William Stallings
Data and Computer Communications
chapter 19 - Distributed Applications

Data Communications and Networking

Chapter 19
Application Layer Protocols

Yang Xianchun
Department of Computer science and Technology
Nanjing University
xcy@nju.edu.cn
http://cs.nju.edu.cn/yangxc

Application Layer Protocols

- PDU Definition Language (OSI Presentation Layer)
  - ASN.1 (ISO 8824 and ITU-T X.680~X.683)
  - ASN.1 Encoding Rules (ISO 8825 and ITU-T X.690~X.693)
- OSI Application Layer
  - FTAM, MHS, VT, DS and CMIP
- TCP/IP Application Layer
  - SNMP
  - STMP and MIME
  - HTTP
  - DNS
  - FTP, TFTP and NFS
  - BOOTP and DHCP

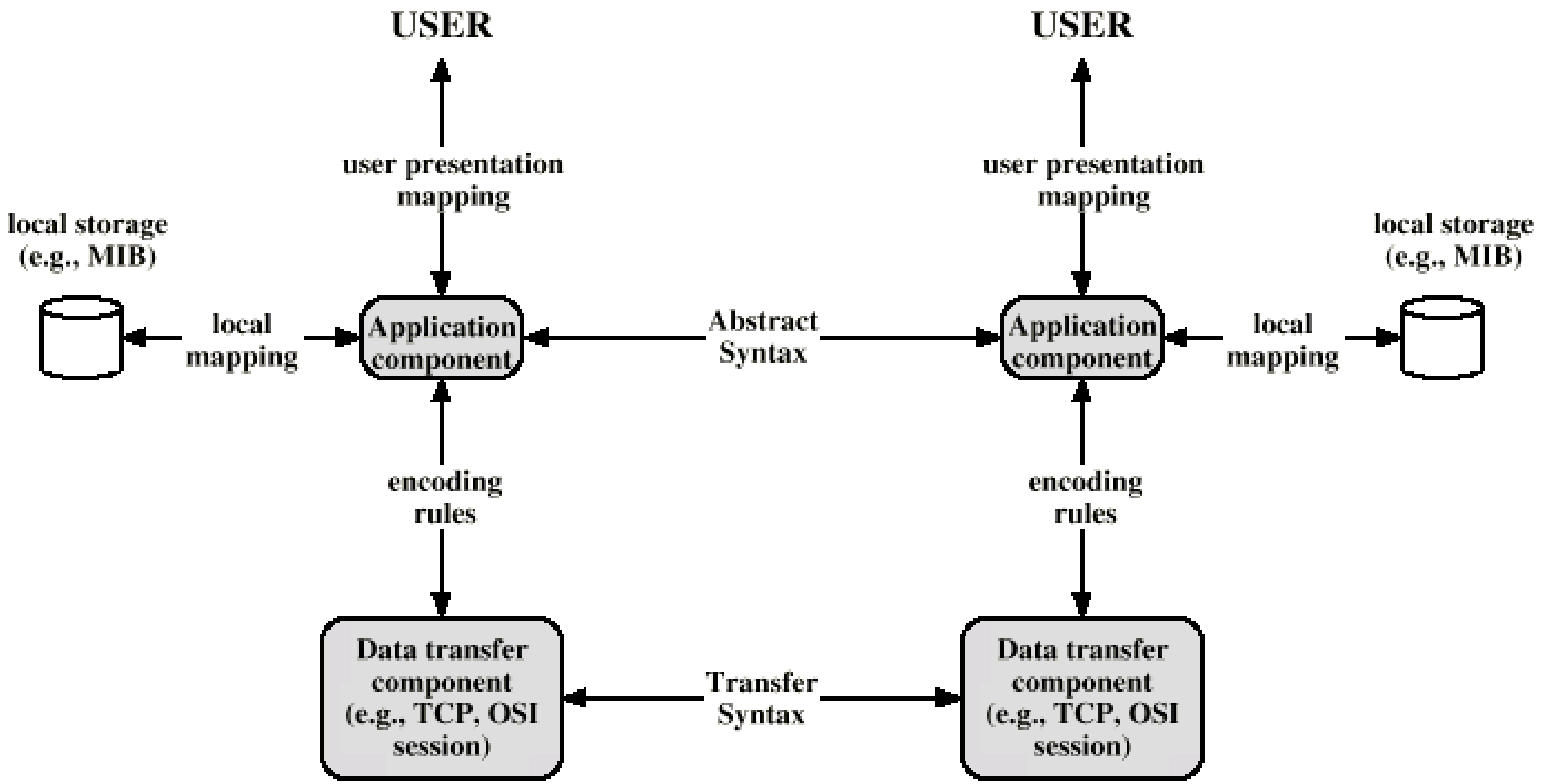
Abstract Syntax Notation One ASN.1

- Used to define format of PDUs
- Representation of distributed information
- Representation of operations performed on transmitted data

Terms Relevant to ANS.1

- Abstract Syntax
    - Describes generic structure of data
- Data Type
    - Named set of values
- Encoding
    - Sequence of octets used to represent data value
- Encoding Rules
    - Mapping from one syntax to another
- Transfer Syntax
    - Way data represented in bit patterns while in transit

Use of Abstract and Transfer Syntaxes


USER
USER
user presentation mapping
user presentation mapping
local storage (e.g., MIB)
local storage (e.g., MIB)
local mapping
Application component
Abstract Syntax
Application component
local mapping
encoding rules
encoding rules
Data transfer component (e.g., TCP, OSI session)
Transfer Syntax
Data transfer component (e.g., TCP, OSI session)

ASN.1 Concepts

- Abstract Syntax
  - Exchange of information between application components in different systems
  - Define the data elements in local storage
- Transfer Syntax
  - Define the representation of the data to be exchanged between data transfer components
  - Describer the data values in a binary form
- Application Component
  - Map the information represented using an abstract syntax into some form for presentation to the human user
  - Convert the data definition to a form suitable for local storage
  - Translate between the abstract syntax of application and a transfer syntax by means of encoding rules

Module definition

- Structured definition of a data structure using ASN.1
- Name of module used as abstract syntax name

Form of Modules

- <modulereference>DEFINITIONS::=

```
  BEGIN
     EXPORTS
     IMPORTS
     AssignmentsList
   End
```

- EXPORTS
  - Definitions which may be imported by other modules
- IMPORTS
  - Definitions to be imported from other modules
- AssignmentList
  - Type assignments, value assignments, macro definitions
  - <name>::=<description>

Lexical Conventions

- Layout not significant
- Comments delimited by pair of hyphens (--) at start and pair of hyphens or end of line end of comment
- Identifiers, type references and module names consist of upper and lower case letters, digits and hyphens
- Identifier starts with lower case letter
- Type reference or module name begins with upper case letter
- Built in type consists of all upper case letters

Abstract Data Types

- Collection of values
- Simple
  - Atomic
  - No components
- Structured
  - Has components
- Tagged
  - Derived from other types
- Other
  - Include CHOICE and ANY types (see later)

Tag Classes (1)

- Every data type (except CHOICE and ANY) has associated tag
- Universal (see Table 19.2)
  - Generally useful
  - Application independent
  - Defined in standard
    - Basic Type   e.g. Boolean, Integer, Real, Bit String
    - Object Type
    - Character String Type
    - Miscellaneous Type
    - Structured Type e.g. SEQUENCE OF , SET OF, SET

Tag Classes (2)

- Application wide
  - Relevant to particular application
- Context specific
  - Relevant to particular application
  - Applicable in limited context
- Private
  - User defined

CHOICE and ANY

- Data types without tags
  - When value assigned, type also assigned
  - Type assigned at run time
- CHOICE
  - List of alternative known types
  - Only one type used to create value
- ANY
  - Arbitrary value
  - Arbitrary type

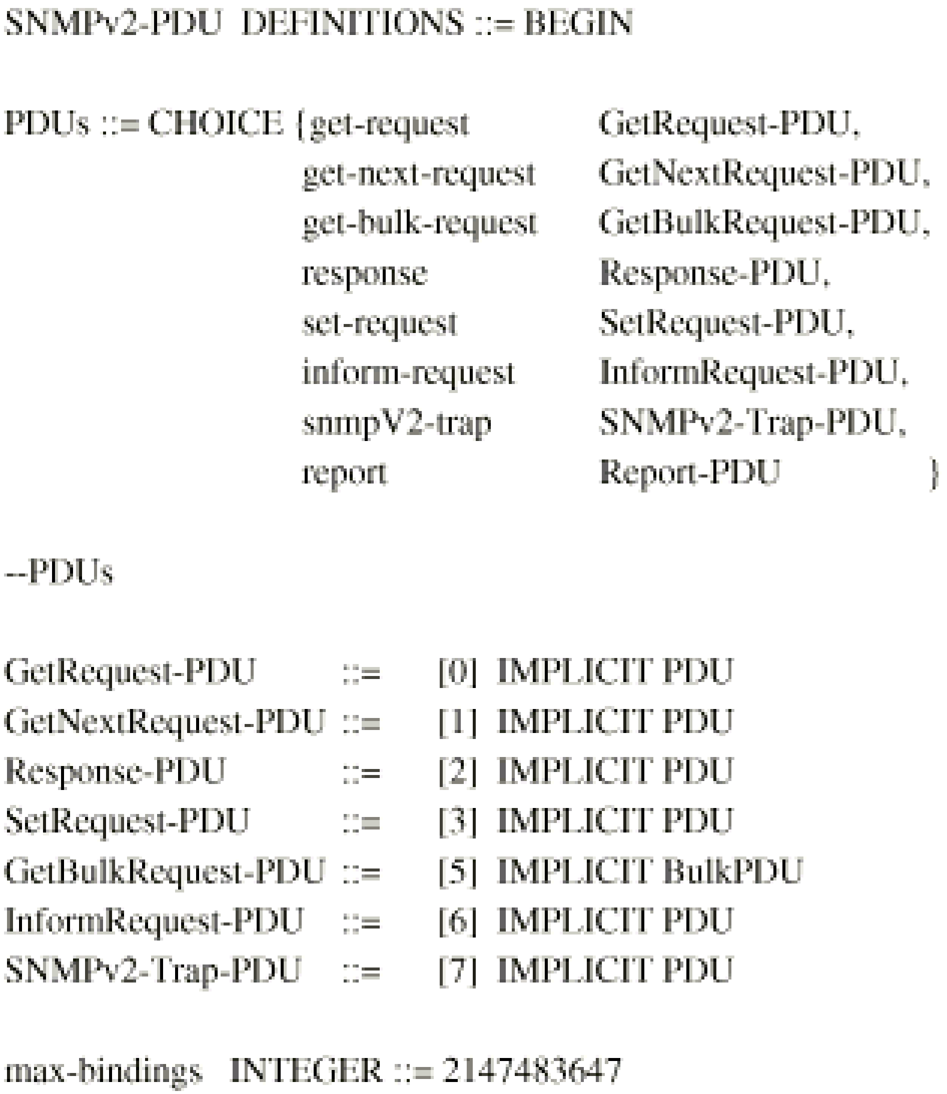
Subtypes (1)

- Derived from parent type
- Restricted subset of values
- May be nested
- Single value subtype
  - Explicit listing of all valid values
- Contained subtype
  - Used to form new subtype from existing subtypes
  - Includes all values of subtypes it contains
- Value range subtype
  - Real and Integer only
  - Specify endpoints of range

Subtypes (2)

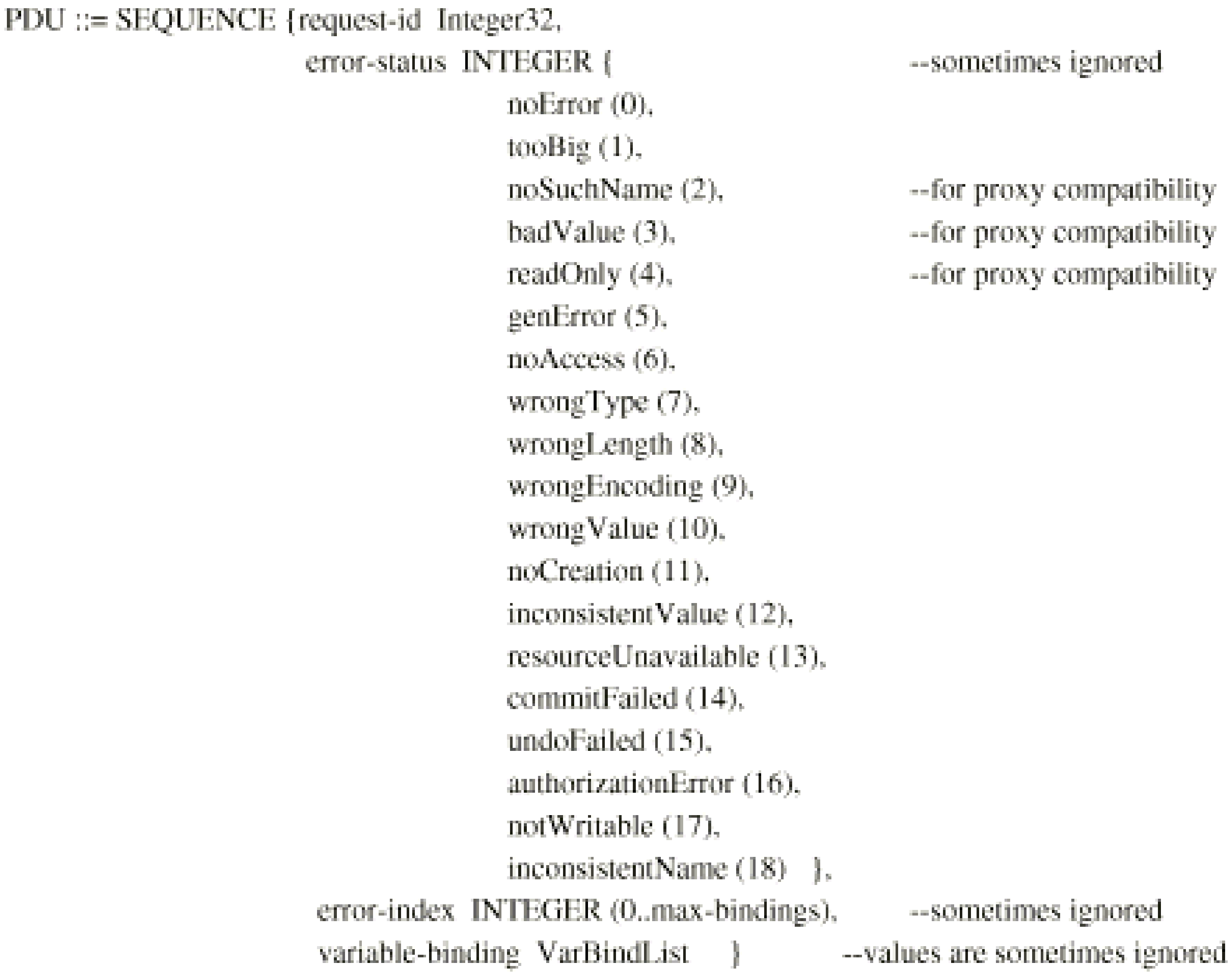
- Permitted alphabet constraint
  - Only character string
  - All values that can be constructed using sub-alphabet
- Size constrained
  - Limits number of items in type
  - e.g. number of bits in bit type
- Inner type constraint
  - Applied to SEQUENCE, SEQUENCE OF, SET, SET OF, CHOICE
  - Only values from parent that satisfy one or more constraints

PDU Example (part 1)



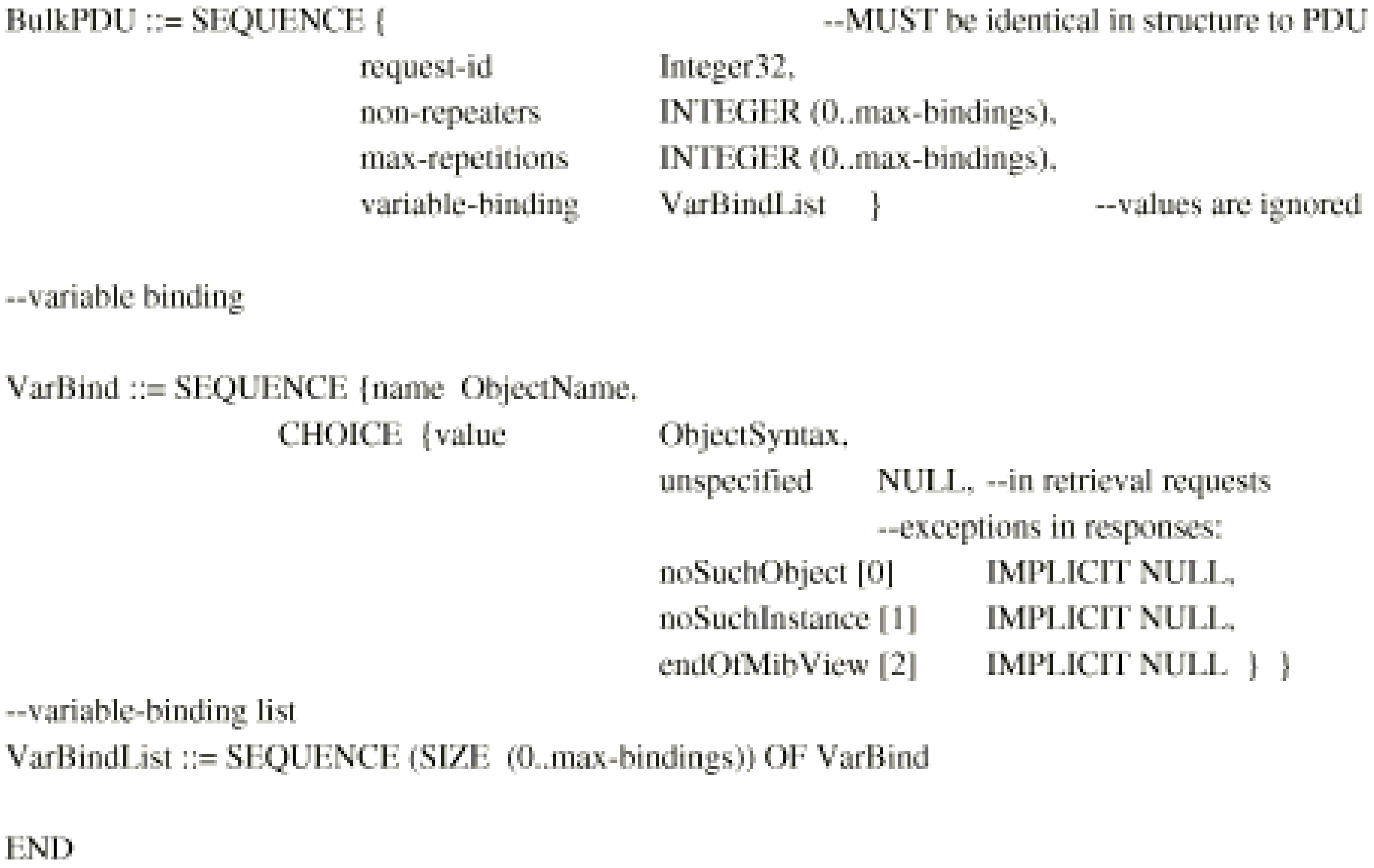
```
SNMPv2-PDU  DEFINITIONS ::= BEGIN

PDUs ::= CHOICE {get-request          GetRequest-PDU,
                 get-next-request     GetNextRequest-PDU,
                 get-bulk-request     GetBulkRequest-PDU,
                 response             Response-PDU,
                 set-request          SetRequest-PDU,
                 inform-request       InformRequest-PDU,
                 snmpV2-trap          SNMPv2-Trap-PDU,
                 report               Report-PDU          }

--PDUs

GetRequest-PDU       ::=    [0]  IMPLICIT PDU
GetNextRequest-PDU   ::=    [1]  IMPLICIT PDU
Response-PDU         ::=    [2]  IMPLICIT PDU
SetRequest-PDU       ::=    [3]  IMPLICIT PDU
GetBulkRequest-PDU   ::=    [5]  IMPLICIT BulkPDU
InformRequest-PDU    ::=    [6]  IMPLICIT PDU
SNMPv2-Trap-PDU      ::=    [7]  IMPLICIT PDU

max-bindings   INTEGER ::= 2147483647
```

