



# **INTERNET MANAGEMENT PROTOCOLS**

## **STATE OF THE ART / RECENT DEVELOPMENTS**

TUTORIAL T6 - PRESENTED AT IM'2003  
COLORADO SPRINGS, USA  
24 MARCH 2003

**AIKO PRAS**  
UNIVERSITY OF TWENTE  
THE NETHERLANDS

pras@cs.utwente.nl  
<http://wwwhome.cs.utwente.nl/~pras>

<http://www.simpleweb.org/tutorials>



# OVERVIEW

## BACKGROUND

- HISTORY, GOALS & STANDARDS

## • STRUCTURE OF MANAGEMENT INFORMATION

- SCALARS
- TABLES

## • MANAGEMENT INFORMATION BASES

- OVERVIEW
  - MIB-II
  - SNMPv2, IF & IP MIB

## • SIMPLE NETWORK MANAGEMENT PROTOCOL

- VERSION 1
- VERSION 2
- VERSION 3

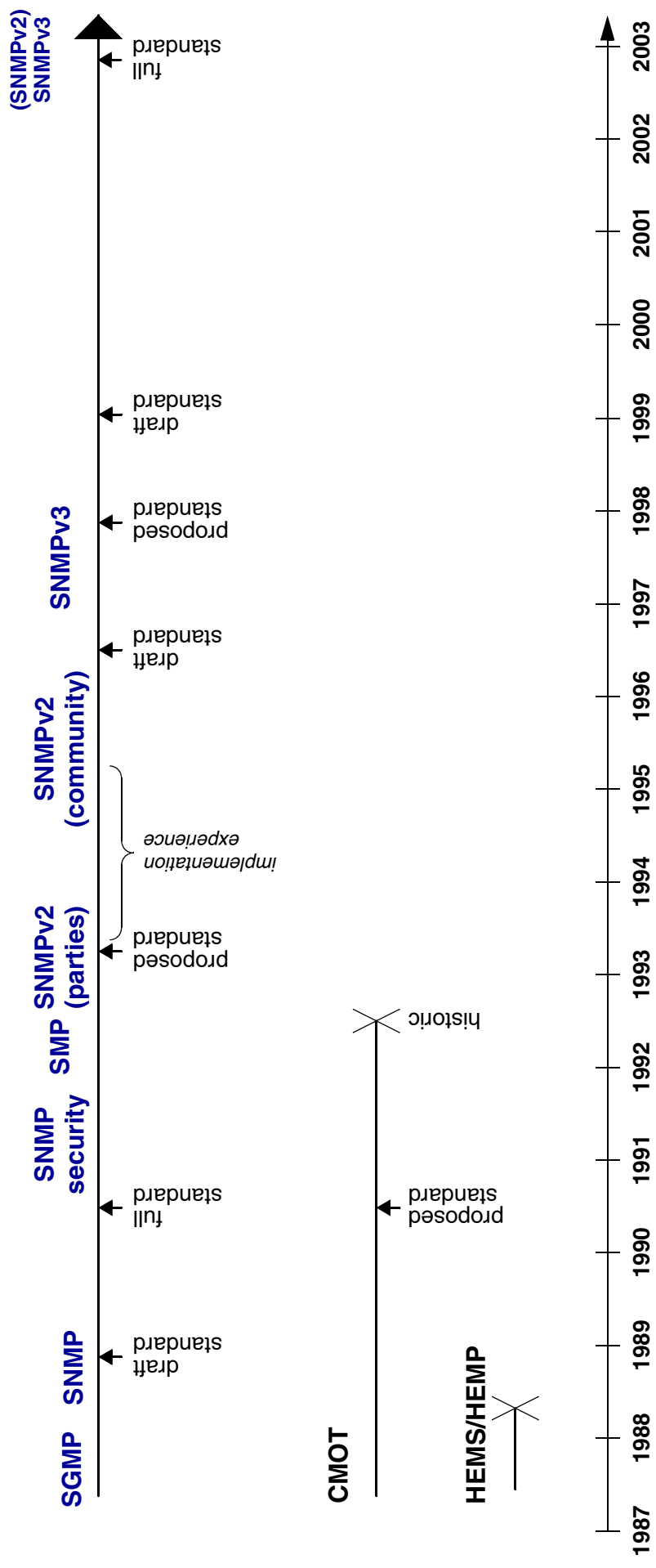
AGENTX  
DISMAN

## RECENT DEVELOPMENTS

## FURTHER INFORMATION



# SNMP HISTORY





## **SNMP GOALS**

### **UBIQUITY**

- PCs AND CRAYS

### **INCLUSION OF MANAGEMENT SHOULD BE INEXPENSIVE**

- SMALL CODE
- LIMITED FUNCTIONALITY

### **MANAGEMENT EXTENSIONS SHOULD BE POSSIBLE**

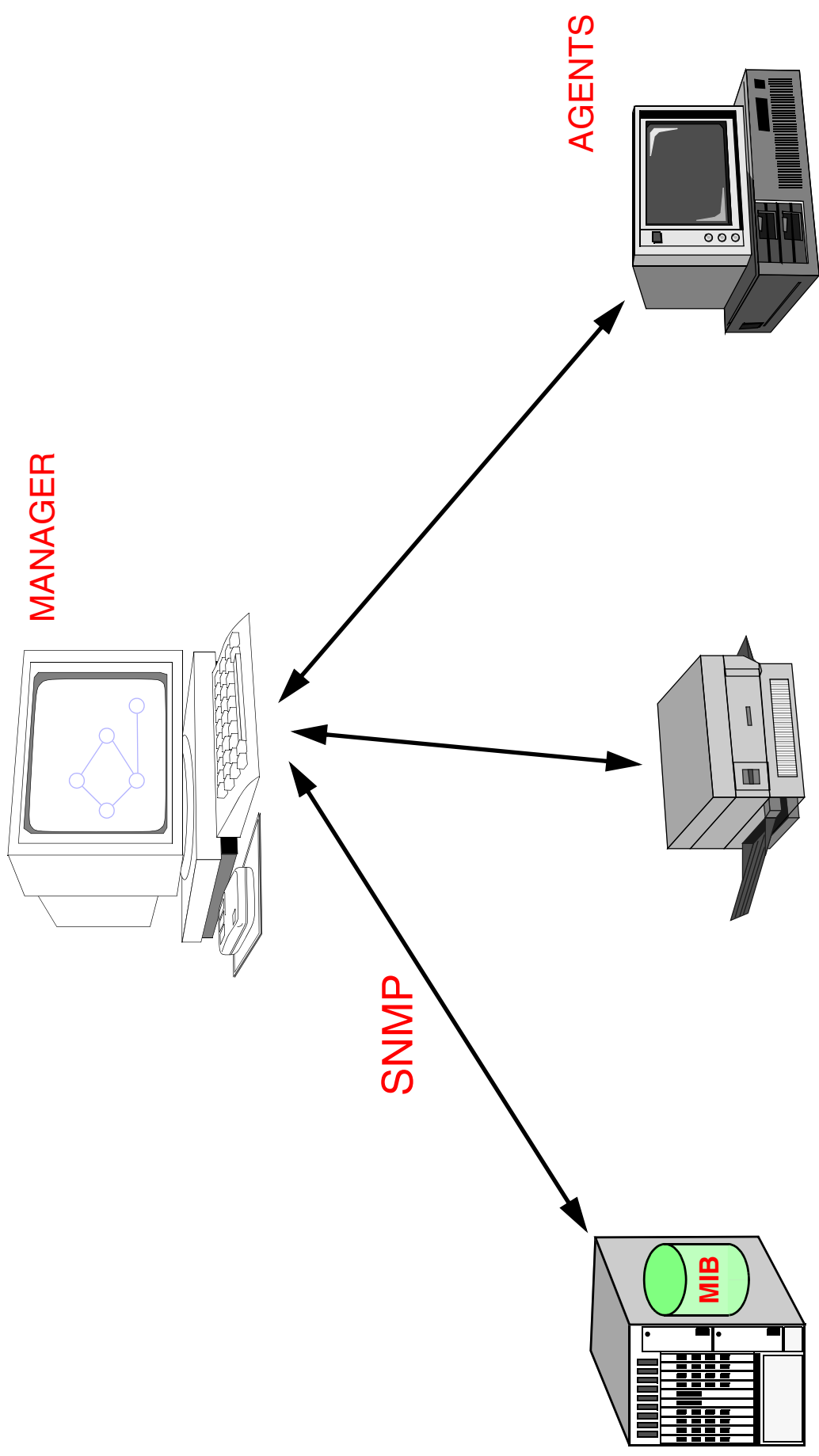
- NEW MIBs

### **MANAGEMENT SHOULD BE ROBUST**

- CONNECTIONLESS TRANSPORT

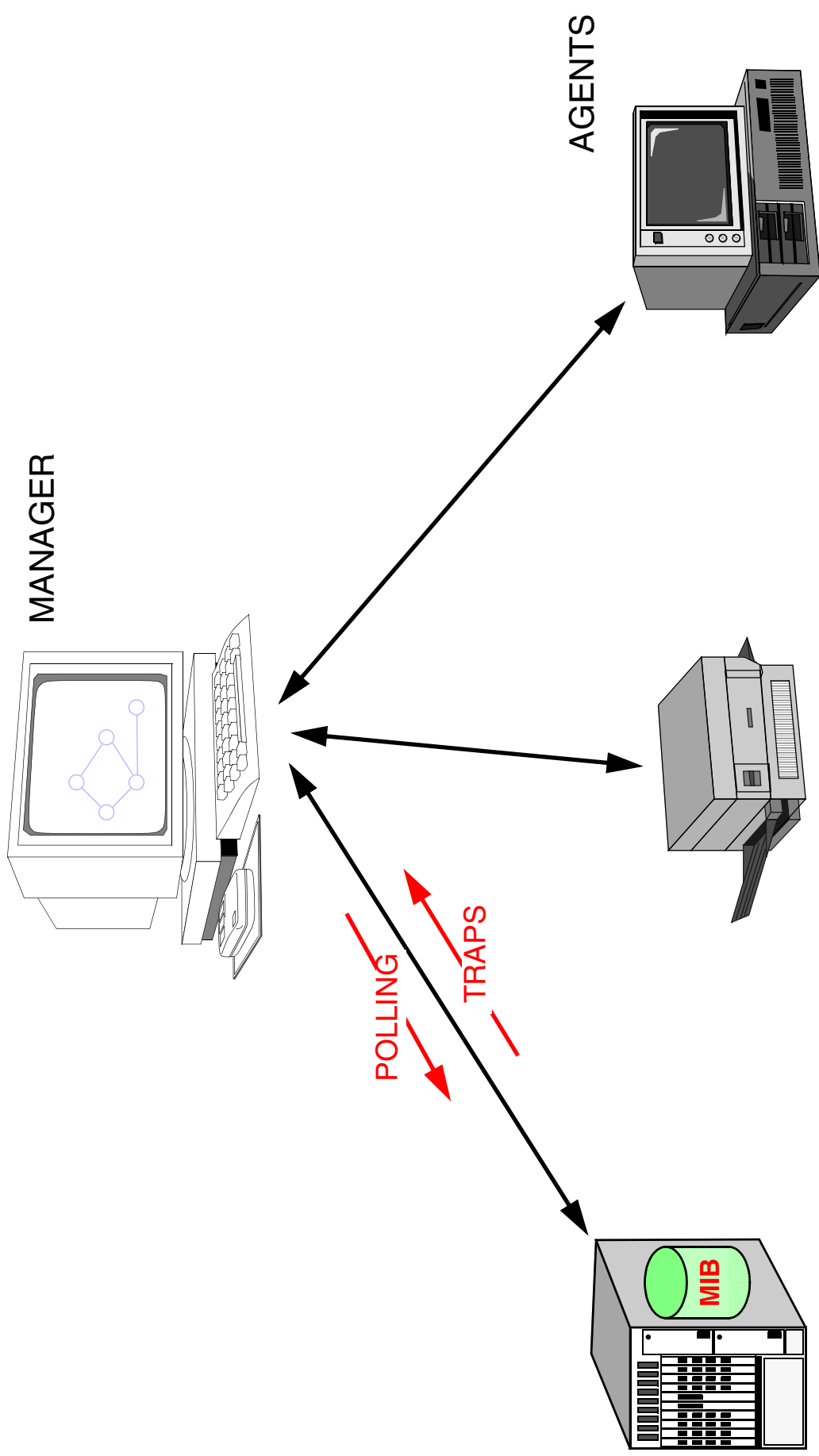


# PRINCIPLE OPERATION



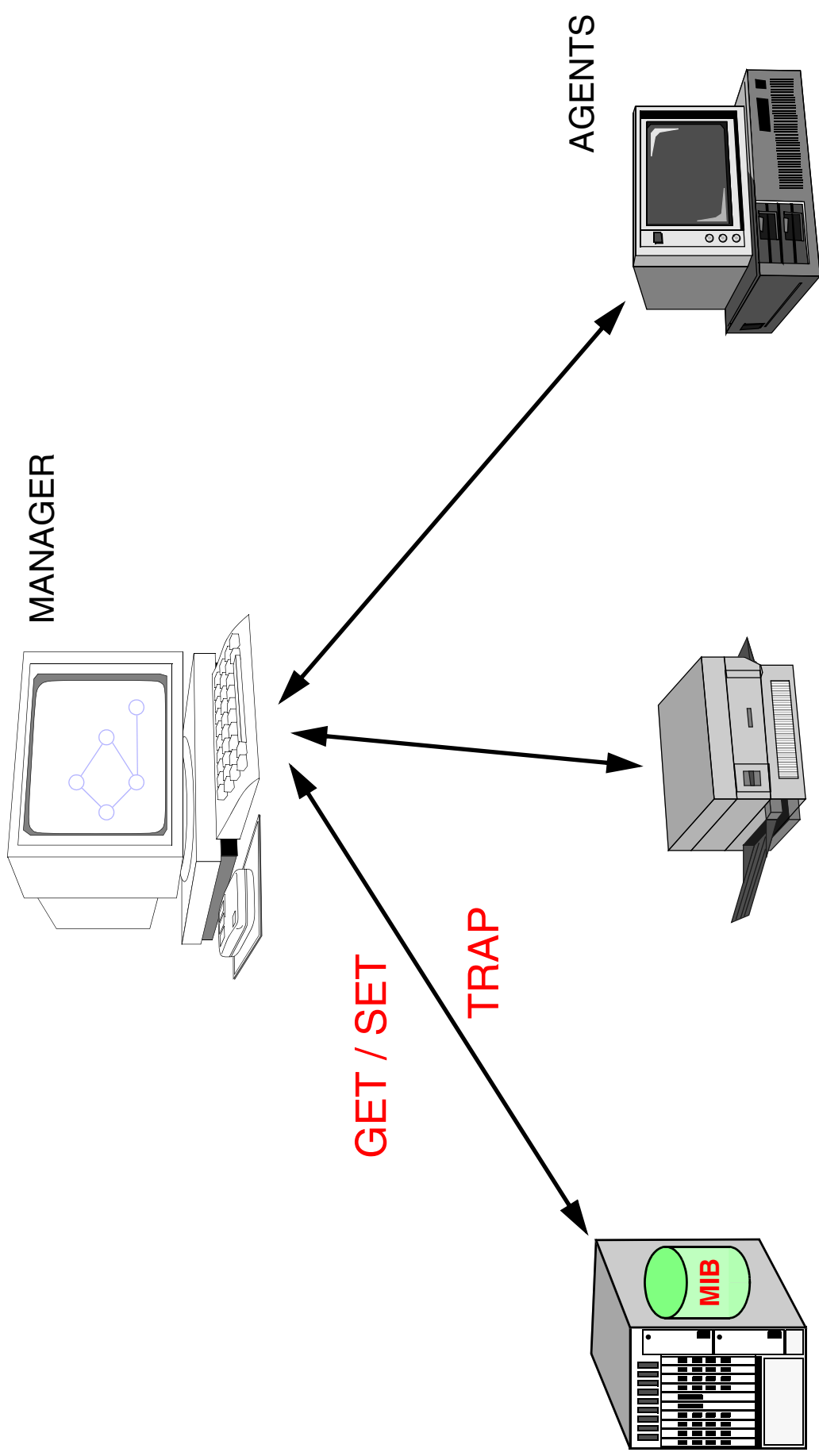


# PRINCIPLE OPERATION



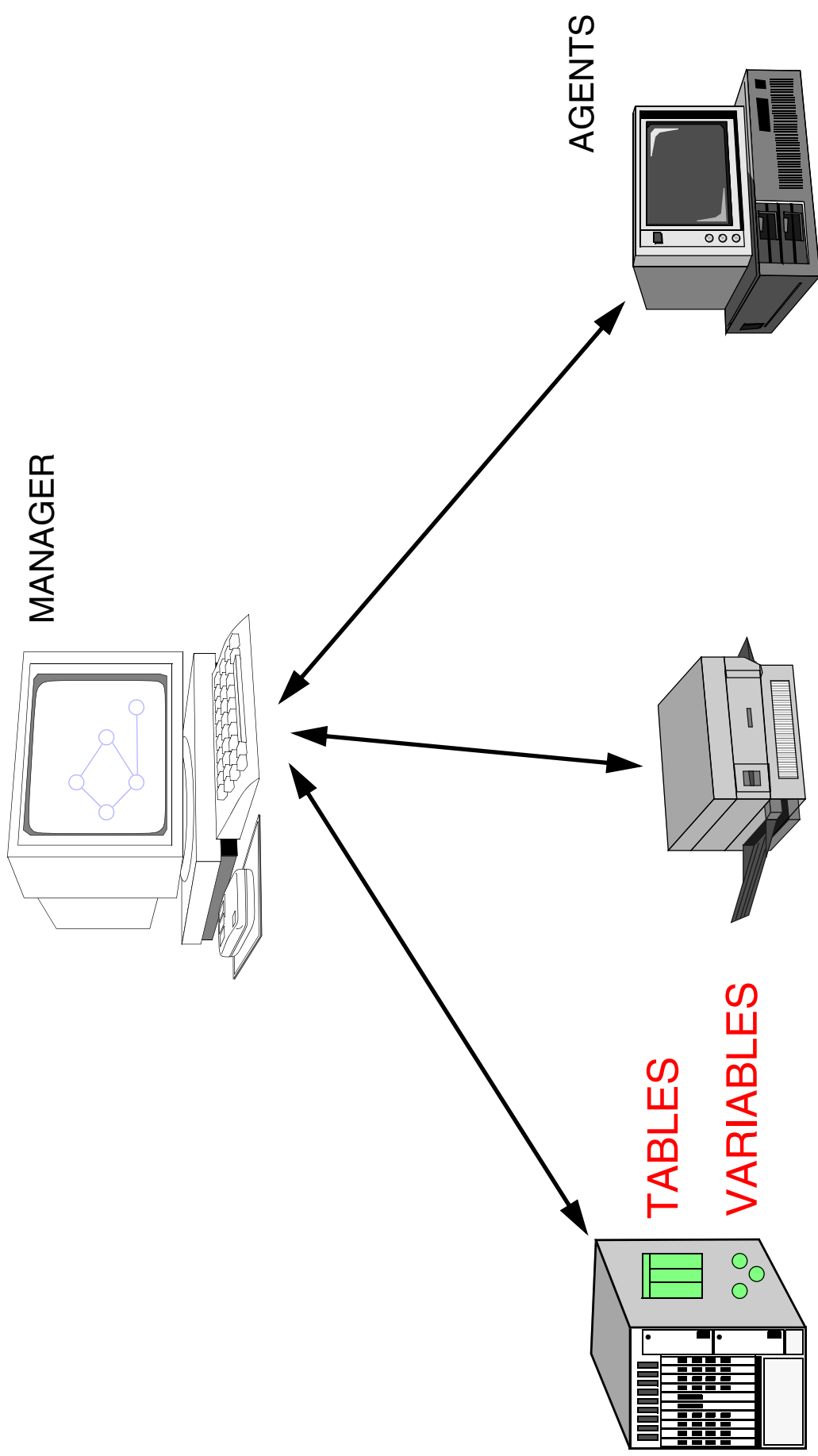


# PRINCIPLE OPERATION



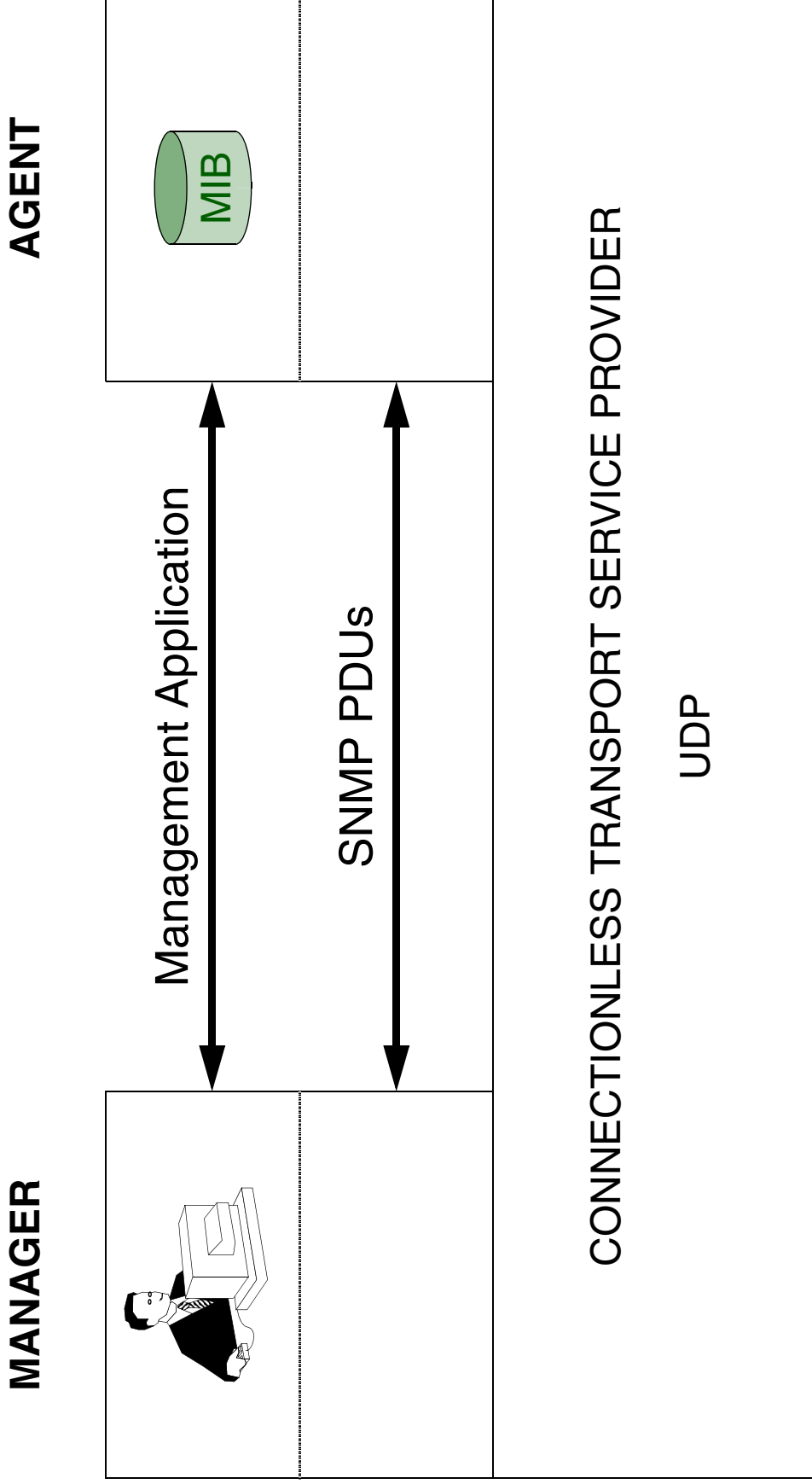


# PRINCIPLE OPERATION





# SNMP STRUCTURE





## **STANDARDS**

### **SMI**

- STRUCTURE OF MANAGEMENT INFORMATION
  - TWO VERSIONS
  - RFC 1155, RFC 2578, ...

### **MIBs**

- MANAGEMENT INFORMATION BASES
  - A LARGE NUMBER OF MIBs EXIST
  - RFC 1213, ...

