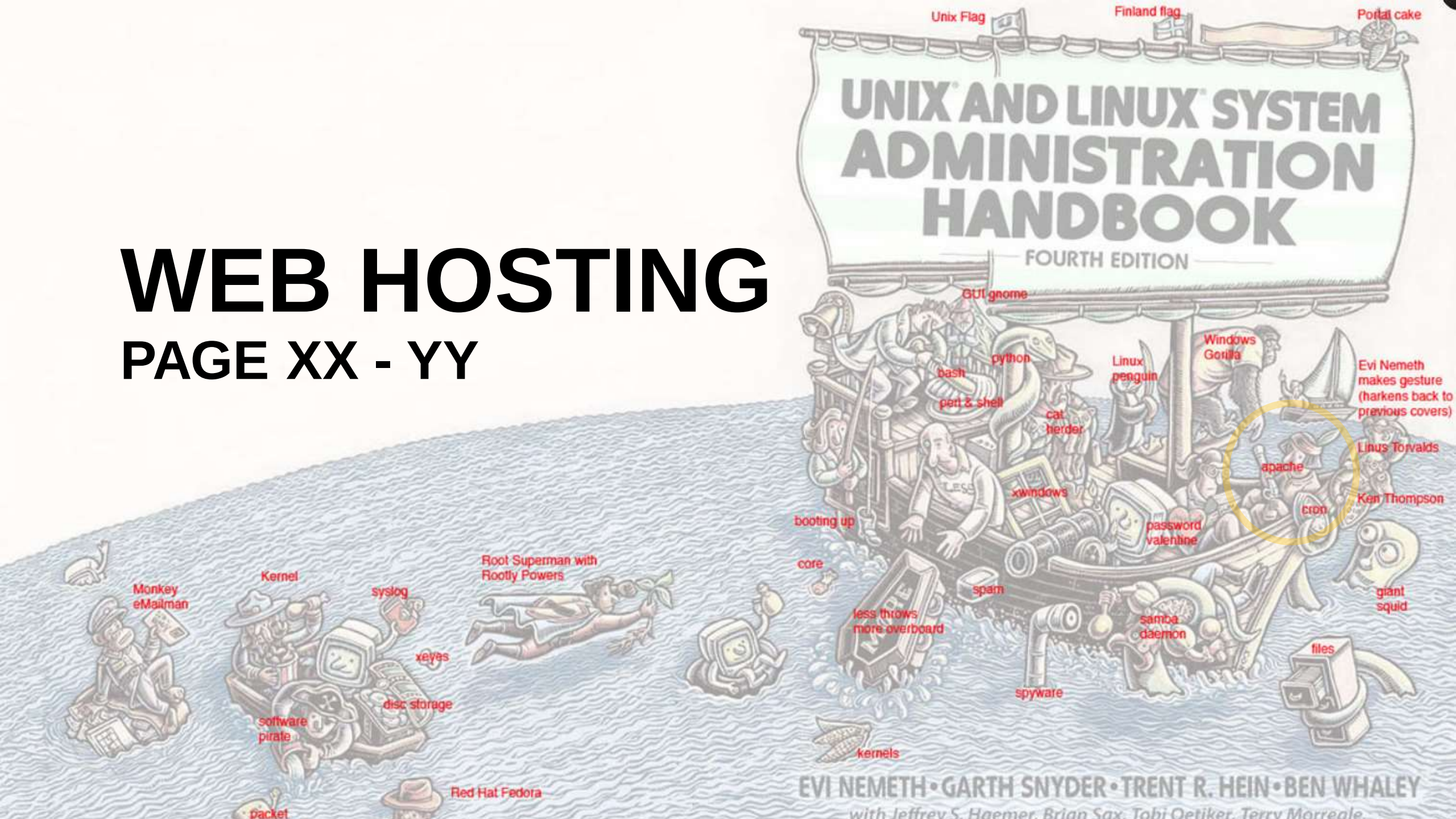
WEB SERVERS

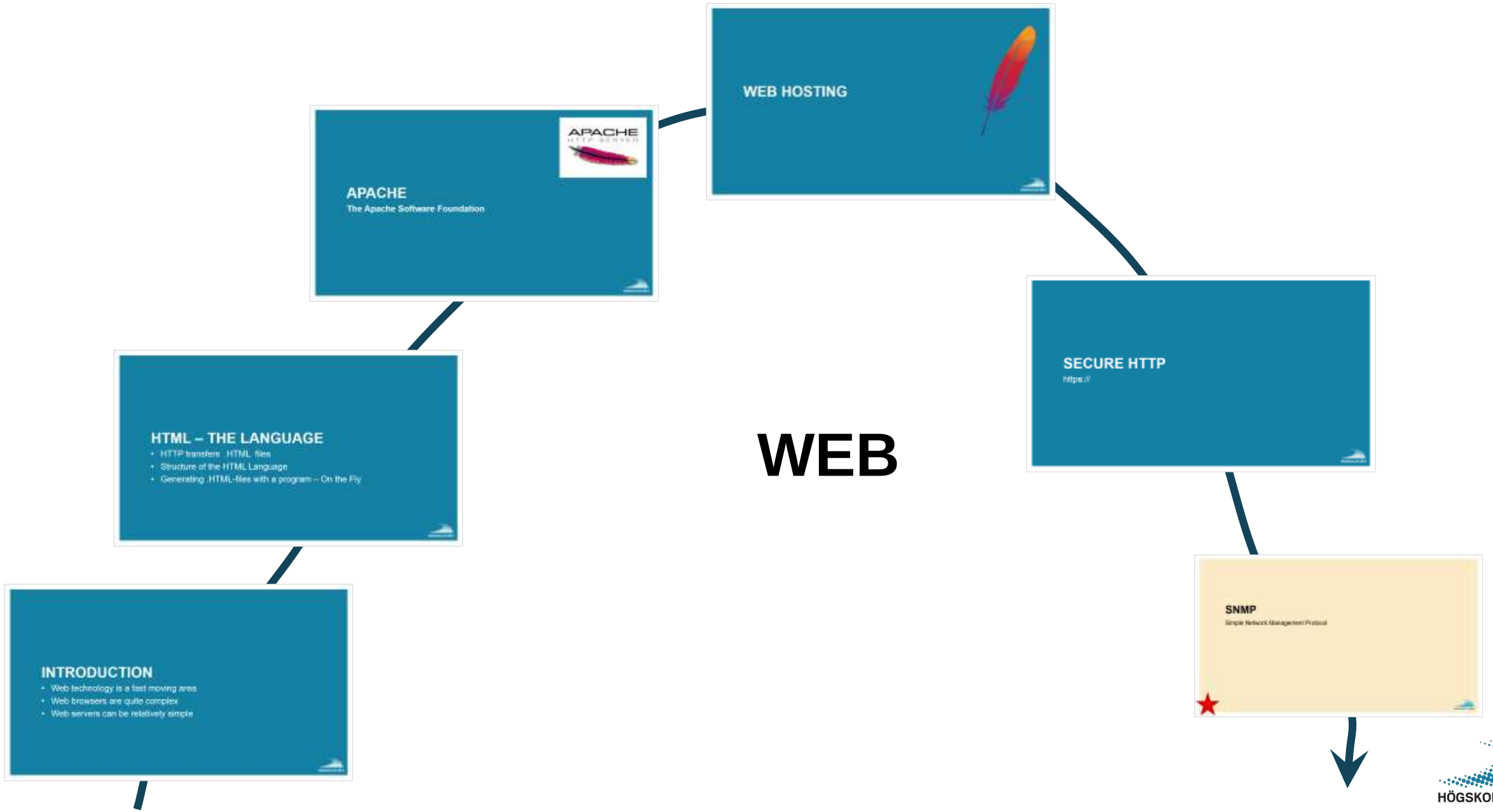
INT202 – Internet Infrastructure Applications

HÖGSKOLAN VÄST

WEB HOSTING

PAGE XX - YY





HÖGSKOLAN VÄST

INTRODUCTION

- Web technology is a fast moving area
- Web browsers are quite complex
- Web servers can be relatively simple

HOW A WEBSERVER WORKS

1. *Client (browser)* contact the webserver
   - HTTP-request (URL)
2. *Server* fetches from the harddrive, or creates on the fly, a file
3. *Server* sends the file as a response
4. Connection closed

URL - UNIFORM RESOURCE LOCATOR

- Address to an object or a service
  - `http://xxx.xxx.yy:port/directory/file.txt`
  - `http://www.hv.se:777/pub/file.txt`
- Protocol or application
- Hostname (computer address)
- TCP-port (opt)
- Directory (opt)
- File name (opt)
- Protocol: *http, https, ftp, file, mailto*