PDU Example (part 2)



```
PDU ::= SEQUENCE {request-id  Integer32,
                  error-status  INTEGER {                         --sometimes ignored
                                noError (0),
                                tooBig (1),
                                noSuchName (2),                   --for proxy compatibility
                                badValue (3),                     --for proxy compatibility
                                readOnly (4),                     --for proxy compatibility
                                genError (5),
                                noAccess (6),
                                wrongType (7),
                                wrongLength (8),
                                wrongEncoding (9),
                                wrongValue (10),
                                noCreation (11),
                                inconsistentValue (12),
                                resourceUnavailable (13),
                                commitFailed (14),
                                undoFailed (15),
                                authorizationError (16),
                                notWritable (17),
                                inconsistentName (18)   },
                  error-index  INTEGER (0..max-bindings),         --sometimes ignored
                  variable-binding  VarBindList    }           --values are sometimes ignored
```

PDU Example (part 3)



```
BulkPDU ::= SEQUENCE {                                      --MUST be identical in structure to PDU
                        request-id           Integer32,
                        non-repeaters        INTEGER (0..max-bindings),
                        max-repetitions      INTEGER (0..max-bindings),
                        variable-binding     VarBindList   }              --values are ignored

--variable binding

VarBind ::= SEQUENCE {name  ObjectName,
                  CHOICE  {value             ObjectSyntax,
                                             unspecified      NULL,  --in retrieval requests
                                                              --exceptions in responses:
                                             noSuchObject [0]         IMPLICIT NULL,
                                             noSuchInstance [1]       IMPLICIT NULL,
                                             endOfMibView [2]         IMPLICIT NULL   }  }
--variable-binding list
VarBindList ::= SEQUENCE (SIZE  (0..max-bindings)) OF VarBind

END
```

Application Layer of OSI

- Application layer divided into a lot of sub-layers and elements
- ASE - Application Service Elements
  - CASE - Common Application Service Elements
    - ACSE - Association Control Service Element
    - RTSE - Reliable Transfer Service Element
    - ROSE - Remote Operation Service Element
  - SASE - Specific Application Service Elements
    - FTAM - File Transfer, Access and Management
    - MHS - Message Handling System
    - VT - Virtual Terminal
    - DS - Directory Services
    - CMIP - Common Management Information Protocol

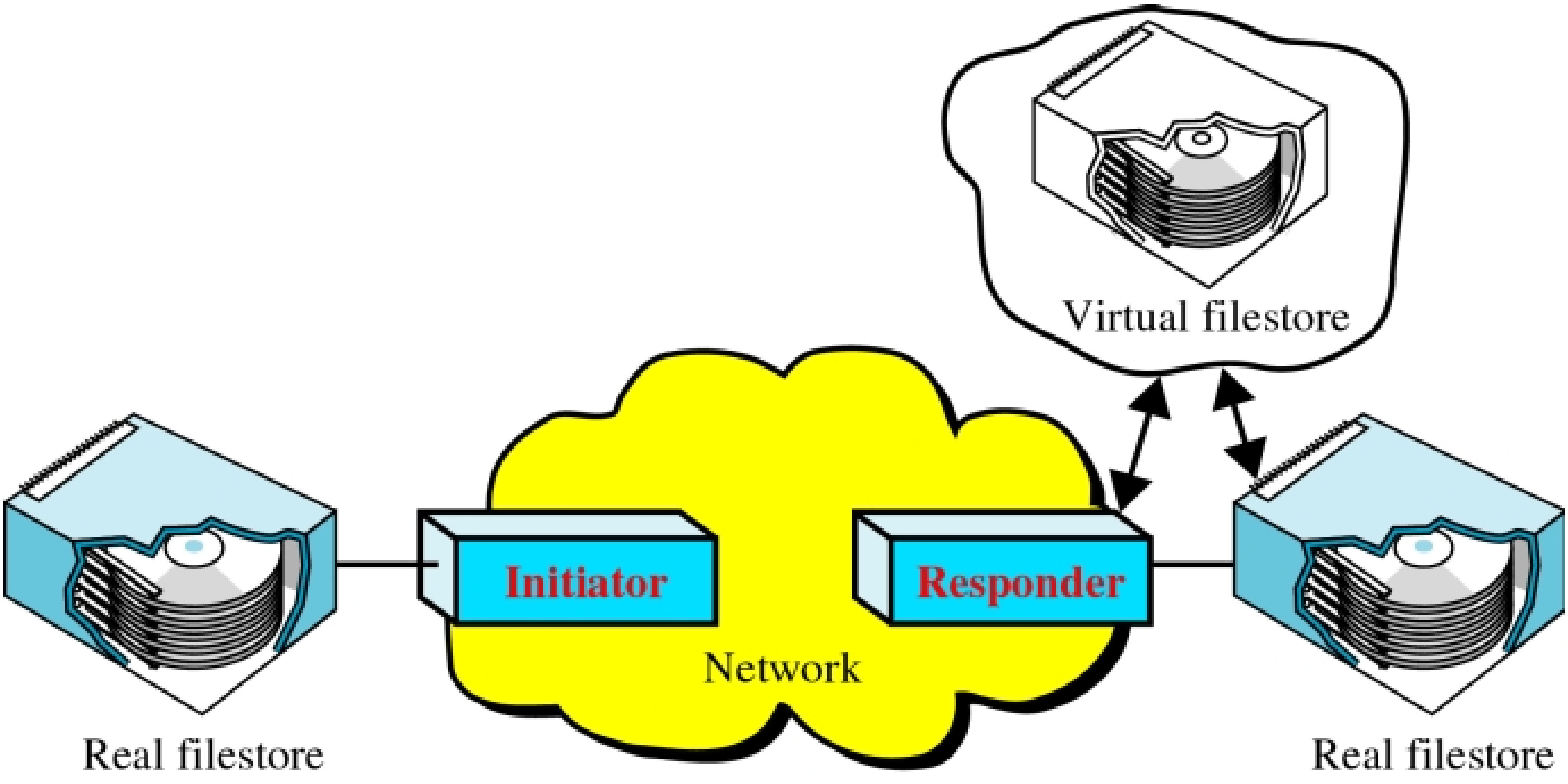
FTAM

- Consists of
  - Virtual Filestores
  - Virtual File Services
  - File Protocol Specification
- Virtual Filestore
  - Non-implementation-specific model for files and databases
  - Can be used as an intermediary for file transfer, access, and management
  - The Concept is similar to that behind ASN.1 for data

Virtual File Model

- FTAM is based on **asymmetric access** of a virtual file
  - Each transaction requires an initiator and a responder
  - Initiator requests FTAM of a file from responder
  - Responder create a virtual file model of its actual file and allows initiator to use the virtual model rather than real file
- Virtual file model is software
  - Can be designed to be independent of hardware and operating system
  - Also create a secure separation between the file that initiator is allowed access to and others in the same real storage

Virtual File Storage



Attribute and Content

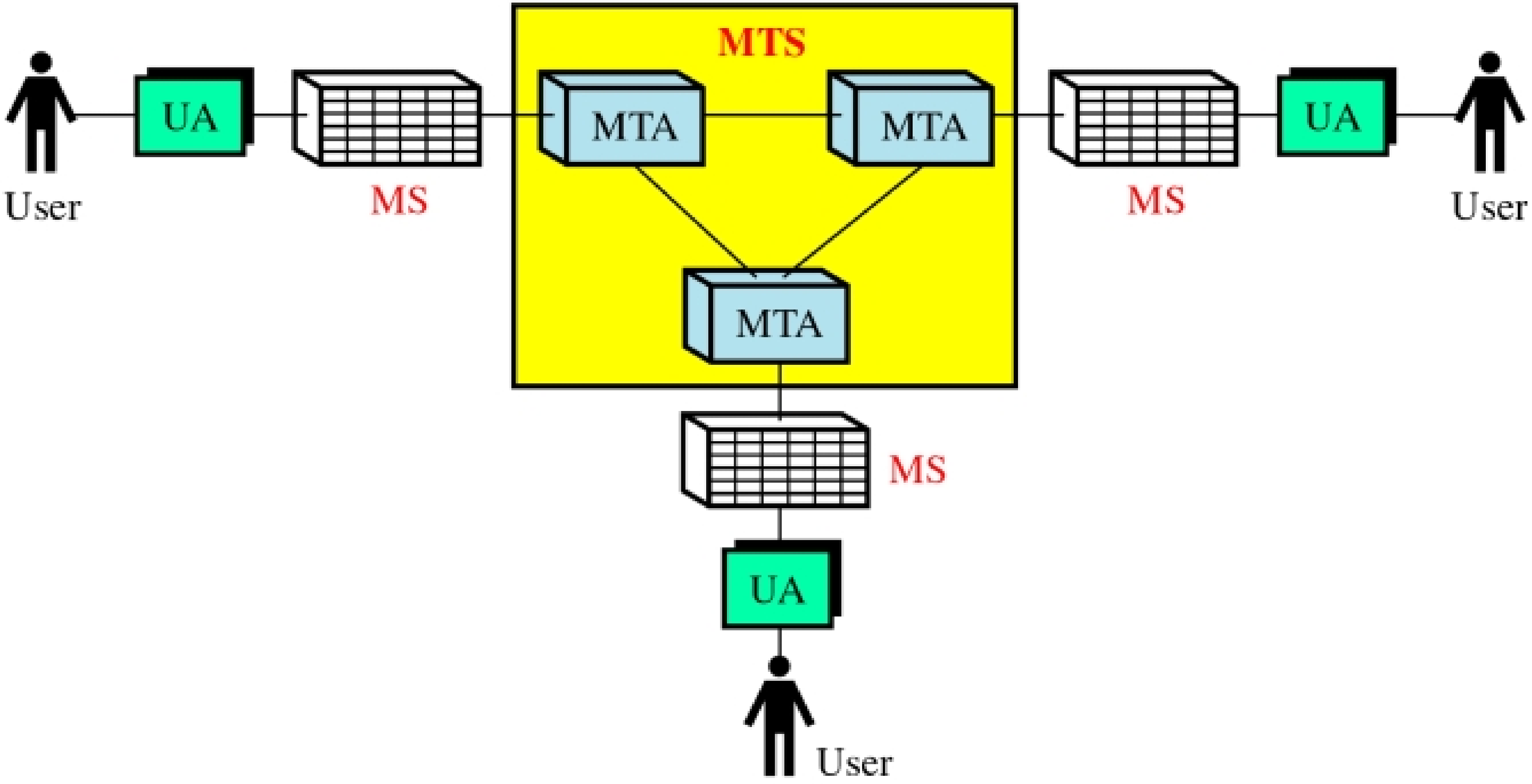
- Creation of a virtual filestore based on two aspects of the file
    - Attribute
        - a set of properties or security measures used to control either the contents or access
    - Content
- FTAM distinguishes between two different types of attribute
    - Per-content
        - related to the contents of the file
    - Per-access
        - Security measures that control access to the file

MHS

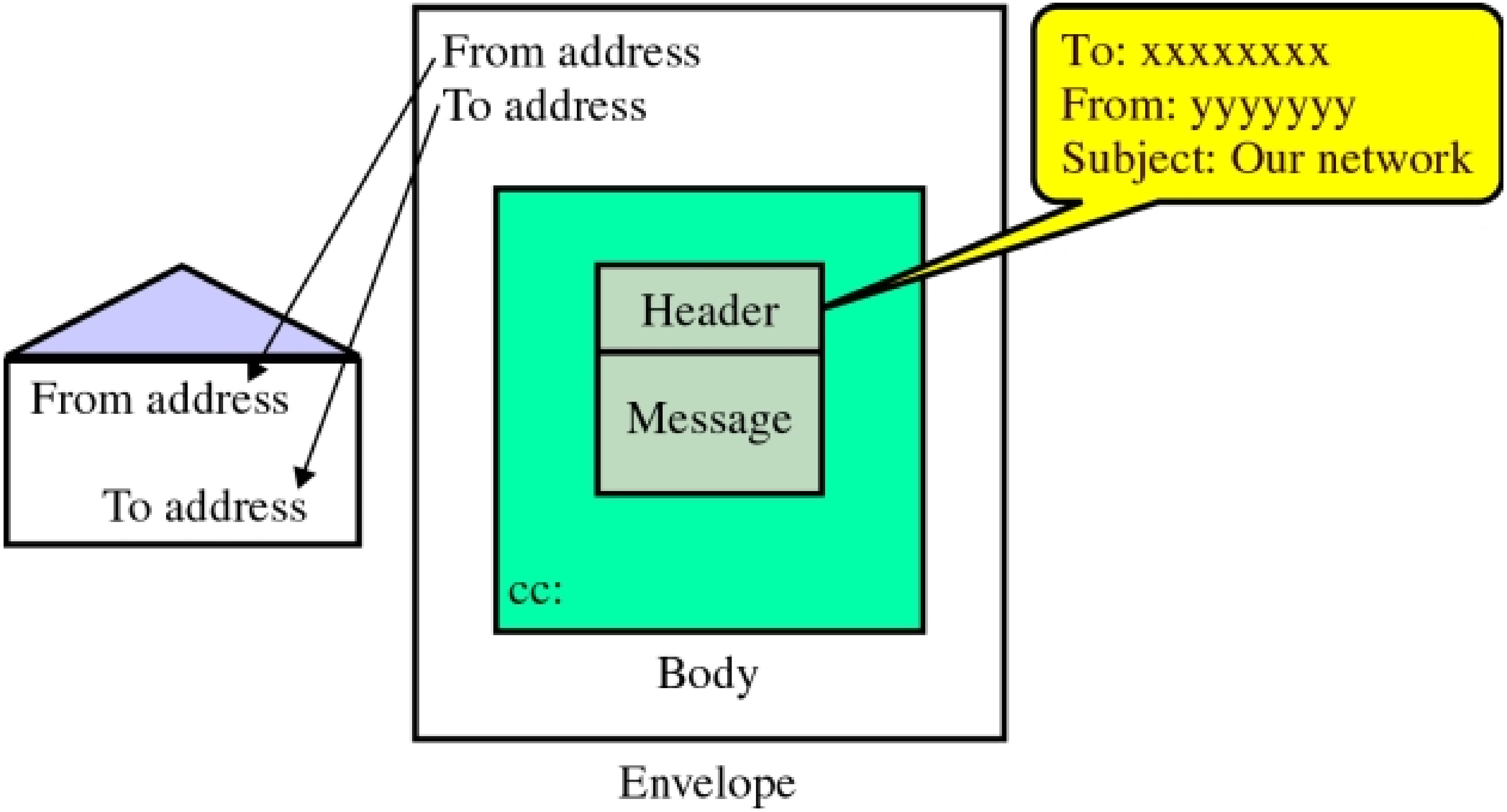
- MHS is an OSI protocol
  - underlies electronic mail and store-and-forward handling
  - Derived from ITU-T X.400 series
  - Used to send any message that can be delivered in a store-and-forward manner
- Store-and-forward delivery
  - Provides a delivery service that forward the message when a link became available
  - Instead of opening an active channel between the sender and the receiver
  - Similar to regular postal system

Structure of MHS

MS	Message storage (mailbox)	MTA	Message transfer agent
MTS	Message transfer system	UA	User agent

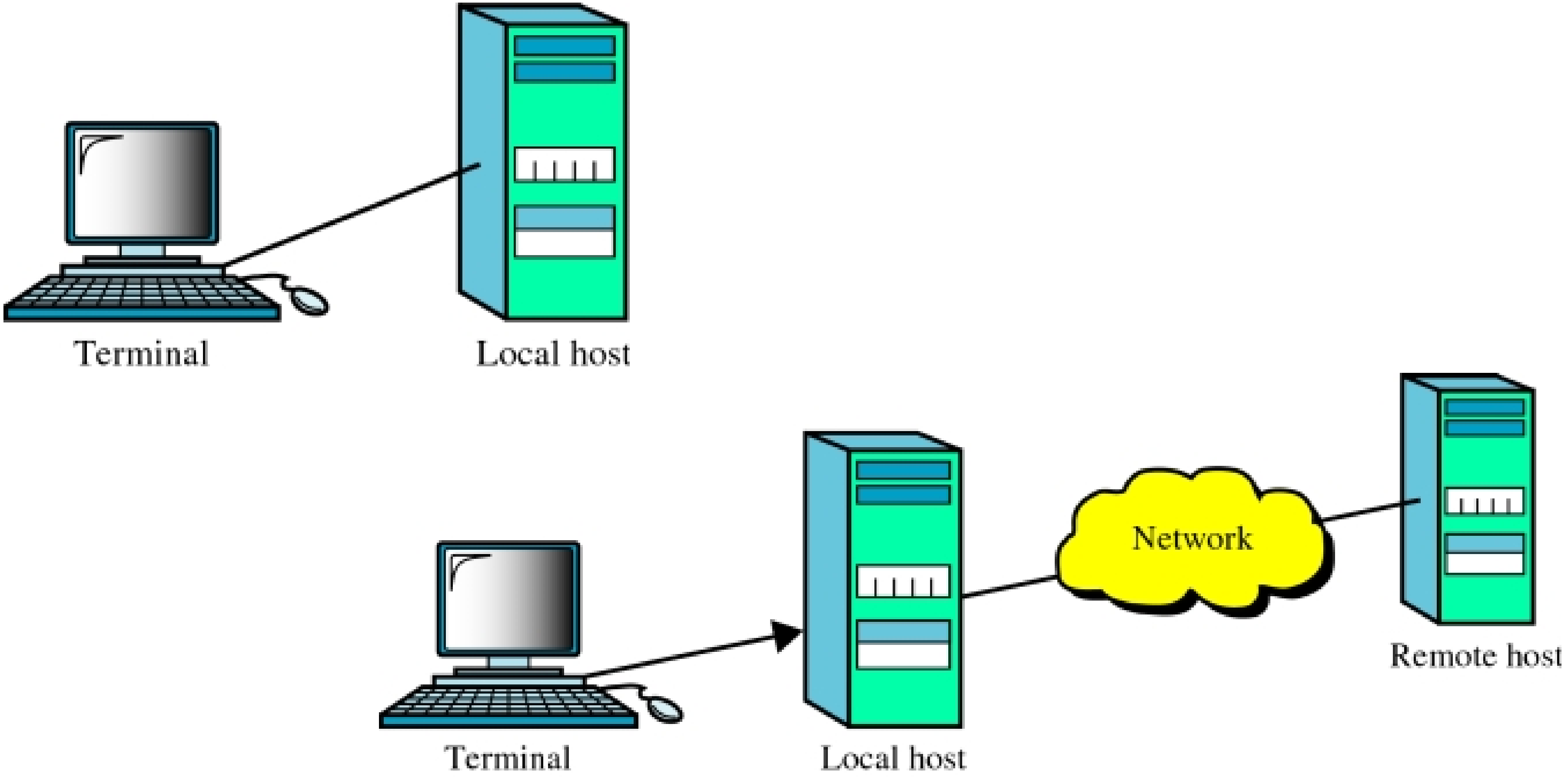


Message Format



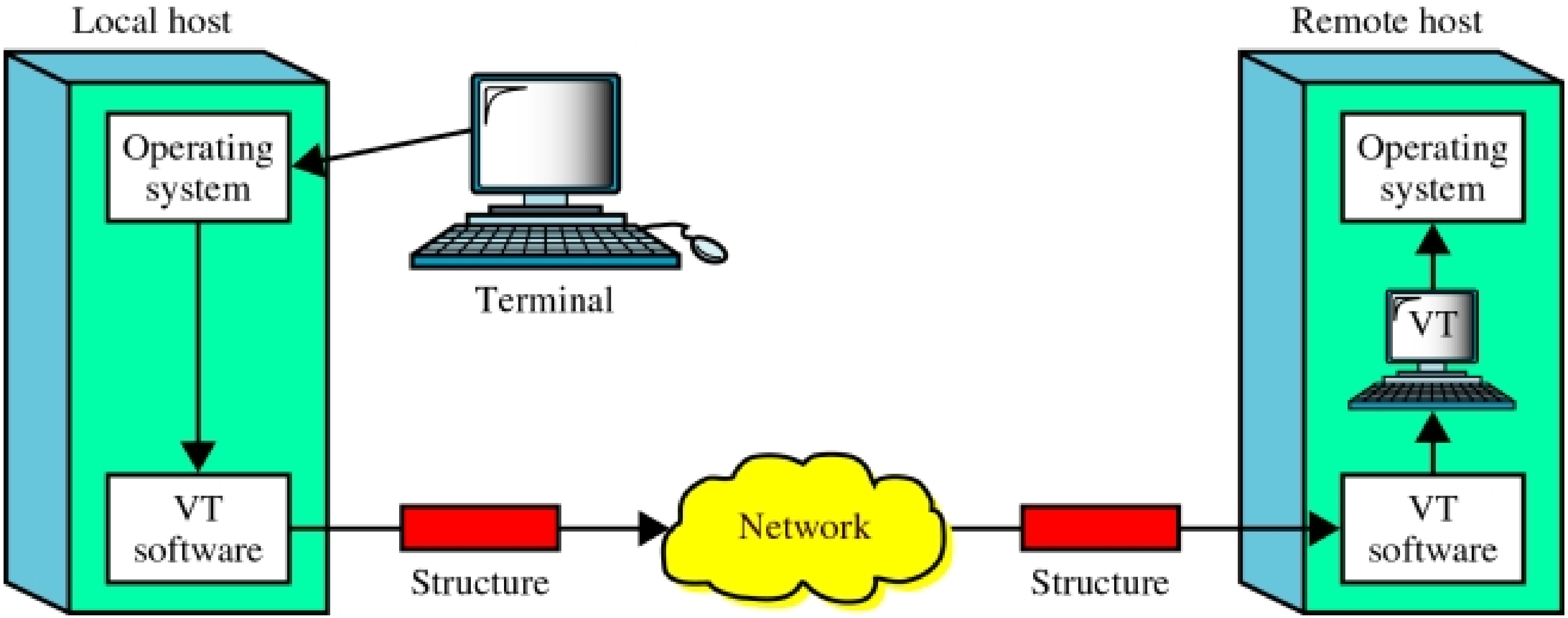
VT

- Local access and remote access



VT

- Virtual Terminal



Directory Services

- Designed according to ITU-T X.500 standard
- Directory
  - A global source of information about many different object
- OSI directory service
  - An application program used to represent and locate objects contained in an OSI directory
- Objects
  - people, organizations
  - logical groups, programs, files