### **SNMP**

- SIMPLE NETWORK MANAGEMENT PROTOCOL
  - NAME IS USED IN A MORE GENERAL SENSE
    - VERSION 1: HISTORIC (RFC 1157)
    - VERSION 3: STANDARD (RFC 3411-3416)



**SMI**

**STRUCTURE OF MANAGEMENT INFORMATION**

**RFC 1155: SMIV1**

**RFC 1212: CONCISE MIB DEFINITIONS**

**RFC 2578: SMIV2**

**RFC 2579: TEXTUAL CONVENTIONS**

**MAKES THE DEFINITION OF (NEW) MIBs EASIER**



## **SMI**

MANAGEMENT INFORMATION WITHIN MANAGED SYSTEMS  
MUST BE REPRESENTED AS:

- SCALARS
- TABLES

(= TWO DIMENSIONAL ARRAYS OF SCALARS)

THE SNMP PROTOCOL CAN ONLY EXCHANGE  
(A LIST OF) SCALARS

DEFINED IN TERMS OF ASN.1 CONSTRUCTS



# SMI: DATA TYPES FOR SCALARS

## SMIV1

### SIMPLE TYPES:

INTEGER  
OCTET STRING  
OBJECT IDENTIFIER

## SMIV2

INTEGER  
OCTET STRING  
OBJECT IDENTIFIER

### APPLICATION-WIDE TYPES:

-  
Integer32

-  
Gauge  
Counter  
-  
TimeTicks  
IpAddress  
Opaque  
NetworkAddress

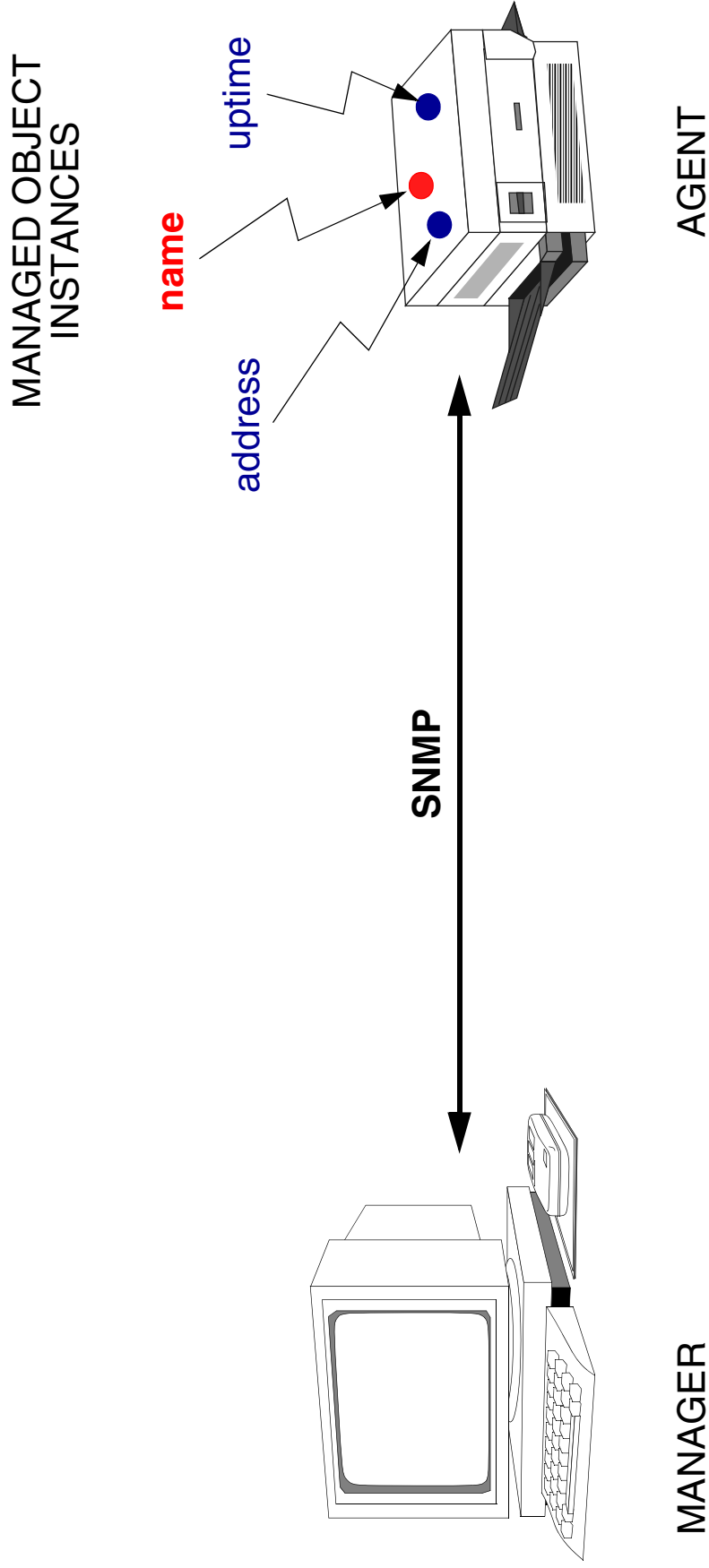
Unsigned32  
Gauge32  
Counter32  
Counter64  
TimeTicks  
IpAddress  
Opaque  
-

### PSEUDO TYPES:

-  
BITS



# EXAMPLE OF SCALAR OBJECTS



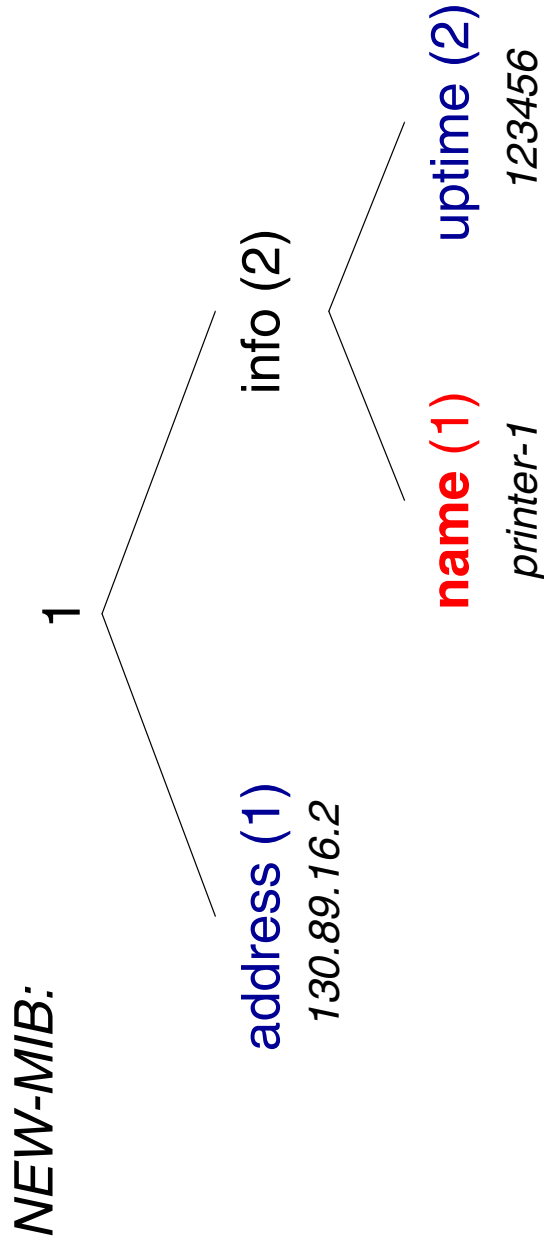
MANAGER

AGENT



## OBJECT NAMING

### INTRODUCE NAMING TREE



THE LEAVES OF THE TREE REPRESENT THE MANAGED OBJECTS

NODES ARE INTRODUCED FOR NAMING PURPOSES



## OBJECT NAMING

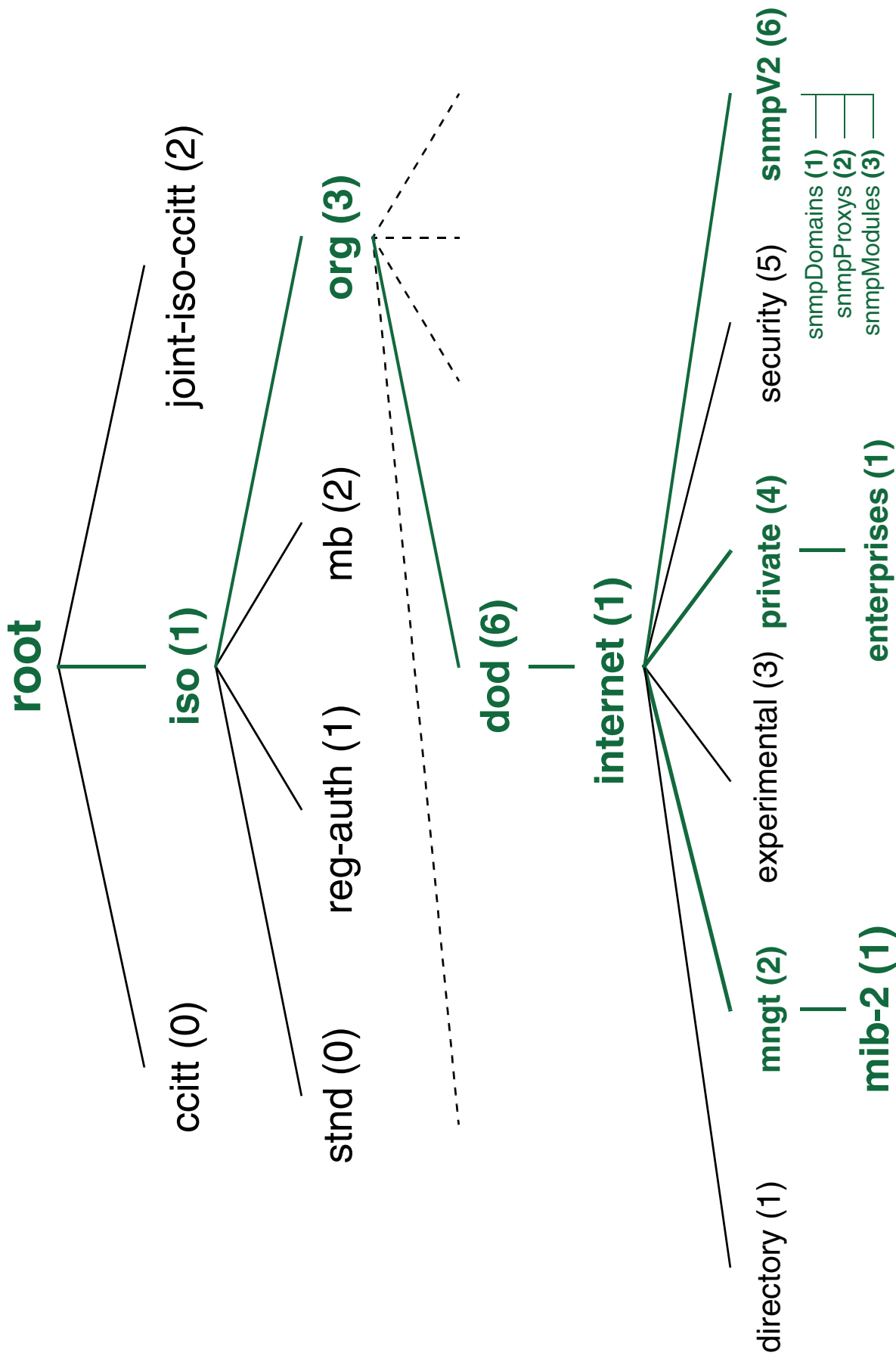
- **address**
  - Object ID = 1.1
  - Object Instance = 1.1.0
  - Value of Instance = 130.89.16.2
- **info**
  - Object ID = 1.2
- **name**
  - Object ID = 1.2.1
  - Object Instance = 1.2.1.0
  - Value of Instance = *printer-1*
- **uptime**
  - Object ID = 1.2.2
  - Object Instance = 1.2.2.0
  - Value of Instance = 123456

### *ALTERNATIVE:*

Object ID = NEW-MIB info uptime



# OBJECT NAMING: MIBS





# OBJECT TYPE DEFINITION

**OBJECT-TYPE:**

- INTEGER
- OCTET STRING
- OBJECT IDENTIFIER
- BITS
- IpAddress
- Integer32
- Counter32
- Counter64
- Gauge32
- TimeTicks
- Opaque
- New Type

## **SYNTAX**

- read-only
- read-write
- read-create
- accessible-for-notify
- not-accessible

## **MAX-ACCESS**

- current
- deprecated
- obsolete

## **STATUS**

""

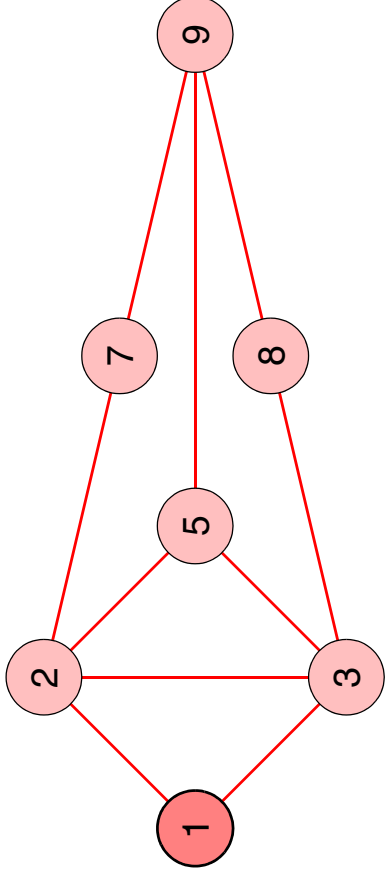
## **DESCRIPTION**



# TABLES

EXAMPLE: ROUTING TABLE

destination	next
2	2
3	3
5	2
7	2
8	3
9	3



TO RETRIEVE INDIVIDUAL TABLE ENTRIES

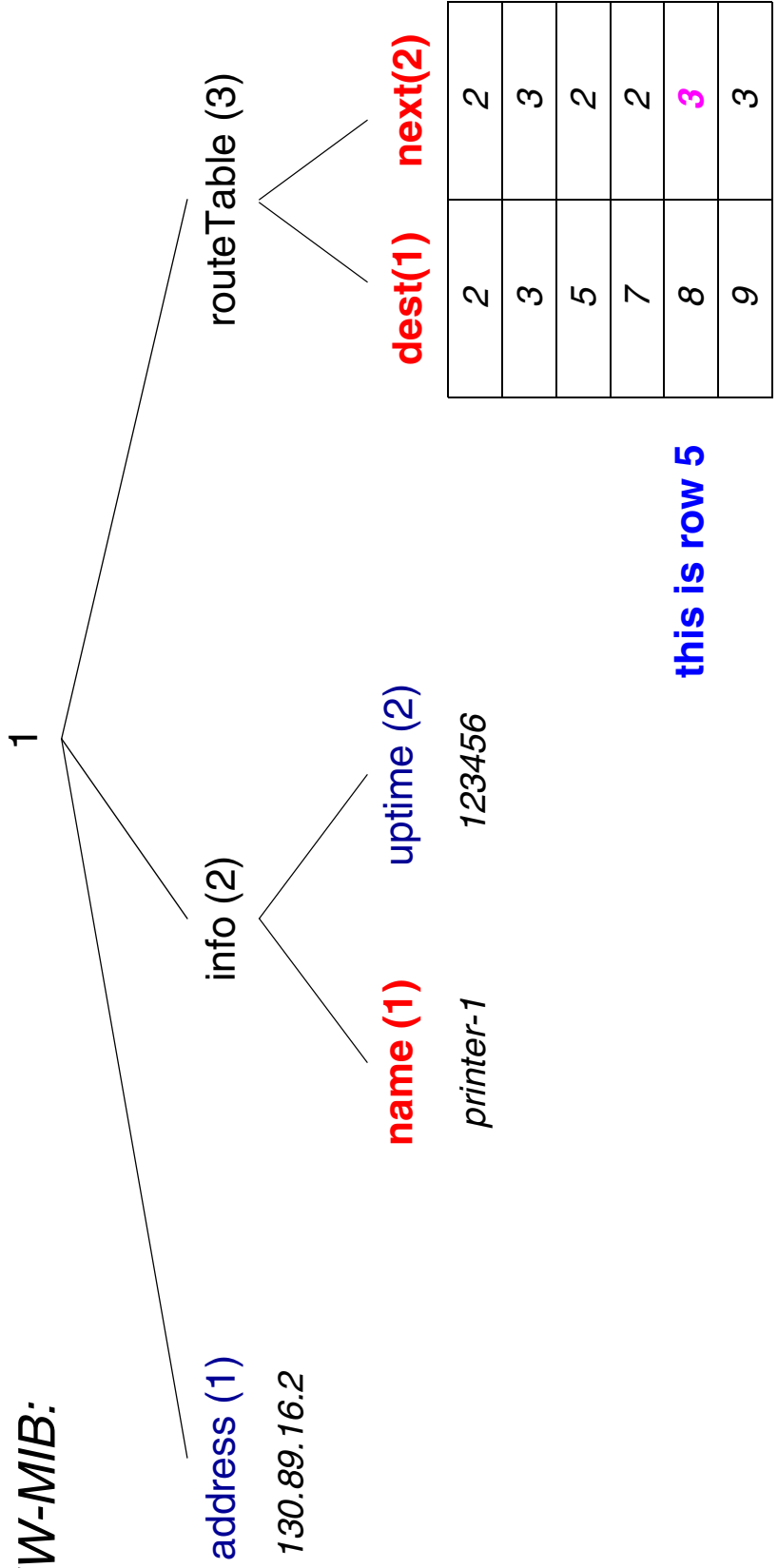
EACH ENTRY SHOULD GET AN IDENTIFIER



# NAMING OF TABLE ENTRIES - I

**POSSIBILITY 1** (NOT BEING USED BY SNMP): **USE ROW NUMBERS**

**NEW-MIB:**

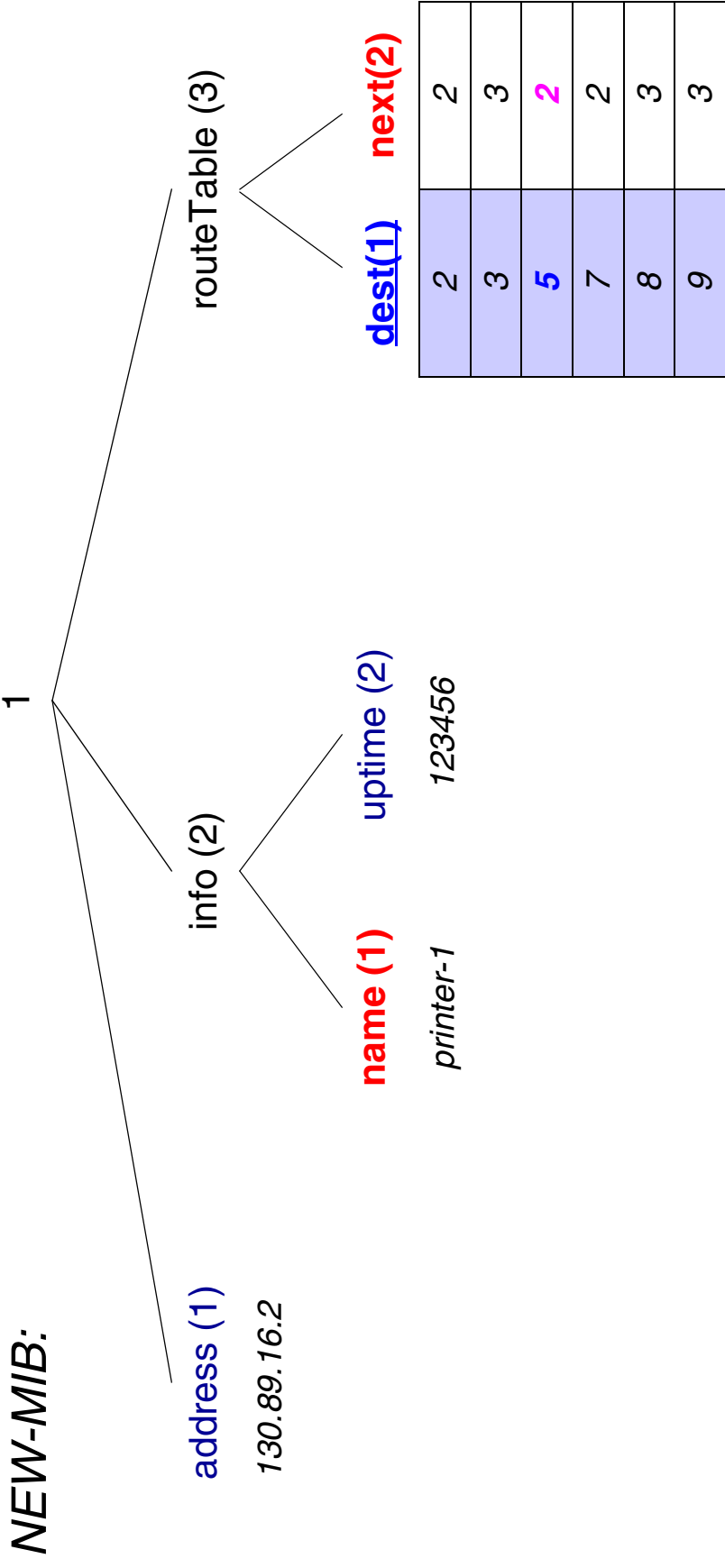


**EXAMPLE: THE VALUE OF NEW-MIB routeTable next 5 IS 3**



# NAMING OF TABLE ENTRIES - II

POSSIBILITY 2 (USED BY SNMP): INTRODUCE AN INDEX COLUMN

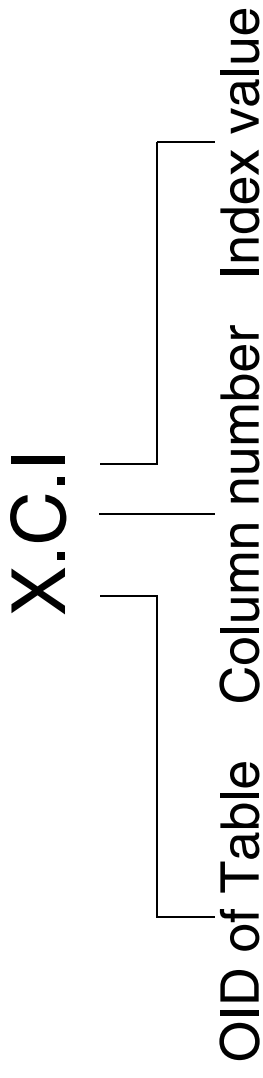


EXAMPLE: THE VALUE OF NEW-MIB routeTable next 5 IS 2



# TABLE INDEXING

## GENERAL SCHEME



### EXAMPLES:

OID of Table = 1.3

1.3.1.5  $\Rightarrow$  5

1.3.2.5  $\Rightarrow$  2

1.3.1.9  $\Rightarrow$  9

1.3.2.9  $\Rightarrow$  3

1.3.2.7  $\Rightarrow$  2

1.3.1.1  $\Rightarrow$  entry does not exist

1.3.2.1  $\Rightarrow$  entry does not exist



## TABLE INDEXING - NON-INTEGGER INDEX

AN INDEX NEED NOT BE AN INTEGER

routeTable (3)

<u>dest (1)</u>	<b>next (2)</b>
130.89.16.1	130.89.16.1
130.89.16.4	130.89.16.4
130.89.16.23	130.89.16.1
130.89.19.121	130.89.16.1
192.1.23.24	130.89.16.4
193.22.11.97	130.89.16.4

### EXAMPLES:

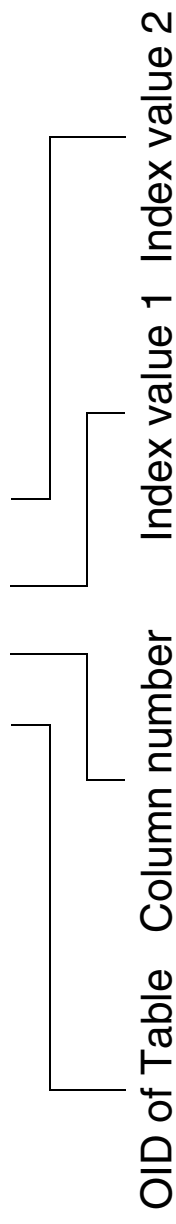
OID of Table = 1.3  
1.3.1.130.89.16.23 => 130.89.16.23  
1.3.2.130.89.16.23 => 130.89.16.1  
1.3.1.193.22.11.97 => 193.22.11.97  
1.3.2.193.22.11.97 => 130.89.16.4  
1.3.2.130.89.19.121 => 130.89.16.1



# TABLE INDEXING - MULTIPLE INDEX FIELDS

## USE OF MULTIPLE INDEX FIELDS

X.C.I1.I2





## TABLE INDEXING - MULTIPLE INDEX FIELDS: EXAMPLE

### EXAMPLE:

1 = low costs  
2 = high reliability

routeTable (3)

dest (1)    policy (2)    next (3)

130.89.16.23	1	130.89.16.23
130.89.16.23	2	130.89.16.23
130.89.19.121	1	130.89.16.1
192.1.23.24	1	130.89.16.1
192.1.23.24	2	130.89.16.4
193.22.11.97	1	130.89.16.1

1.3.3.192.1.23.24.1 => 130.89.16.1

1.3.3.192.1.23.24.2 => 130.89.16.4



## DEFINITION OF NEW TYPES

### TEXTUAL CONVENTIONS

TO REFINE SEMANTICS OF EXISTING TYPES

EXAMPLE:

```
RunState ::= TEXTUAL CONVENTION  
STATUS current  
DESCRIPTION "..."  
SYNTAX INTEGER{  
    running(1)  
    runnable(2)  
    waiting(3)  
    exiting(4) }
```



## **TEXTUAL CONVENTIONS**

- PhysAddress
- MacAddress
  - TruthValue
- AutonomousType
- InstancePointer
- VariablePointer
  - RowPointer
  - RowStatus
  - TimeStamp
  - TimeInterval
- DateAndTime
- StorageType
  - TDomain
  - TAddress
- Inet-Address...



## NOTIFICATION TYPES

- SMIV2:
- MIBs MAY NOW INCLUDE NOTIFICATION TYPE MACROS

### EXAMPLE:

```
LinkUp NOTIFICATION-TYPE
OBJECTS {ifIndex}
STATUS current
DESCRIPTION
    "A linkUp trap signifies that the
    entity has detected that the
    ifOperStatus object has changed to Up"
::= {snmpTraps 4}
```



## PROBLEMS WITH SMIV1 / v2

- SMIV2 RELIED ON 1988 VERSION OF ASN.1
- TOOLS FOR SMIV2 RELATIVELY COMPLEX
- CERTAIN DATA TYPES WERE MISSING IN SMIV2  
64 bit integers, ...
- LIMITED FACILITIES TO REUSE DEFINITIONS
- SMIV2 DID NOT ALLOW FOR EXTENSIONS
- NEW, POSSIBLY INCOMPATIBLE VARIANTS APPEARED  
SPPI, ...