REQUEST

- A TCP connection is created to the server
- Client sends text command (one or a few lines).
  For example: **GET /index.html**
- Server sends HTML-file back

```
telnet www.google.se 80
GET /
```

HTTP V1.1

- Supports “name based virtual hosts”
- Supports “multiple requests in the same session”

telnet *www.hv.se* 80

GET /index.html HTTP/1.1

Host: www.hv.se

<Emtpy line>

HTTP V2 / HTTP V3

HTTP v2

- 2015
- Single connection
- Server push, header compression, etc

HTTP v3

- QUIC protocol (Not TCP !!)
- Header compression
- Multiplexing

QUIC is designed to obsolete TCP at the transport layer for many applications, thus earning the protocol the occasional nickname "TCP/2”

HTML – THE LANGUAGE

- HTTP transfers .HTML files
- Structure of the HTML Language
- Generating .HTML-files with a program – On the Fly

HTML-KOD

```
<!DOCTYPE html>
<html>
 <body>

   <h1>My First Heading</h1>

   <p>My first paragraph.</p>

   <p>
     <a href="http://www.google.com"> Att googla </a>
   </p>

   <p>
   <img src="chef.jpg">
   </p>

 </body>
</html>
```

index.html fil

RESULTERANDE BILD AV HTML-KOD (Rendering)

HÖGSKOLAN VÄST

GENERATE HTML “ON THE FLY”

DYNAMIC PAGES

HTML files can be generated “on the fly”
- using external programs or by the webserver.

- PHP (**P**HP **H**ypertext **P**reprocessor)
  - A common language to generate dynamic content
  - Can be embedded in the HTML code.

- CGI (**C**ommon **G**ateway **I**nterface)
  - A script is executed on the server (could be Python)
  - Standard output from the script is sent to the browser

HÖGSKOLAN VÄST

PHP – EXAMPLE (TEST.PHP)

```
<html>
<head><title>Test of PHP</title></head>
<p>Here is some generated text:</p>
<?php
        phpinfo();
?>
<p>Ending text.</p>
</html>
```

HÖGSKOLAN VÄST

CGI EXAMPLE

/var/www/cgi-bin/thedate.py

```
#!/usr/bin/python3
from datetime import datetime

print("Content-type: text/plain")
print( " " )
justnu = datetime.now()
print("Today is", justnu)
```

/var/www/cgi-bin/thedate.sh

```
#!/bin/bash


echo Content-type: text/plain
echo " "
justnu=`date`
echo "Today is " justnu
```