Distributed Access Mechanism

- A directory is a distributed database
  - Each host holds only a part of the information
- Access mechanism of directory
  - Users know only one entry port from which all information may be retrieved
  - All of this information appears to be stored in a single database, located in a single host

Structure of Directory Service

- DIB - Directory Information Base
  - Stored as a set of entries, each describing a object
  - Entries structured as a tree with different levels
- DUA - Directory User Agent
  - Passes a request for information to a DSA
- DSA - Directory System Agent
  - If the DSA knows the whereabouts of information sought, it either fills the request or passes it onto another DSA with the necessary access, and so on
  - The requested information is retrieved and passed back through the successive DSAs to the DAU
  - If a DSA dose not know how to fill a request, it has three options
    - forward, broadcast, and return a report of failure

Network Management

- Networks are becoming indispensable
- More complexity makes failure more likely
- Require automatic network management tools
- Standards required to allow multi-vendor networks
- Covering:
  - Services
  - Protocols
  - Management information base (MIB)

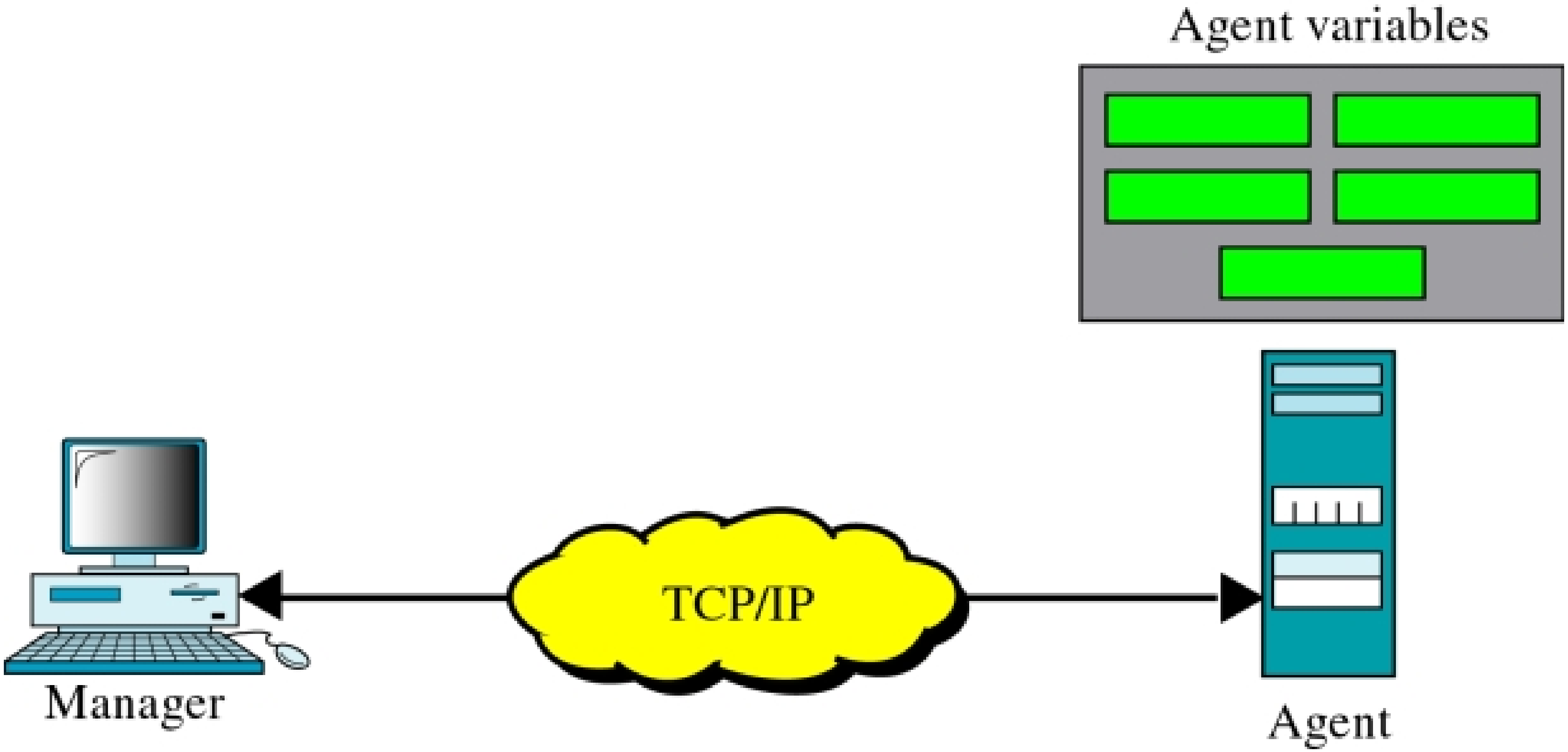
Network Management Systems

- Collection of tools for network monitoring and control
- Single operator interface
- Powerful, user friendly command set
- Performing most or all management tasks
- Minimal amount of separate equipment
  - i.e. use existing user equipment
- View entire network as unified architecture
- Active elements provide regular feedback

Key Elements

- Management station or manager
- Agent
- Management information base
- Network management protocol

Manager and Agent



Management Station

- Stand alone system or part of shared system
- Interface for human network manager
- Set of management applications
  - Data analysis
  - Fault recovery
- Interface to monitor and control network
- Translate manager's requirements into monitoring and control of remote elements
- Data base of network management information extracted from managed entities

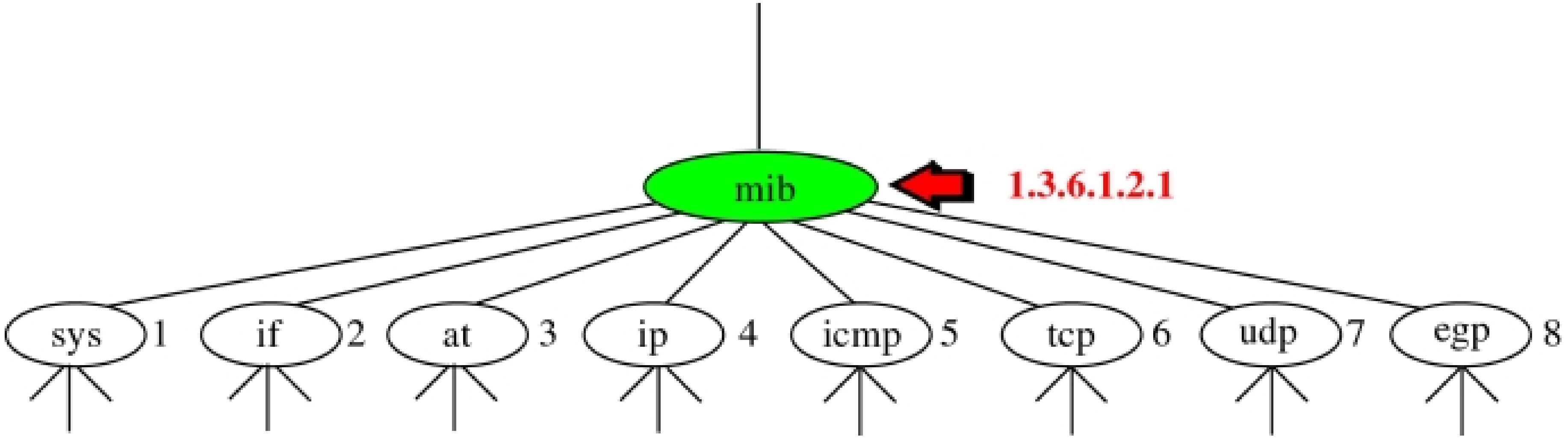
Agent

- Hosts, bridges, hubs, routers equipped with agent software
- Allow them to be managed from management station
- Respond to requests for information
- Respond to requests for action
- Asynchronously supply unsolicited information

Management Information Base

- MIB
- Representation of network resources as objects
- Each object is a data variable representing one aspect of managed object
- MIB functions is collection of access points at agent for management of station
- Objects standardized across class of system
  - Bridge, router etc.

MIB



Network Management Protocol

- Link between management station and agent
- TCP/IP uses Simple Network Management Protocol (SNMP)
- OSI uses Common Management Information Protocol (CMIP)
- SNMPv2 (enhanced SNMP) for OSI and TCP/IP

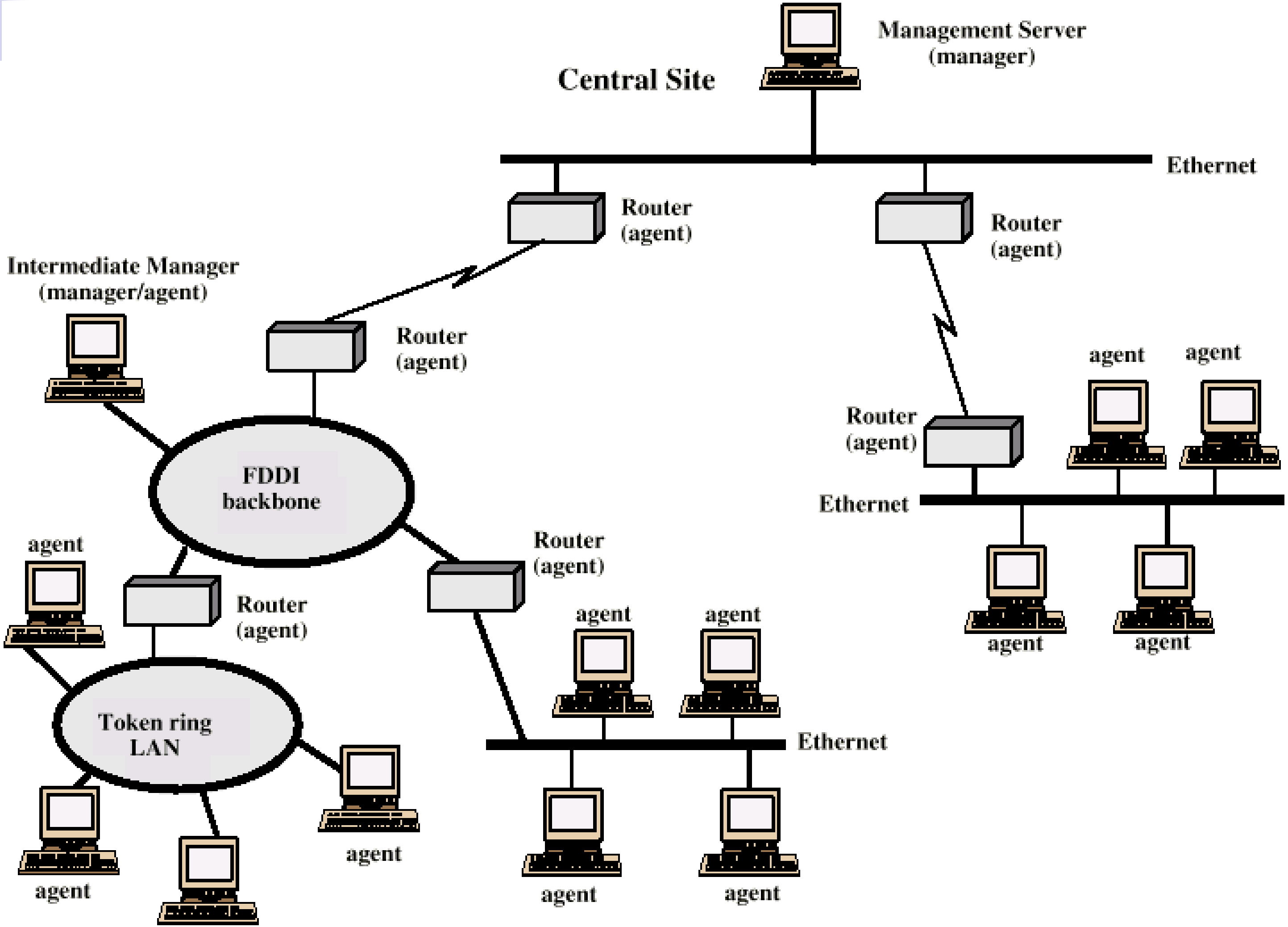
Protocol Capabilities

- Get
- Set
- Notify

Management Layout

- May be centralized in simple network
- May be distributed in large, complex network
  - Multiple management servers
  - Each manages pool of agents
  - Management may be delegated to intermediate manager

Example of Distributed Network Management Configuration



CMIP

- Developed by ISO and ITU-T working together
- Provides Common Management Information Services (CMIS)

CMIP Management

- Occurrence of all CMIP management
  - by monitoring and manipulating communication between OSI entities called managed objects
- Managed object
  - a network resource such as workstation, switch, routing hardware or software, queuing program
- CIMP allows two OSI management-service user
  - perform actions on managed objects
    - change their status for efficiency or testing purpose
  - collect data about the status of those objects
- Evaluate system performance and identify problems
  - by keeping logs of collected data
    - number of bytes of data processed by a router over given time
  - by changing managed object setting and monitoring response

CMIS

- Fault management
- Accounting management
- Configuration and name management
- Performance management
- Security management

CMISE

- Common Management Information Service Elements
- Three categories
  - Management Association Services
    - M-INITIALIZE, M-TERMINATE, M-ABORT
  - Management Notification Services
    - M-EVENT-REPORT
  - Management Operation Services
    - M-GET, M-CANCEL-GET, M-SET, M-ACTION, M-CREATE, M-DELETE

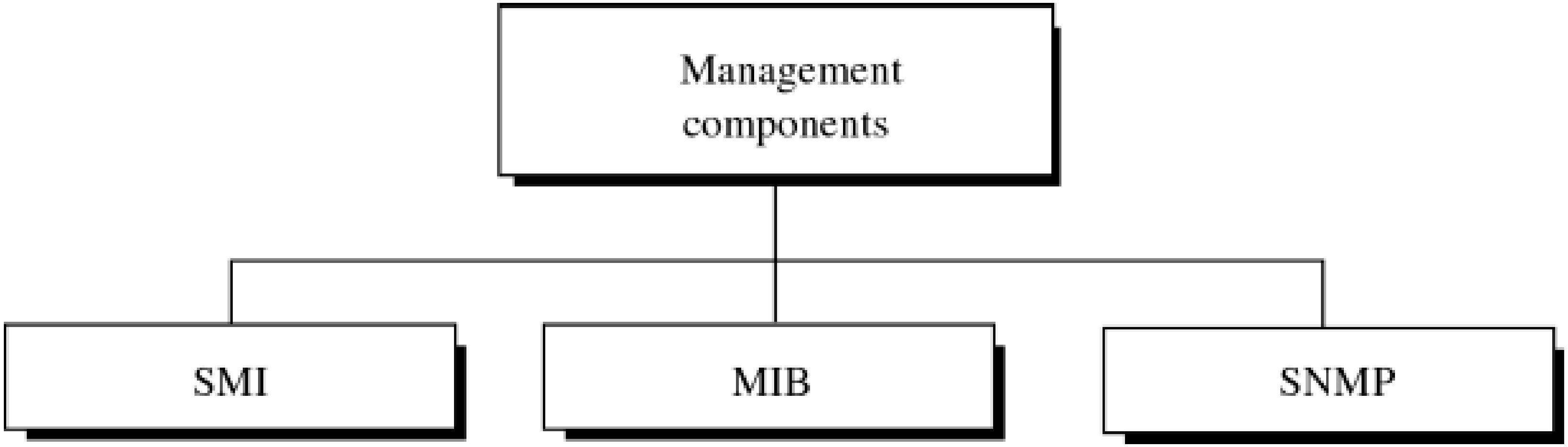
SNMPv1

- August 1988 SNMP specification issued
- Stand alone management stations and bridges, routers workstations etc supplied with agents
- Defines limited, easily implemented MIB of scalar variables and two dimensional tables
- Streamlined protocol
- Limited functionality
- Lack of security
- SNMPv2 1993, revised 1996
  - RFC 1901-1908

SNMPv2 (1)

- Framework on which network management applications can be built
  - e.g fault management, performance monitoring, accounting
- Protocol used to exchange management information
- Each player maintains local MIB
  - Structure defined in standard - SMI
- At least one system responsible for management
  - Houses management applications

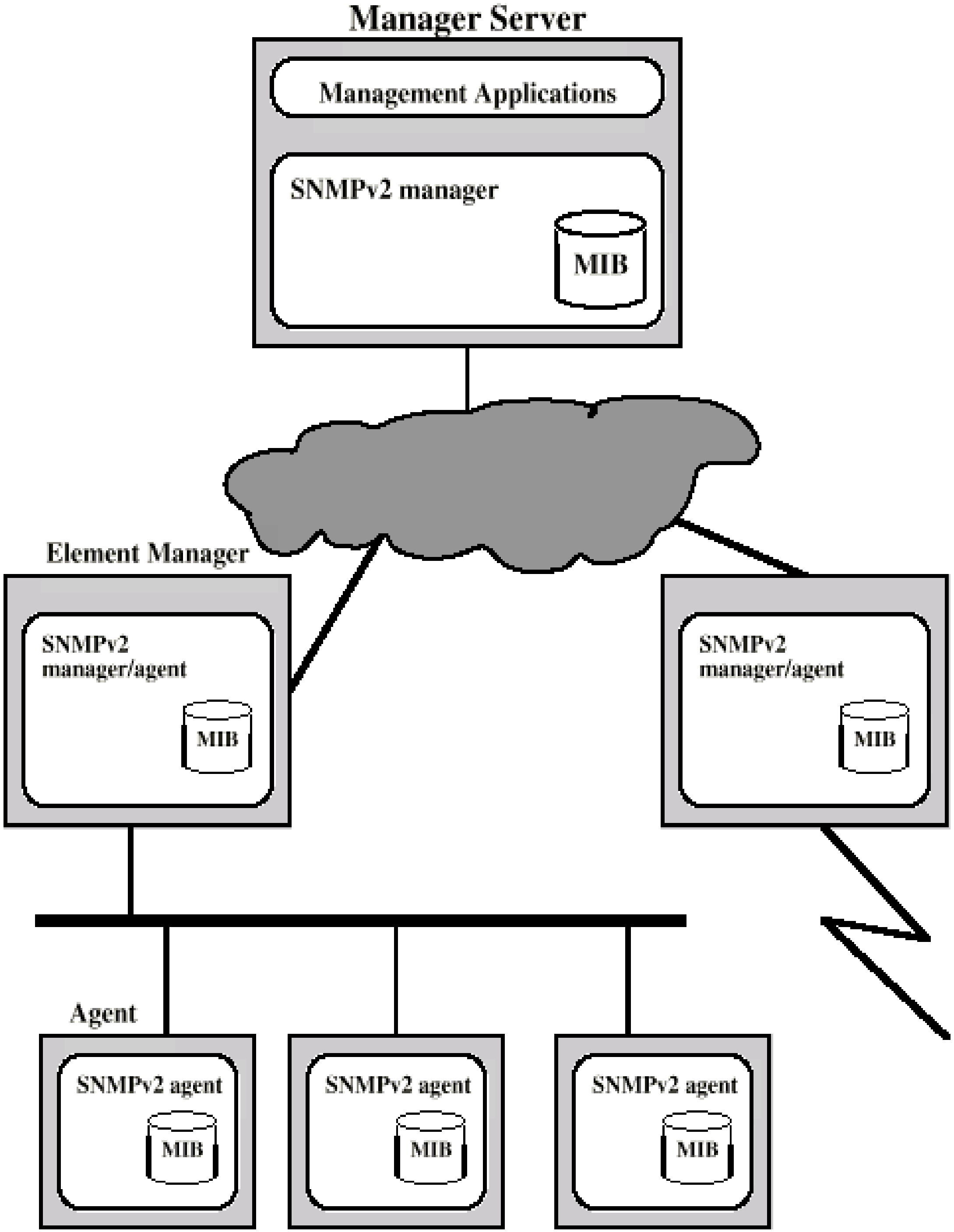
Internet Management Components



SNPMv2 (2)