TO RESOLVE THESE PROBLEMS

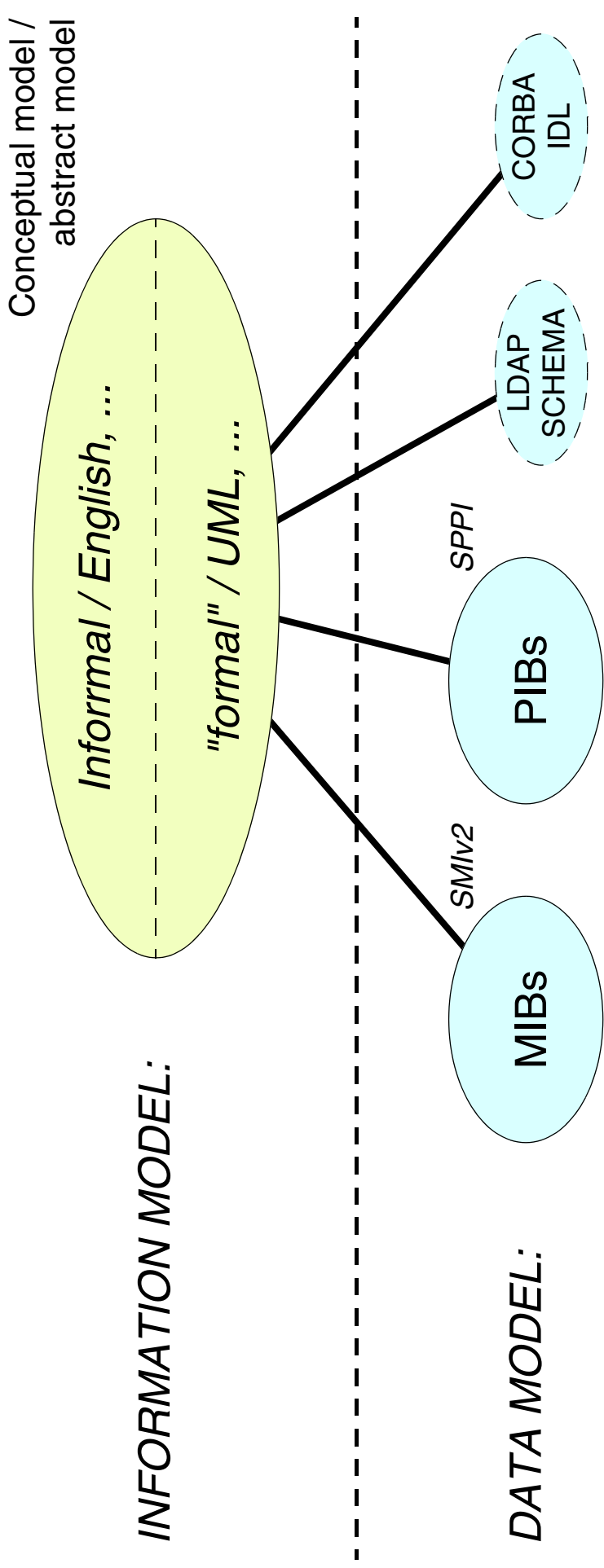
THE IRTF NMRG DEFINED:

SMI Next Generation (ng)

THE CHALLENGE:  
CREATE A COMMON **DATA DEFINITION LANGUAGE**,  
INDEPENDENT OF SPECIFIC PROTOCOLS

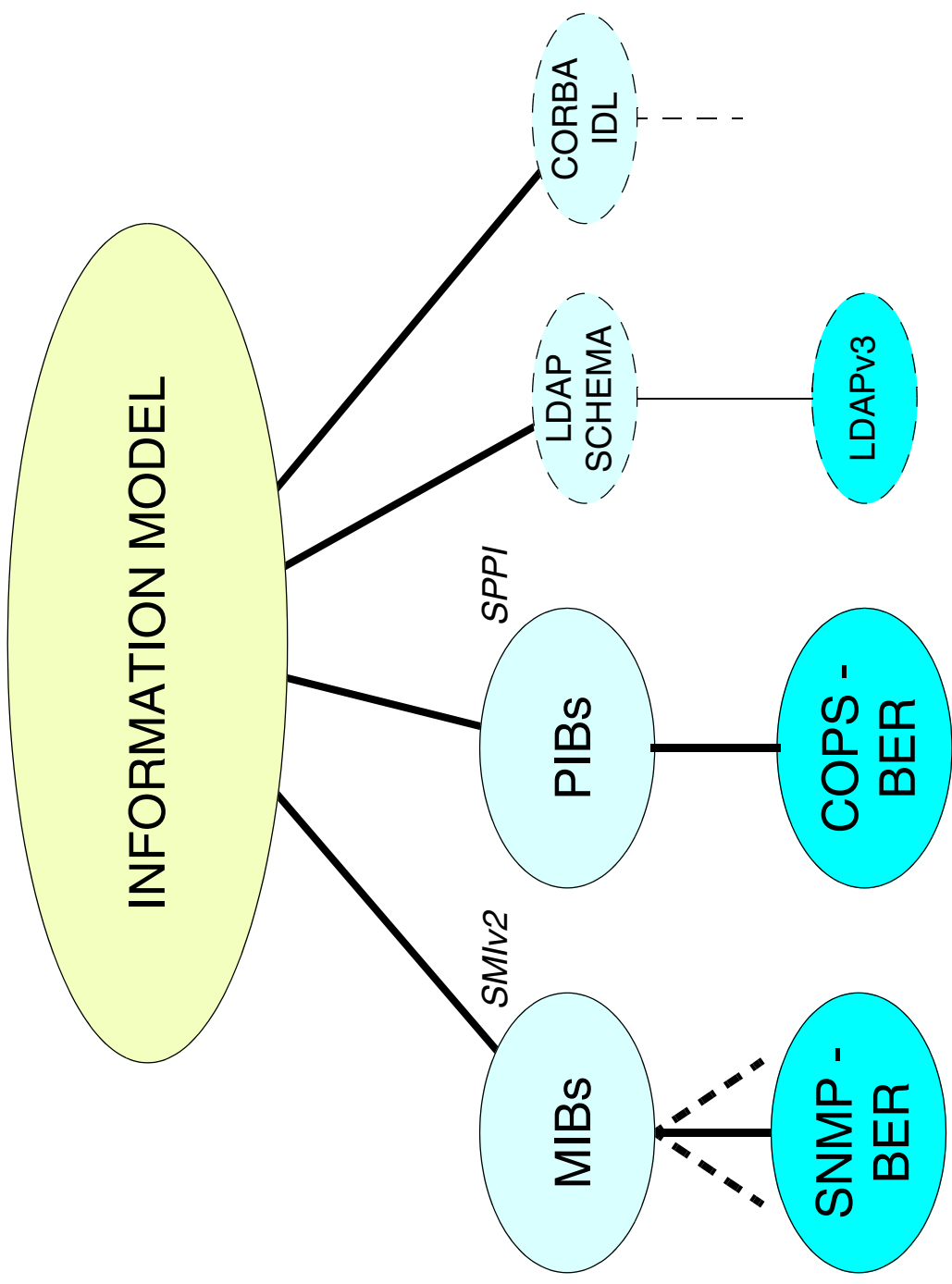


# DATA VERSUS INFORMATION MODEL





# DATA MODEL & "TRANSFER" PROTOCOL"

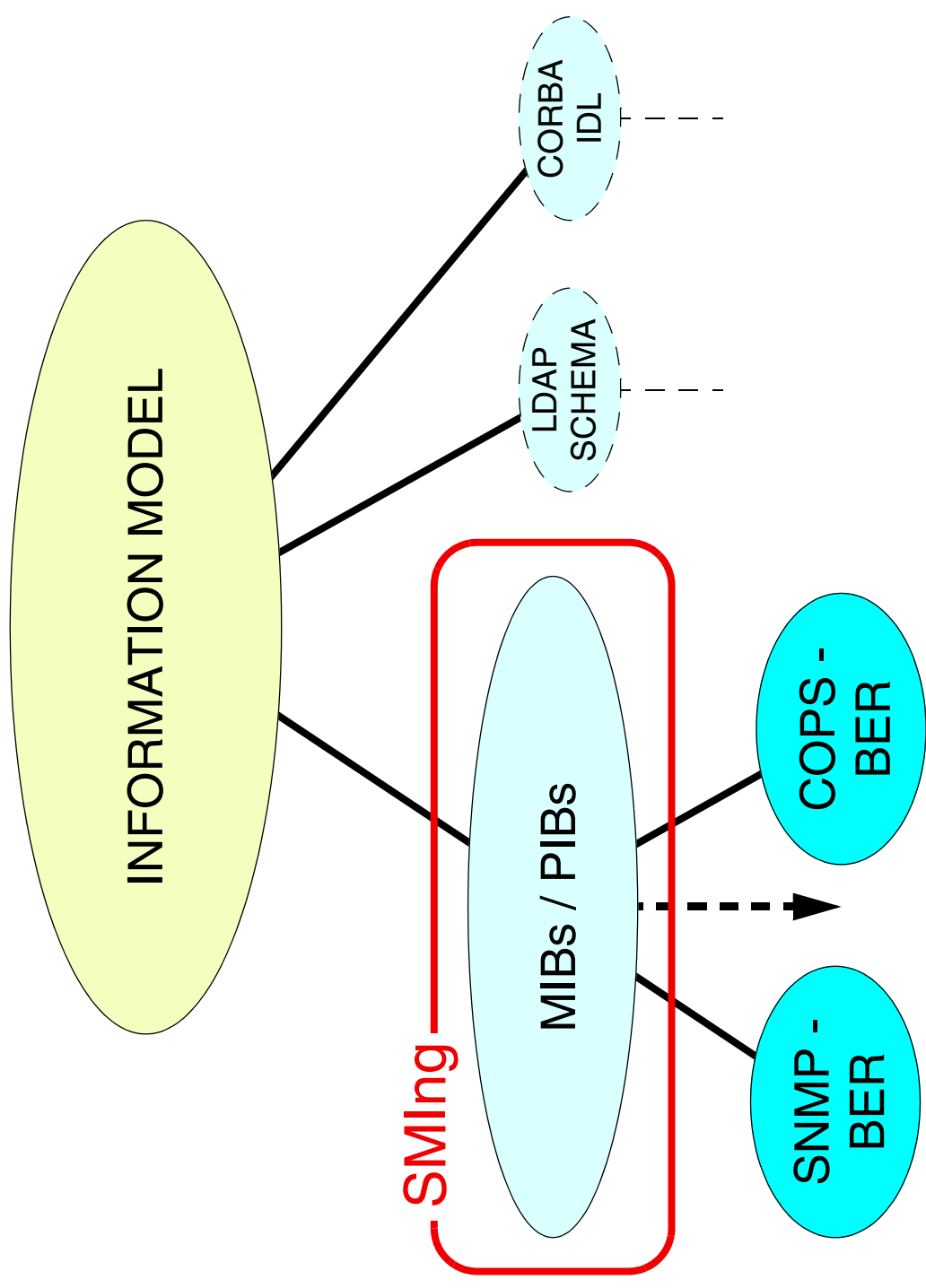


DATA MODEL:

"TRANSFER"  
PROTOCOL:



# ROLE OF SMIng



DATA MODEL:



## **SMIPv3**

IRTF-NMRG PROPOSAL FORWARDED TO IETF  
SMIng WORKING GROUP FORMED

SMIPv3

2000



## **SMIV3 OBJECTIVES**

ALLOW ARBITRARILY NESTED DATA STRUCTURES  
EASE REUSABILITY OF COMPLEX STRUCTURED TYPES

DETAILED LIST OF OBJECTIVES: RFC 3216

### **HOWEVER:**

PROTOCOL INDEPENDENCE NO LONGER DESIRABLE  
LANGUAGE EXTENSIBILITY NO LONGER DESIRABLE

**FINALLY SMIV3 IS A RELATIVE SMALL UPDATE OF SMIV2**



## MIBs

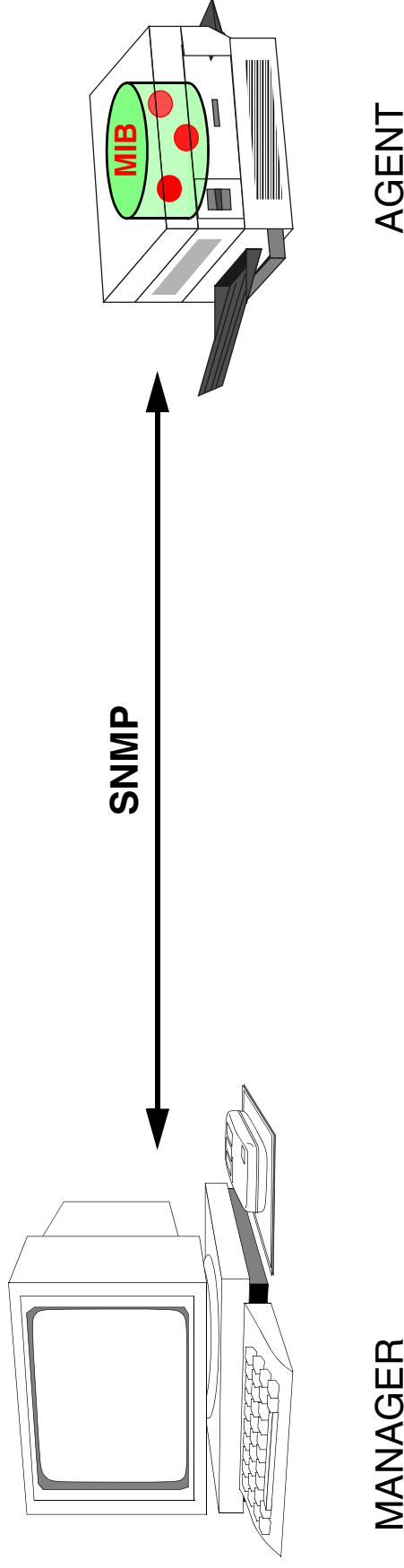
### MANAGEMENT INFORMATION BASES

CONTAIN THE MANAGED OBJECTS (VARIABLES)

THAT REPRESENT THE RESOURCES OF A SYSTEM

AND WHICH MAY BE MONITORED AND MODIFIED BY A (REMOTE) MANAGER

TO CONTROL THE BEHAVIOUR OF THAT SYSTEM





## **MIB DEFINITION AND MIB INSTANCE**

MIB DEFINITIONS SHOULD BE KNOWN BY:

- THE IMPLEMENTORS OF THE MANAGED SYSTEM
- THE MANAGER

THE MIB IS INSTANTIATED WITHIN THE MANAGED SYSTEM



## MODULARITY

THE MANAGED OBJECTS OF A SYSTEM  
ARE USUALLY DEFINED IN MULTIPLE MIB DEFINITIONS

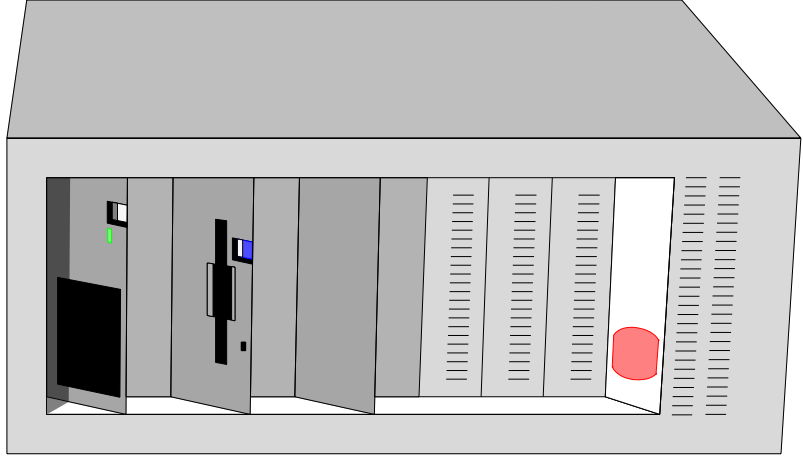
### MODULES

- DIFFERENT MODULES CAN BE DEFINED BY DIFFERENT TEAMS
- MANAGEMENT FUNCTIONALITY CAN GRADUALLY BE EXTENDED
  - DIFFERENT TYPES OF SYSTEMS  
CAN SUPPORT DIFFERENT MIB MODULES
- VENDORS CAN EXTEND THE MANAGEMENT FUNCTIONALITY  
VIA PROPRIETARY MIBS

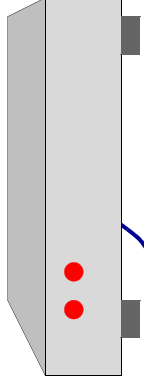


# HARDWARE MIBS

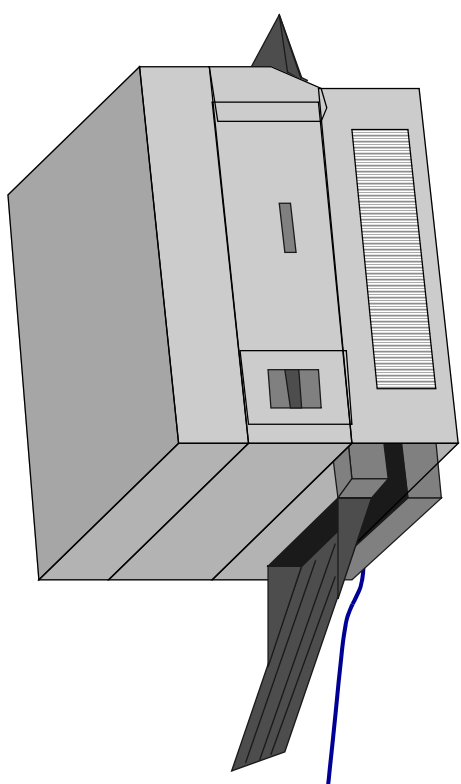
**HOST RESOURCES MIB**



**MODEM MIB**

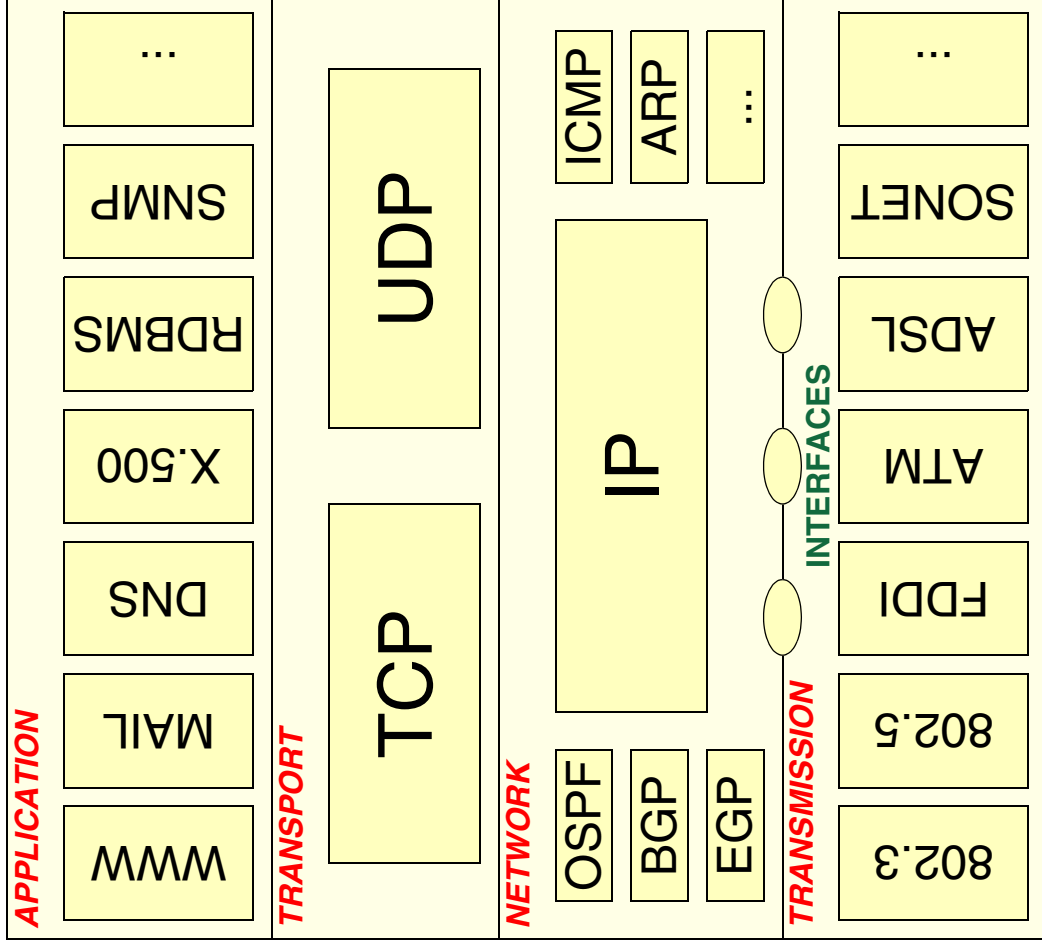


**PRINTER MIB**



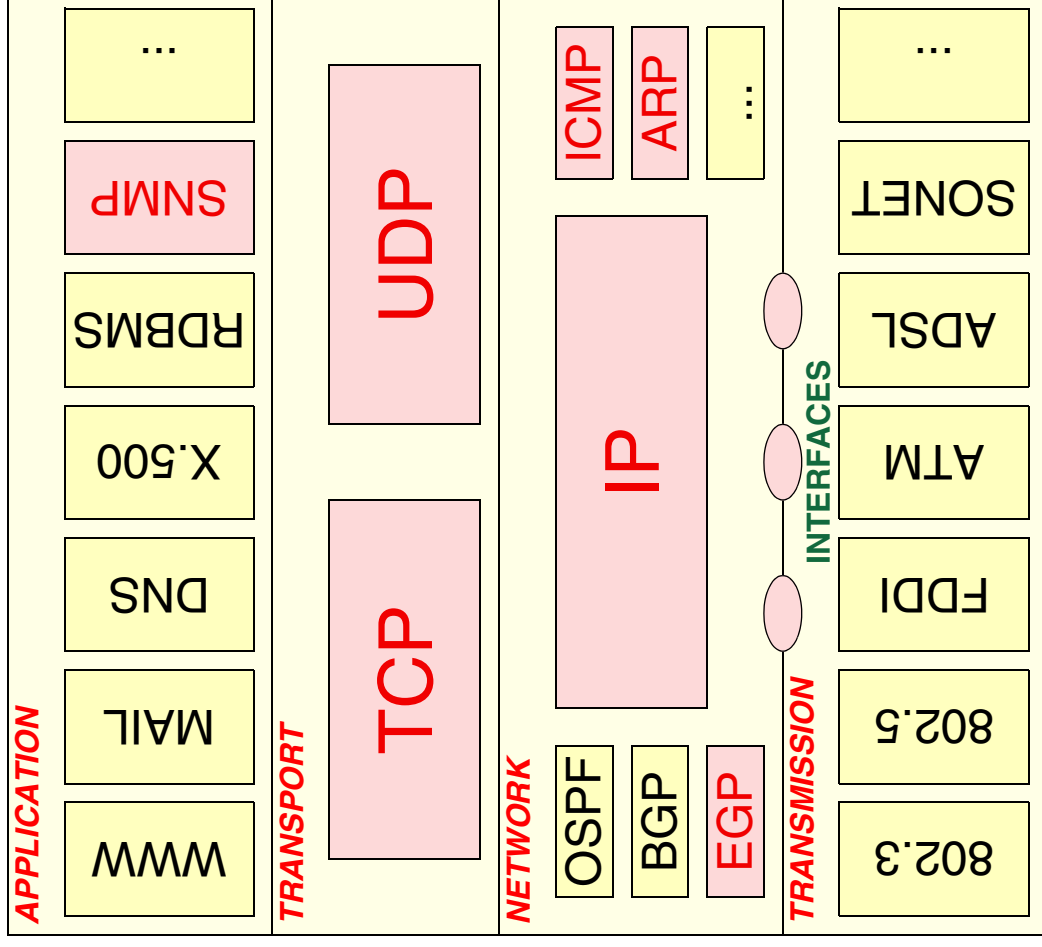


# PROTOCOL MIBS



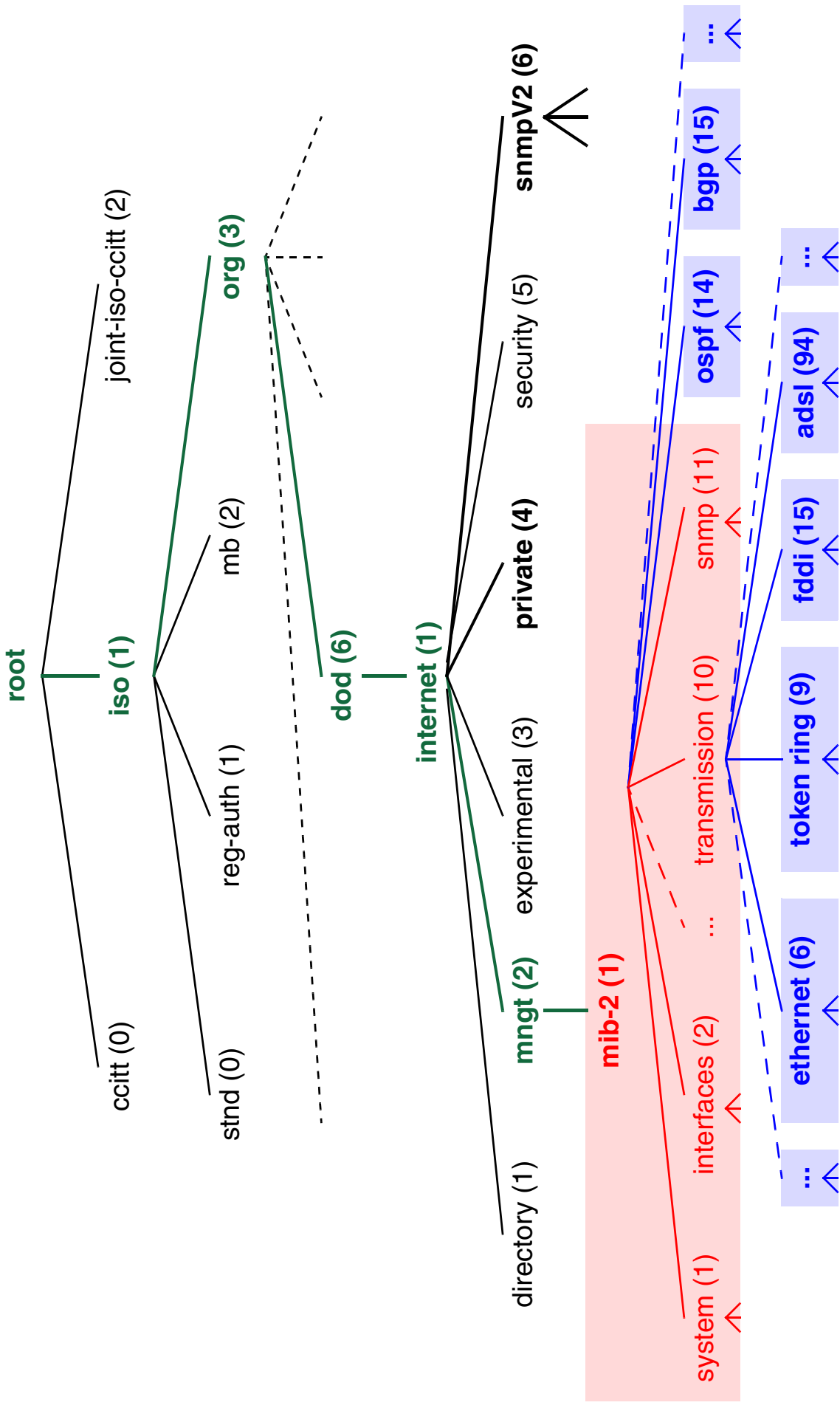


## PROTOCOL MIBS - EXAMPLE: MIB-II





# NAMING OF MIBS





## **MIB-II**

DEFINES THE VARIABLES TO MANAGE THE  
TCP/IP PROTOCOL STACK

170 VARIABLES

RFC 1213  
SMIV1

ENHANCEMENT OF MIB-I

RFC 1156

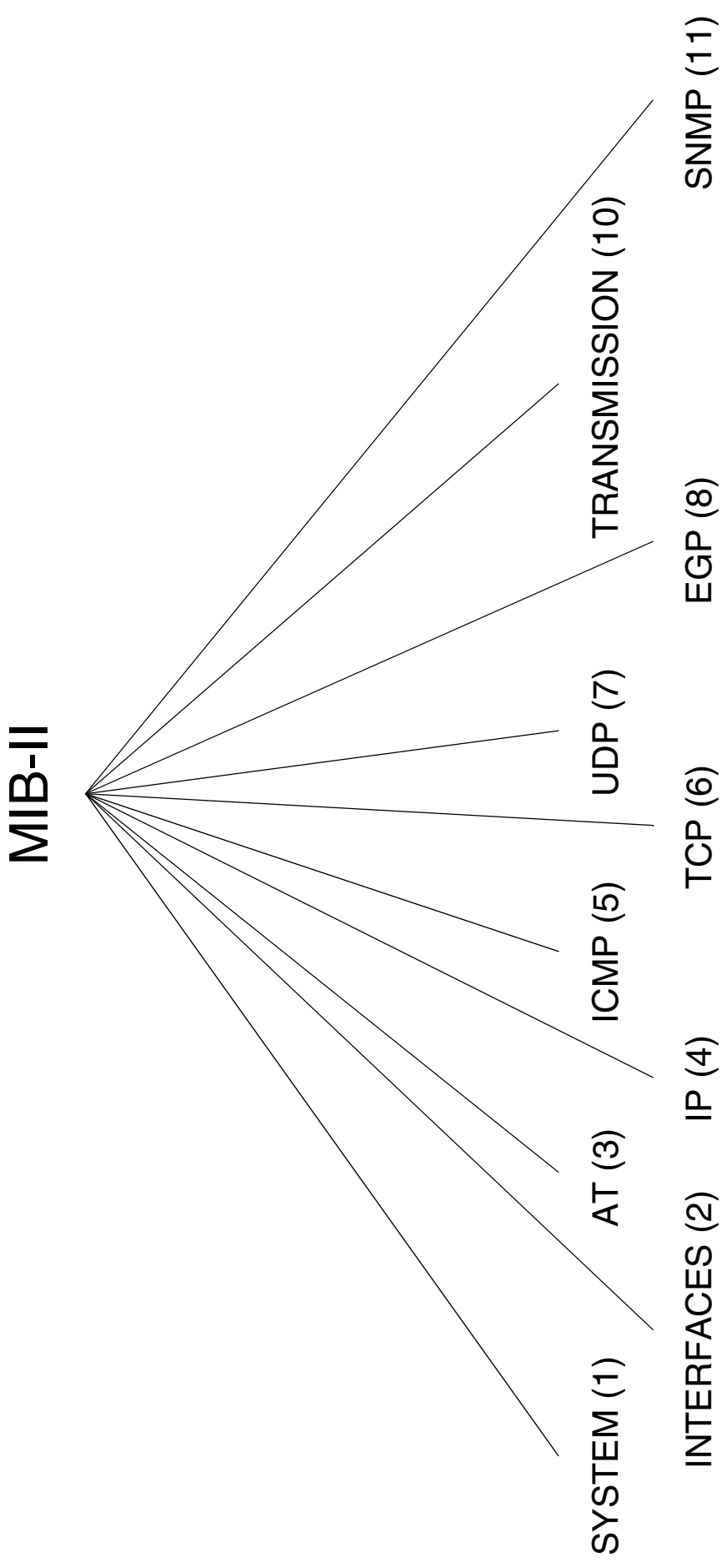


## **DESIGN CRITERIA**

- ESSENTIAL FOR FAULT OR CONFIGURATION MANAGEMENT
- ONLY WEAK CONTROL OBJECTS
- SMALL NUMBER OF OBJECTS
  - AVOID REDUNDANCY
  - EVIDENCE OF UTILITY
- DO NOT DISTURB NORMAL OPERATION
- NO IMPLEMENTATION SPECIFIC ISSUES