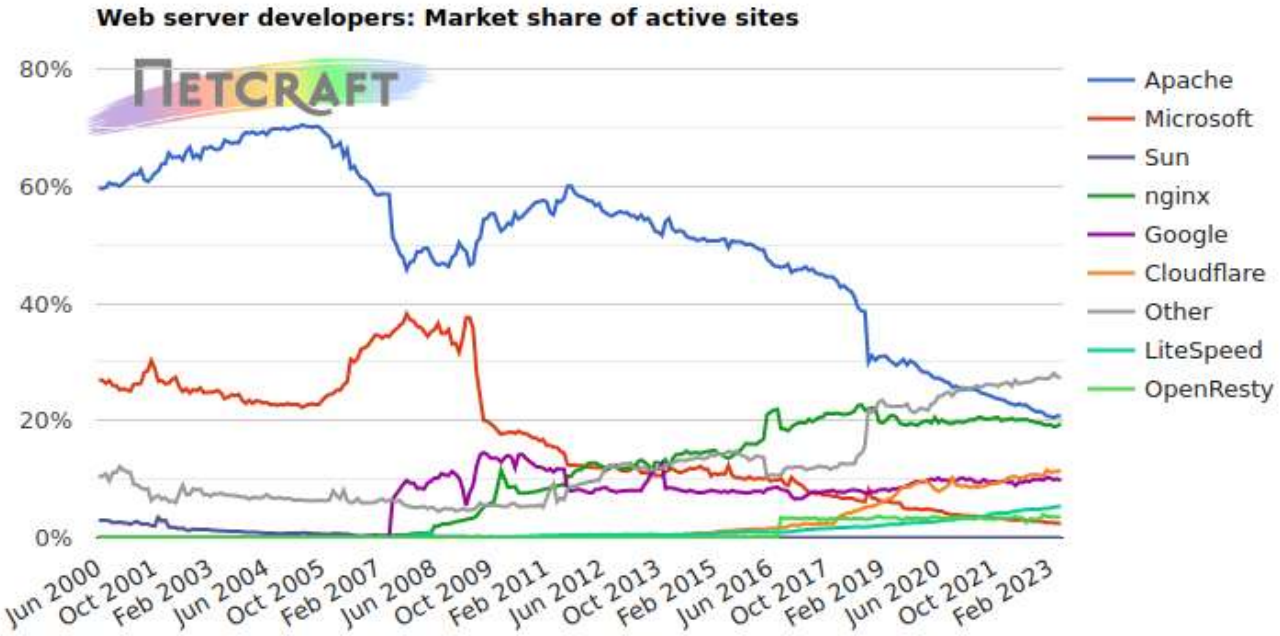
HÖGSKOLAN VÄST

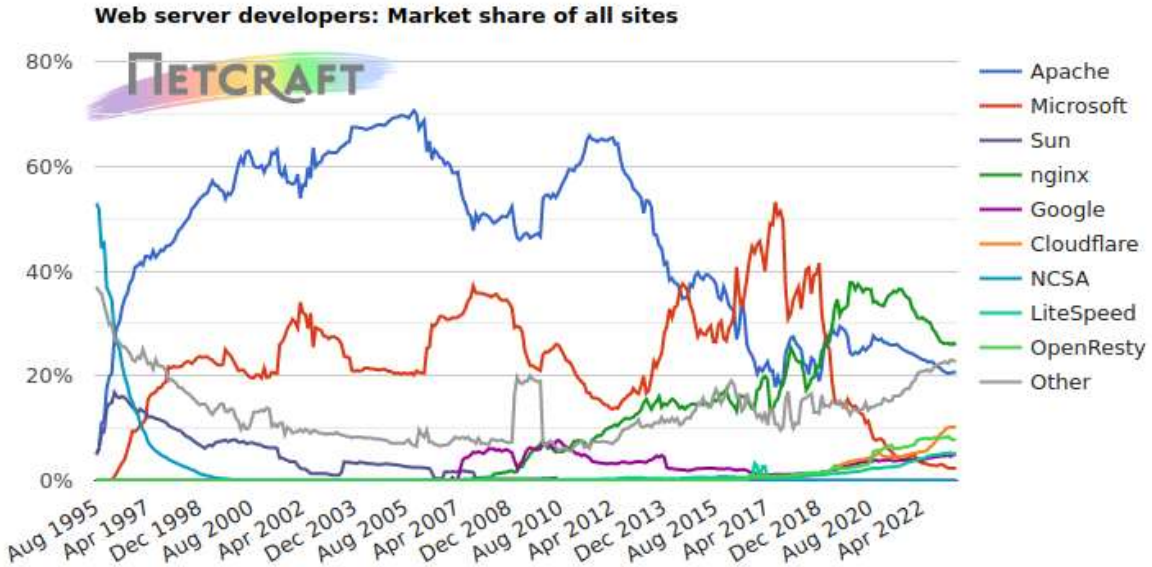
APACHE

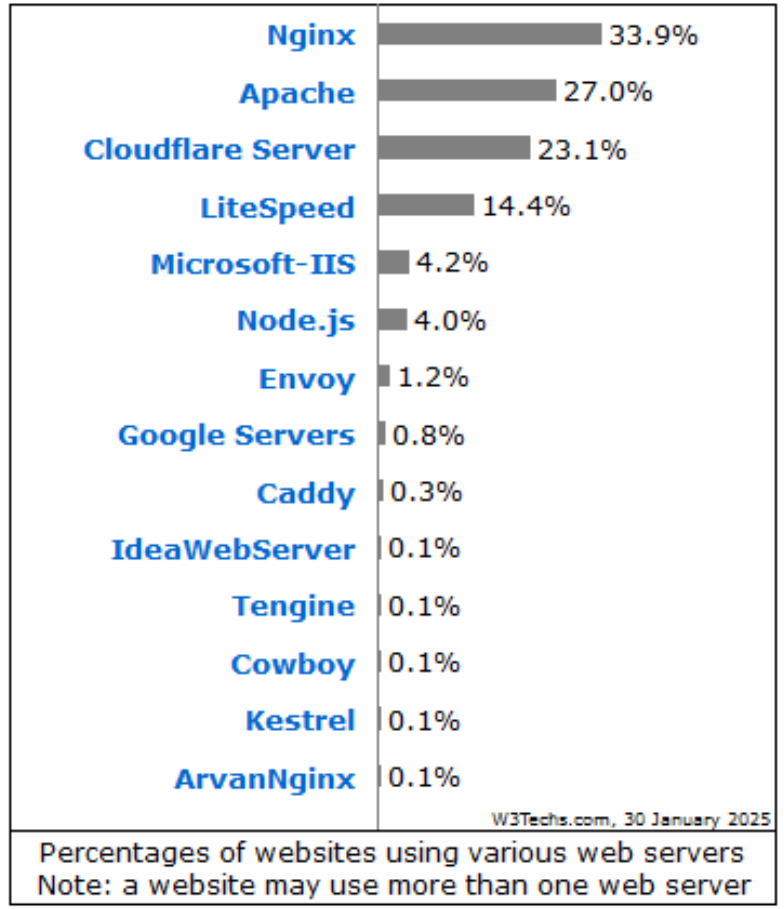
The Apache Software Foundation

HÖGSKOLAN VÄST

WEB SERVERS







https://w3techs.com/technologies/overview/web_server/all

HÖGSKOLAN VÄST

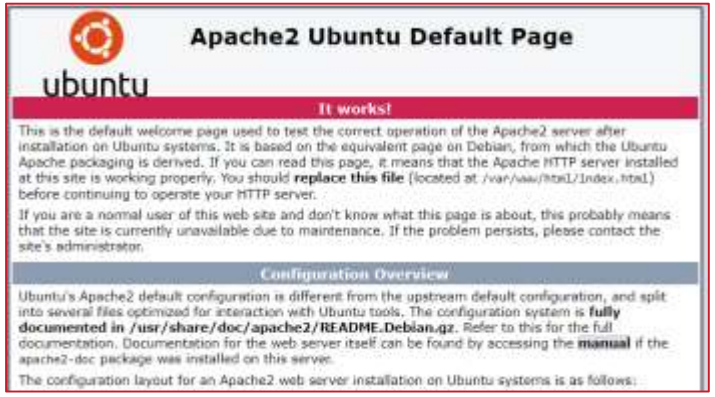
APACHE CONFIGURATION

- Deamon name **apache2** (/usr/sbin/apache2)
- Files: **apt install apache2**
  - /etc/apache2/apache2.conf
  - /etc/apache2/*/*
  - /var/log/syslog
  - /var/log/apache2/*
  - /var/www/*/*

CONTENTS

404 Not Found

- If no file is specified, **index.htm[l]** is used
  - www.hv.se/file.txt
  - www.hv.se/index.html
  - www.hv.se/
- If the file is not found in /var/www/html
  - The webserver shows */var/www/error/noindex.html*



GLOBAL SETTINGS

- ServerRoot ”/etc/apache2”
  - Points to the configuration file directory
- Listen 80
  Listen 443
  - “Default” TCP port to listen to
- LoadModule
  - Extra functionality (modularity)
- Directory
  - Options, Indexes, AllowOverride

VIRTUAL HOST CONFIGURATION BLOCK

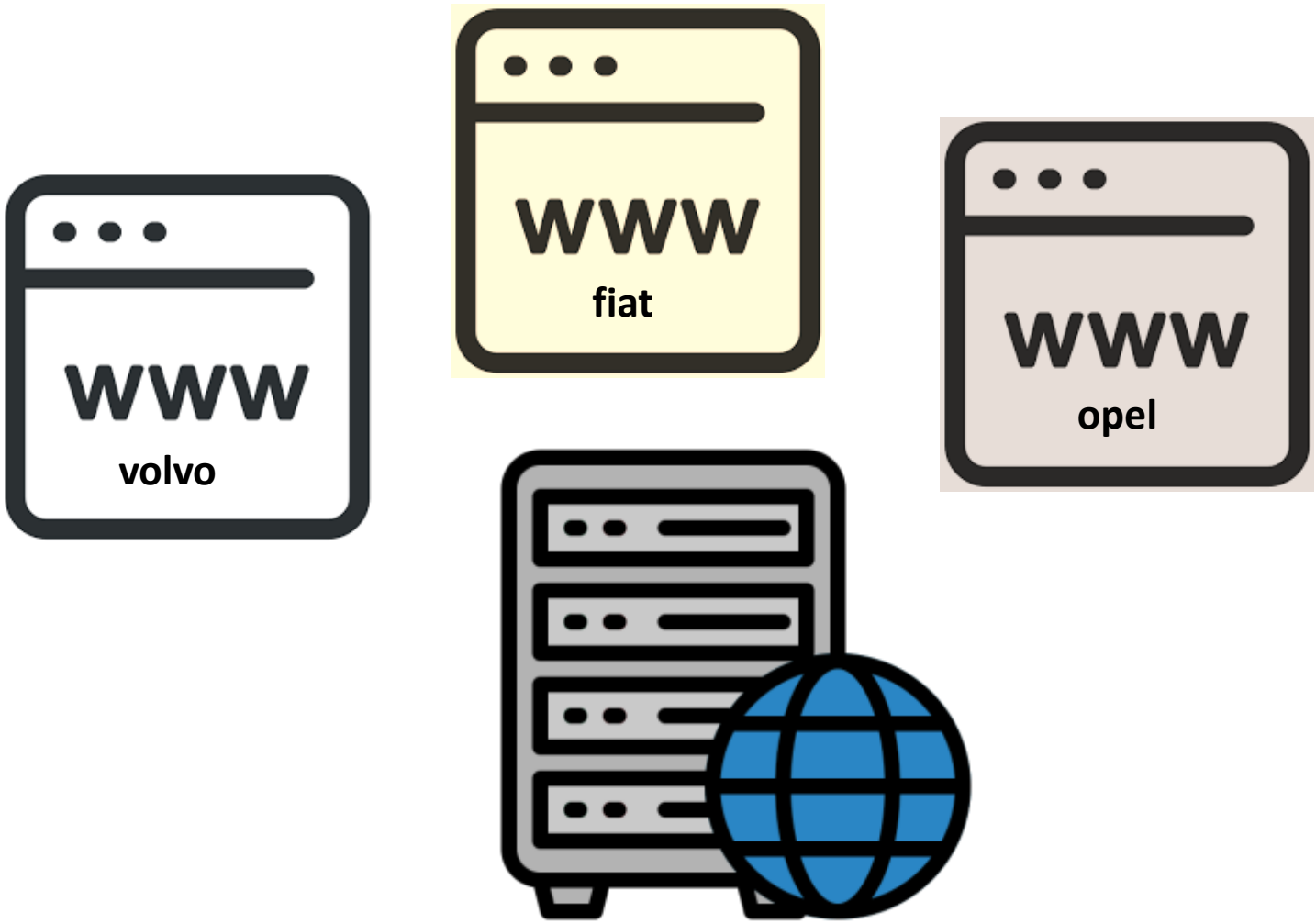
<VirtualHost *:80>	To handles the HTTP request to our virtual host
ServerAdmin asdf0001@student.hv.se	E-mail of the admin for error reporting
ServerName www.grp99.lab.hv.se	The name of our new created domain
DocumentRoot “/var/www/html”	Defines the root directory where the website's files
ErrorLog …	Location of the log server error
CustomLog …	Access log, records incoming requests.
</VirtualHost>	Closes the VH block the end

HÖGSKOLAN VÄST

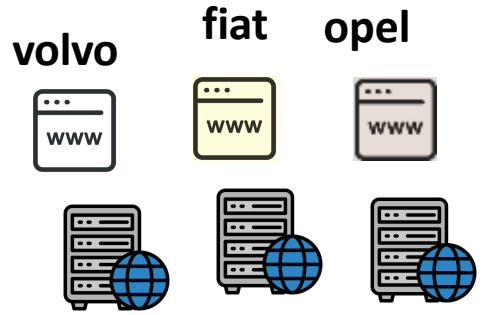
WEB HOSTING

HÖGSKOLAN VÄST

WEB HOSTING



DOMAIN–SERVER CONNECTION



Normal (old way): one server – one domain

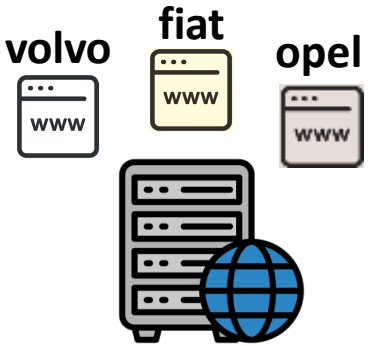
- Ex: *www.sunet.se* was one computer with one IP-address