- Support central or distributed management
- In distributes system, some elements operate as manager and agent
- Exchanges use SNMP v2 protocol
  - Simple request/response protocol
  - Typically uses UDP
    - Ongoing reliable connection not required
    - Reduces management overhead

SNMPv2 Managed Configuration



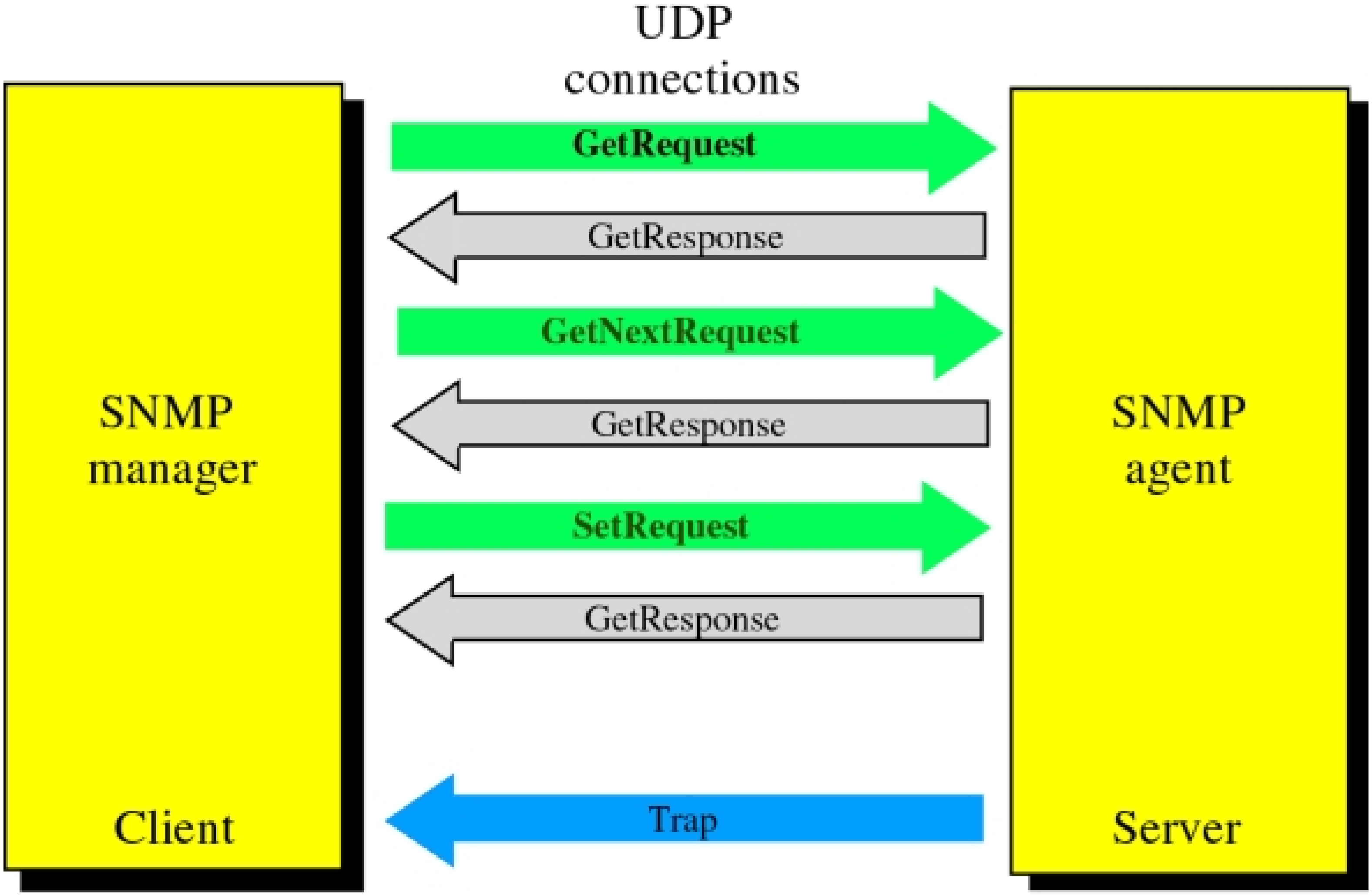
Structure of Management Information

- SMI - Structure of Management Information
  - a language for defining management information
- Defines general framework with which MIB defined and constructed
- Identifies data types
- How resources are represented and named
- Encourages simplicity and extensibility
- Scalars and two dimensional arrays of scalars (tables) only

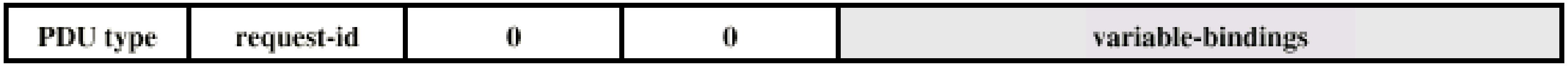
Protocol Operation

- Exchange of messages
- Outer message header deals with security
- Seven types of PDU

SNMPv1 Messages (5 types)

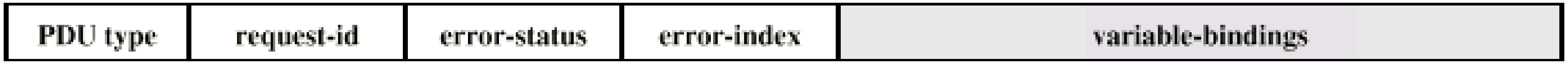


SNMPv2 PDU Formats



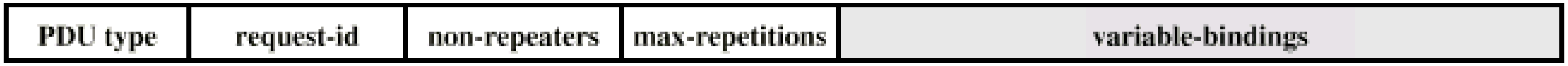
PDU type	request-id	0	0	variable-bindings

(a) GetRequest-PDU, GetNextRequest-PDU, SetRequest-PDU, SNMPv2-Trap-PDU, InformRequest-PDU



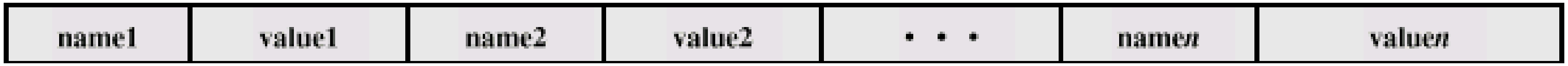
PDU type	request-id	error-status	error-index	variable-bindings

(b) Response-PDU



PDU type	request-id	non-repeaters	max-repetitions	variable-bindings

(c) GetBulkRequest-PDU



name1	value1	name2	value2	• • •	name*n*	value*n*

(d) variable-bindings

SNMPv3

- Addresses security issues of SNMP v1/2
- RFC 2570-2575
- Proposed standard January 1998
- Defines overall architecture and security capability
- To be used with SNMPv2

SNMP v3 Services

- Authentication
  - Part of User-Based Security Model (USM)
  - Assures that message:
    - Came from identified source
    - Has not been altered
    - Has not been delayed or replayed
- Privacy (part of USM)
  - Encrypted messages using DES
- Access control(defined in VACM View-Based Access Control Model)
  - Can configure agents to provide a number of levels of access to MIB
  - Access to information
  - Limit operations

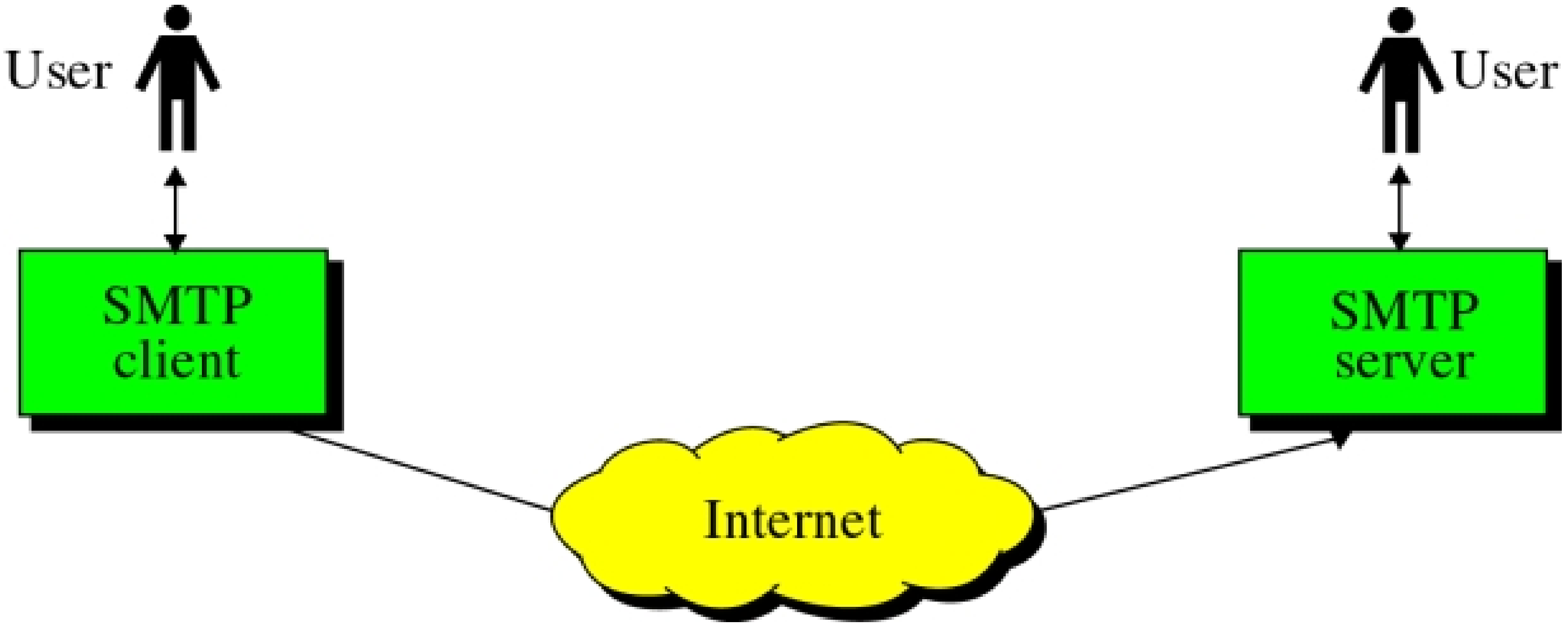
Electronic Mail

- Most heavily used application on any network
- Simple Mail Transfer Protocol (SMTP)
  - TCP/IP
  - Delivery of simple text messages
- Multi-purpose Internet Mail Extension (MIME)
  - Delivery of other types of data
  - Voice, images, video clips

SMTP

- RFC 821
- Not concerned with format of messages or data
  - Covered in RFC 822 (see later)
- SMTP uses info written on envelope of mail
  - Message header
- Does not look at contents
  - Message body
- Except:
  - Standardize message character set to 7 bit ASCII
  - Add log info to start of message
    - Shows path taken

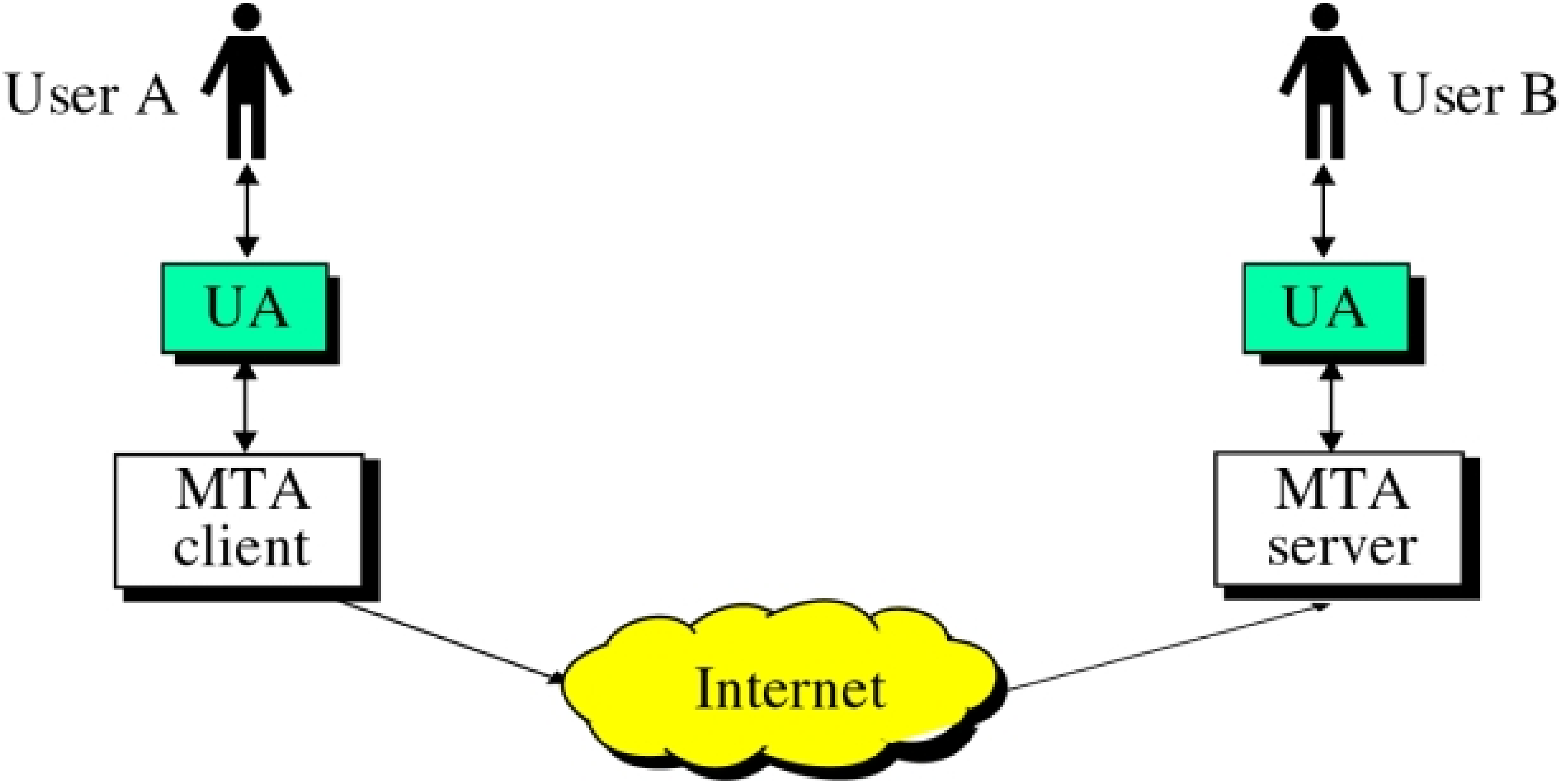
STMP Concept



Basic Operation

- Mail created by user agent program (mail client)
  - Message consists of:
    - Header containing recipient's address and other info
    - Body containing user data
- Messages queued and sent as input to SMTP sender program
  - Typically a server process (daemon on UNIX)

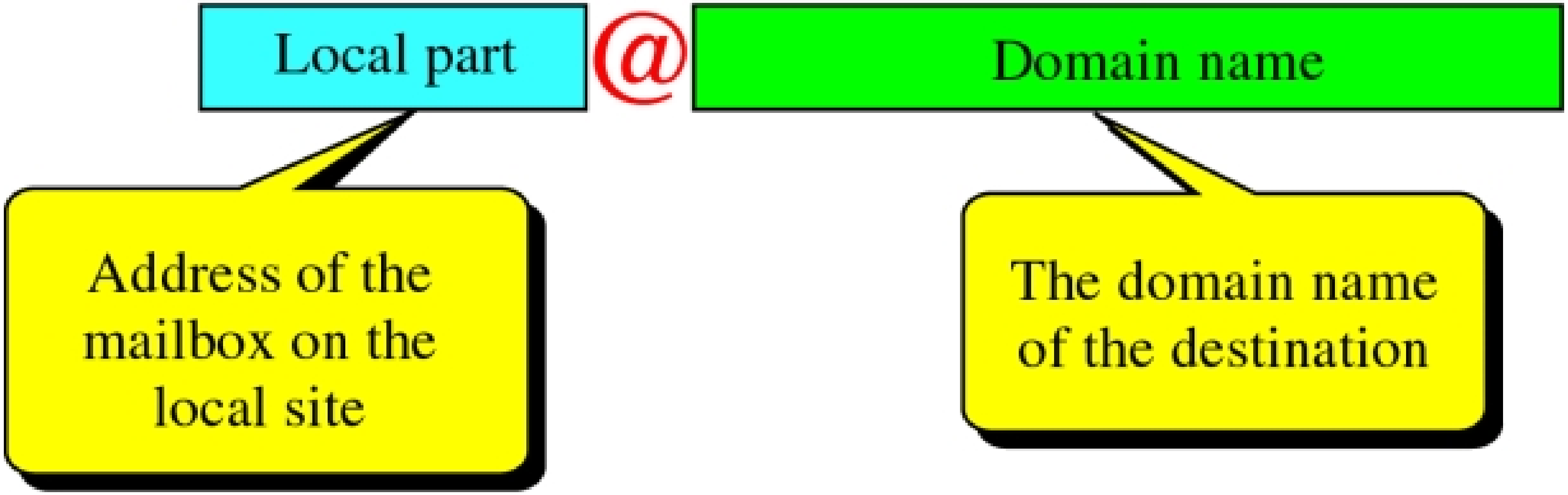
User Agent and Mail Transfer Agent



Mail Message Contents

- Each queued message has:
  - Message text
    - RFC 822 header with message envelope and list of recipients
    - Message body, composed by user
  - A list of mail destinations
    - Derived by user agent from header
    - May be listed in header
    - May require expansion of mailing lists
    - May need replacement of mnemonic names with mailbox names
- If BCCs indicated, user agent needs to prepare correct message format

E-mail Address



Local part
@
Domain name
Address of the mailbox on the local site
The domain name of the destination

SMTP Sender

- Takes message from queue
- Transmits to proper destination host
  - Via SMTP transaction
  - Over one or more TCP connections to port 25
- Host may have multiple senders active
- Host should be able to create receivers on demand
- When delivery complete, sender deletes destination from list for that message
- When all destinations processed, message is deleted

Optimization

- If message destined for multiple users on a given host, it is sent only once
  - Delivery to users handled at destination host
- If multiple messages ready for given host, a single TCP connection can be used
  - Saves overhead of setting up and dropping connection

Possible Errors

- Host unreachable
- Host out of operation
- TCP connection fail during transfer
- Sender can re-queue mail
  - Give up after a period
- Faulty destination address
  - User error
  - Target user changed address
  - Redirect if possible
  - Inform user if not

SMTP Protocol - Reliability

- Used to transfer messages from sender to receiver over TCP connection
- Attempts to provide reliable service
- No guarantee to recover lost messages
- No end to end acknowledgement to originator
- Error indication delivery not guaranteed
- Generally considered reliable