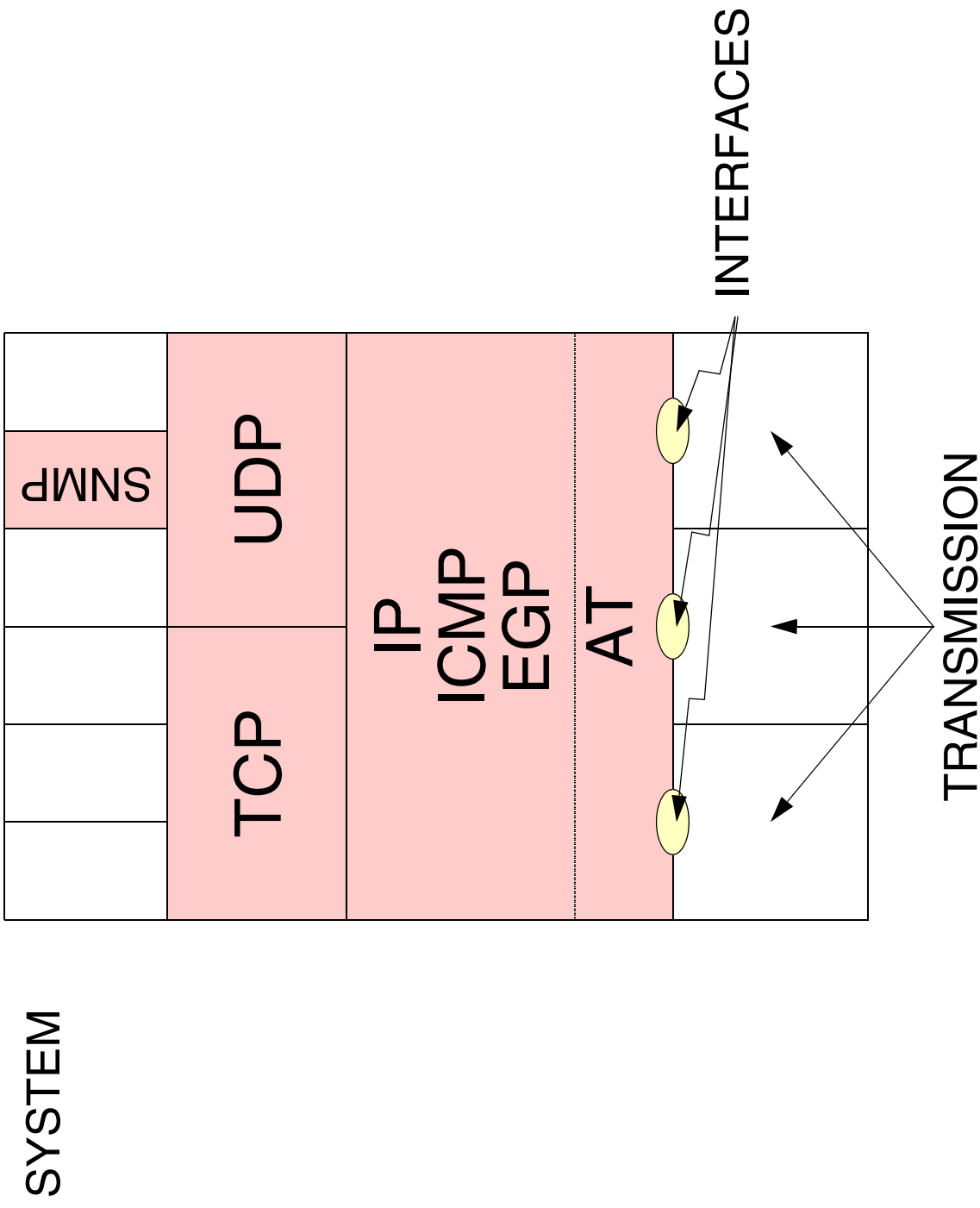


# STRUCTURE





# MIB-II GROUPS IN A PROTOCOL STACK





## **NEW VERSIONS**

SYSTEM GROUP → SNMPv2 MIB (RFC 3418)

INTERFACES (IF) GROUP → IF-MIB (RFC 2863)

ADDRESS TRANSLATION (AT) GROUP → DEPRECATED

IP & ICMP GROUPS → IP-MIB (RFC 2011)

TCP GROUP → TCP-MIB (RFC 2012)

UDP GROUP → UDP-MIB (RFC 2013)

EGP GROUP → OUTDATED (BGP)

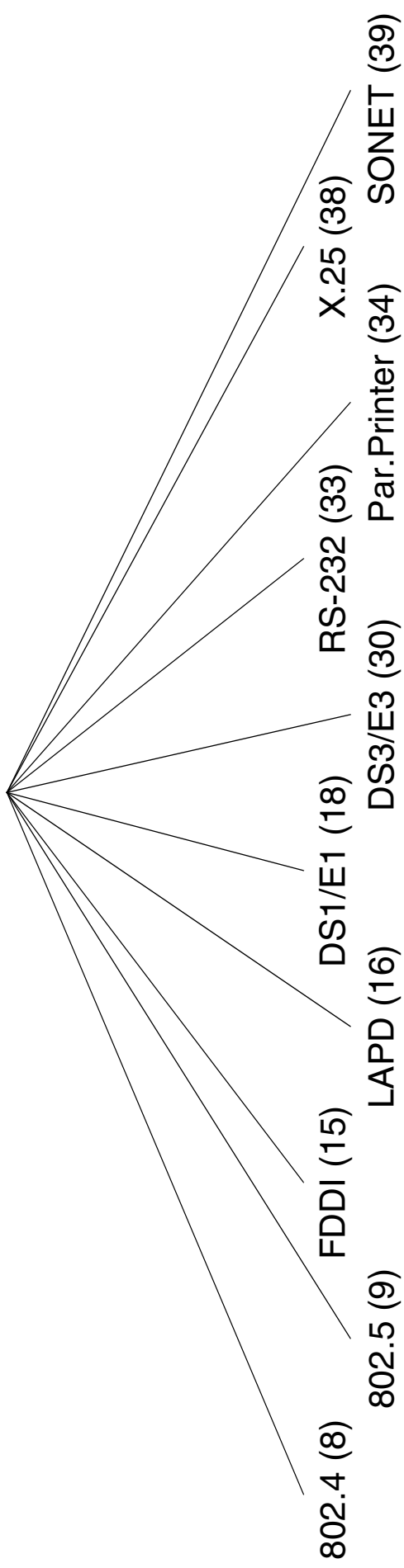
TRANSMISSION GROUP → IS PLACEHOLDER

SNMP GROUP → SNMPv2 MIB (RFC 3418)



# TRANSMISSION GROUP

transmission (10)





## **SNMPv2 MIB**

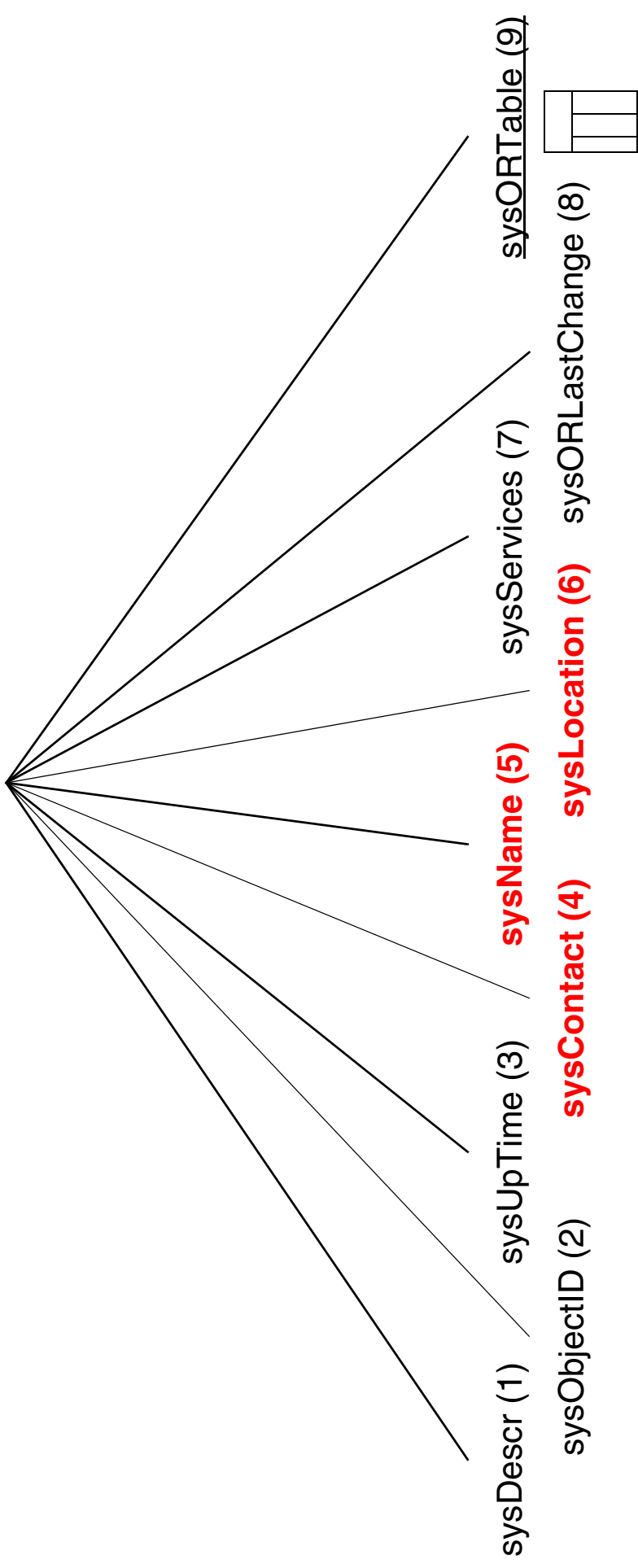
RFC 3418  
STANDARD

- SYSTEM GROUP
- SNMP GROUP
- SNMP MIBObjects GROUP
  - snmpTrap
  - snmpTraps
  - snmpSet (snmpSetSerialNo)



# SYSTEM GROUP

system (1)





## **IF MIB**

**RFC 2863**  
DRAFT STANDARD

**REPLACES IF GROUP OF MIB-II**

- RFC 1213
- RFC1229 (EXTENSIONS TO THE IF GROUP)

**DEFINES THE FOLLOWING MAIN TABLES:**

- ifStackTable
- ifTable
- ifXTable



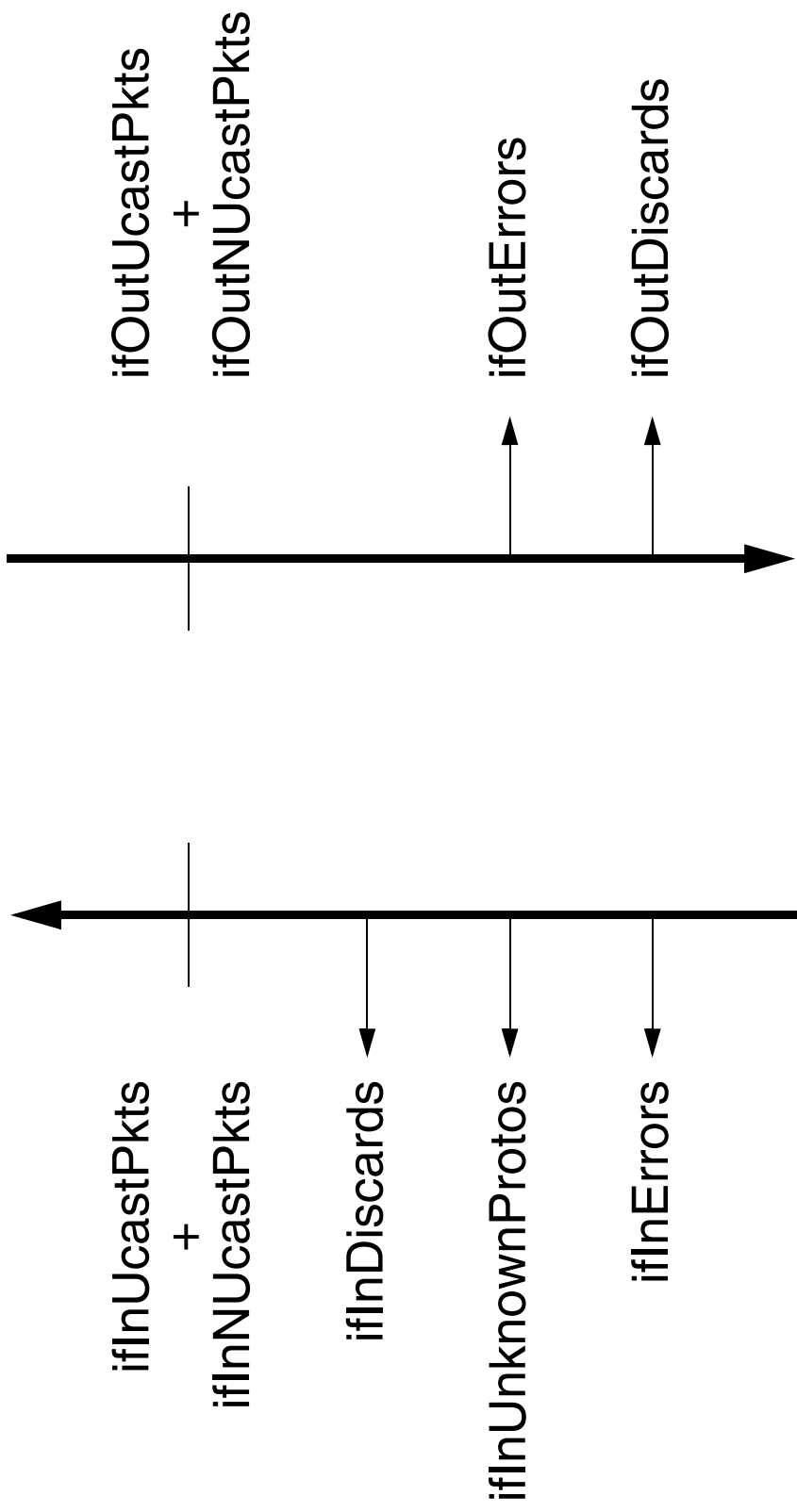
## ifTable

ifIndex	1	2		3
ifDescr				
ifType				
ifMtu				
ifSpeed				
ifPhysAddress				
<b>ifAdminStatus</b>				
ifOperStatus				
ifLastChange				
ifInOctets				
ifInUcastPkts				
ifInNUcastPkts				
ifInDiscards				
ifInErrors				
ifInUnknownProtos				
ifOutOctets				
ifOutUcastPkts				
ifOutNUcastPkts				
ifOutDiscards				
ifOutErrors				
ifOutLen				
ifSpecific	●	●		●