If one server is not enough

- Load balancing between hardware
- DNS load-balancing (*www* is associated with several IP-addresses)

If one server is too much - one server can be responsible for several domains

- Better resource usage (at low loads)
- Web hotels



HÖGSKOLAN VÄST

SHARING DOMAINS ON ONE SERVER TWO+1 SOLUTIONS

1. IP-based virtual hosting
   - Virtual interfaces
   - Several IP-addresses for each computer
   - Each IP-address has its own *DNS name*

2. Name-based virtual hosting
   - Different *DNS names* points to the same IP-address
   - HTTP 1.1 “Host:” header line in request informs the server about the actual domain to serve

3. Port-based virtual hosting
   - Different ports lead to different web pages

kubernetes

HÖGSKOLAN VÄST

IP-BASED VIRTUAL HOSTING

Example: The server has two IP addresses (172.20.30.40 and 172.20.30.50) which resolve to the names www.site1.com and www.site2.org respectively.

```
Listen 80

<VirtualHost 172.20.30.40>
   DocumentRoot /var/www/site1
   ServerName www.site1.com
</VirtualHost>

<VirtualHost 172.20.30.50>
   DocumentRoot /var/www/site2
   ServerName www.site2.org
</VirtualHost>
```

IP-BASED VIRTUAL HOSTING (2)
ERROR

- **CREATE A VIRTUAL INTERFACE**
- The (first) networkcard is **/dev/eth0** or **/dev/ens18**

- It is bound to an IP-address in the file
  - /etc/netplan/…

- Create copies ”subinterfaces” …
  - ifcfg-eth0:0
  - ifcfg-eth0:1
  - ifcfg-eth0:2

PORT-BASED VIRTUAL HOSTING ERROR

Example: The server has one IP addresses, but map the names www.site1.com and www.site2.org to 8000 and 8080 respectively.

```
Listen 8000
Listen 8080

<VirtualHost *:8000>
        DocumentRoot /var/www/site1
        ServerName www.site1.com
</VirtualHost>

<VirtualHost *:8080>
        DocumentRoot /var/www/site2
        ServerName www.site2.org
</VirtualHost>
```

NAME BASED VIRTUAL HOSTS

- No need to touch the network interfaces
- Saves IP-addresses
- Requires HTTP/1.1 support

- Map all domains to same IP in the DNS-server (CNAME or multiple A records):

```
www.site1.com          IN A       ipnumber
www.site2.org          IN A       ipnumber
```

- **Better** use of DNS

```
www.site1.com          IN CNAME   server1337.lab.hv.se
www.site2.org          IN CNAME   server1337.lab.hv.se
server1337.lab.hv.se   IN A       ipnumber
```

HÖGSKOLAN VÄST

APACHE CONFIGURATION EXAMPLE

```
# Ensure that Apache listens on port 80
Listen 80
# Listen for virtual host requests on all IP addresses
NameVirtualHost *:80

<VirtualHost *:80>
        DocumentRoot /www/site1
        ServerName www.site1.com

        # Other directives here
</VirtualHost>

<VirtualHost *:80>
        DocumentRoot /www/site2
        ServerName www.site2.org

        # Other directives here
</VirtualHost>
```

SECURE HTTP

https://

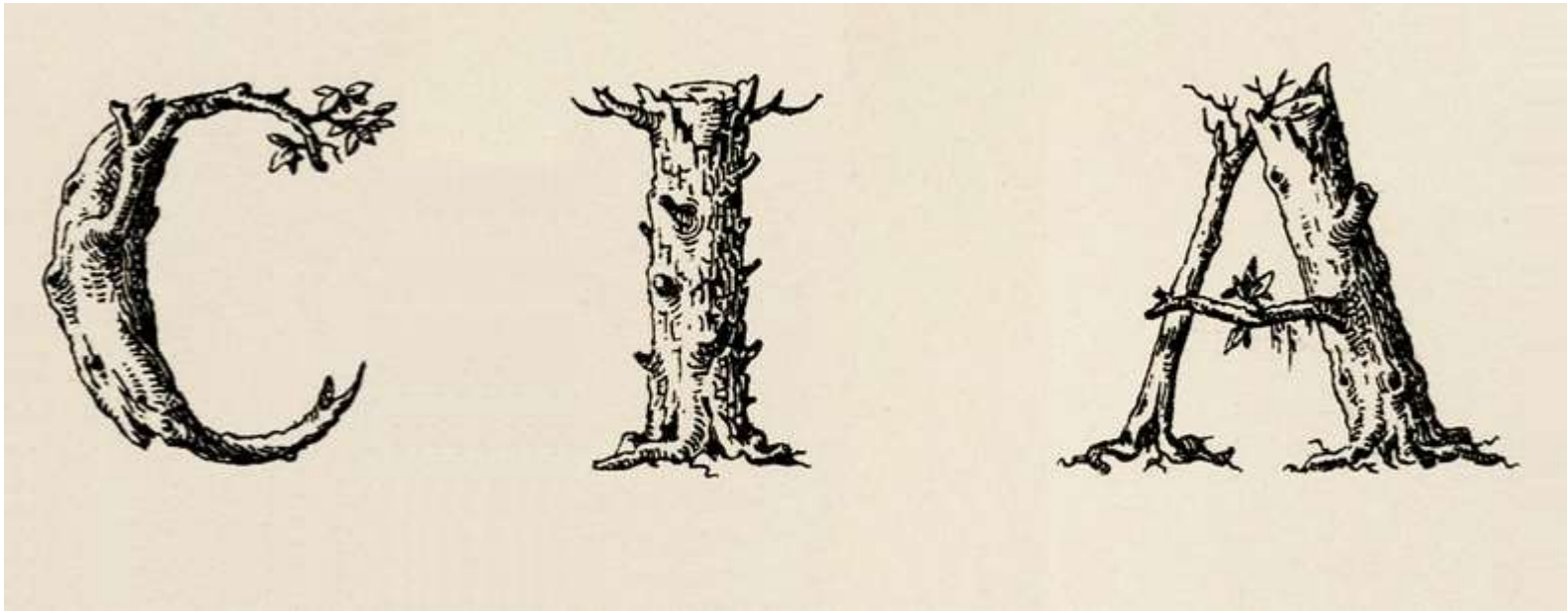
HÖGSKOLAN VÄST

HTTP vs HTTPS

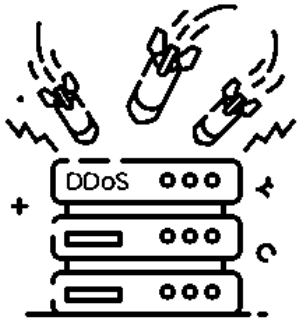
- **HTTP** is sending everything in clear text
- That is of course bad
  - Login information and passwords can be snooped
  - Credit card numbers can be spied on
  - Anyone who can be man-in-the-middle can inject data to your browser, for ex. Malware

- **HTTPS** uses SSL/TLS (Secure Sockets Layers)
  - Same as *ssh* does
  - Normally uses port 443 (instead of 80)