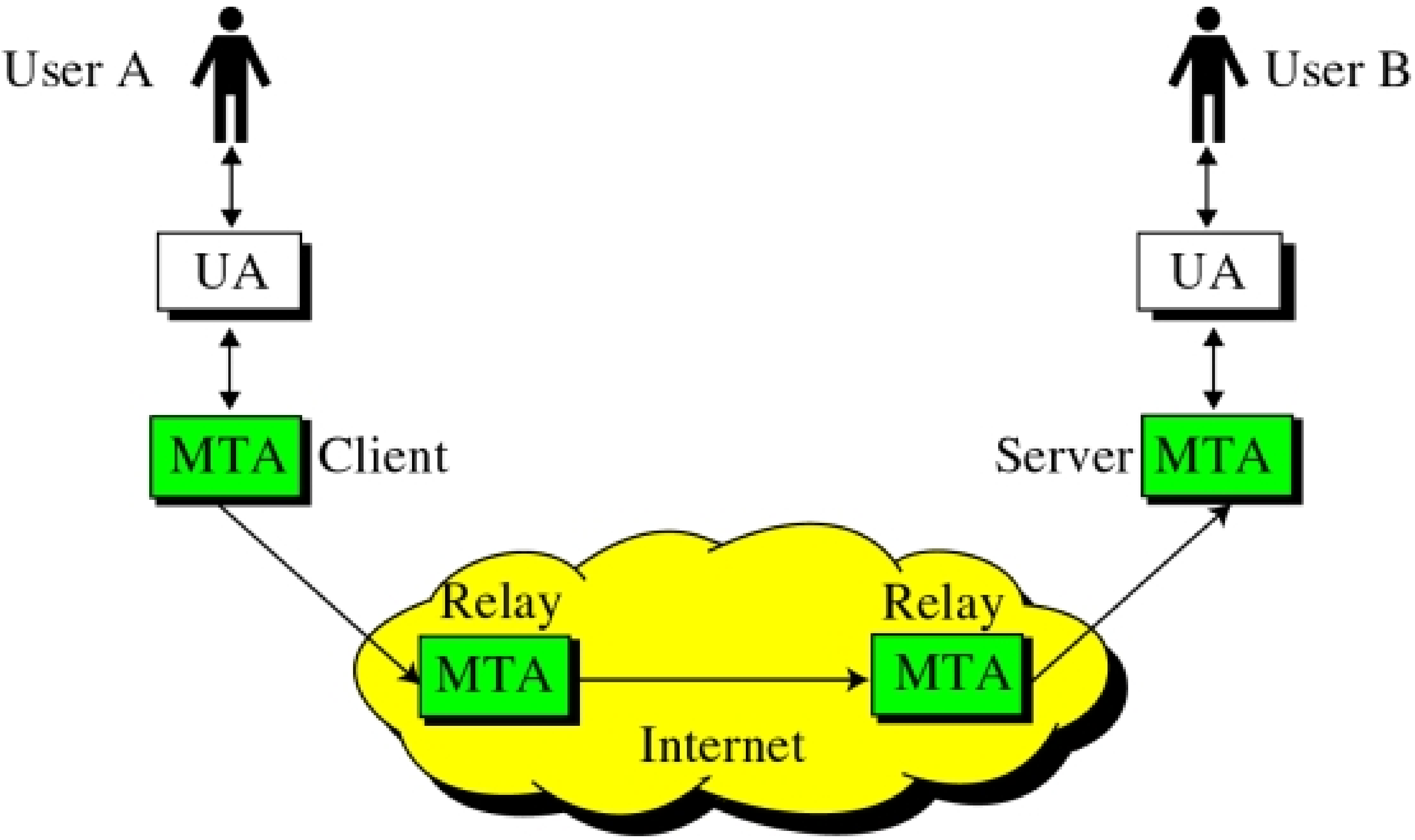
SMTP Receiver

- Accepts arriving message
- Places in user mailbox or copies to outgoing queue for forwarding
- Receiver must:
  - Verify local mail destinations
  - Deal with errors
    - Transmission
    - Lack of disk space
- Sender responsible for message until receiver confirm complete transfer
  - Indicates mail has arrived at host, not user

SMTP Forwarding

- Mostly direct transfer from sender host to receiver host
- May go through intermediate machine via forwarding capability
  - Sender can specify route
  - Target user may have moved

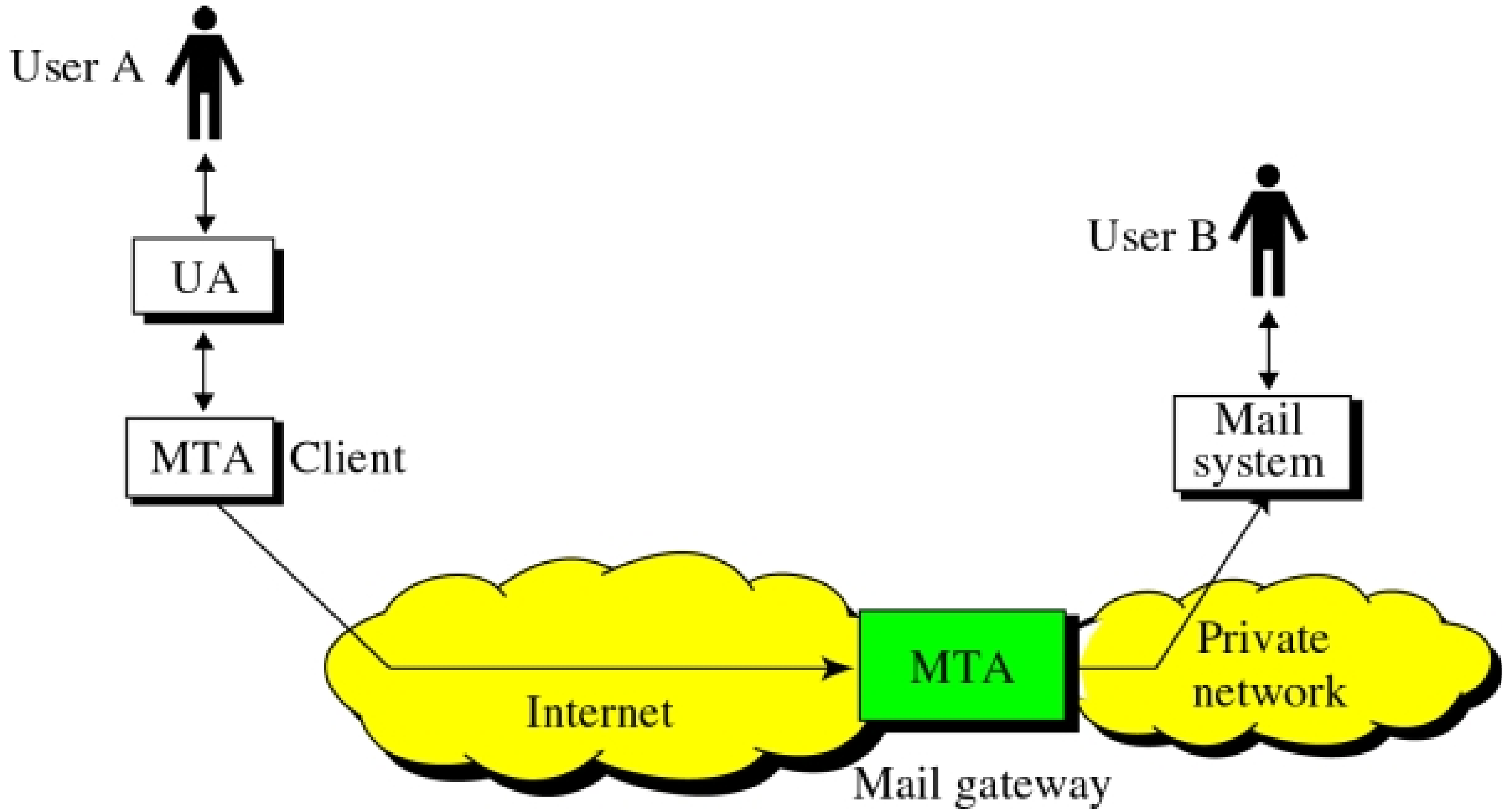
Relay MTAs


User A
User B
UA
UA
MTA
Client
Server
MTA
Relay
MTA
Relay
MTA
Internet

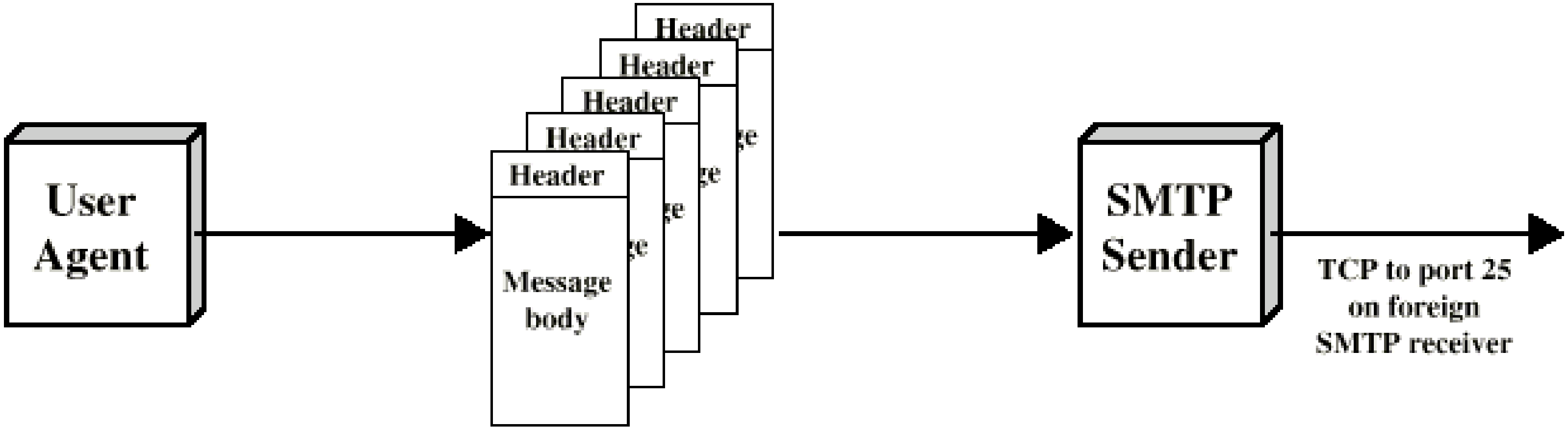
Conversation

- SMTP limited to conversation between sender and receiver
- Main function is to transfer messages
- Rest of mail handling beyond scope of SMTP
  - May differ between systems

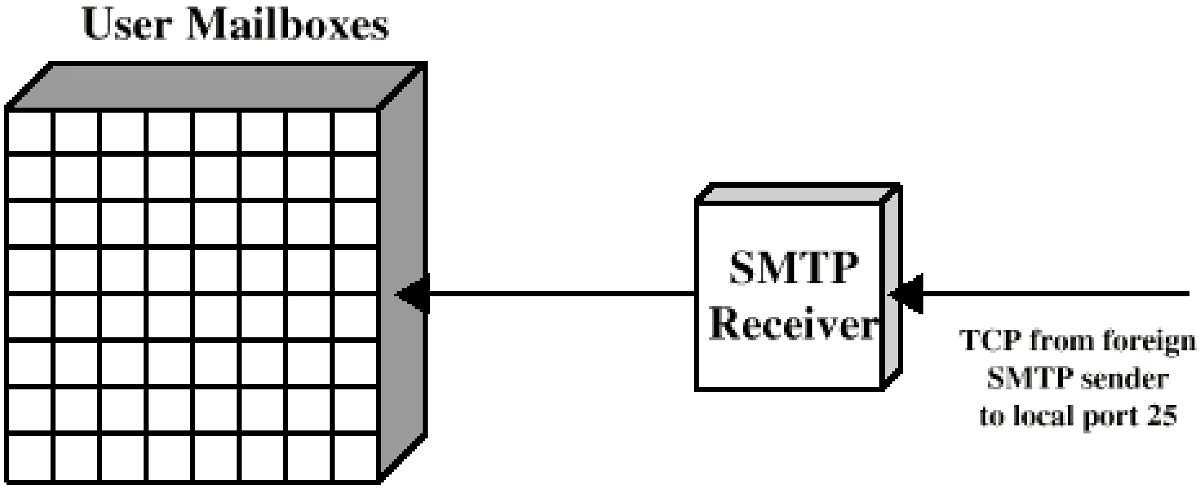
Mail Gateways



SMTP Mail Flow

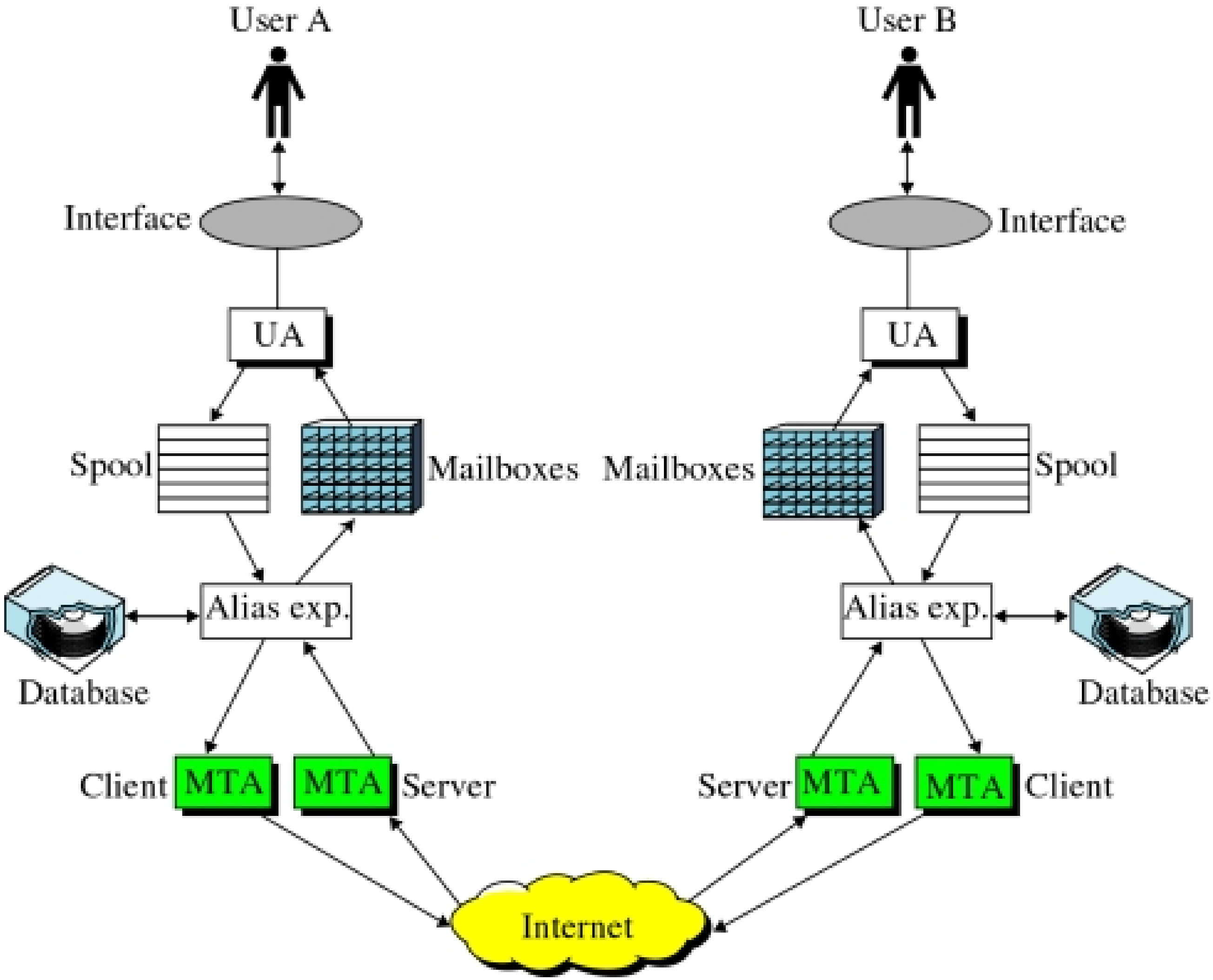


(a) Outgoing Mail



(b) Incoming Mail

Two Directional Electronic Mail



SMTP System Overview

- Commands and responses between sender and receiver
- Initiative with sender
  - Establishes TCP connection
- Sender sends commands to receiver
- e.g. HELO<SP><domain><CRLF>
- Each command generates exactly one reply
- e.g. 250 requested mail action ok; completed

SMTP Replies

- Leading digit indicates category
  - Positive completion reply (2xx)
  - Positive intermediate reply (3xx)
  - Transient negative completion reply (4xx)
  - Permanent negative completion reply (5xx)

Operation Phases

- Connection setup
- Exchange of command-response pairs
- Connection termination

Connection Setup

- Sender opens TCP connection with receiver
- Once connected, receiver identifies itself
  - 220 <domain> service ready
- Sender identifies itself
  - HELO
- Receiver accepts sender's identification
  - 250 OK
- If mail service not available, step 2 above becomes:
  - 421 service not available

Mail Transfer

- Sender may send one or more messages to receiver
- MAIL command identifies originator
  - Gives reverse path to used for error reporting
  - Receiver returns 250 OK or appropriate fail/error message
- One or more RCPT commands identifies recipients for the message
  - Separate reply for each recipient
- DATA command transfers message text
  - End of message indicated by line containing just period (.)

Closing Connection

- Two steps
- Sender sends QUIT and waits for reply
- Then initiate TCP close operation
- Receiver initiates TCP close after sending reply to QUIT

Format for Text Messages RFC 822

- Message viewed as having envelope and contents
- Envelope contains information required to transmit and deliver message
- Message is sequence of lines of text
  - Uses general memo framework
  - Header usually keyword followed by colon followed by arguments

Example Message

Date:Tue, 16 Jan 1996 10:37:17 (EST)

From: “William Stallings” <ws@host.com>

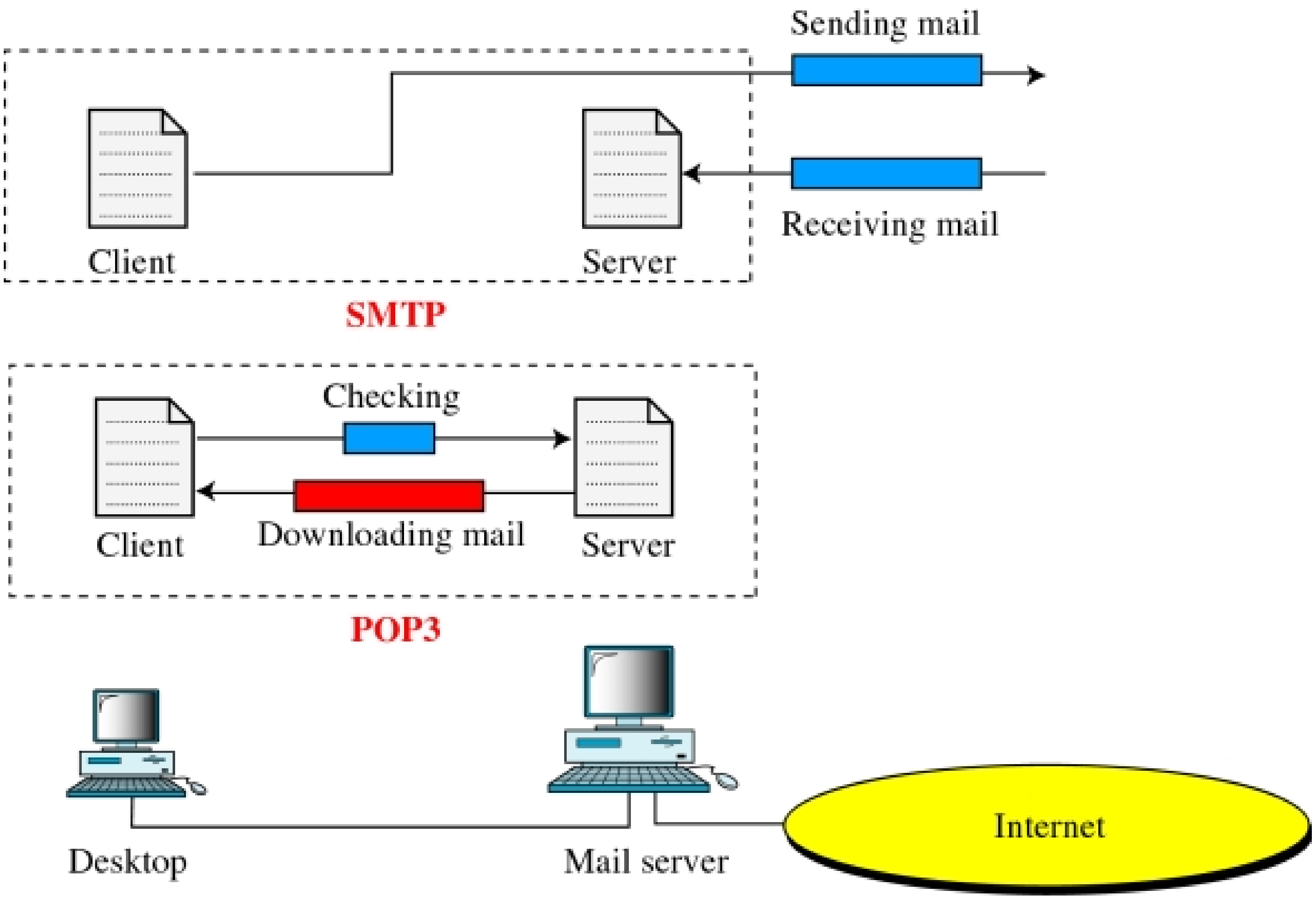
Subject:The syntax of RFC 822

To: Smith@otherhost.com

Cc: Jones@Yet-another_host.com

This is the main text, delimited from the header by a blank line.