# IF PACKET COUNT





## **IP MIB**

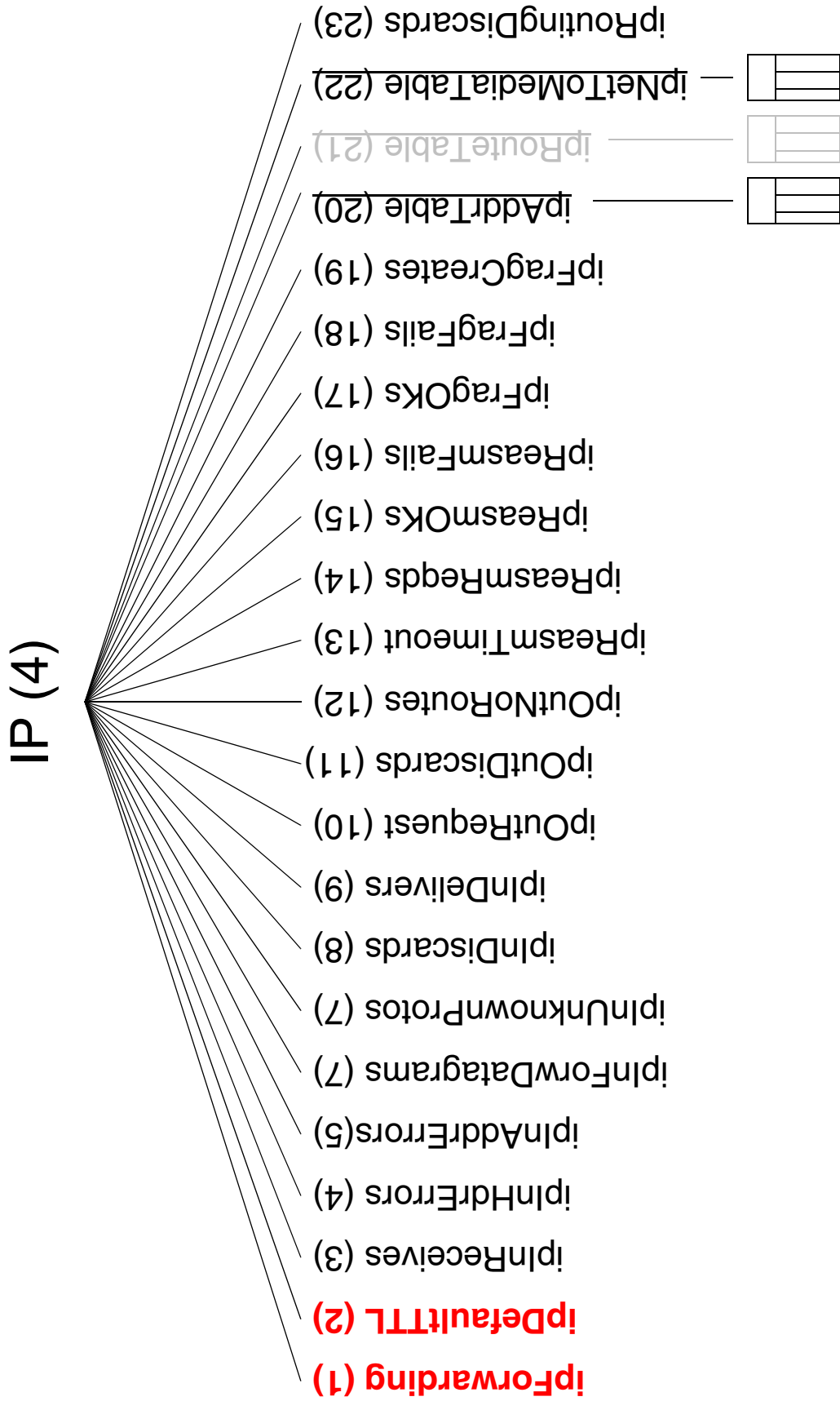
RFC 2011

PROPOSED STANDARD

- IP GROUP
- ICMP GROUP
- IP MIB Conformance

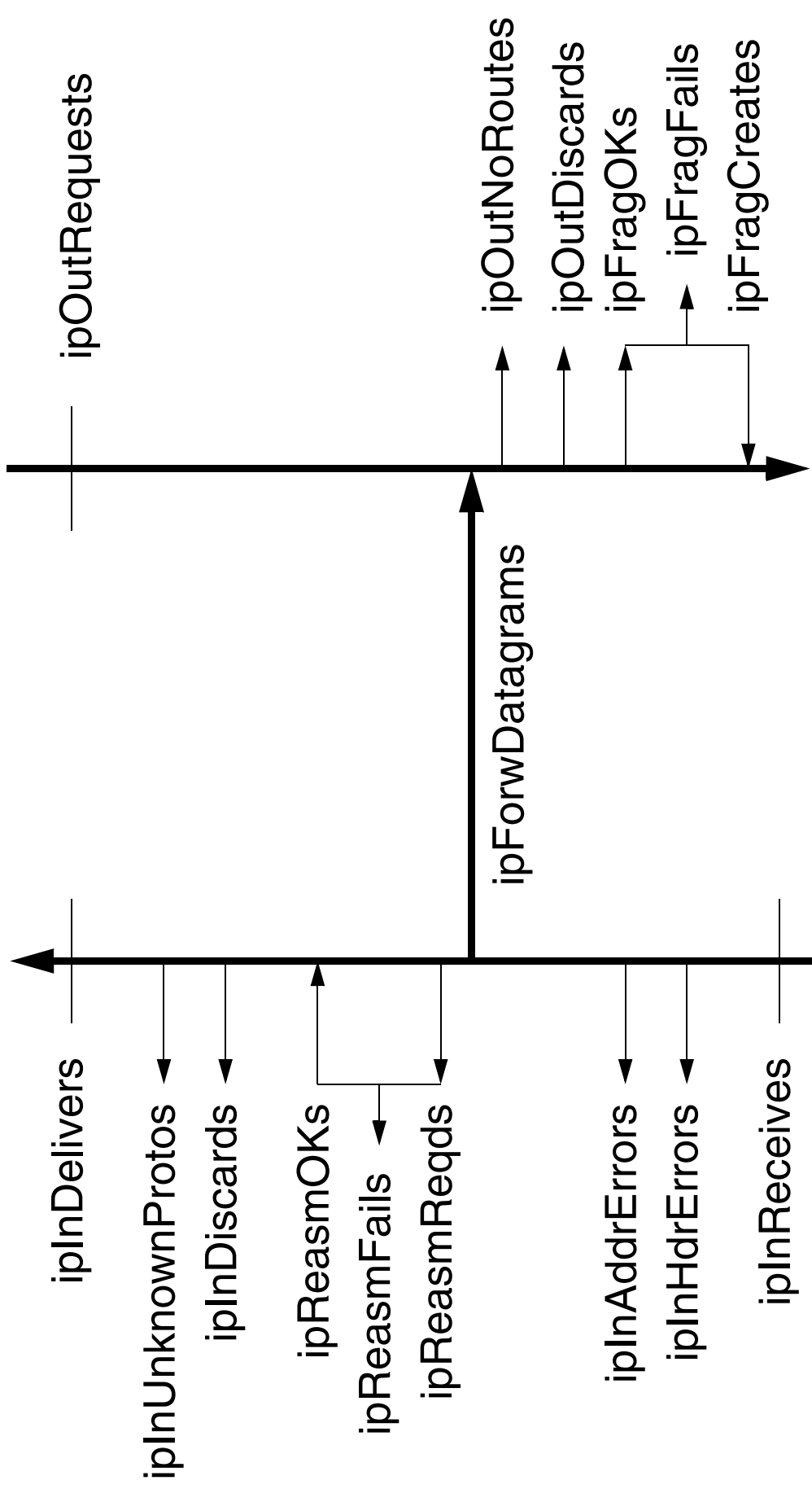


# IP GROUP





# IP PACKET COUNT





# **SNMP PROTOCOL**

## **VERSION 1**

## **VERSION 2**

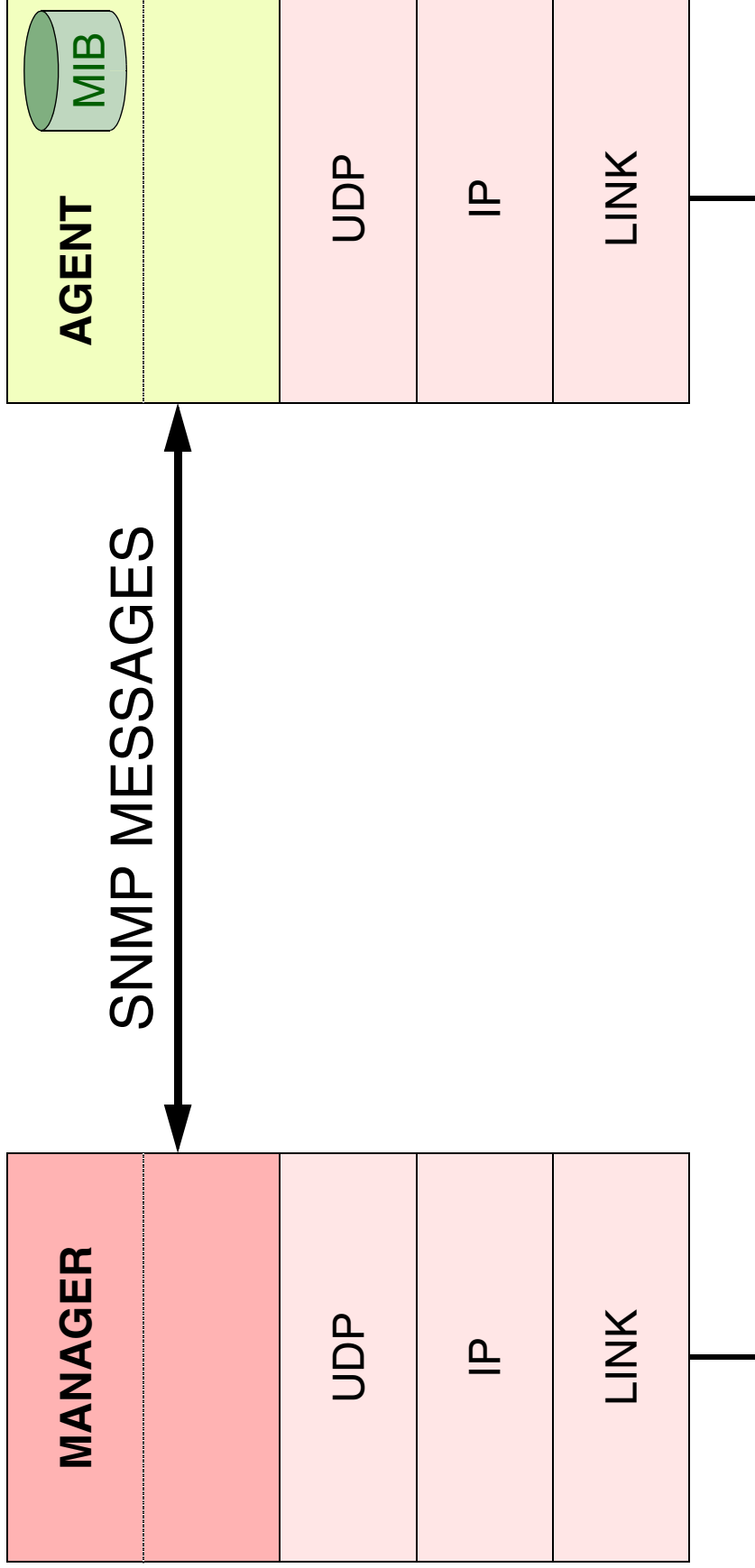
- HISTORY
- PROTOCOL OPERATIONS

## **VERSION 3**

- ARCHITECTURE
- MESSAGE STRUCTURE
- SECURITY

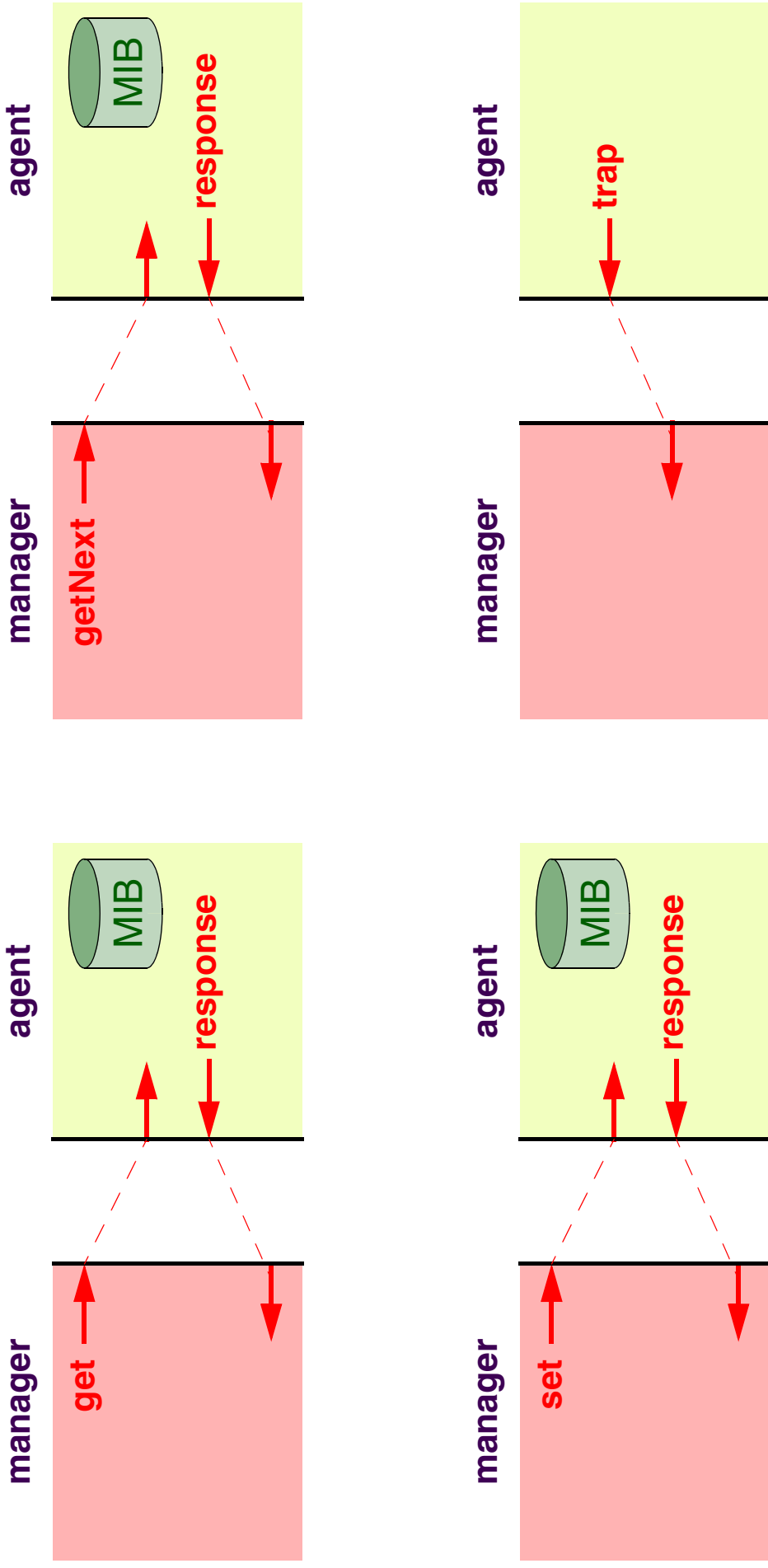


# SNMPV1





# OVERVIEW OF PDUS





# MESSAGE & PDU STRUCTURE

*variable bindings:*

NAME 1	VALUE 1	NAME 2	VALUE 2	...	NAME <i>n</i>	VALUE <i>n</i>
--------	---------	--------	---------	-----	---------------	----------------

**SNMP PDU:**

PDU TYPE *	REQUEST ID	ERROR STATUS	ERROR INDEX	VARIABLE BINDINGS
------------	------------	--------------	-------------	-------------------

**SNMP message:**

VERSION	COMMUNITY	SNMP PDU
---------	-----------	----------



# SNMPv2

## OVERVIEW:

RFCs

LIMITATIONS OF SNMPv1

HISTORY OF SNMPv2

- HIERARCHIES
- SECURITY

SNMPv2 PROTOCOL OPERATIONS



## SNMPv2 RFCs

### COMMUNICATION MODEL

- DRAFT STANDARD
- RFC 3416, RFC3417

### SECURITY MODEL - SNMPv2C:

- COMMUNITY BASED SNMP
- SAME 'SECURITY MECHANISMS' AS SNMPv1
  - HISTORIC
  - RFC 1901

### SECURITY MODEL - SNMPv2U:

- USER BASED SECURITY (AUTHENTICATION / ENCRYPTION / ACCESS CONTROL)
  - HISTORIC
  - RFC 1909, RFC1910

### INFORMATION MODEL:

- STANDARD
- RFC2578, RFC2579, RFC2580

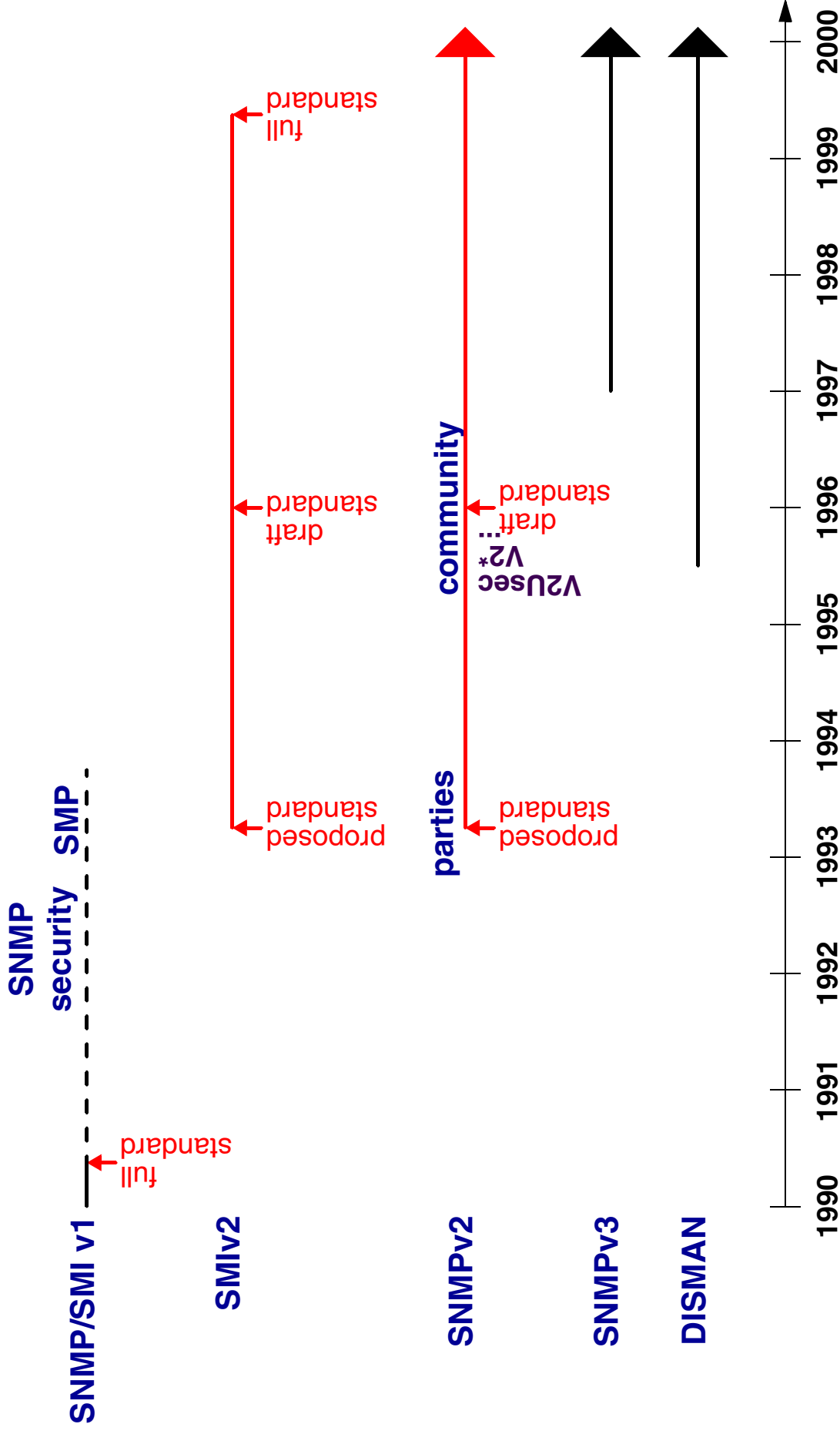


## **LIMITATIONS OF SNMPv1**

- LIMITED ERROR CODES
- LIMITED NOTIFICATIONS
- LIMITED PERFORMANCE
- TRANSPORT DEPENDENCE
- LACK OF HIERARCHIES
  - LACK OF SECURITY
- UNDOCUMENTED RULES (SMIPv1)
- LIMITED DATA TYPES (SMIPv1)



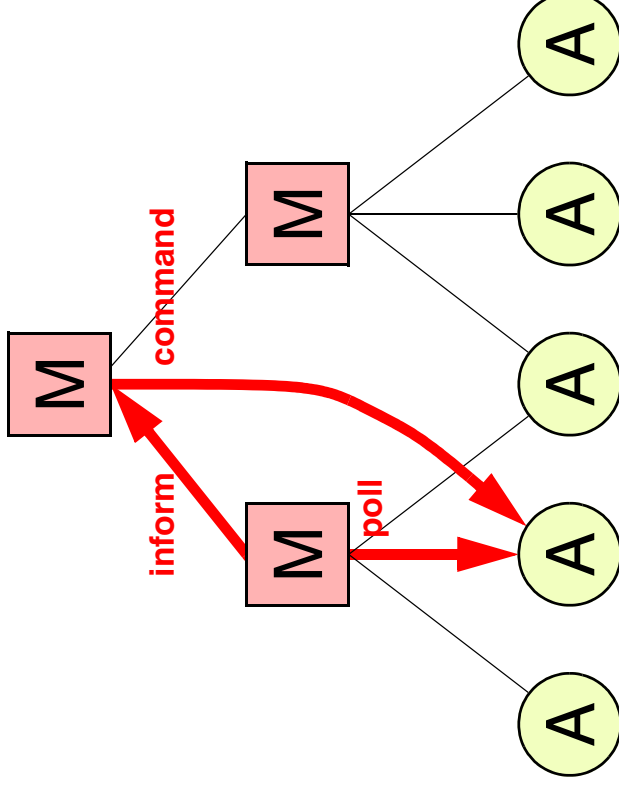
# HISTORY OF SNMPv2





## HIERARCHIES: ORIGINAL IDEA

### MANAGER TO MANAGER (M2M) MIB



- STANDARD MIB APPROACH
- LIMITED FUNCTIONALITY
- RUN-TIME BEHAVIOUR MUST BE DEFINED AT IMPLEMENTATION TIME



## **HIERARCHIES: STATUS**

WORK HAS MOVED TO A SEPARATE  
DISTRIBUTED MANAGEMENT GROUP  
(DISMAN)

THREE APPROACHES ARE STANDARDIZED:

- MIB BASED (EXPRESSION, EVENT AND NOTIFICATION LOG MIB)
- SCRIPT BASED (SCRIPT AND SCHEDULE MIB)
- REMOTE OPERATIONS BASED (REMOPS MIB)



## SNMPv2 SECURITY: WHAT HAPPENED?

**APRIL 1993:**

PROPOSED STANDARD  
FOUR EDITORS  
SECURITY BASED ON *PARTIES*  
FIRST PROTOTYPES APPEARED SOON

**JUNE 1995:**

*PROPOSED STANDARD REJECTED BY TWO OF THE ORIGINAL EDITORS!*

**AUGUST 1995:**

GENERAL AGREEMENT THAT PARTY BASED MODEL WAS TOO COMPLEX!  
MANY NEW PROPOSALS APPEARED:

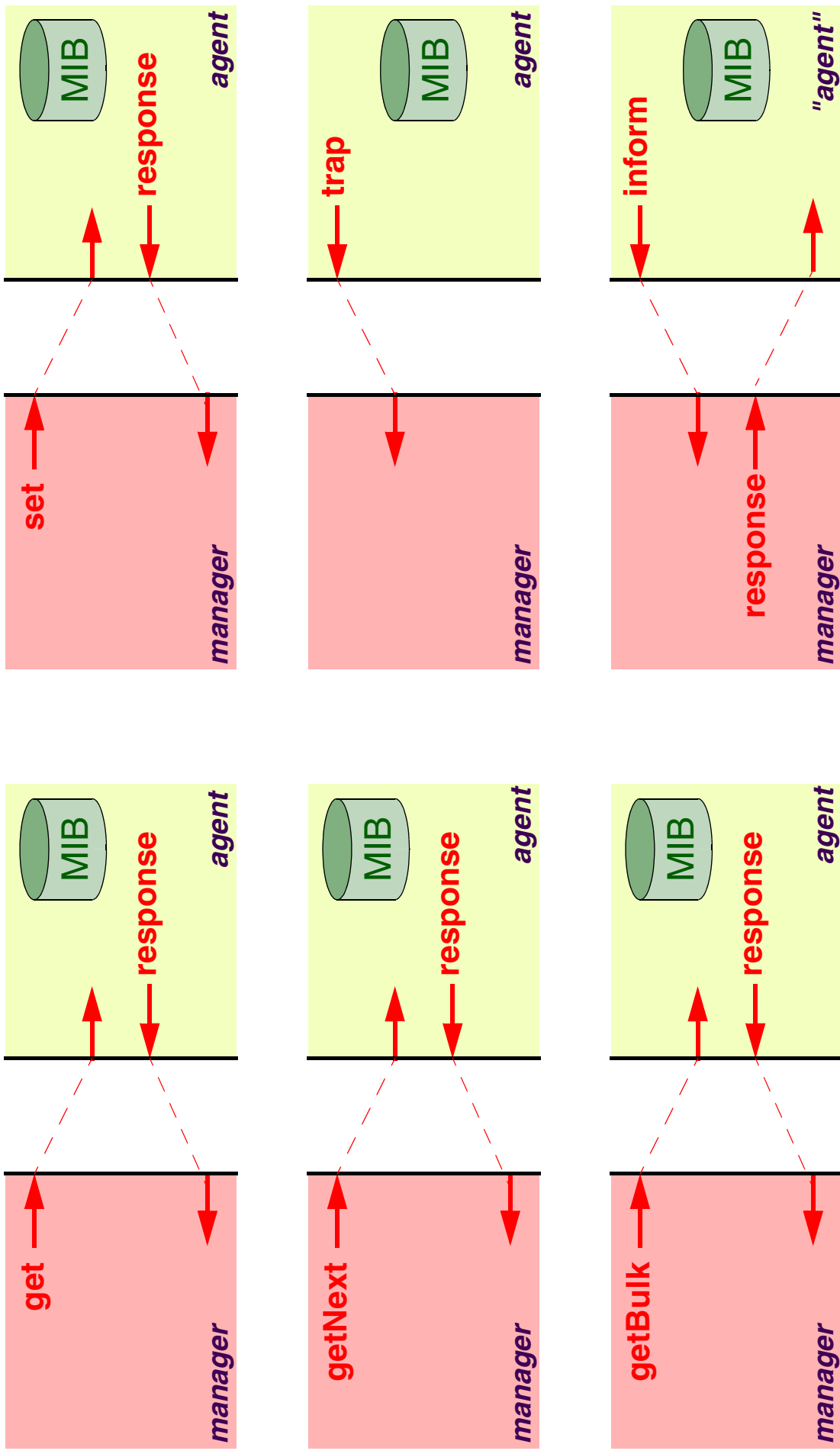
- SNMPv2C: COMMUNITY BASED
- SNMPv2U: USER BASED
- ...

**1997:**

NEW SNMPv3 WORKING GROUP WAS FORMED  
WITH NEW EDITORS

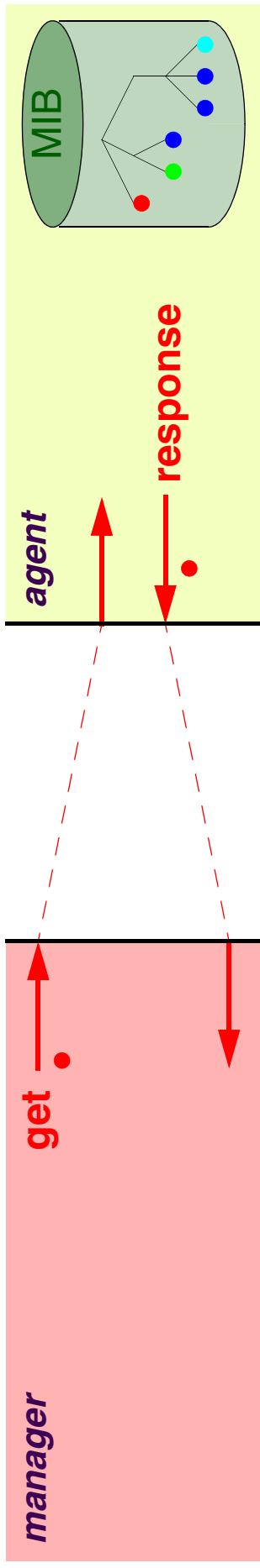


# SNMPv2 PROTOCOL OPERATIONS





# GET



SIMILAR TO SNMPv1, EXCEPT FOR "EXCEPTIONS"

POSSIBLE EXCEPTIONS:

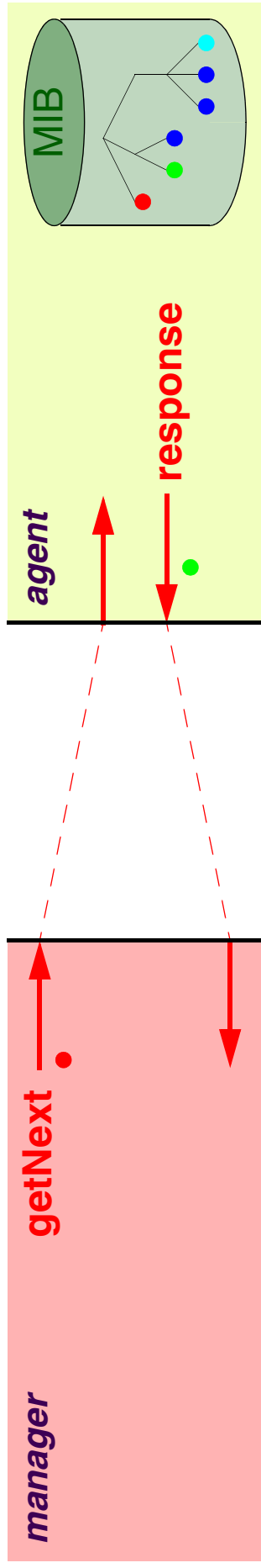
- noSuchObject
- noSuchInstance

EXCEPTIONS ARE CODED WITHIN THE VARBINDS

EXCEPTIONS DO NOT RAISE ERROR STATUS AND INDEX



# GET-NEXT



SIMILAR TO SNMPv1, EXCEPT FOR "EXCEPTIONS"

POSSIBLE EXCEPTIONS:

- `endOfMibView`

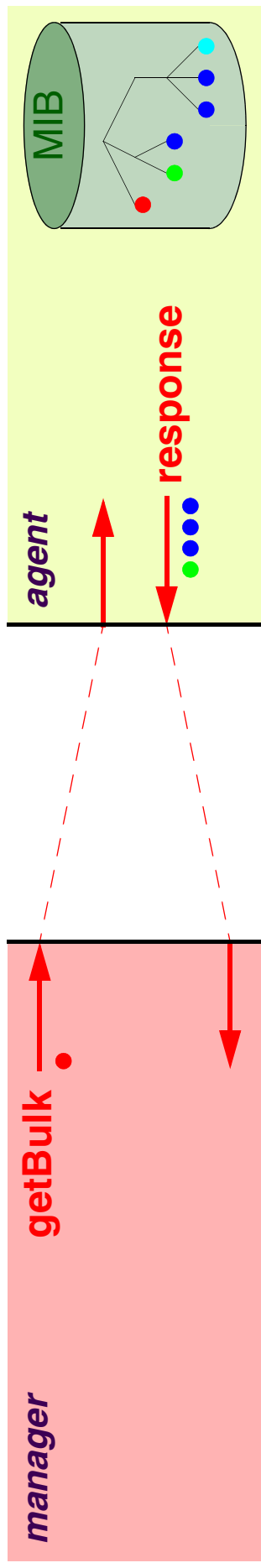
## EXAMPLE

`getNext(7.4.0)`

`response(error-status => noError, 7.4.0 => endOfMibView)`



# GET-BULK



NEW IN SNMPv2

TO RETRIEVE A LARGE NUMBER OF VARBINDS

IMPROVES PERFORMANCE!