C I A



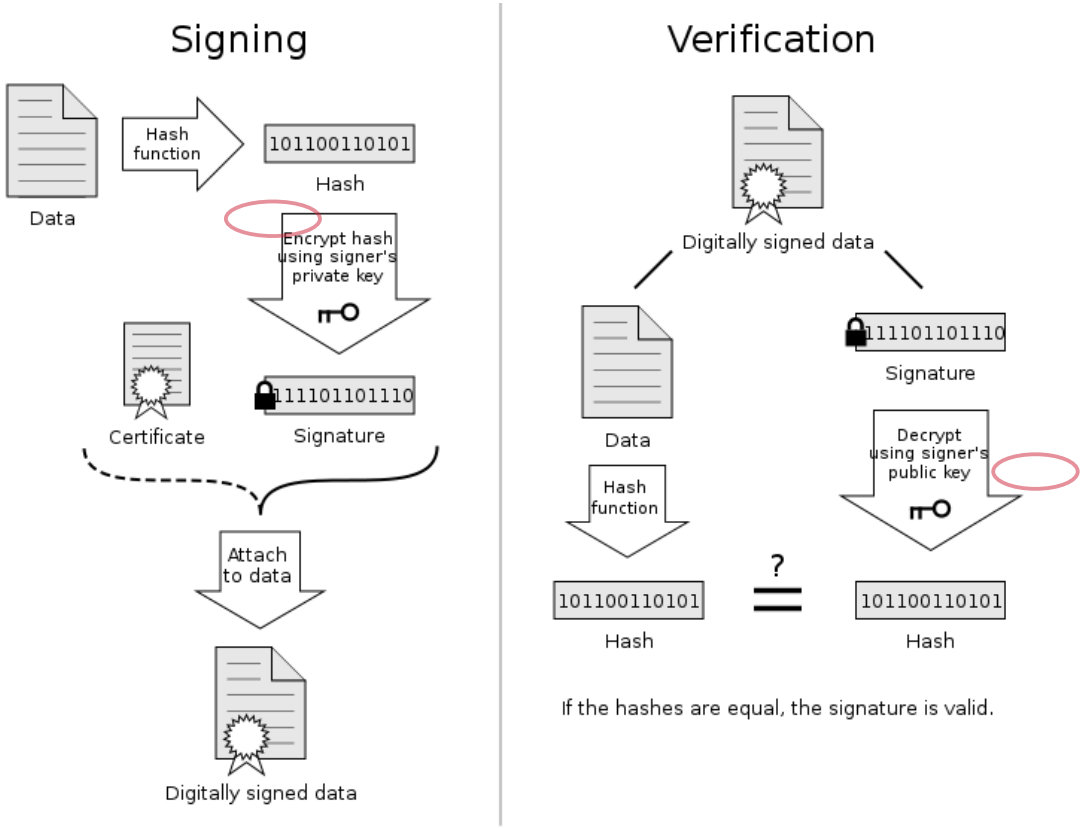
Read, **Write** or **Change,**



Attack

PUBLIC & PRIVATE IS A PAIR

Asymmetric keypair (not symmetric)



http://en.wikipedia.org/wiki/Digital_signature

HÖGSKOLAN VÄST

LEVELS OF SECURITY

- Unregistered (self signed) HTTPS-server
  - Traffic is encrypted
- Registered (signed) HTTPS-server
  - Server is also authenticated

… which really means that it has access to the unique private key guaranteed by a **trusted third party** to belong to a certain organization or company…

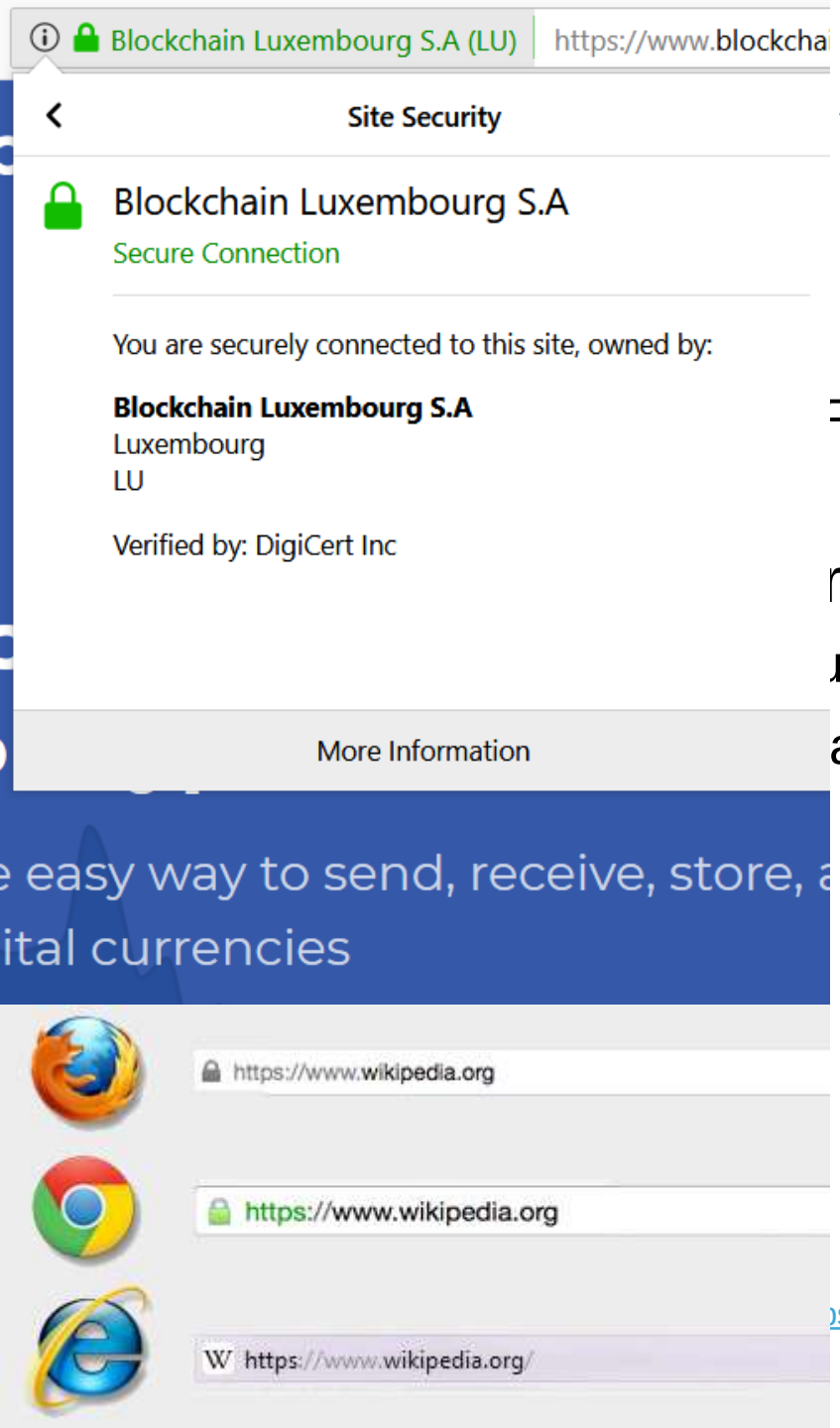
REGISTERING PROCESS

1. The owner of a web site
   - Generate an CSR – Certificate Signing Request

2. A trusted source (CA – Certificate Authority)
   - Signs the certificate and sends it back

3. Web browser or OS have built in lists of CA:s
   - E.g. Verisign, DigiCert, Addtrust, Globalsign

WEB BROWSER



on a secure page

ad lock icon

rom a trusted source or not

ured URL-field

arning

s://blog.easynews.com/http-vs-https-whats-the-difference/

HÖGSKOLAN VÄST

GENERATING AND SELF SIGNING A CERT

openssl

- https://www.sslshopper.com/article-most-common-openssl-commands.html

```
openssl req -x509 -nodes -days 365 -sha256 \
  -newkey rsa:2048 -keyout privateKey.key \
  -out certificate.csr
```

- Important information:
  - Common name: *machinename.grpX.lab.hv.se*
  - Email address

SNMP

Simple Network Management Protocol

HÖGSKOLAN VÄST

SNMPWALK

```
root@bt:/# snmpwalk -v 1 -c private -On 192.168.1.1
.1.3.6.1.2.1.1.1.0 = STRING: Cisco IOS Software, C3560 Software (C3560-
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Tue 30-Aug-05 14:19 by yenanh
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.9.1.615
.1.3.6.1.2.1.1.3.0 = Timeticks: (1830548590) 211 days, 20:51:25.90
.1.3.6.1.2.1.1.4.0 = STRING:
.1.3.6.1.2.1.1.5.0 = STRING:
.1.3.6.1.2.1.1.6.0 = STRING:
.1.3.6.1.2.1.1.7.0 = INTEGER: 6
.1.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
.1.3.6.1.2.1.2.1.0 = INTEGER: 38
.1.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
.1.3.6.1.2.1.2.2.1.1.2 = INTEGER: 2
```

172.