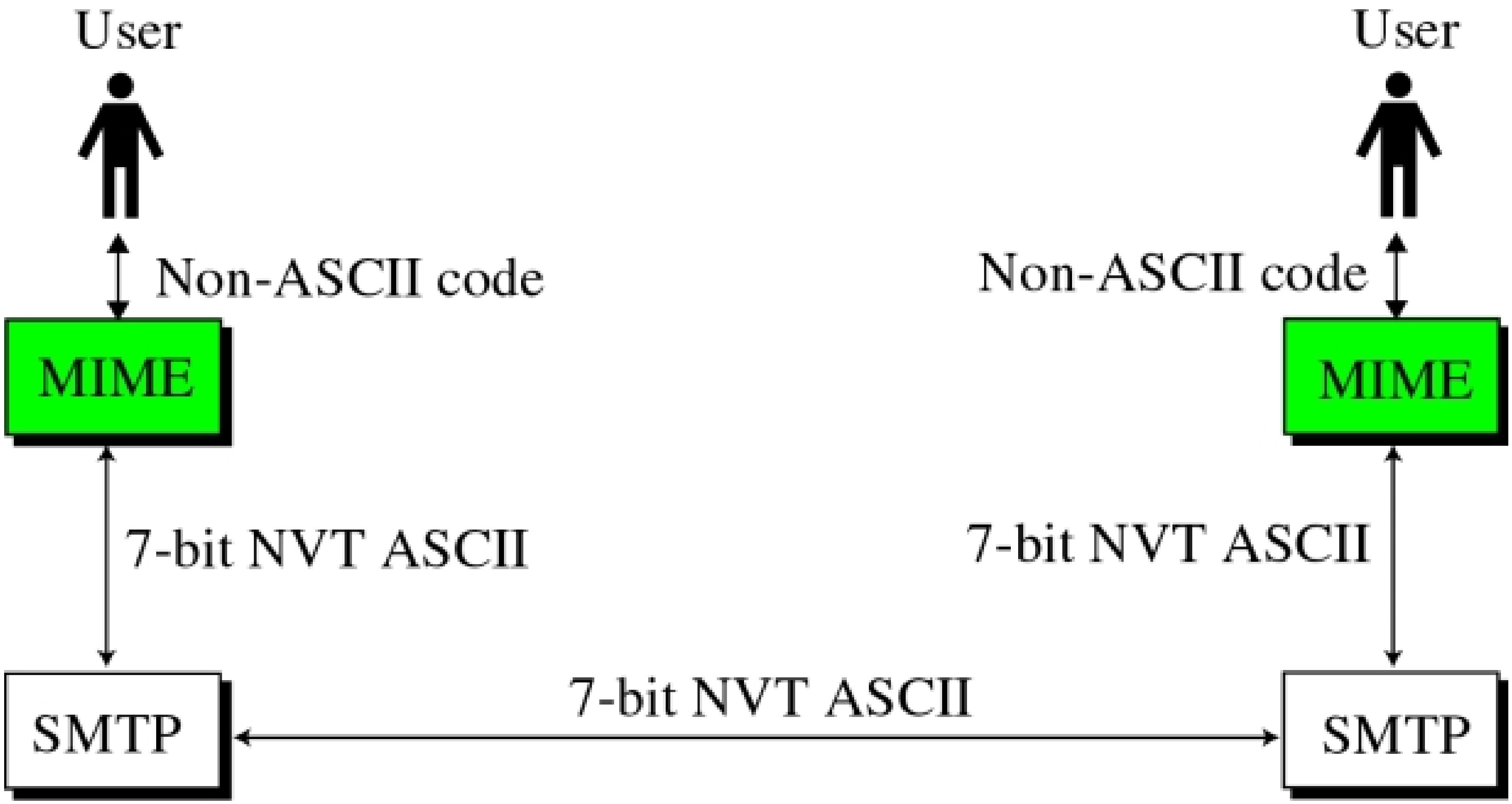
SMTP and POP3 (Post Office Protocol v3)



Multipurpose Internet Mail Extension (MIME)

- Extension to RFC822
- SMTP can not transmit executables
  - Uuencode and other schemes are available
    - Not standardized
- Can not transmit text including international characters (e.g. â, å, ä, è, é, ê, ë)
  - Need 8 bit ASCII
- Servers may reject mail over certain size
- Translation between ASCII and EBCDIC not standard
- SMTP gateways to X.400 can not handle none text data in X.400 messages
- Some SMTP implementations do not adhere to standard
  - CRLF, truncate or wrap long lines, removal of white space, etc.

MIME


User
User
Non-ASCII code
Non-ASCII code
MIME
MIME
7-bit NVT ASCII
7-bit NVT ASCII
SMTP
7-bit NVT ASCII
SMTP

Overview of MIME

- Five new message header fields
  - MIME version
  - Content type
  - Content transfer encoding
  - Content Id
  - Content Description
- Number of content formats defines
- Transfer encoding defined

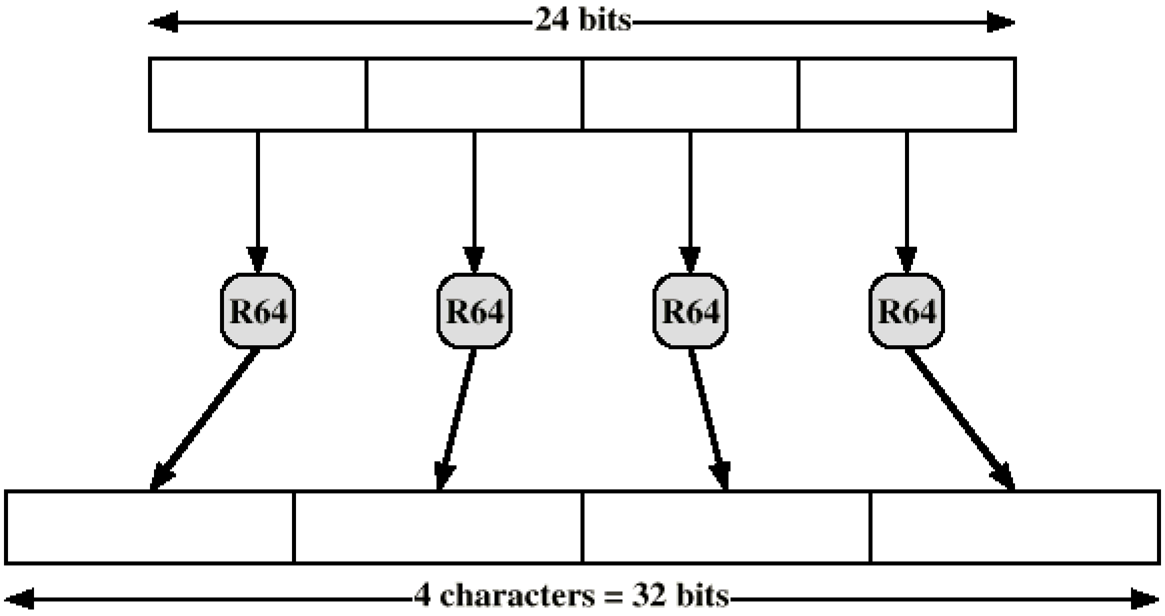
Content Types

- Text body
- Multipart
  - Mixed, Parallel, Alternative, Digest
- Message
  - RFC 822, Partial, External-body
- Image
  - jpeg, gif
- Video
  - mpeg
- Audio
  - Basic
- Application
  - Postscript
  - octet stream

MIME Transfer Encoding

- Reliable delivery across wide largest range of environments
- Content transfer encoding field
  - Six values
  - Three (7bit, 8bit, binary) no encoding done
    - Provide info about nature of data
- Quoted-printable
  - Data largely printable ASCII characters
  - Non-printing characters represented by hex code
- Base64
  - Maps arbitrary binary input onto printable output
- X-token
  - Named nonstandard encoding

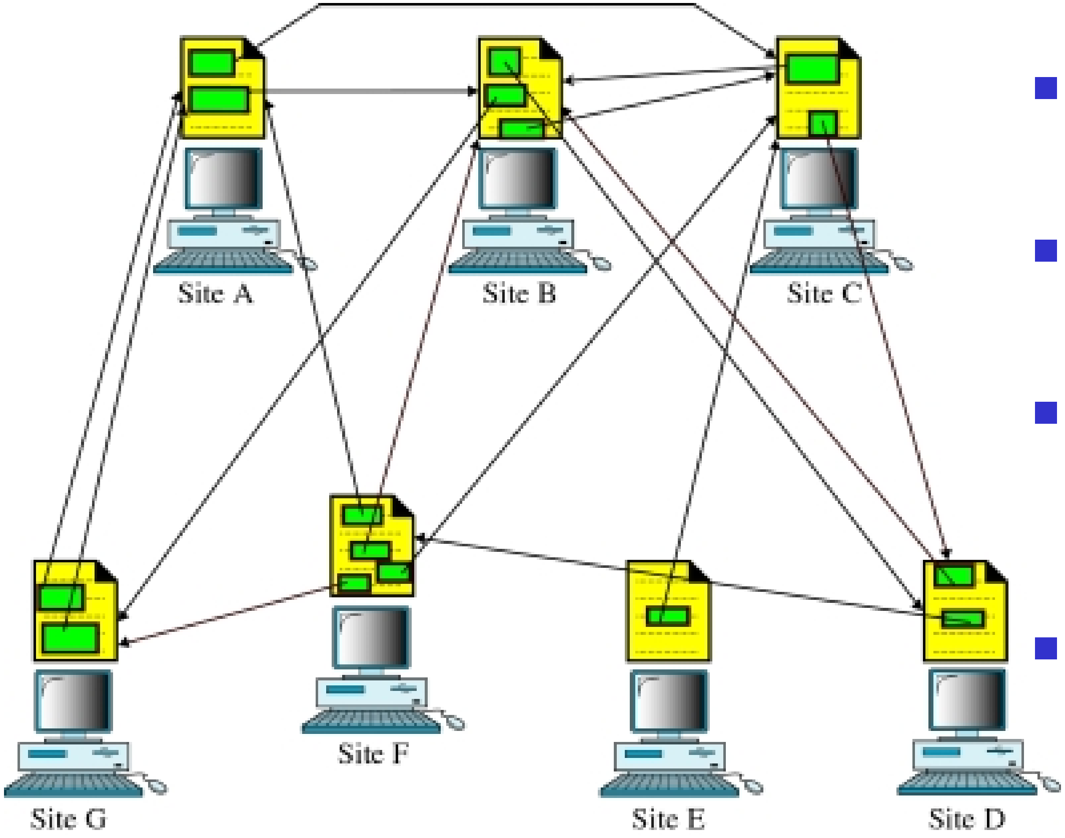
Base 64 Encoding



Hypertext Transfer Protocol HTTP

- Underlying protocol of the World Wide Web
- Not a protocol for transferring hypertext
  - For transmitting information with efficiency necessary for hypertext jumps
- Can transfer plain text, hypertext, audio, images, and Internet accessible information

World Wide Web - WWW

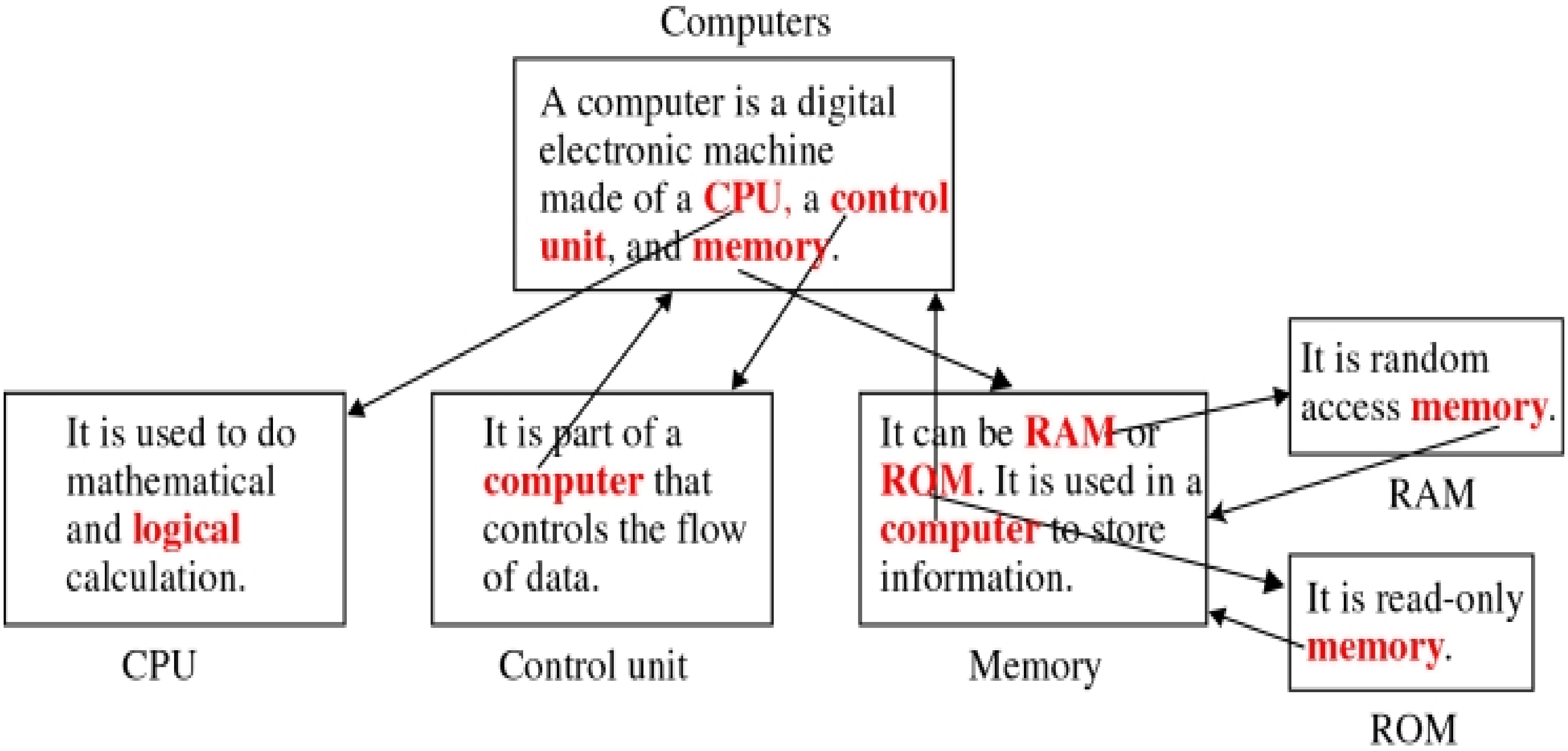


- A repository of information spread all over the world and linked together
- Distributed services based on client-server architecture
- Clients use browser to access the services
- Services spread every-where in global, where is called Web site
- WWW uses concepts of hypertext and hypermedia

Hypertext and Hypermedia

- In a ***hypertext environment***, information is stored in a set of ***documents***
- Documents are linked together using the concept of ***pointers***
- An ***item*** can be associated with another document by means of the pointer
- Reader who is browsing through a document can move to other documents by clicking the items
- ***Hypertext document*** contain only text, ***hypermedia document*** can contain image, graphics and audio
- A hypertext or a hypermedium available on Web is called a ***page***
- Main page for an organization or an individual is known as a ***homepage***

Hypertext



Computers

A computer is a digital electronic machine made of a **CPU**, a **control unit**, and **memory**.

It is used to do mathematical and **logical** calculation.

CPU

It is part of a **computer** that controls the flow of data.

Control unit

It can be **RAM** or **ROM**. It is used in a **computer** to store information.

Memory

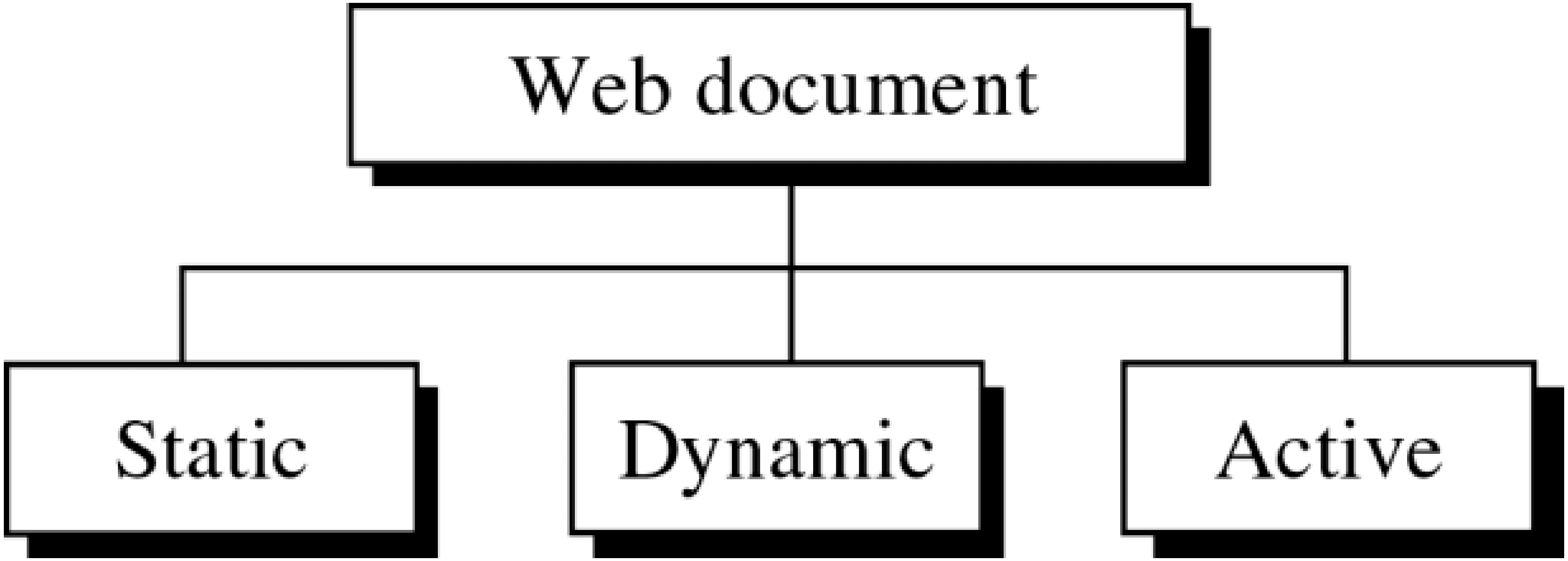
It is random access **memory**.

RAM

It is read-only **memory**.