## GET-BULK

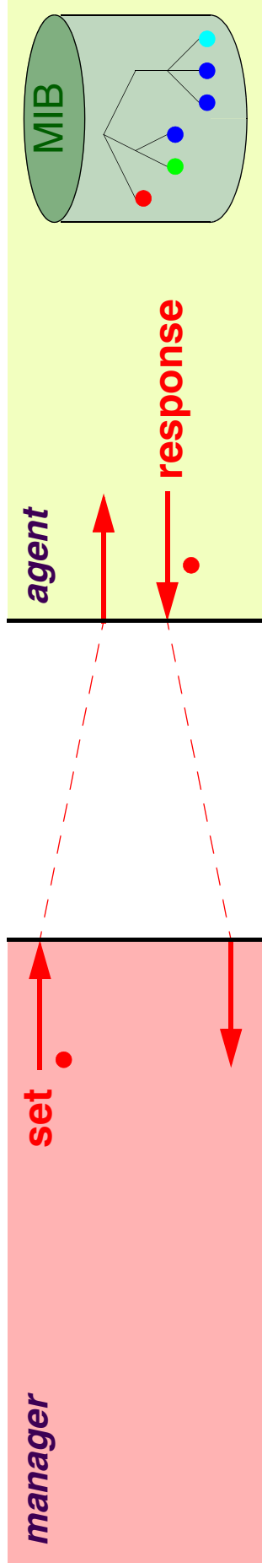
getBulk REQUEST HAS TWO ADDITIONAL PARAMETERS:

- non-repeators
- max-repetitions

- THE FIRST N ELEMENTS (non-repeators) OF THE VARBIND LIST ARE TREATED AS IF THE OPERATION WAS A NORMAL `getnext` OPERATION
- THE NEXT ELEMENTS OF THE VARBIND LIST ARE TREATED AS IF THE OPERATION CONSISTED OF A NUMBER (max-repetitions) OF REPEATED `getnext` OPERATIONS



# SET



SIMILAR TO SNMPv1

CONCEPTUAL TWO PHASE COMMIT:

- PHASE 1: PERFORM VARIOUS CHECKS
- PHASE 2: PERFORM THE ACTUAL SET

MANY NEW ERROR CODES ARE DEFINED



## **SET: NEW ERROR CODES**

### **SNMPv1**

#### **PHASE 1:**

badValue  
badValue  
badValue  
badValue  
badValue  
noSuchName  
noSuchName  
noSuchName  
noSuchName

genErr  
genErr

#### **PHASE 2:**

genErr  
genErr

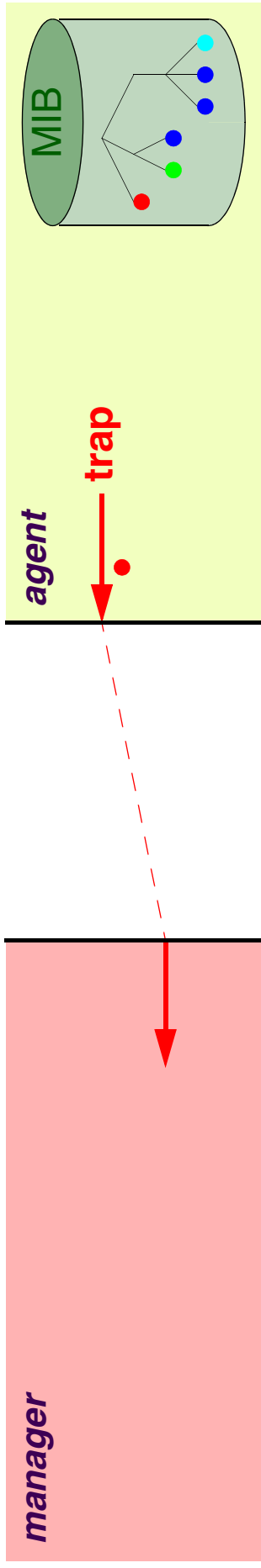
### **SNMPv2**

wrongValue  
wrongEncoding  
wrongType  
wrongLength  
inconsistentValue  
noAccess  
notWritable  
noCreation  
inconsistentName  
resourceUnavailable  
genErr

CommitFailed  
undoFailed



# TRAP



## SNMPv1:

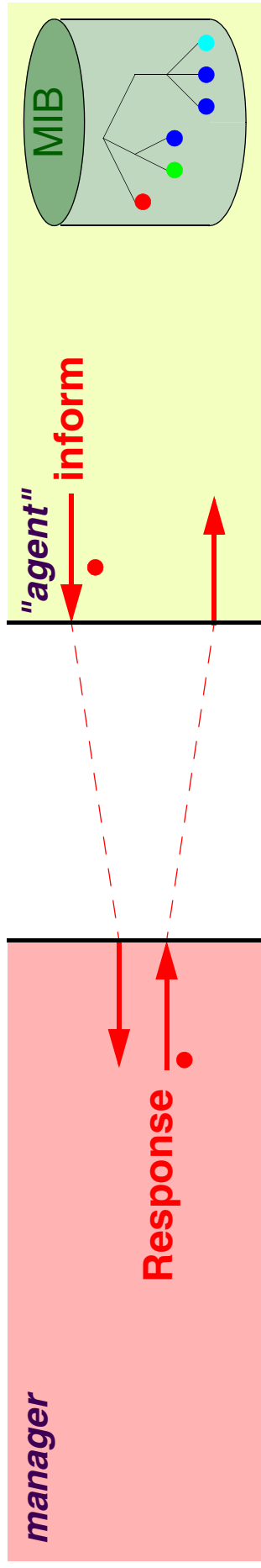
- COLD START
- WARM START
- LINK DOWN
  - LINK UP
- AUTHENTICATION FAILURE
- EGP NEIGHBOR LOSS

## SNMPv2:

- MIBs MAY NOW INCLUDE NOTIFICATION TYPE MACROS
- FIRST TWO VARBINDS: `sysUptime` AND `snmpTrapOID`
  - USES SAME FORMAT AS OTHER PDUS



# INFORM



## CONFIRMED TRAP

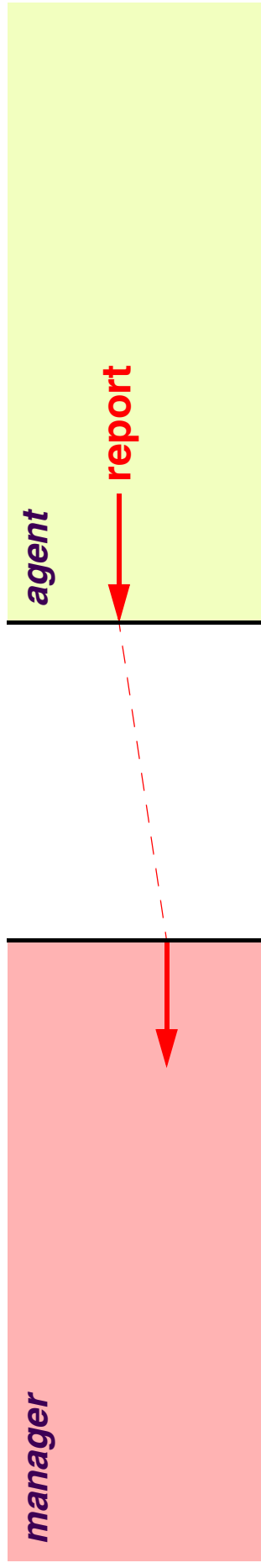
ORIGINALLY TO INFORM A HIGHER LEVEL MANAGER

SAME FORMAT AS TRAP PDU

POSSIBLE ERROR: `tooBig`



# REPORT



NEW PDU TO SIGNAL PROTOCOL EXCEPTIONS / ERRORS  
NO SEMANTICS DEFINED IN SNMPv2



# **SNMPv3**

## **OVERVIEW:**

DESIGN DECISIONS

ARCHITECTURE

SNMP MESSAGE STRUCTURE

SECURE COMMUNICATION

- USER SECURITY MODEL (USM)

ACCESS CONTROL

- VIEW BASED ACCESS CONTROL MODEL (VACM)

RFCs



## **DESIGN DECISIONS**

ADDRESS THE NEED FOR SECURITY SET SUPPORT

DEFINE AN ARCHITECTURE THAT ALLOWS FOR LONGEVITY OF SNMP

ALLOW THAT DIFFERENT PORTIONS OF THE ARCHITECTURE  
MOVE AT DIFFERENT SPEEDS TOWARDS STANDARD STATUS

ALLOW FOR FUTURE EXTENSIONS

KEEP SNMP AS SIMPLE AS POSSIBLE

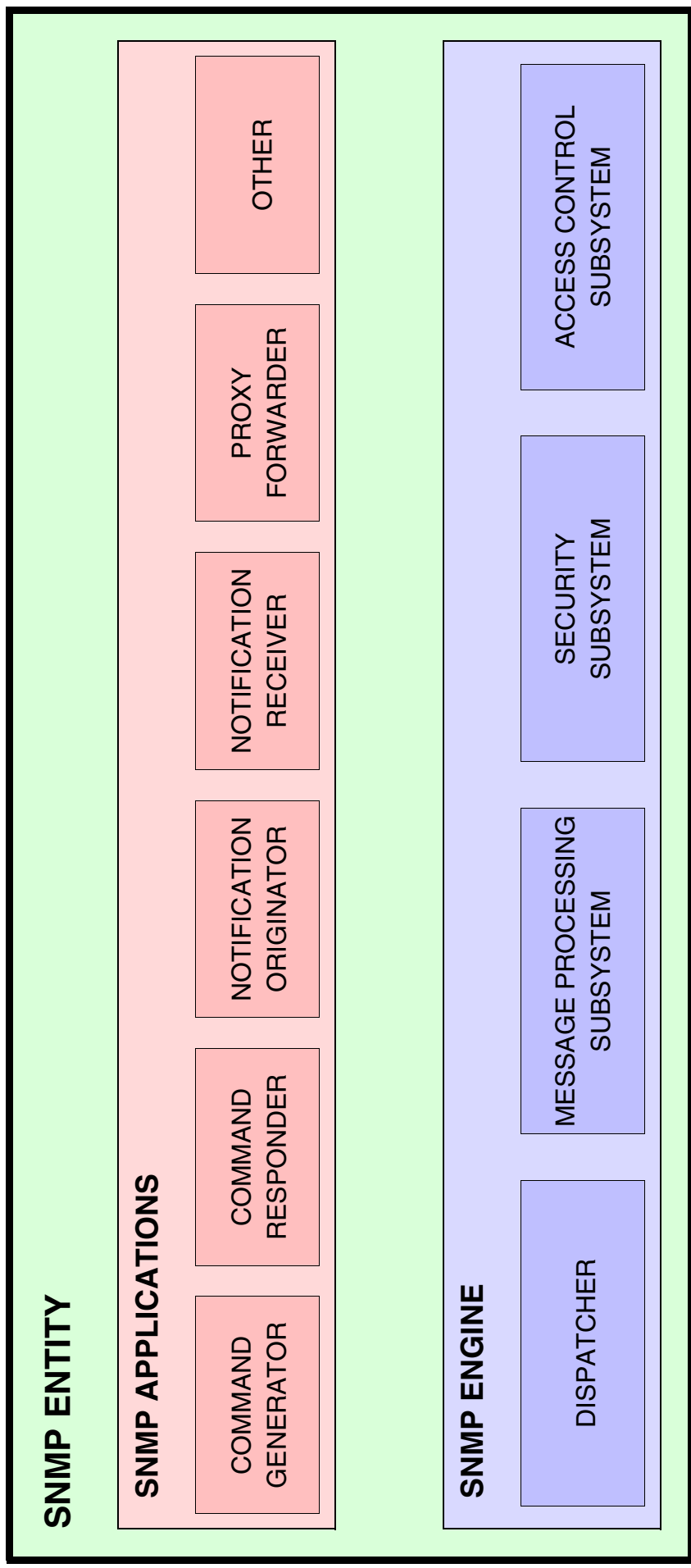
ALLOW FOR MINIMAL IMPLEMENTATIONS

SUPPORT ALSO THE MORE COMPLEX FEATURES,  
WHICH ARE REQUIRED IN LARGE NETWORKS

RE-USE EXISTING SPECIFICATIONS, WHENEVER POSSIBLE

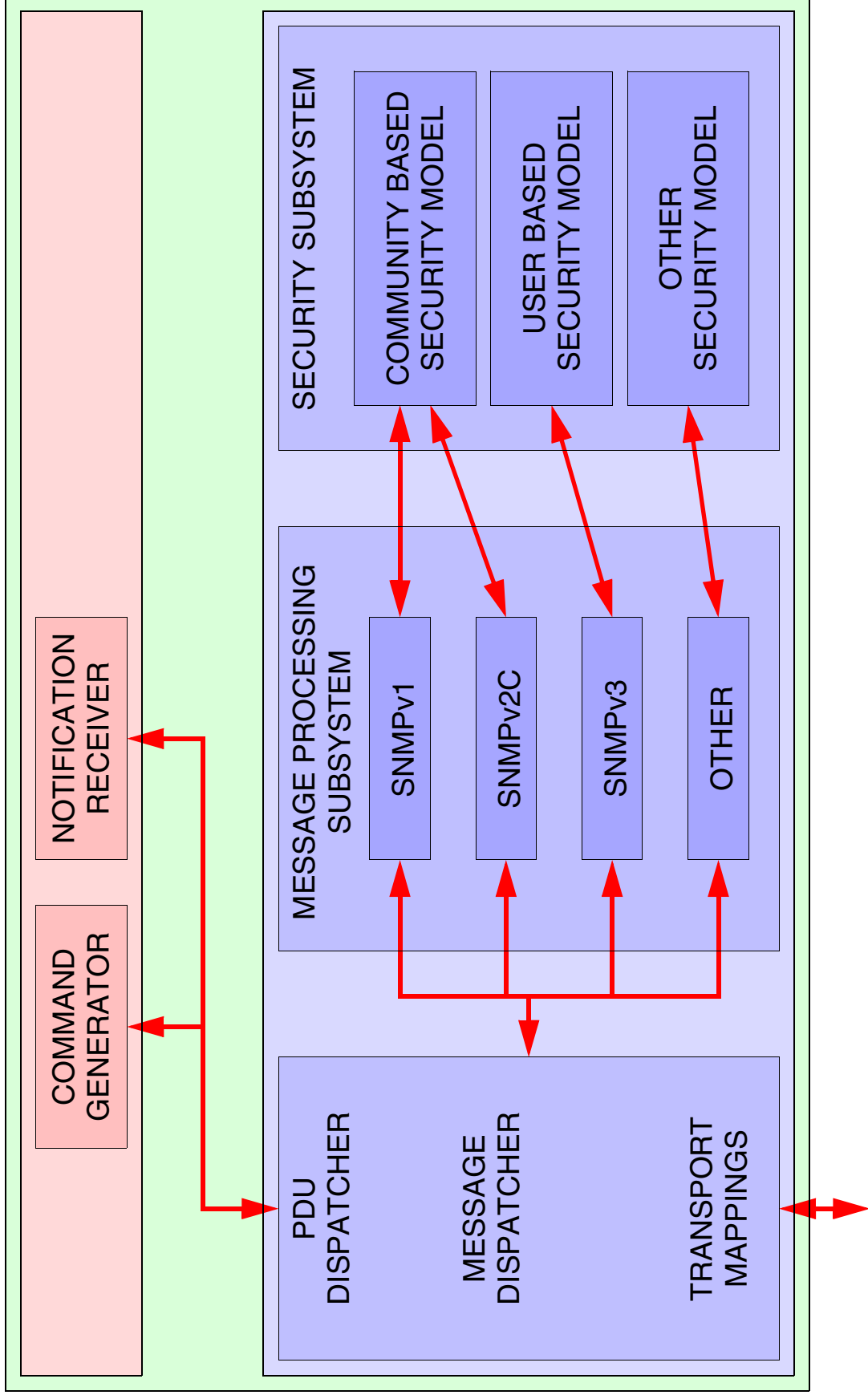


# SNMPv3 ARCHITECTURE



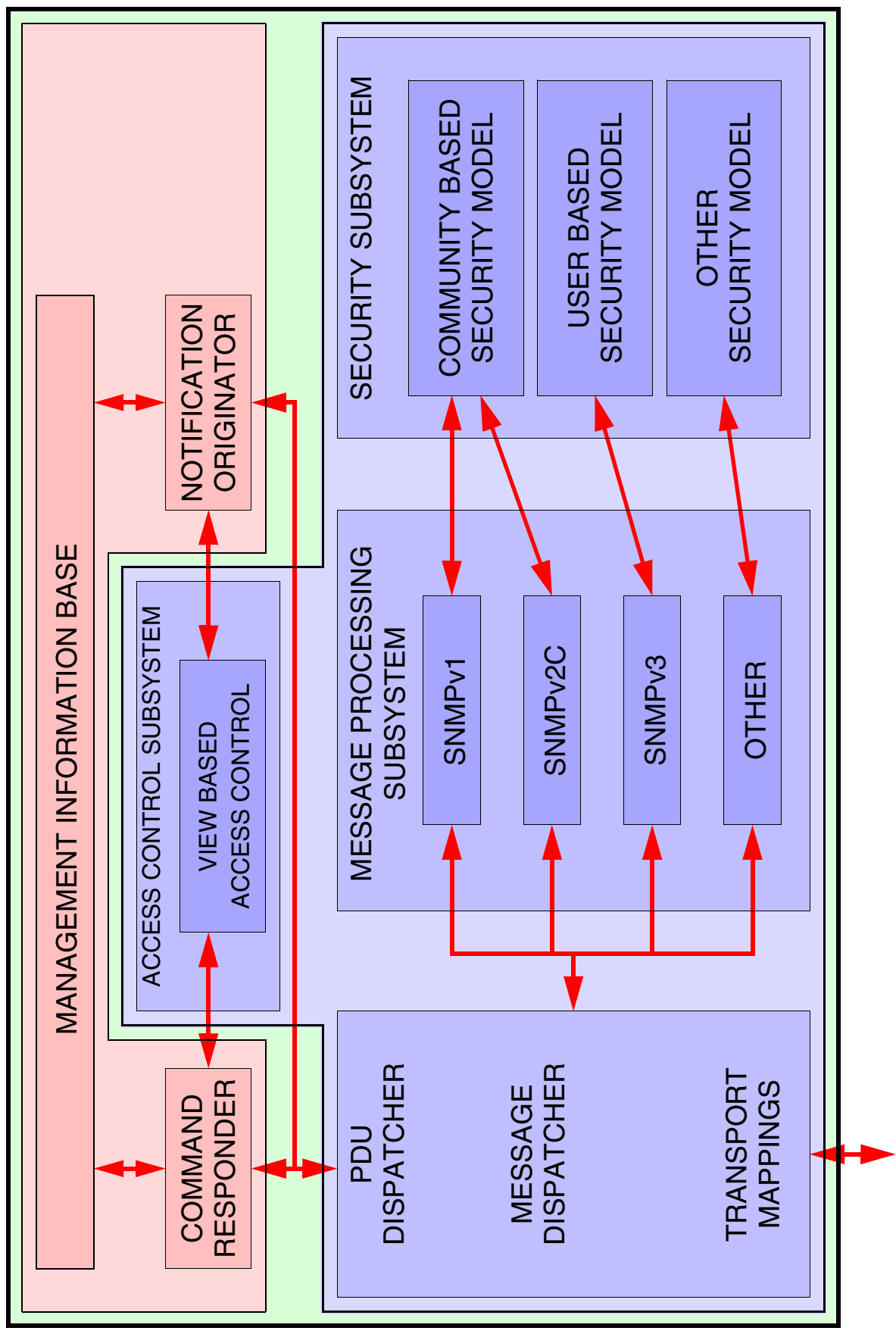


# SNMPv3 ARCHITECTURE: MANAGER



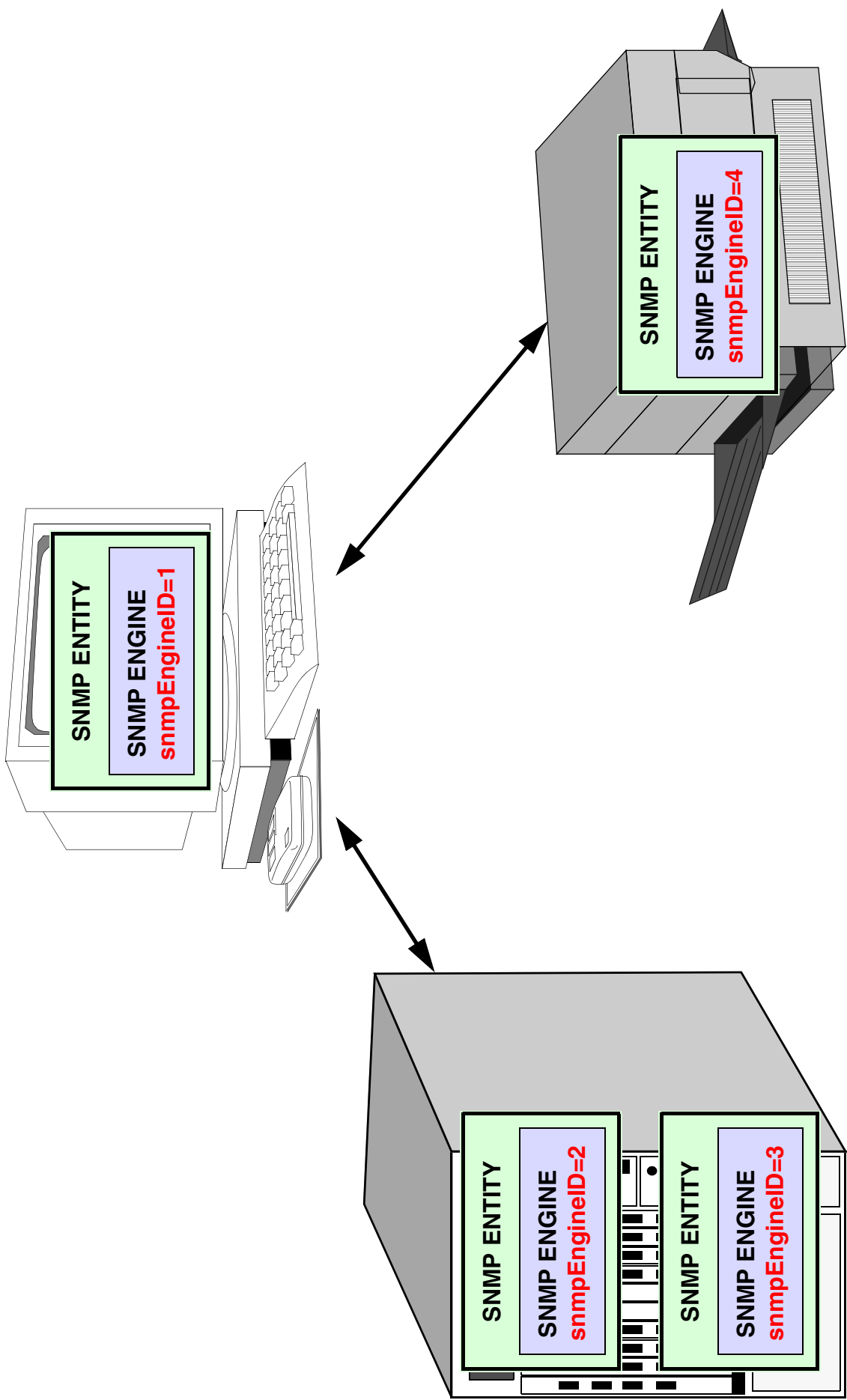


# SNMPv3 ARCHITECTURE: AGENT





# CONCEPTS: snmpEngineID





## **CONCEPTS: snmpEngineID**

SYNTAX DEFINED VIA TEXTUAL CONVENTION

OCTET STRING (5..32)

THE VALUE OF snmpEngineID MAY BE DETERMINED BY:

- HUMAN OPERATOR
- AUTOMATIC ALGORITHM

**AUTOMATIC ALGORITHM USES:**

- PRIVATE ENTERPRISE NUMBER
- IPV4 ADDRESS / IPV6 ADDRESS / MAC ADDRESS

TEXTUAL CONVENTION DEFINED IN SNMP FRAMEWORK MIB



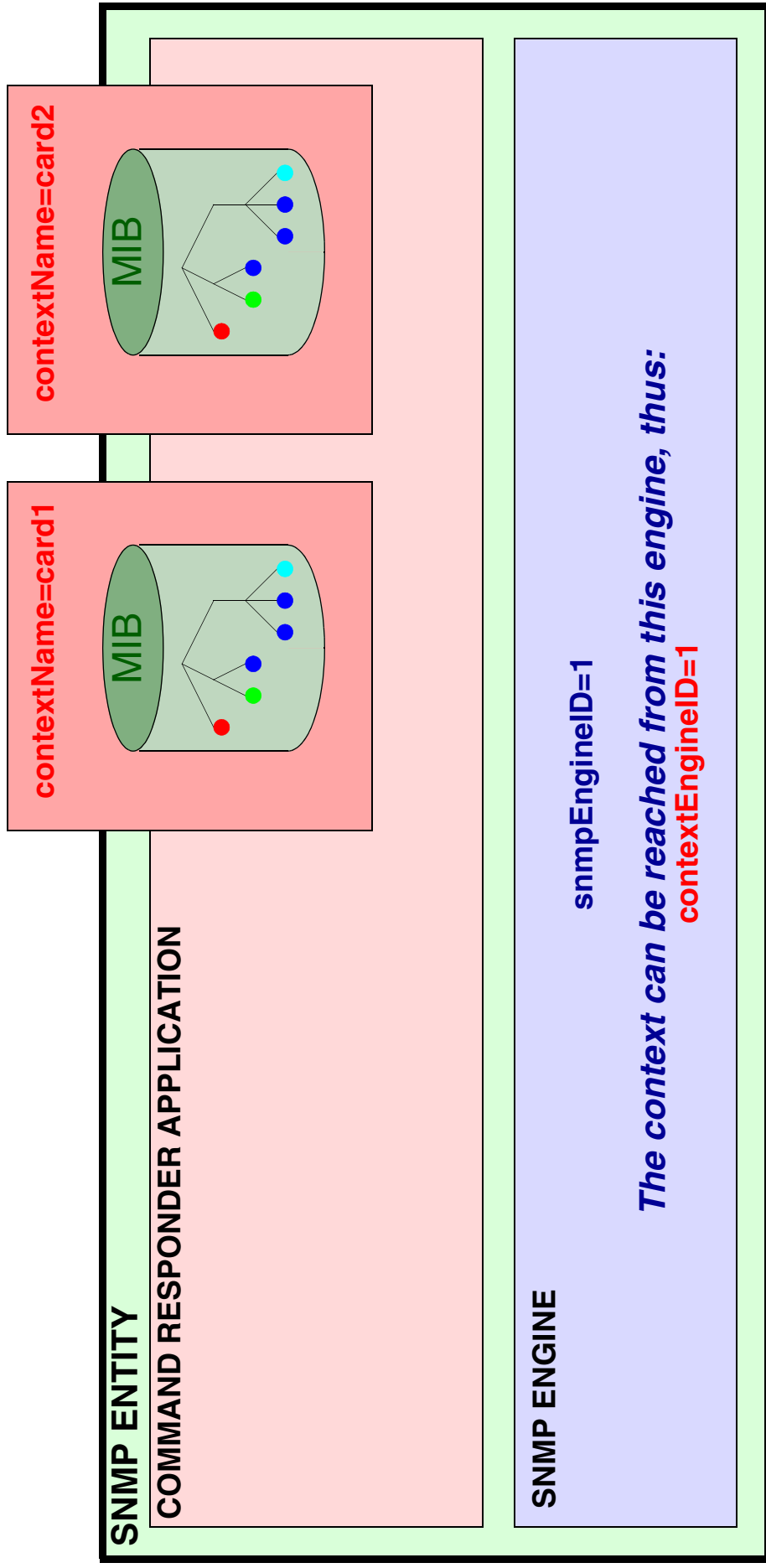
## CONCEPTS: snmpEngineID

### THE TERM EngineID IS FREQUENTLY USED

SnmpEngineID	The textual convention.
snmpEngineID	The identifier of an SNMP engine.
securityEngineID	Parameter of primitives in the architecture. The <i>authoritative</i> SNMP entity (which is the receiver of a confirmed PDU, the sender of a trap).
contextEngineID	Parameter in messages. Identifies the engine associated with the data.
msgAuthoritativeEngineID	Parameter in messages. USM security parameter.
usmUserEngineID	An object in the snmpUsmMIB. In a simple agent, this is the agent's own snmpEngineID. It may also be the snmpEngineID of a remote SNMP engine with which this user can communicate.
usmStatsUnknownEngineID	An object in the snmpUsmMIB.
snmpCommunityContextEngineID	An object in the communityMIB.
entLogicalContextEngineID	An object in the entityMIB.
snmpProxyContextEngineID	An object in the proxyMIB.