File Edit Setup Control Window Help

```
[root@ARNAG ]# snmpwalk -v2c -c Savvi148 172.20.1.25 .1.3.6.1.2.1.31.1.1.1.18
IF-MIB::ifAlias.1 = STRING:
IF-MIB::ifAlias.10 = STRING: DMZ
IF-MIB::ifAlias.15 = STRING: Net_LAN
IF-MIB::ifAlias.20 = STRING: Admin_LAN
IF-MIB::ifAlias.30 = STRING: Checkpoint_Failover
IF-MIB::ifAlias.40 = STRING: Internet
IF-MIB::ifAlias.50 = STRING: Savvi
IF-MIB::ifAlias.666 = STRING: DEAD_VLAN
IF-MIB::ifAlias.5137 = STRING:
IF-MIB::ifAlias.5138 = STRING:
IF-MIB::ifAlias.5139 = STRING:
IF-MIB::ifAlias.5140 = STRING:
IF-MIB::ifAlias.5141 = STRING:
IF-MIB::ifAlias.5142 = STRING:
IF-MIB::ifAlias.10101 = STRING: USDR1FW01 Eth-s4

[root@ARNAG ]# snmpwalk -v2c -c Savvi148 172.20.1.25 .1.3.6.1.2.1.2
IF-MIB::ifDescr.1 = STRING: Vlan1
IF-MIB::ifDescr.10 = STRING: Vlan10
IF-MIB::ifDescr.15 = STRING: Vlan15
IF-MIB::ifDescr.20 = STRING: Vlan20
IF-MIB::ifDescr.30 = STRING: Vlan30
IF-MIB::ifDescr.40 = STRING: Vlan40
IF-MIB::ifDescr.50 = STRING: Vlan50
IF-MIB::ifDescr.666 = STRING: Vlan666
IF-MIB::ifDescr.5137 = STRING: StackPort1
IF-MIB::ifDescr.5138 = STRING: StackSub-St1-1
IF-MIB::ifDescr.5139 = STRING: StackSub-St1-2
IF-MIB::ifDescr.5140 = STRING: StackPort2
IF-MIB::ifDescr.5141 = STRING: StackSub-St2-1
IF-MIB::ifDescr.5142 = STRING: StackSub-St2-2
IF-MIB::ifDescr.10101 = STRING: GigabitEthernet1/0/1
IF-MIB::ifDescr.10102 = STRING: GigabitEthernet1/0/2
IF-MIB::ifDescr.10103 = STRING: GigabitEthernet1/0/3
IF-MIB::ifDescr.10104 = STRING: GigabitEthernet1/0/4
IF-MIB::ifDescr.10105 = STRING: GigabitEthernet1/0/5
IF-MIB::ifDescr.10106 = STRING: GigabitEthernet1/0/6
IF-MIB::ifDescr.10107 = STRING: GigabitEthernet1/0/7
```

```
$ snmpget -v3 -l authPriv -u usr-md5-des -A authkey1 -X privkey1 demo.snmplabs.com IF-MIB::ifInOctets.1 IF-MIB::ifOutOctets.1
```

GET TABLE ROW

SNMPGET

```
from pysnmp.hlapi import *

iterator = getCmd(
    SnmpEngine(),
    UsmUserData('usr-none-none'),
    UdpTransportTarget(('demo.snmplabs.com', 161)),
    ContextData(),
    ObjectType(ObjectIdentity('IF-MIB', 'ifInOctets', 1)),
    ObjectType(ObjectIdentity('IF-MIB', 'ifOutOctets', 1))
)

errorIndication, errorStatus, errorIndex, varBinds = next(iterator)

if errorIndication:
    print(errorIndication)

elif errorStatus:
    print('%s at %s' % (errorStatus.prettyPrint(),
                        errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))

else:
    for varBind in varBinds:
        print(' = '.join([x.prettyPrint() for x in varBind]))
```

HÖGSKOLAN VÄST

https://pysnmp.readthedocs.io/en/latest/examples/hlapi/v3arch/asyncore/sync/manager/cmdgen/table-operations.html

```
$ snmpwalk -v3 -lauthPriv -u usr-md5-none -A authkey1 -X privkey1 demo.snmplabs.com IF-MIB::
```

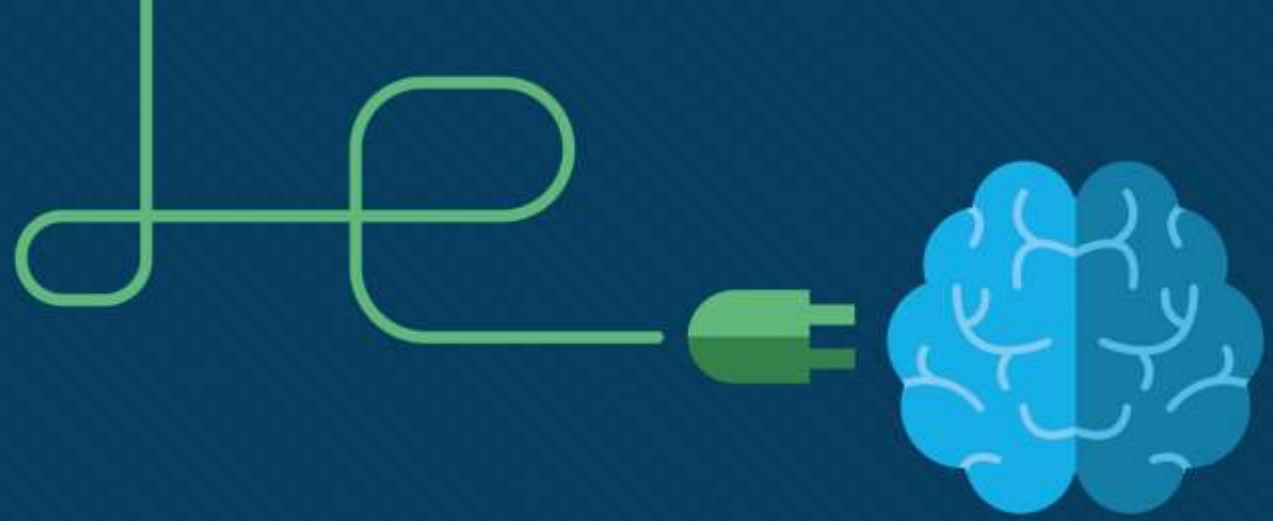
WALK WHOLE MIB

SNMPWALK

```
from pysnmp.hlapi import *

iterator = nextCmd(
    SnmpEngine(),
    UsmUserData('usr-md5-none', 'authkey1'),
    UdpTransportTarget(('demo.snmplabs.com', 161)),
    ContextData(),
    ObjectType(ObjectIdentity('IF-MIB'))
)

for errorIndication, errorStatus, errorIndex, varBinds in iterator:

    if errorIndication:
        print(errorIndication)
        break

    elif errorStatus:
        print('%s at %s' % (errorStatus.prettyPrint(),
                            errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
        break

    else:
        for varBind in varBinds:
            print(' = '.join([x.prettyPrint() for x in varBind]))
```

HÖGSKOLAN VÄST

https://pysnmp.readthedocs.io/en/latest/examples/hlapi/v3arch/asyncore/sync/manager/cmdgen/walking-operations.html



Module 10: Network Management

Johan Larsson et al

Enterprise Networking, Security, and Automation v7.01
(ENSA, ~8 slides)

10.4 SNMP

SNMP Operation

- SNMP agents that reside on managed devices collect and store information about the device and its operation locally in the MIB. The SNMP manager then uses the SNMP agent to access information within the MIB.
- There are two primary SNMP manager requests, get and set. In addition to configuration, a set can cause an action to occur, like restarting a router.

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table; the SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
get-bulk-request	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data. (Only works with SNMPv2 or later.)
get-response	Replies to a **get-request, get-next-request,** and **set-request** sent by an NMS.
set-request	Stores a value in a specific variable.

LÄS

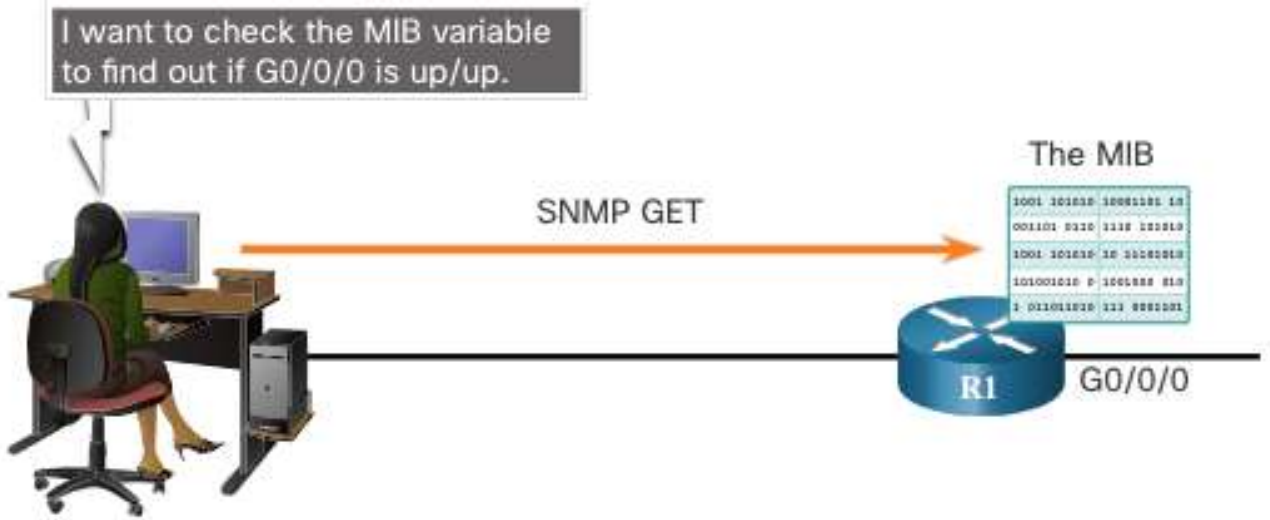
LÄS

SNMP Operation (Cont.)