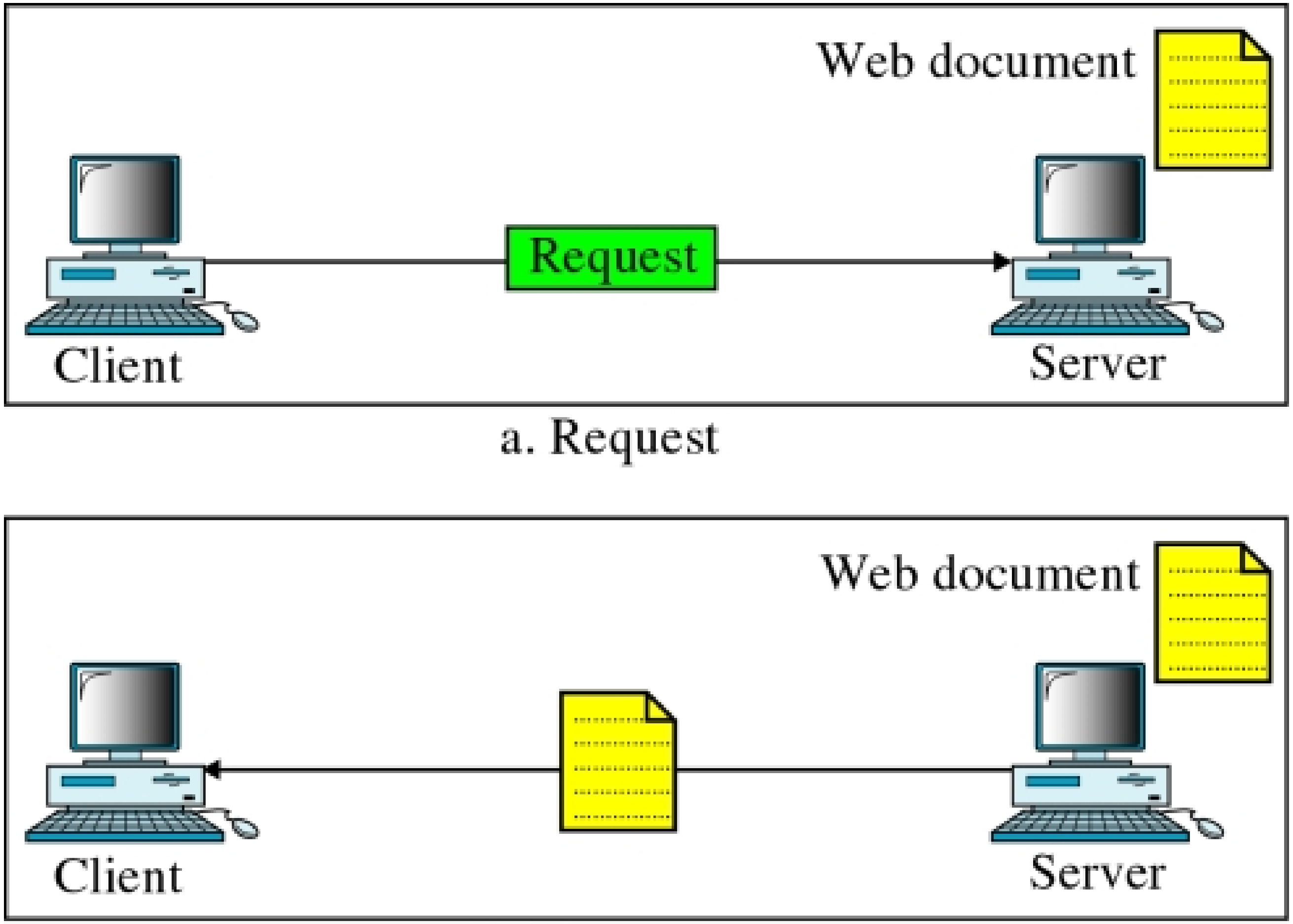
ROM

Categories of Web Documents



Static Document

- Static document are fixed-contents
- Are created and stored in a server

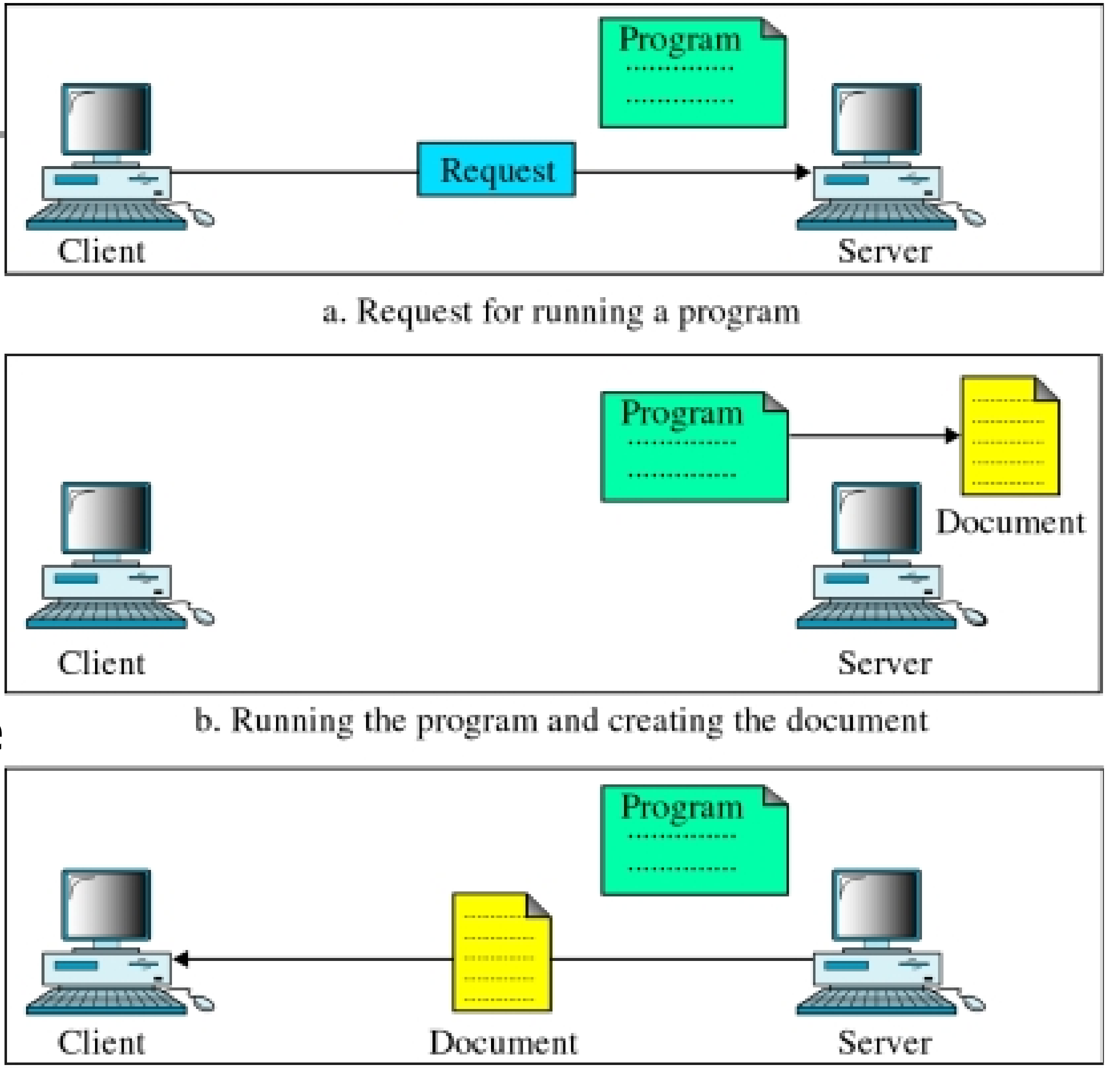


a. Request

b. Response

Dynamic Document

- Not exist in a predefined format
- Created by a Web server running an application program whenever a browser request the document
- Common Gateway Interface (CGI)
  - A standard for building dynamic Web documents
- Most of CGI program have been created using shell scripts in UNIX
- Script is run at server site



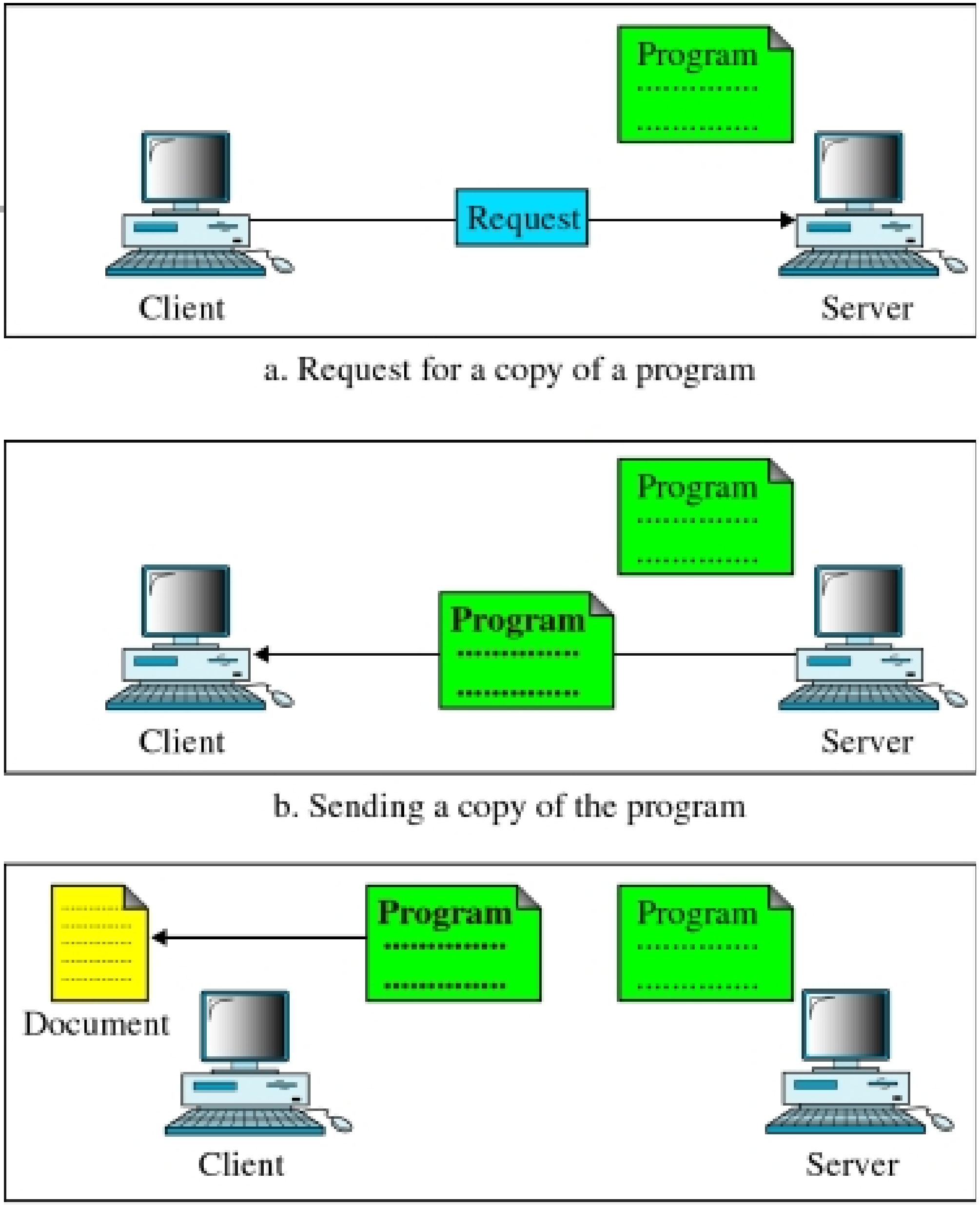
a. Request for running a program

b. Running the program and creating the document

c. Response

Active Document

- Dynamic documents are produced at the server site when they are called
- For many applications, we need the program to be run at the client site.
- These programs are called active document
- Java uses ***applet*** to define an active document



a. Request for a copy of a program

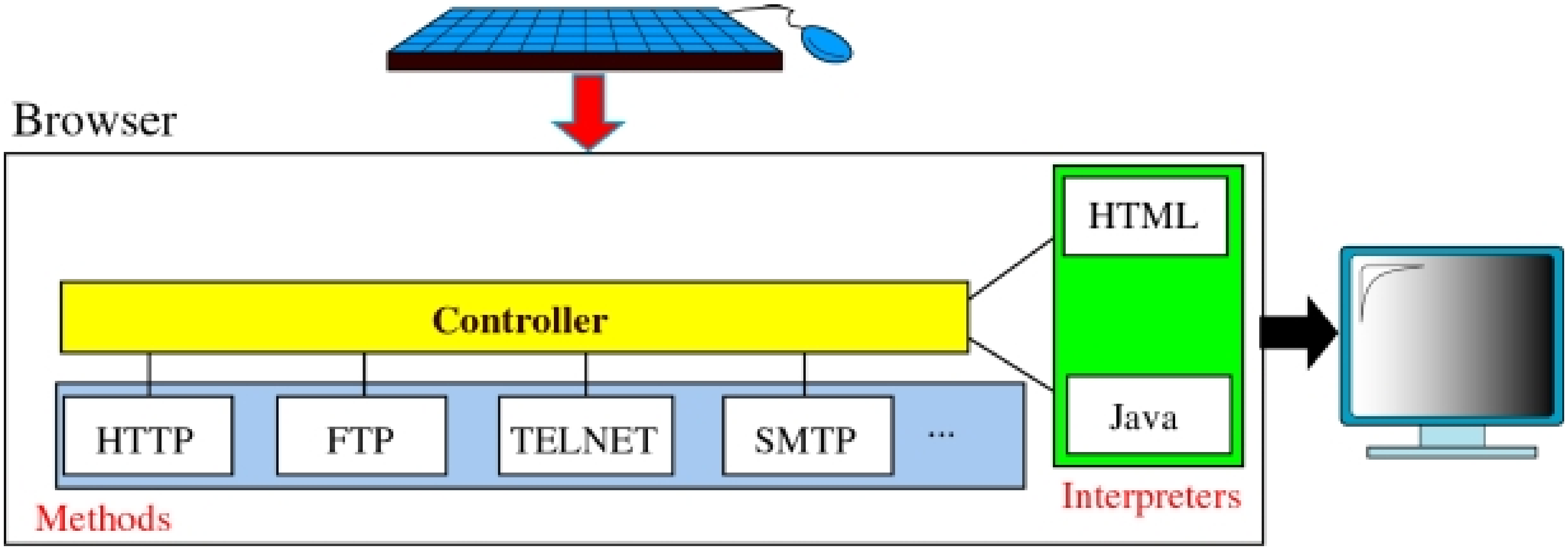
b. Sending a copy of the program

c. Running the program and creating the document

HTML

- HyperText Markup Language
- A markup language for writing Web pages
- Only thing it does is let the browsers format the Web pages

Browser Architecture



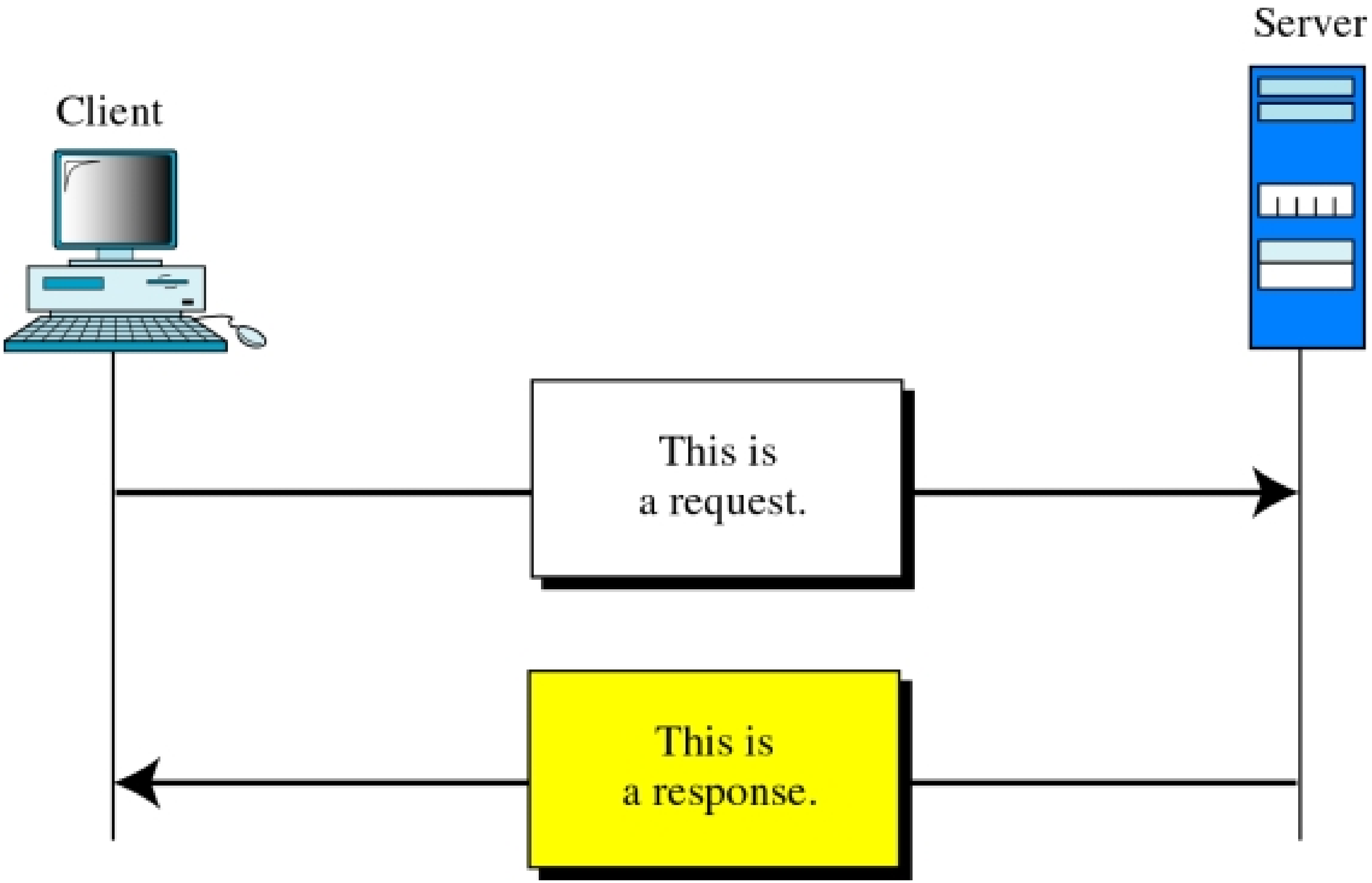
HTTP Overview

- Transaction oriented client/server protocol
- Usually between Web browser (client) and Web server
- Uses TCP connections
- Stateless
  - Each transaction treated independently
  - Each new TCP connection for each transaction
  - Terminate connection when transaction complete

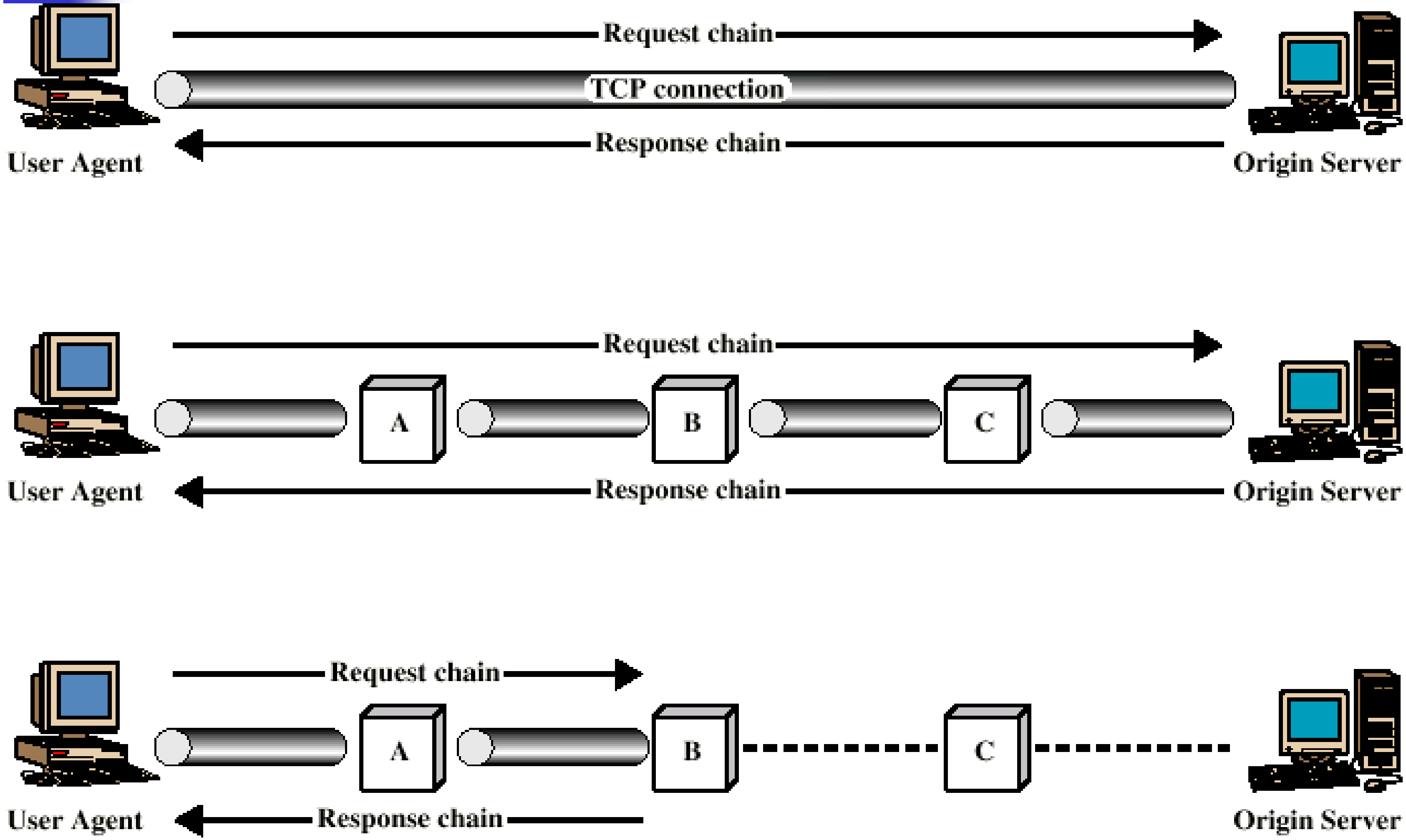
Key Terms

- Cache
- Client
- Connection
- Entity
- Gateway
- Message
- Origin server
- Proxy
- Resource
- Server
- Tunnel
- User agent