# CONCEPTS: Context





# MODULES OF THE SNMPv3 ARCHITECTURE

## DISPATCHER AND MESSAGE PROCESSING MODULE

- SNMPv3 MESSAGE STRUCTURE
- snmpMPDMIB
- RFC 3412

## APPLICATIONS

- snmpTargetMIB
- snmpNotificationMIB
- snmpProxyMIB
- RFC 3413

## SECURITY SUBSYSTEM

- USER BASED SECURITY MODEL
- snmpUsmMIB
- RFC 3414

## ACCESS CONTROL SUBSYSTEM

- VIEW BASED ACCESS CONTROL MODEL
- snmpVacmMIB
- RFC 3415



## SNMPv3 MESSAGE STRUCTURE

msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgSecurityParameters
contextEngineID
contextName
PDU

USED BY MESSAGE PROCESSING SUBSYSTEM

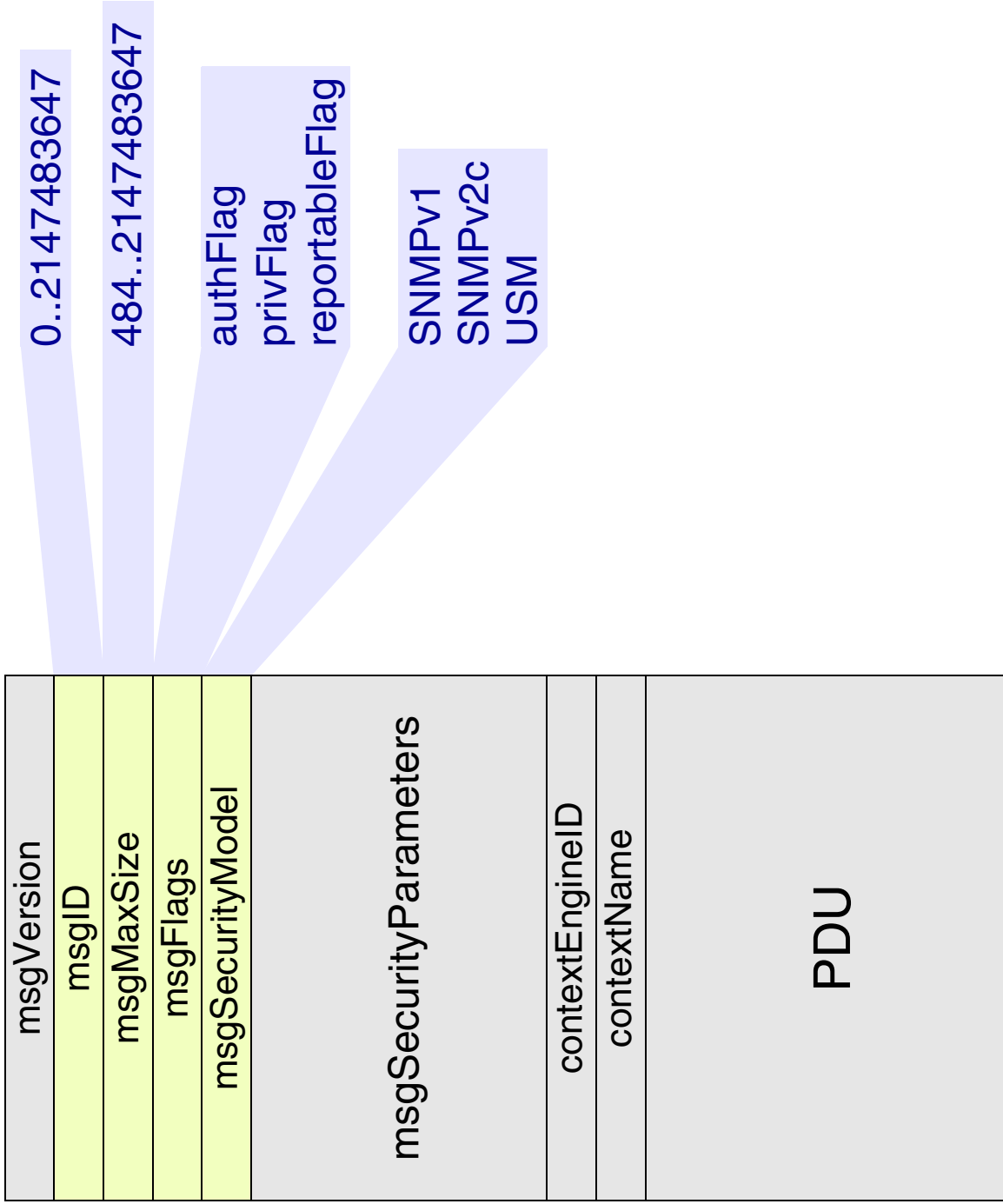
USED BY SNMPv3 PROCESSING MODULE

USED BY SECURITY SUBSYSTEM

USED BY ACCESS CONTROL SUBSYSTEM  
AND APPLICATIONS

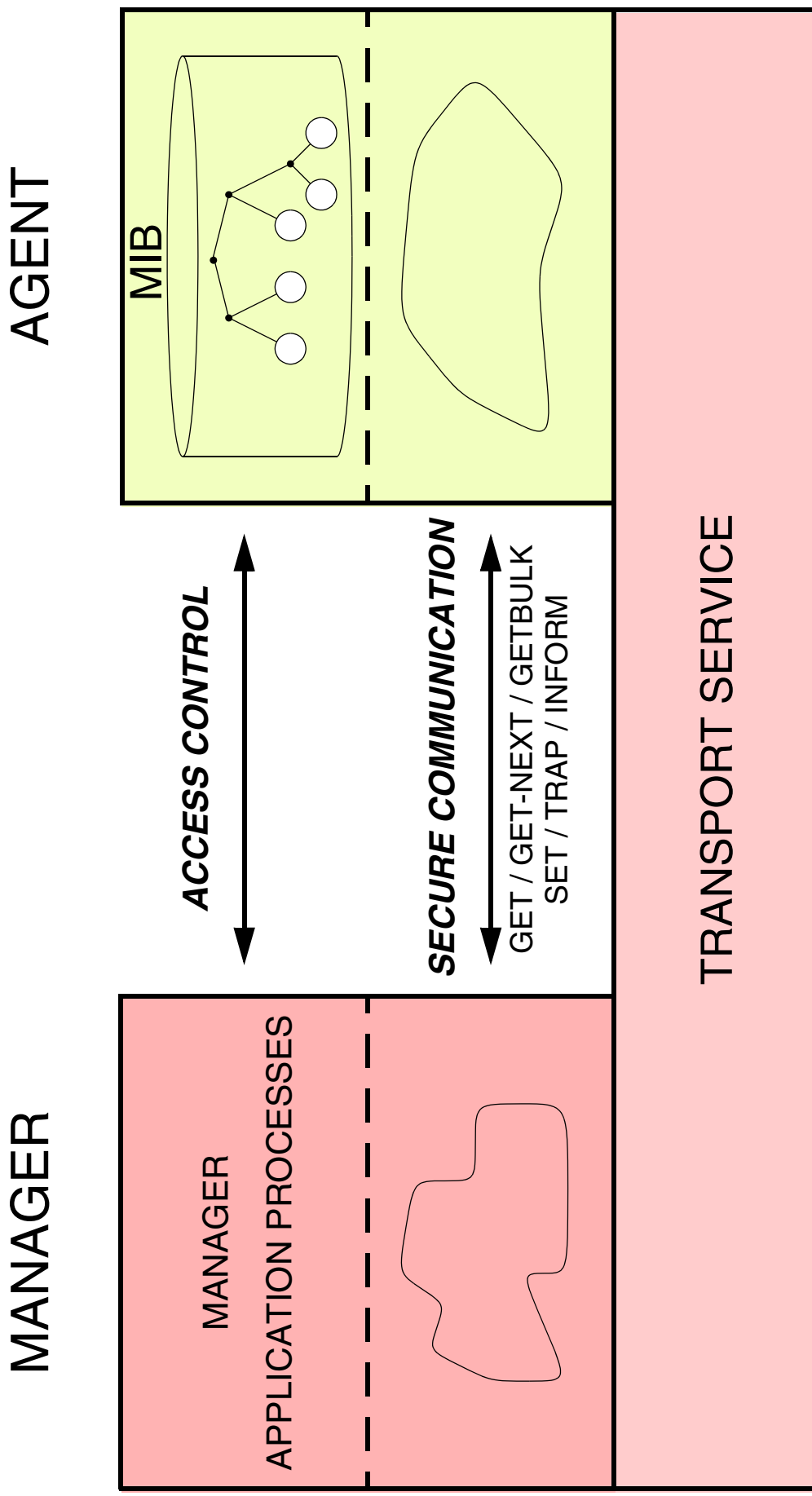


# SNMPv3 PROCESSING MODULE PARAMETERS





# SECURE COMMUNICATION VERSUS ACCESS CONTROL





## **USM: SECURITY THREATS**

<b>THREAT</b>	<b>ADDRESSED?</b>	<b>MECHANISM</b>
REPLAY	YES	TIME STAMP
MASQUERADE	YES	MD5 / SHA-1
INTEGRITY	YES	(MD5 / SHA-1)
DISCLOSURE	YES	DES
DENIAL OF SERVICE	NO	
TRAFFIC ANALYSIS	NO	



# USM MESSAGE STRUCTURE

msgVersion
msgID
msgMaxSize
msgFlags
msgSecurityModel
msgAuthoritativeEngineID
msgAuthoritativeEngineBoots
msgAuthoritativeEngineTime
msgUserName
msgAuthenticationParameters
msgPrivacyParameters
contextEngineID
contextName
<b>PDU</b>

REPLAY

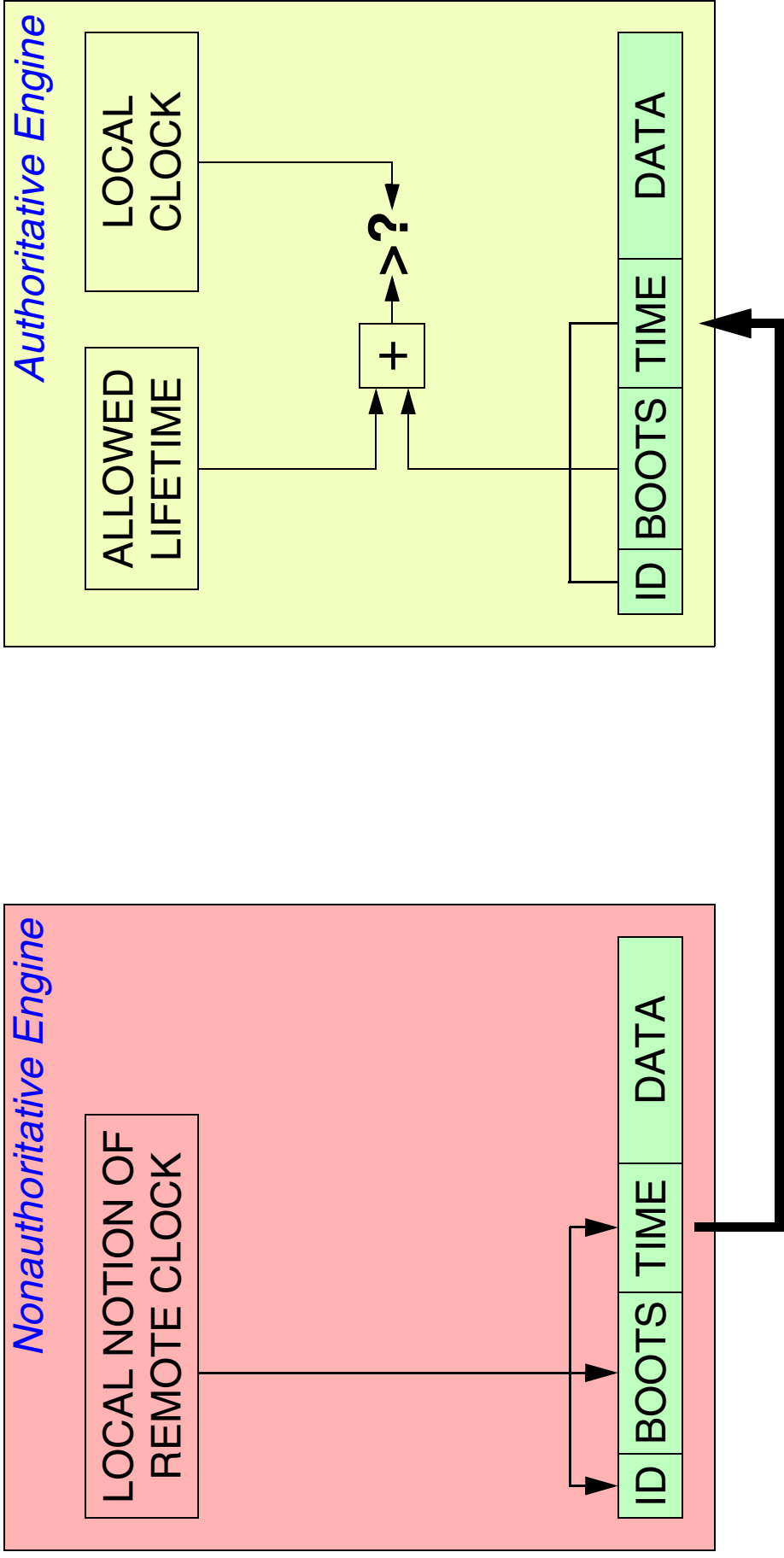
MASQUERADE/INTEGRITY/DISCLOSURE

MASQUERADE/INTEGRITY

DISCLOSURE

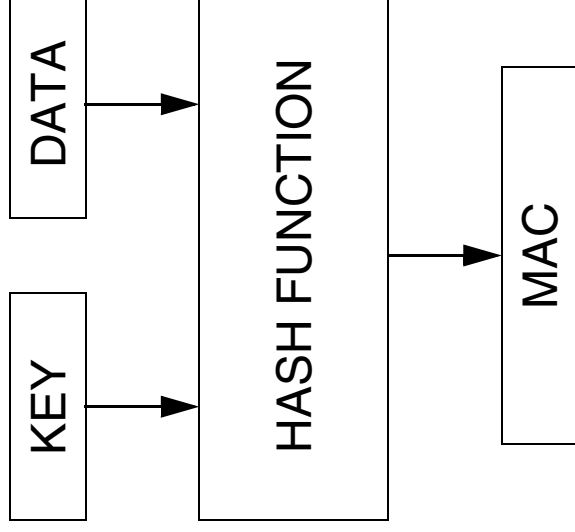


# IDEA BEHIND REPLAY PROTECTION





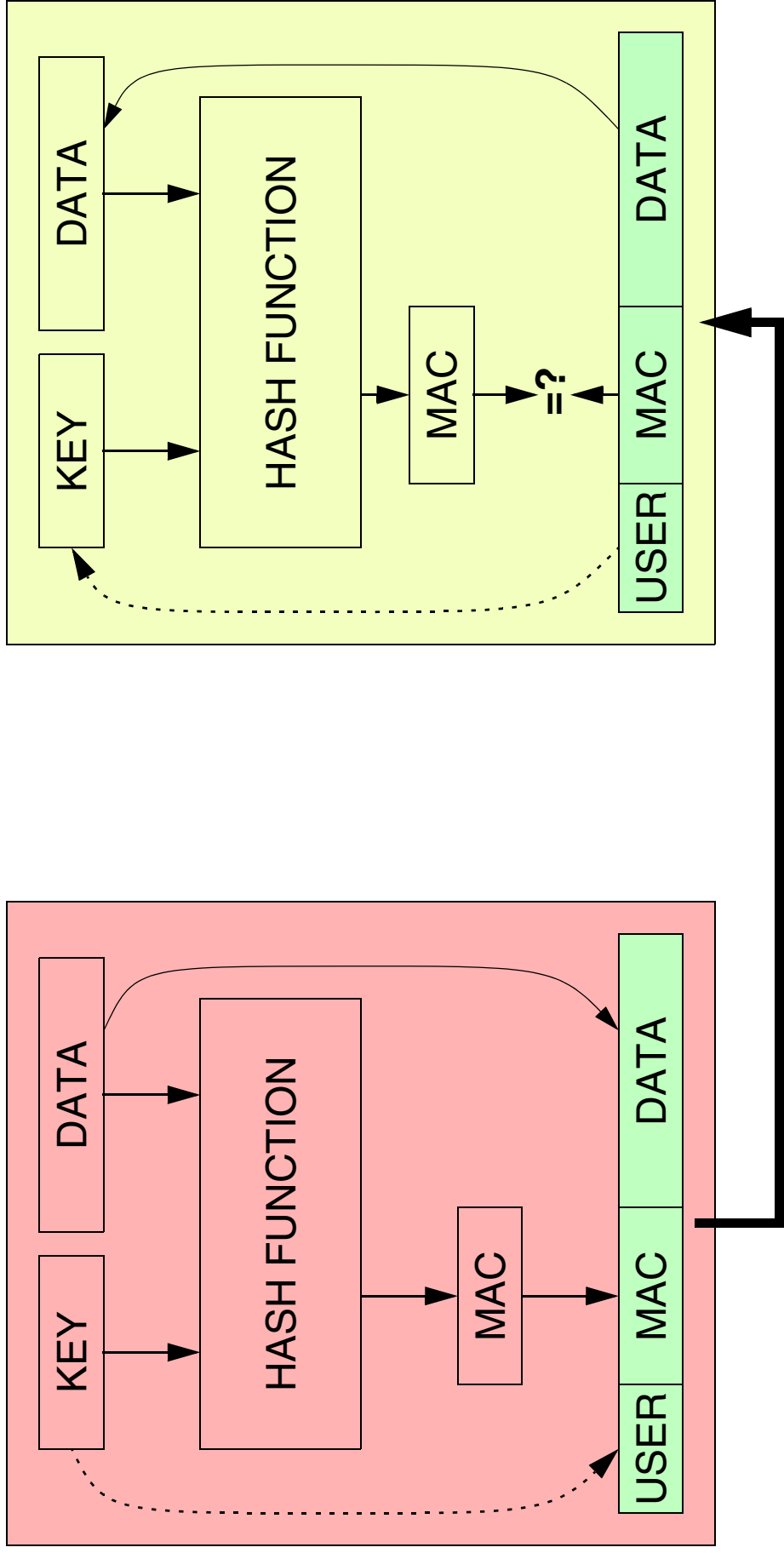
## IDEA BEHIND DATA INTEGRITY AND AUTHENTICATION



ADD THE MESSAGE AUTHENTICATION CODE (MAC) TO THE DATA  
AND SEND THE RESULT

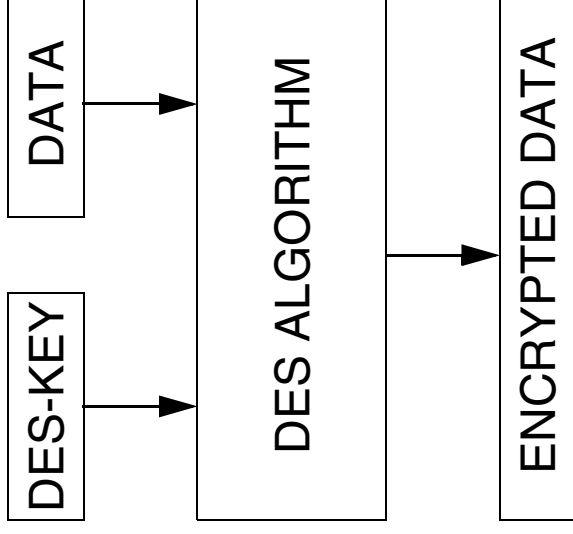


# IDEA BEHIND AUTHENTICATION



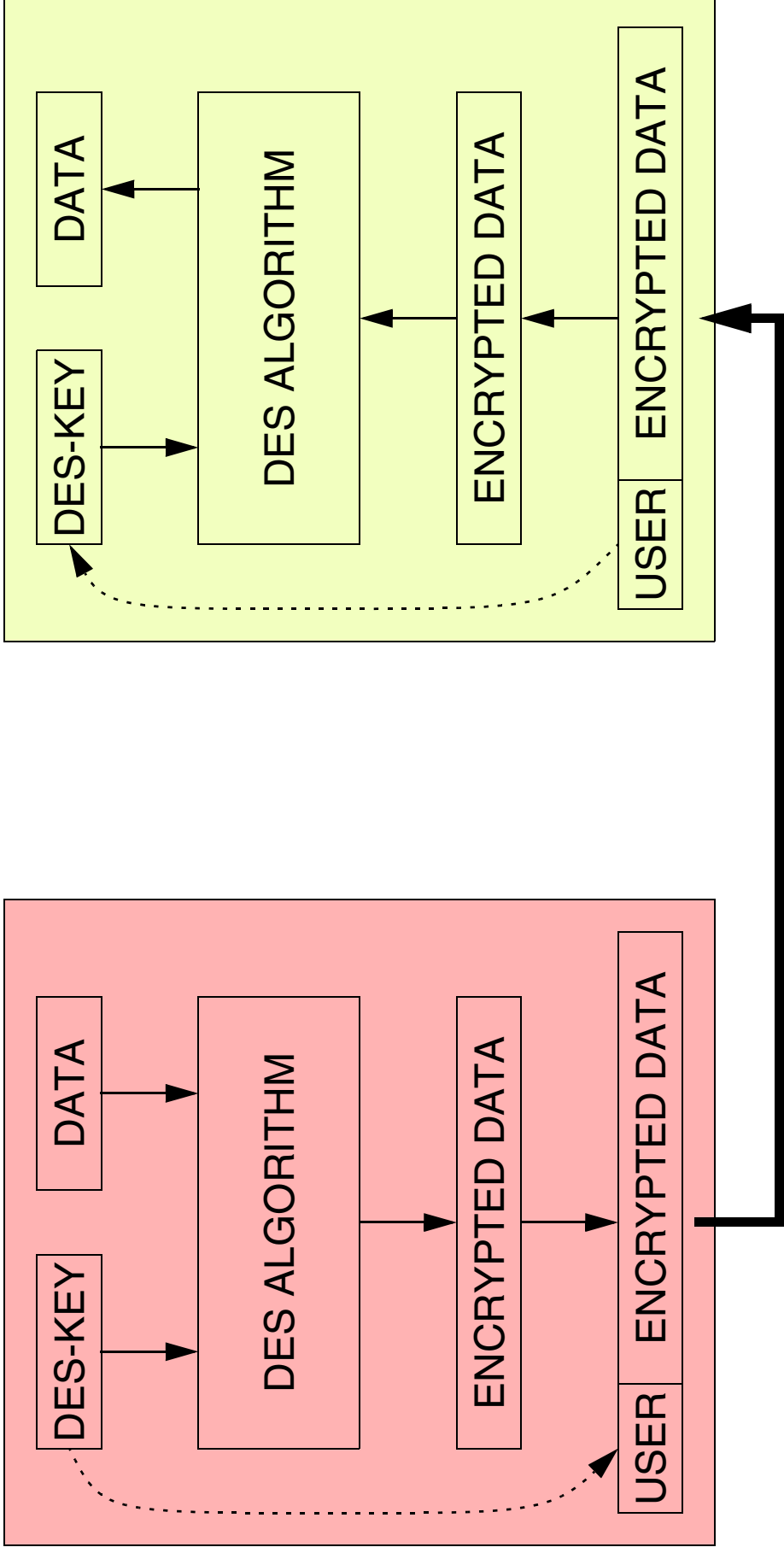


## IDEA BEHIND THE DATA CONFIDENTIALITY (DES)





# IDEA BEHIND ENCRYPTION





# **VIEW BASED ACCESS CONTROL MODEL**

ACCESS CONTROL TABLE

MIB VIEWS

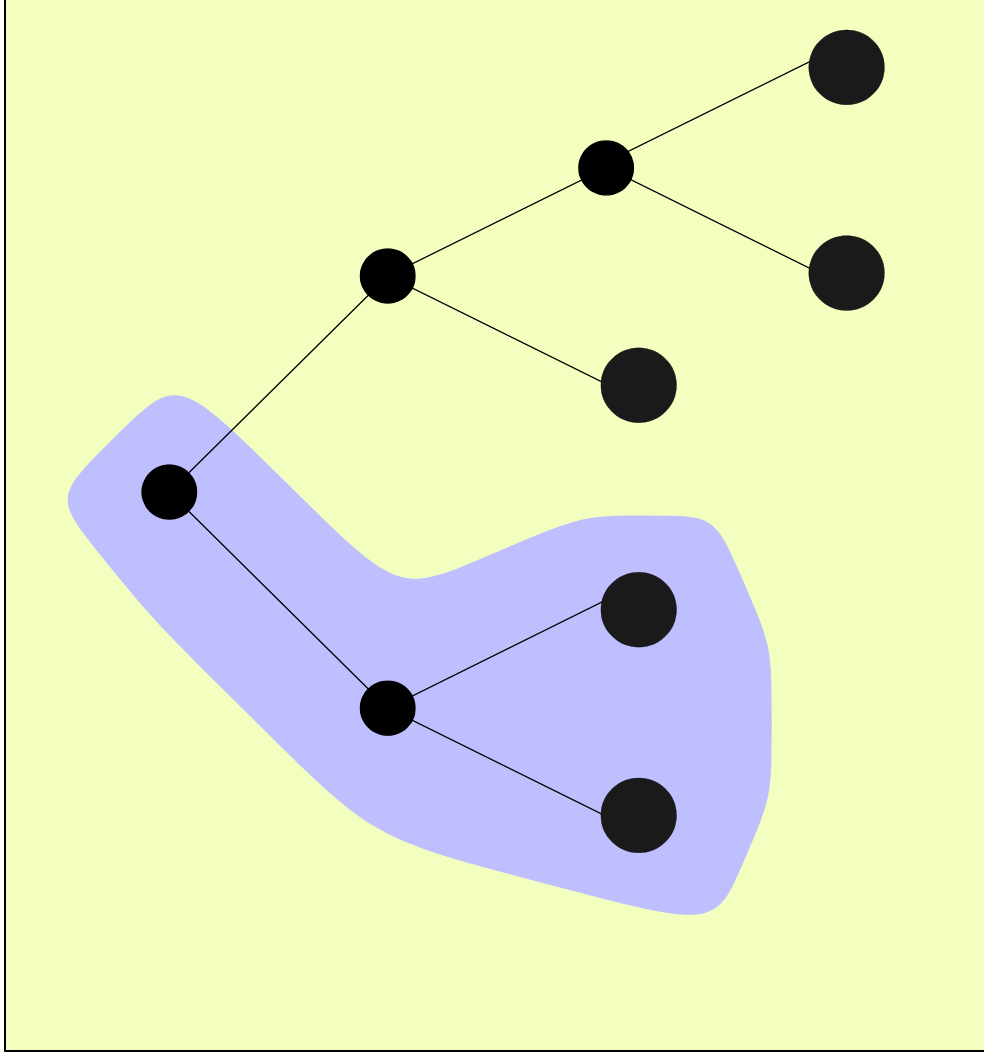


## ACCESS CONTROL TABLES

<b>MIB VIEW</b>	<b>ALLOWED OPERATIONS</b>	<b>ALLOWED MANAGERS</b>	<b>REQUIRED LEVEL OF SECURITY</b>
Interface Table	SET	John	Authentication Encryption
Interface Table	GET / GETNEXT	John, Paul	Authentication
Systems Group	GET / GETNEXT	George	None
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