The SNMP agent responds to SNMP manager requests as follows:

LÄS

- **Get an MIB variable** - The SNMP agent performs this function in response to a GetRequest-PDU from the network manager. The agent retrieves the value of the requested MIB variable and responds to the network manager with that value.
- **Set an MIB variable** - The SNMP agent performs this function in response to a SetRequest-PDU from the network manager. The SNMP agent changes the value of the MIB variable to the value specified by the network manager. An SNMP agent reply to a set request includes the new settings in the device.



SNMP Versions

- SNMPv1 - Legacy standard defined in RFC 1157. Uses a simple community-string based authentication method. Should not be used due to security risks.
- SNMPv2c - Defined in RFCs 1901-1908. Uses a simple community-string based authentication method. Provides for bulk retrieval options, as well as more detailed error messages.
- SNMPv3 - Defined in RFCs 3410-3415. Uses username authentication, provides data protection using HMAC-MD5 or HMAC-SHA and encryption using DES, 3DES, or AES encryption.

Community Strings (≈ lösenord-ish)

SNMPv1 and SNMPv2c use community strings that control access to the MIB. Community strings are plaintext passwords. SNMP community strings authenticate access to MIB objects.

There are two types of community strings:

- **Read-only (ro)** - This type provides access to the MIB variables, but does not allow these variables to be changed, only read. Because security is minimal in version 2c, many organizations use SNMPv2c in read-only mode.
- **Read-write (rw)** - This type provides read and write access to all objects in the MIB.

To view or set MIB variables, the user must specify the appropriate community string for read or write access.

MIB Object ID

The MIB organizes variables hierarchically. Formally, the MIB defines each variable as an object ID (OID). OIDs uniquely identify managed objects. The MIB organizes the OIDs based on RFC standards into a hierarchy of OIDs, usually shown as a tree.

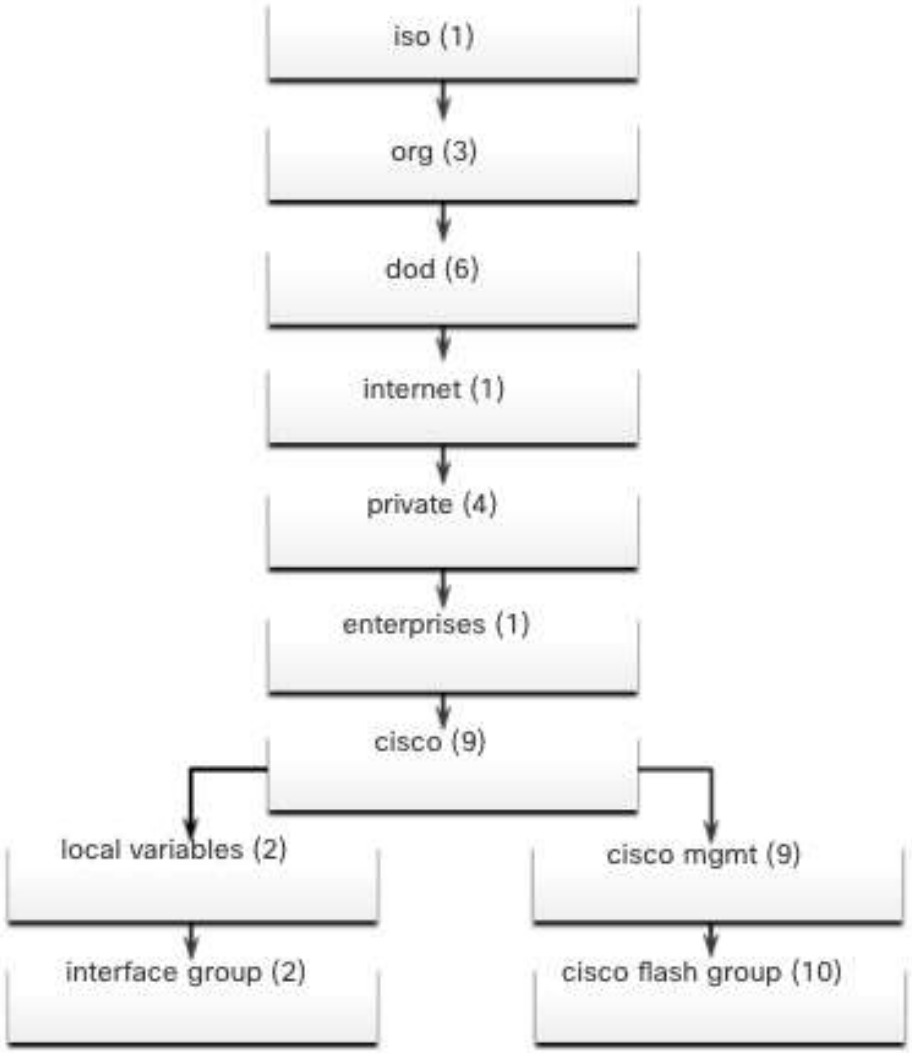
- The MIB tree for any given device includes some branches with variables common to many networking devices and some branches with variables specific to that device or vendor.
- RFCs define some common public variables. Most devices implement these MIB variables. In addition, networking equipment vendors, like Cisco, can define their own private branches of the tree to accommodate new variables specific to their devices.

MIB Object ID (Cont.)

The figure shows portions of the MIB structure defined by Cisco. Note how the OID can be described in words or numbers to help locate a particular variable in the tree.

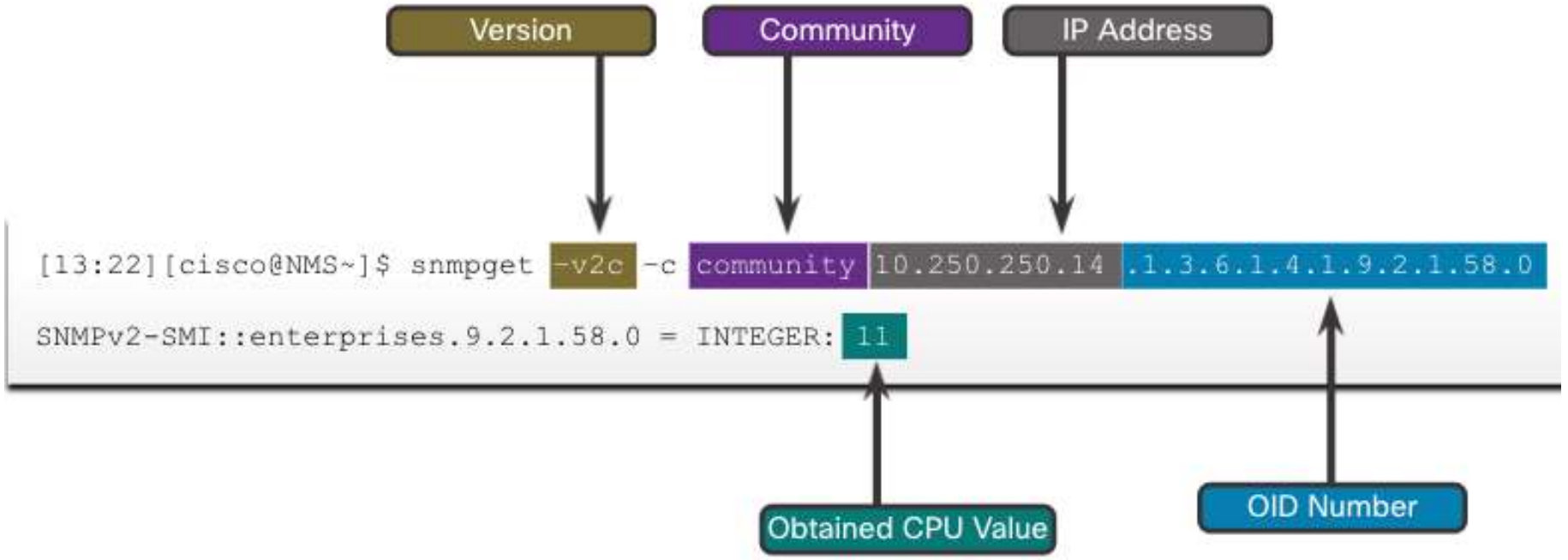
OIDs belonging to Cisco, are numbered as follows: .iso (1).org (3).dod (6).internet (1).private (4).enterprises (1).cisco (9).