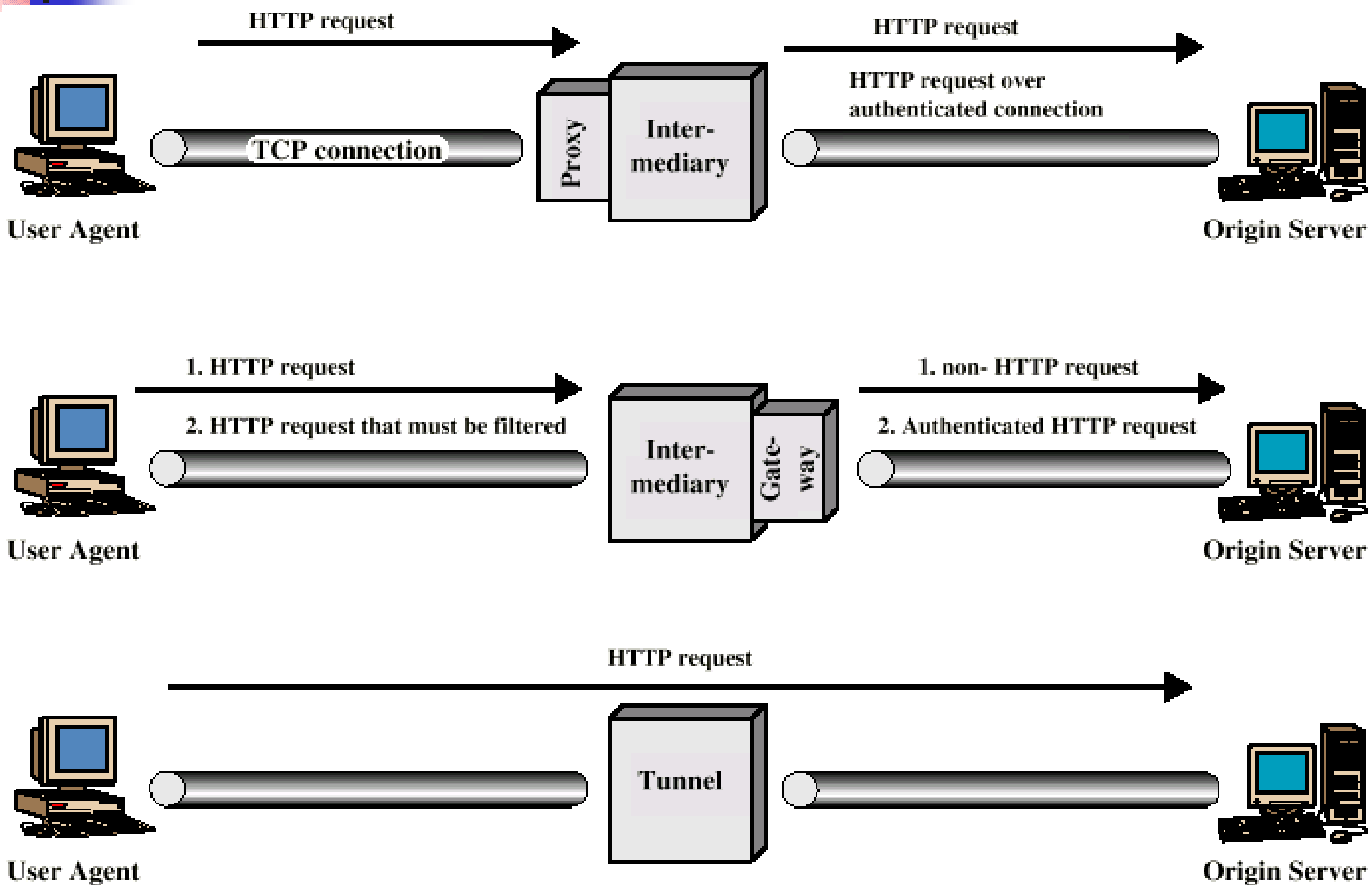
HTTP Transaction



Examples of HTTP Operation

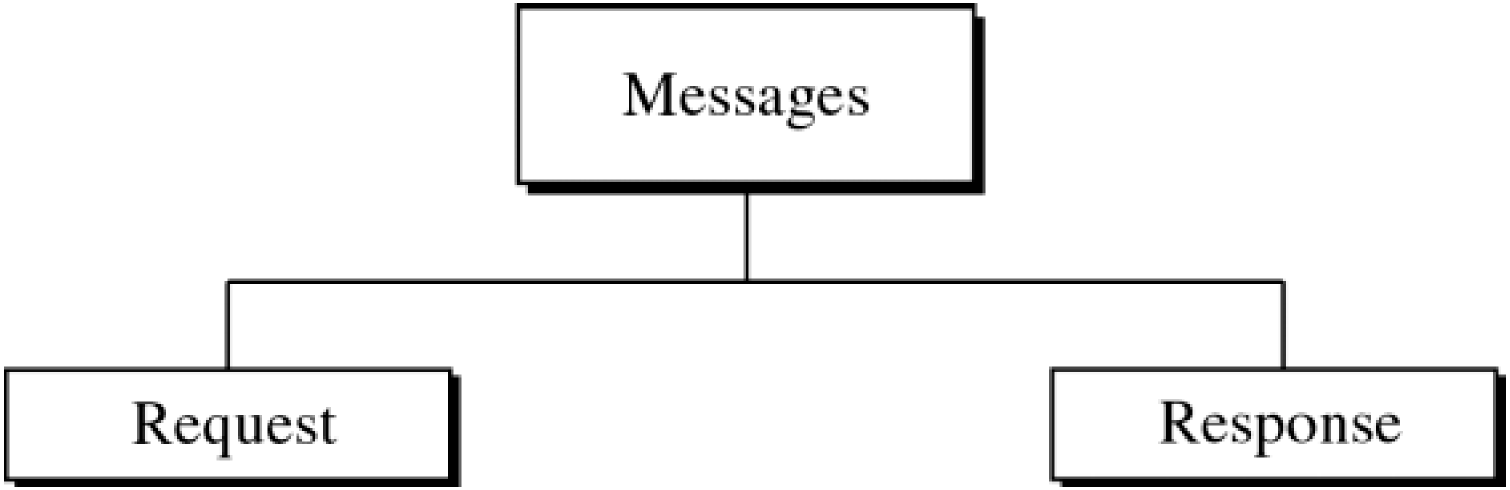


Intermediate HTTP Systems



HTTP Messages

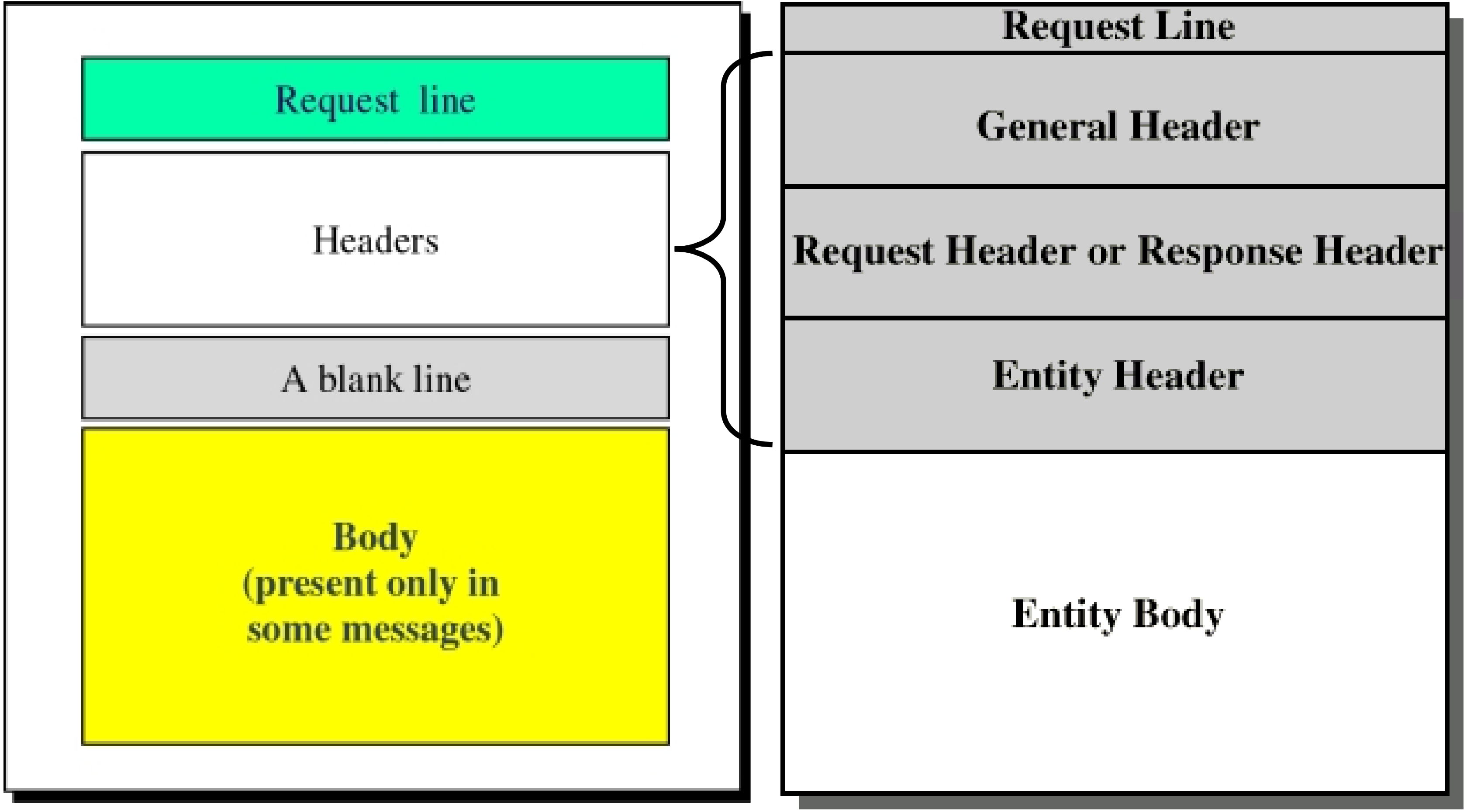
- Requests
  - Client to server
- Responses
  - Server to client



Fields in HTTP Messages

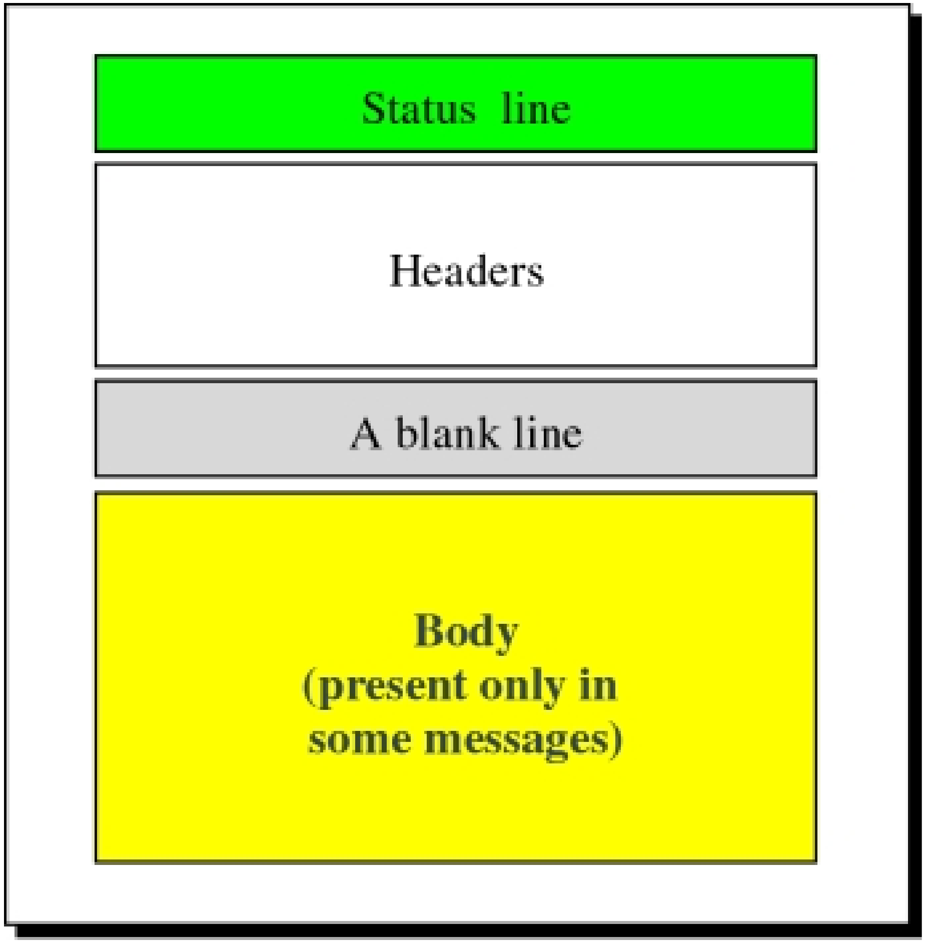
- Request line
- Response line
- General header
- Request header
- Response header
- Entity header
- Entity body

Request Message Structure



Request message

Response Message Structure



Response message

General Header Fields

- Cache control
- Connection
- Data
- Forwarded
- Keep alive
- MIME version
- Pragma
- Upgrade

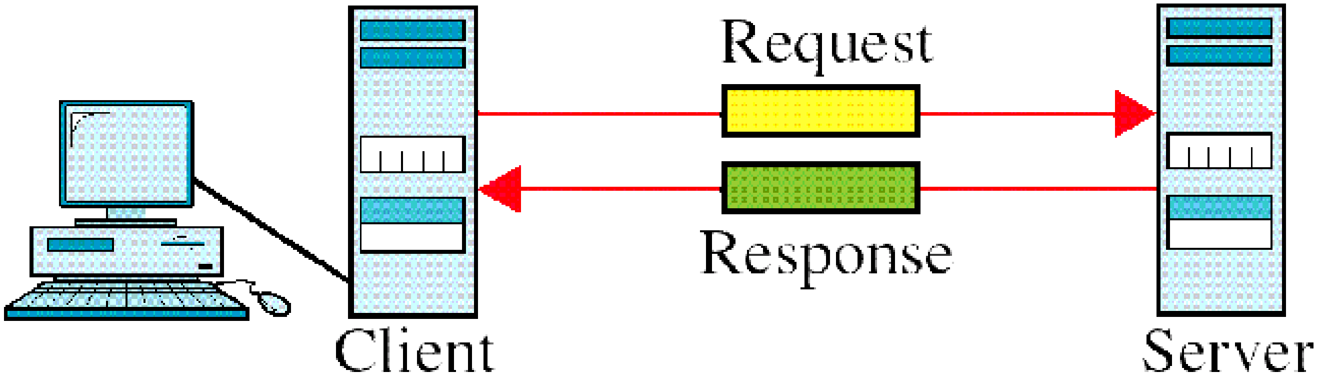
HTTP Commands

Request

Get: request data
Head: ask only for headers
Post: send information

Response

To send the response of get or head to client



Request Methods

- Request-Line = Method <SP> Request_URL <SP> HTTP-Version <CRLF>
- Methods:
  - Options
  - Get
  - Head
  - Post
  - Put
  - Patch
  - Copy
  - Move
  - Delete
  - Link
  - Unlink
  - Trace
  - Wrapped
  - Extension-method

Request Header Field

- Accept
- Accept charset
- Accept encoding
- Accept language
- Authorization
- From
- Host
- If modified since
- Proxy authentication
- Range
- Referrer
- Unless
- User agent

Response Messages

- Status line followed by one or more general, response and entity headers, followed by optional entity body

- Status-Line = HTTP-Version <SP> Status-Code <SP> Reason-Phrase <CRLF>

Status Codes

- Informational
- Successful
- Redirection
- Client error
- Server error

Response Header Fields

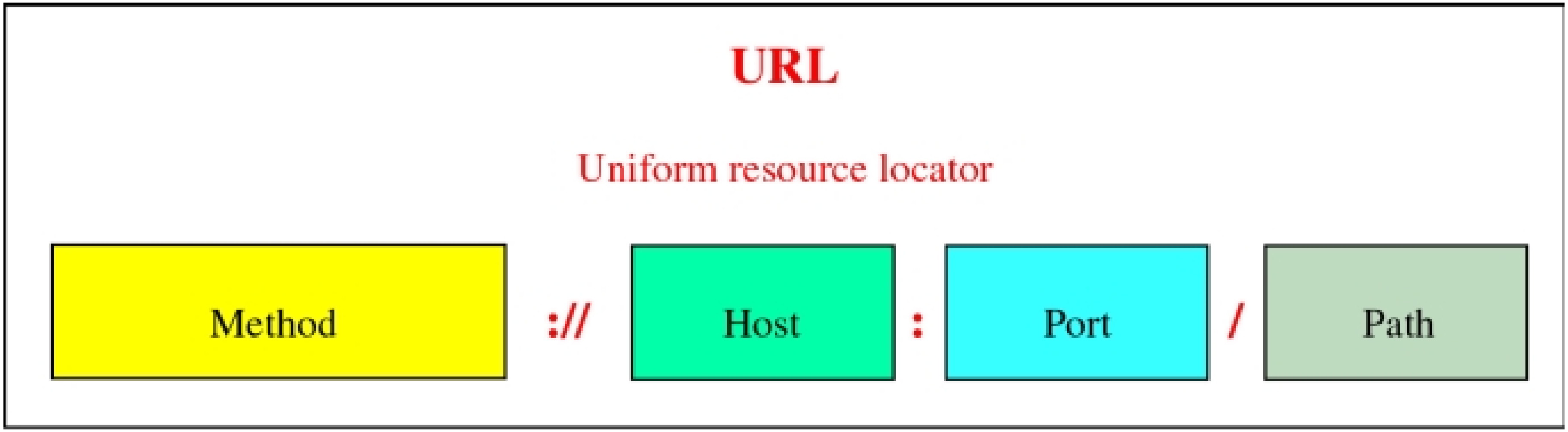
- Location
- Proxy authentication
- Public
- Retry after
- Server
- WWW-Authenticate

Entity Header Fields

- Allow
- Content encoding
- Content language
- Content length
- Content MD5
- Content range
- Content type
- Content version
- Derived from
- Expires
- Last modified
- Link
- Title
- Transfer encoding
- URL header
- Extension header

URL - Uniform Resource Locator

- A standard for specifying any kind of information on the Internet
- URL defines only three things
  - Method - Gopher, FTP, HTTP, news, TELNET
  - Host - alias names usually begin with www
  - Path - pathname of the file where the information is located



URL

Uniform resource locator

Method :// Host : Port / Path

Entity Body

- Arbitrary sequence of octets
- HTTP transfers any type of data including:
  - text
  - binary data
  - audio
  - images
  - video
- Interpretation of data determined by header fields
  - Content encoding, content type, transfer encoding