Therefore, the OID is 1.3.6.1.4.1.9.



SNMP Polling Scenario

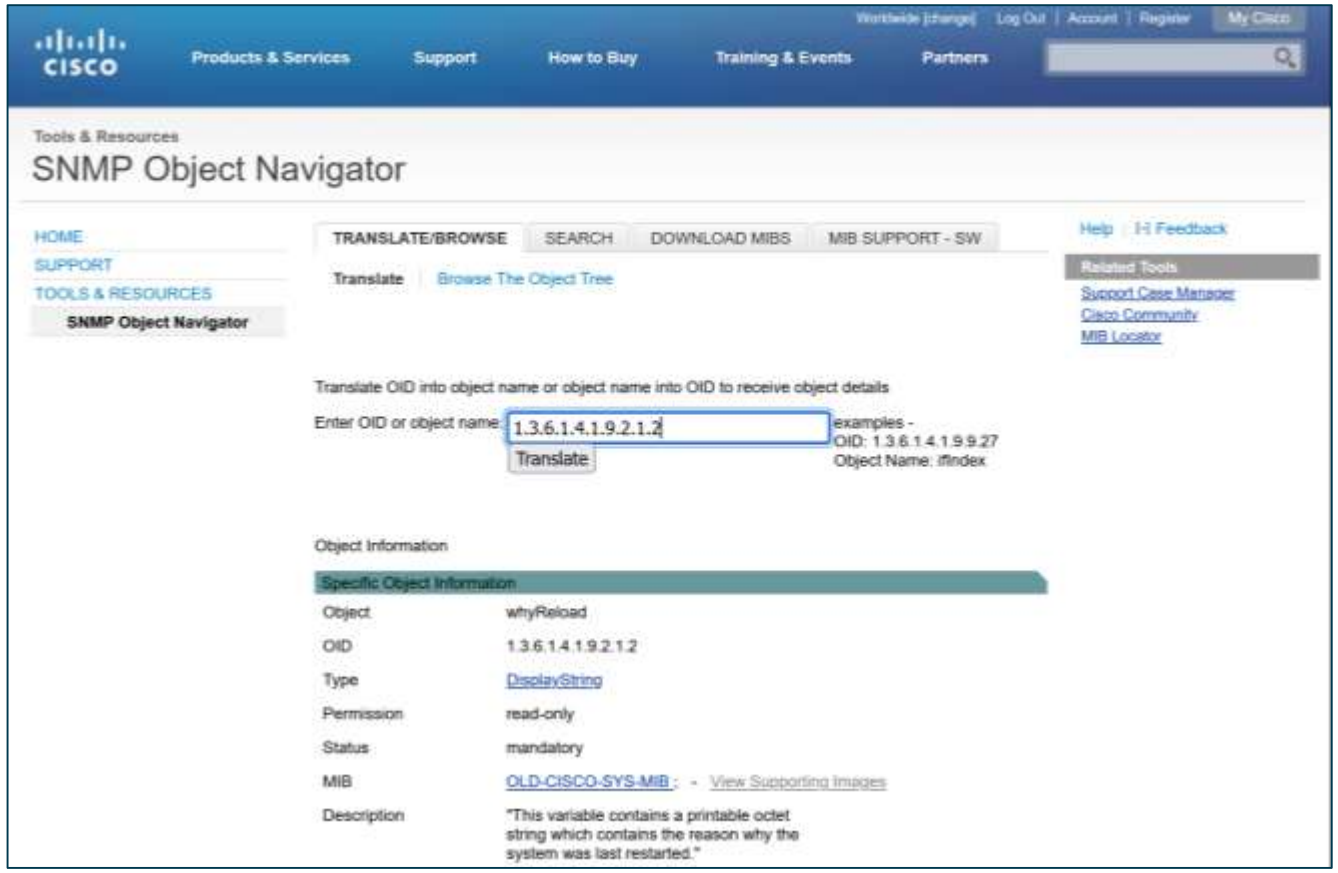
- SNMP can be used is to observe CPU utilization over a period of time by polling devices. CPU statistics can then be compiled on the NMS and graphed. This creates a baseline for the network administrator.
- The data is retrieved via the snmpget utility, issued on the NMS. Using the snmpget utility, you can manually retrieve real-time data, or have the NMS run a report. This report would give you a period of time that you could use the data to get the average.



SNMP Object Navigator

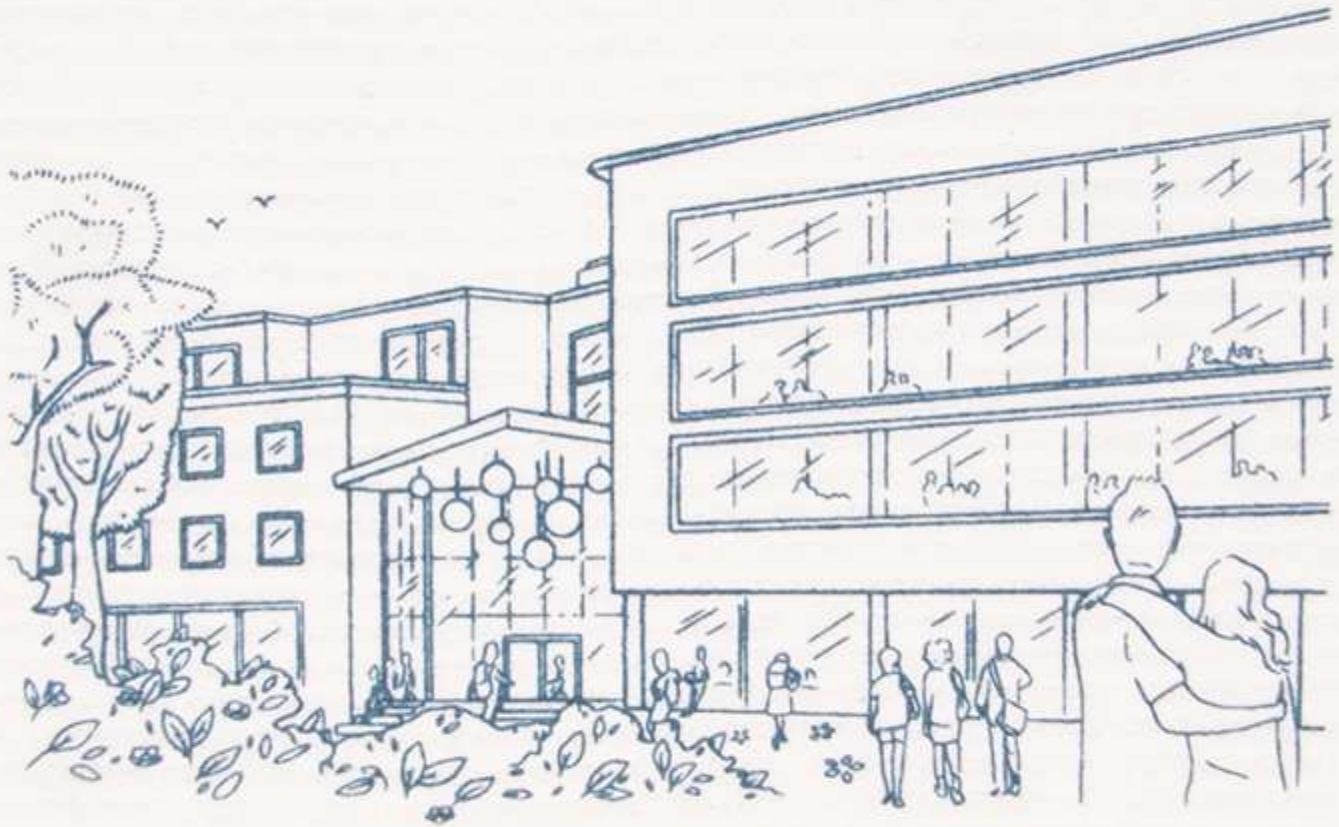
The *snmpget* utility gives some insight into the basic mechanics of how SNMP works.

However, working with long MIB variable names like 1.3.6.1.4.1.9.2.1.2 can be problematic for the average user. More commonly, the network operations staff uses a *network management product* with an easy-to-use GUI, which makes the entire MIB data variable naming transparent to the user.



The Cisco SNMP Navigator on the http://www.cisco.com website allows a network administrator to research details about a particular OID.

https://snmp.cloudapps.cisco.com/Support/SNMP/do/BrowseOID.do



UNIVERSITY WEST

End of Web

UNIVERSITY WEST

HÖGSKOLAN VÄST

HÖGSKOLAN VÄST

UNIVERSITY WEST

HÖGSKOLAN VÄST

UNIVERSITY WEST


UNIVERSITY WEST