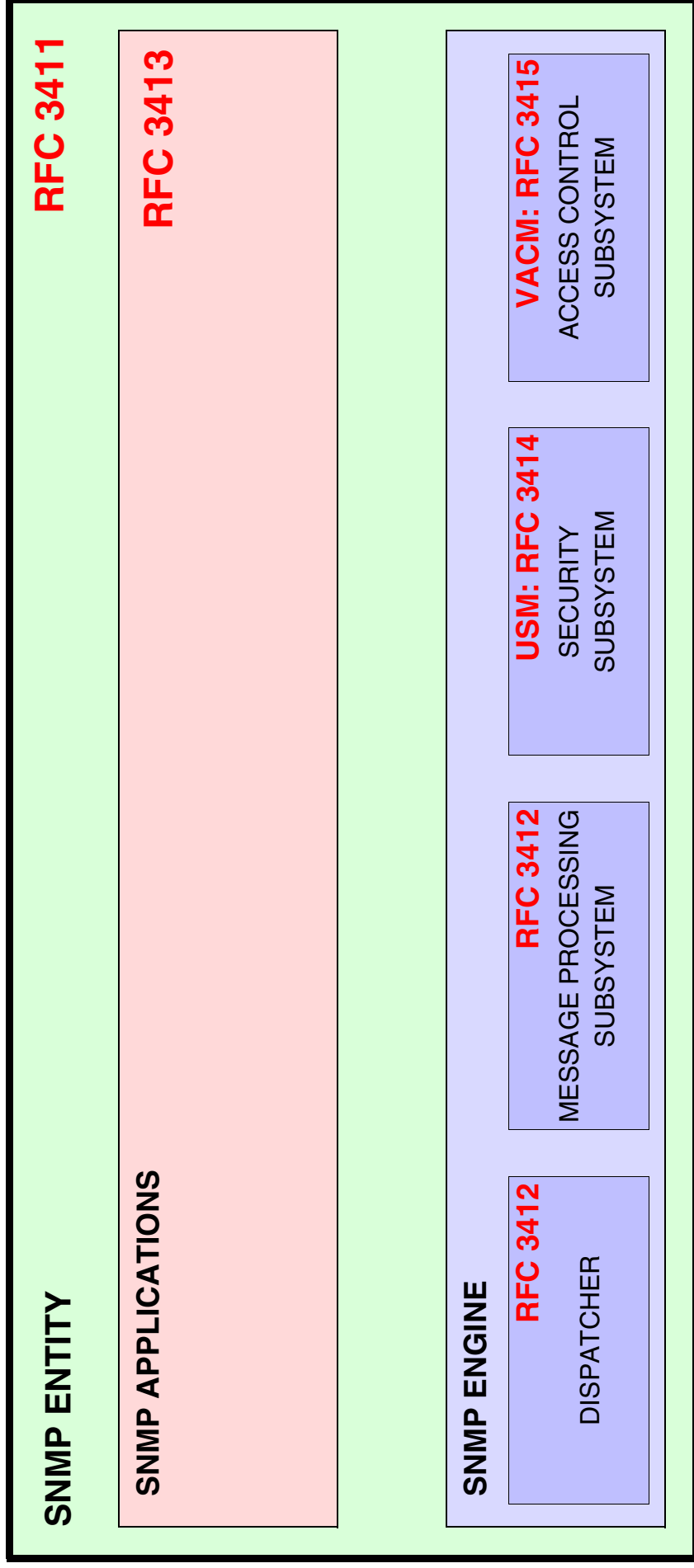


# MIB VIEWS





# SNMPv3 RFCs





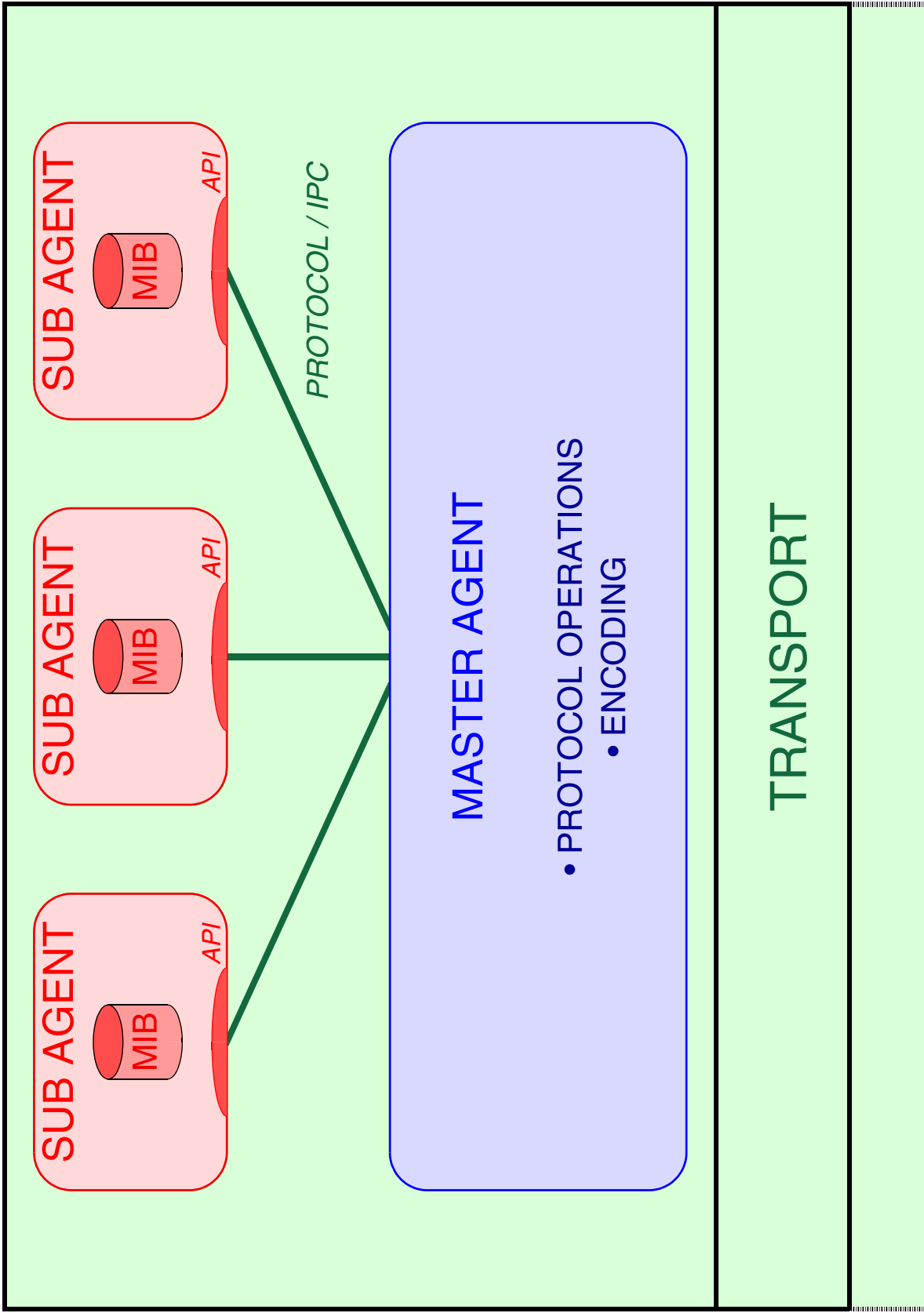
## **EXTENSIBLE AGENTS**

FACILITATE THE EXTENSION OF SNMP AGENTS  
WITH NEW MIB MODULES

- SEPARATE SNMP PROTOCOL ENGINE  
FROM MIB INSTRUMENTATION
  - ALLOW DYNAMIC ADDITION  
OF NEW MIB MODULE IMPLEMENTATIONS
- EXTENSIBLE AGENTS SHOULD BE TRANSPARENT

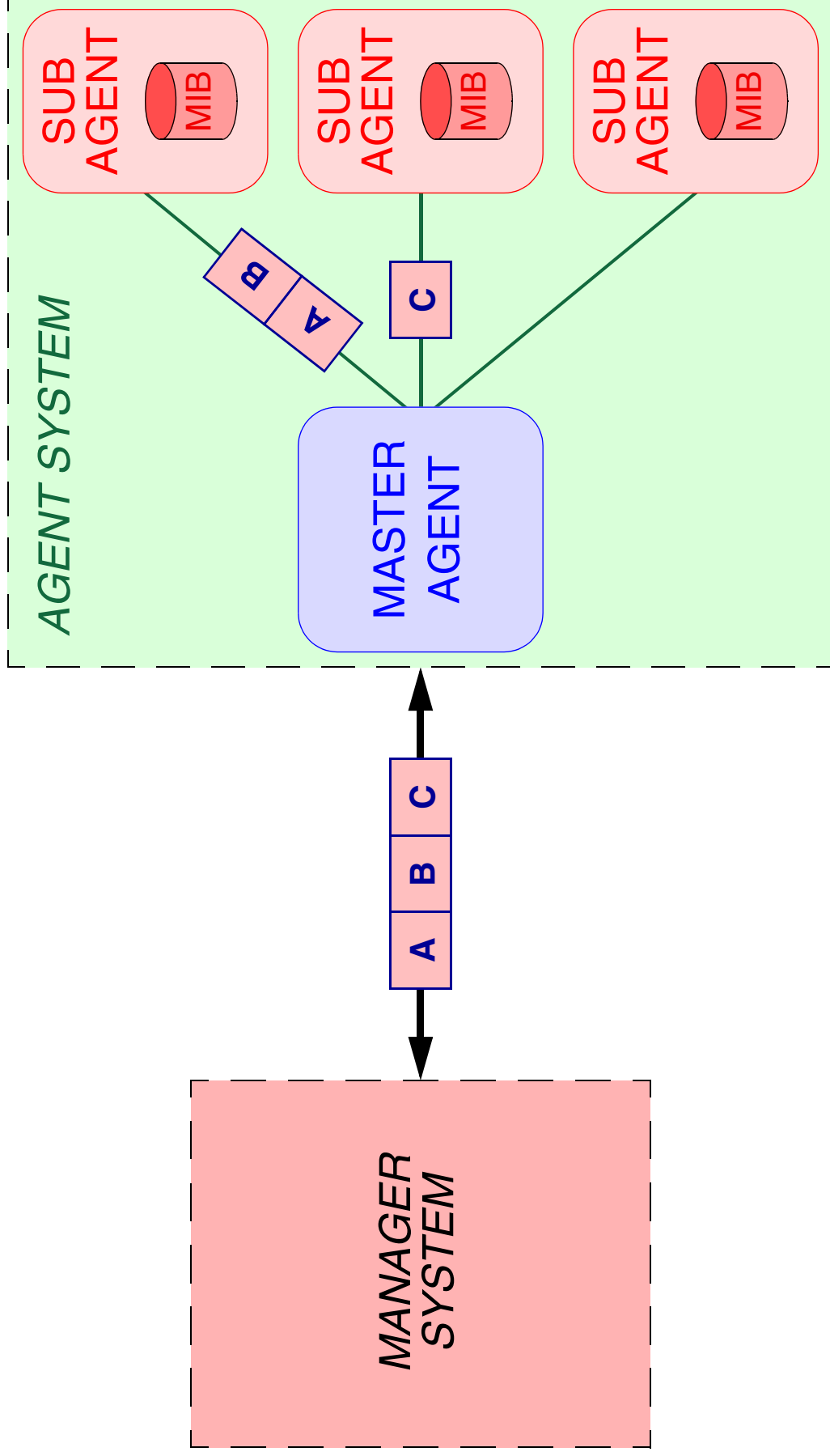


# BASIC STRUCTURE





# SPLITTING OF VARBIND LIST

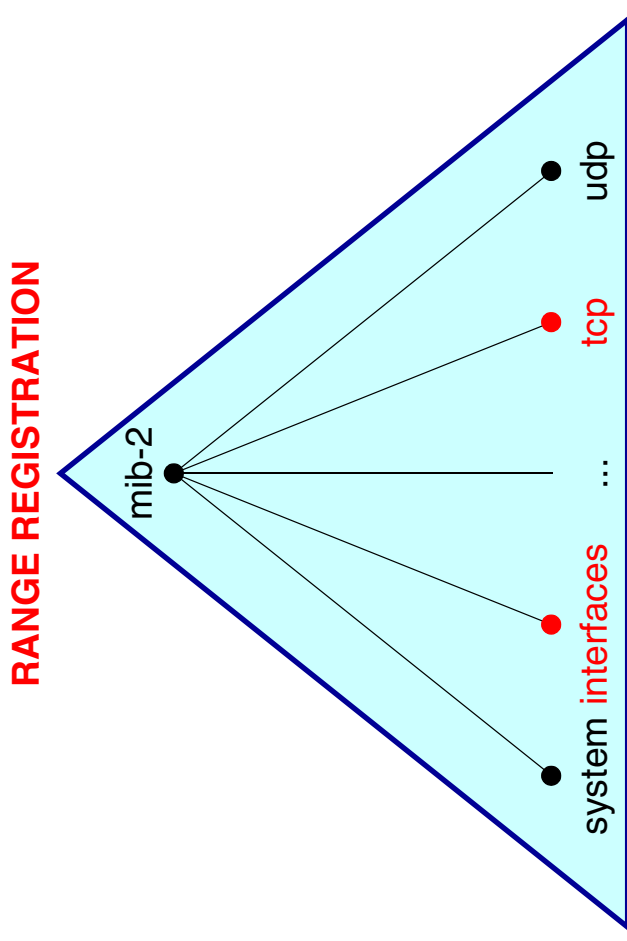
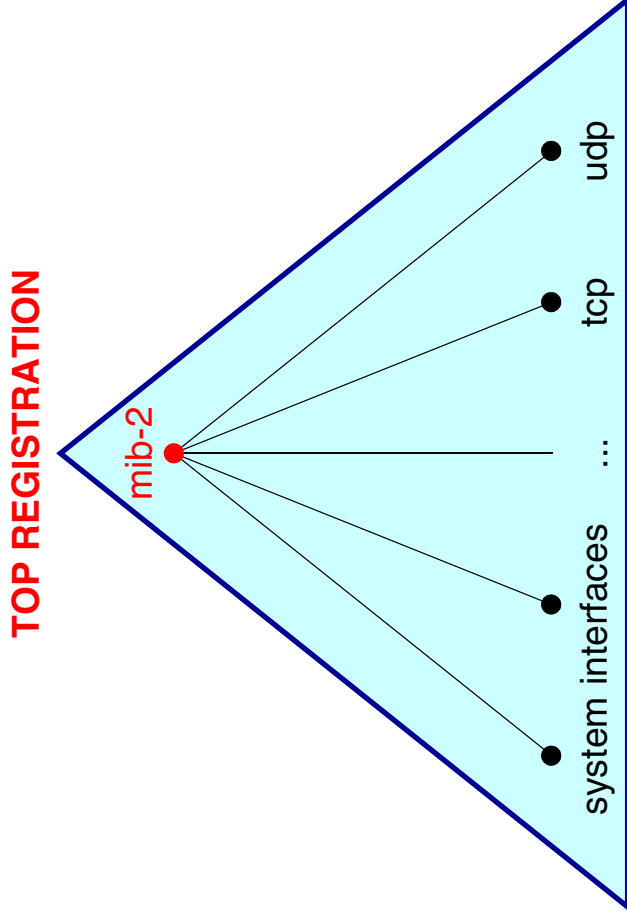




# CHARACTERISTICS

REQUIRES OID REGISTRATION:

- TOP REGISTRATION  
EXAMPLE: REGISTER(mib-2)
- RANGE REGISTRATION  
EXAMPLE REGISTER(interfaces -> tcp)



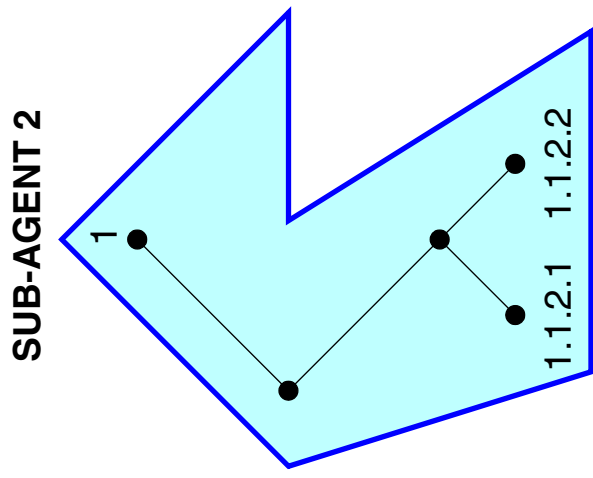
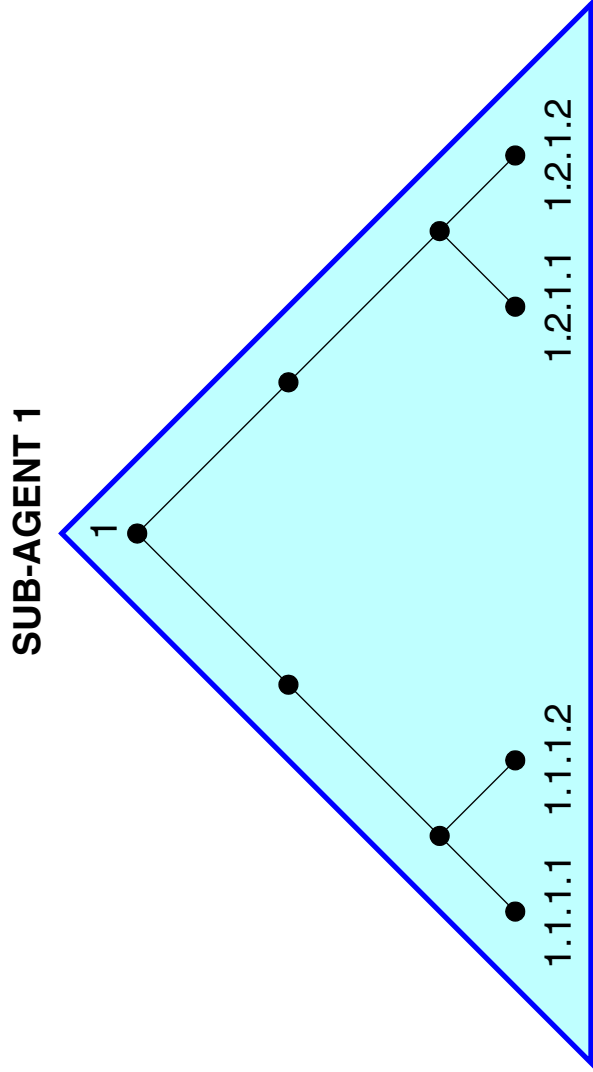


## **POTENTIAL PROBLEMS**

- TABLE ENTRIES MAY BE CREATED AND DELETED  
AT RUN-TIME
- ENTRIES OF A SINGLE TABLE MAY BE LOCATED  
IN DIFFERENT SUBAGENTS
- DUPLICATED OIDS
  - GAPS
  - SETS
- sysUpTime



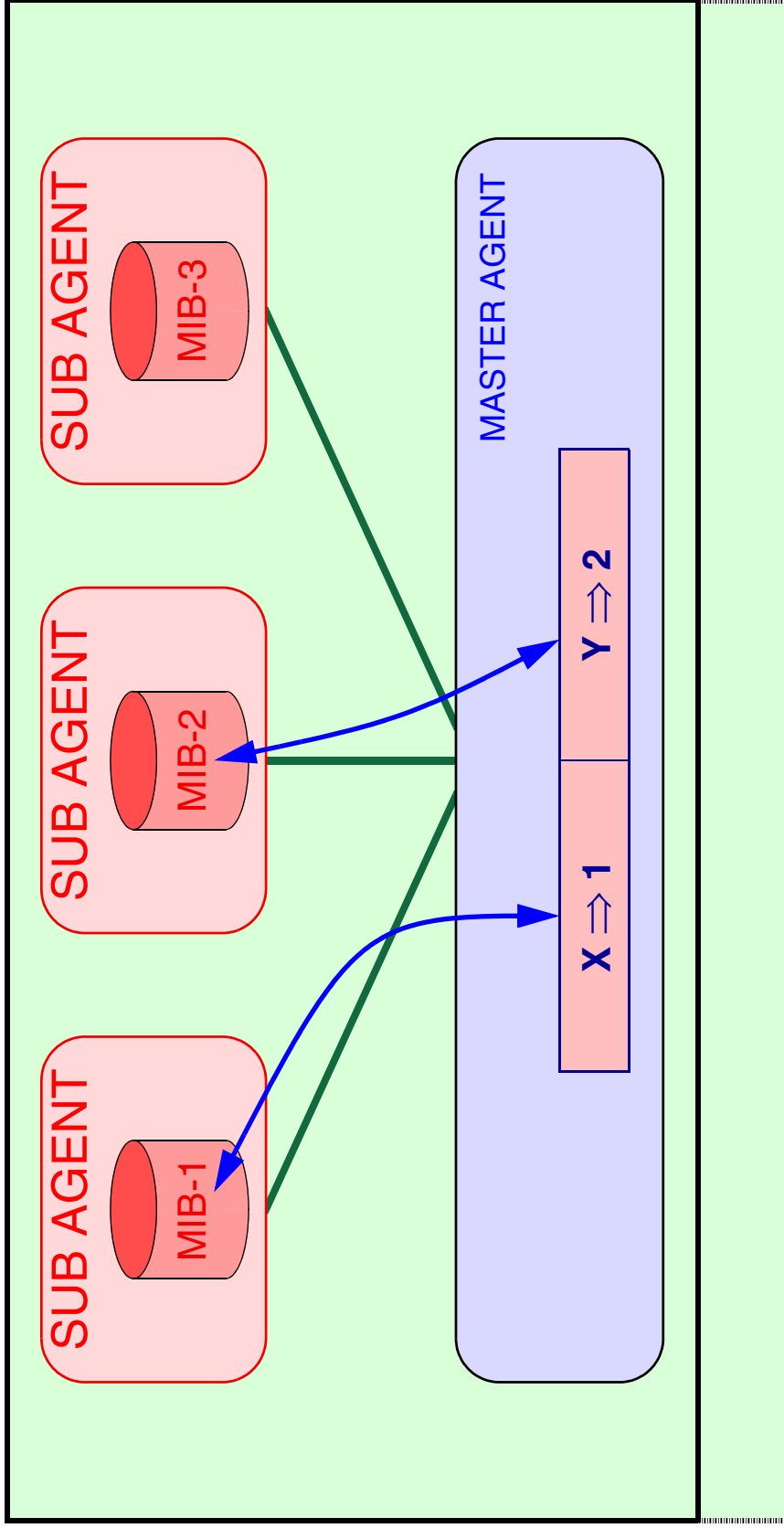
# EXAMPLE: GAPS



GET-NEXT ...



# SETS AND ATOMICITY



## TRANSACTION-LIKE APPROACH

- TEST
- COMMIT
- UNDO / CLEAN



## **HISTORY**

SMUX (1991: RFC 1227)  
SNMP MULTIPLEXING PROTOCOL

DPI (1991-1994: RFC 1228 & RFC 1592)  
DISTRIBUTED PROTOCOL INTERFACE

RESEARCH PROTOTYPES

FOR EXAMPLE: UNIVERSITY OF TWENTE - UT-SNMPv2

COMMERCIAL PRODUCTS

FOR EXAMPLE: SNMP RESEARCH - EMANATE  
(ENHANCED MANAGEMENT AGENT THROUGH EXTENSIONS)

AGENTX (1998-2000: RFC2741 & RFC2742)



## **AGENTX**

### PROPOSED IETF STANDARD

- RFC 2741 & RFC 2742
- <http://www.scguild.com/agentx/>

### HAS EFFICIENT MESSAGE FORMAT AND CODING

#### SUPPORTS

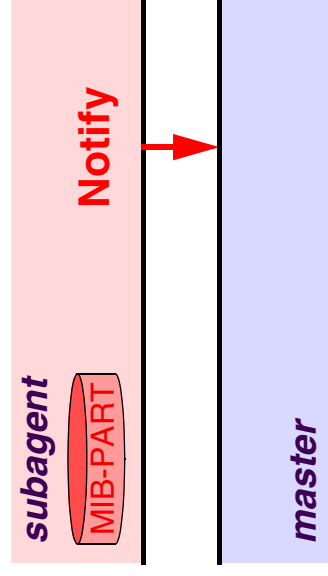
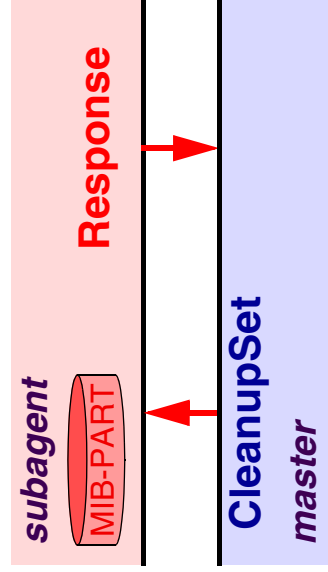
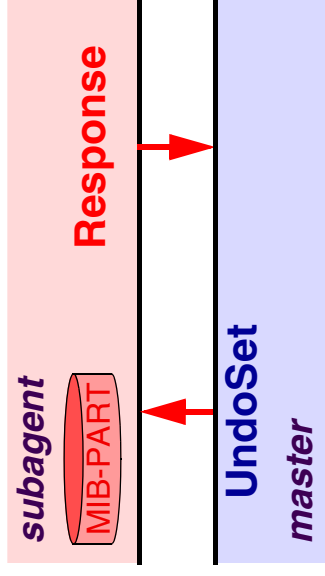
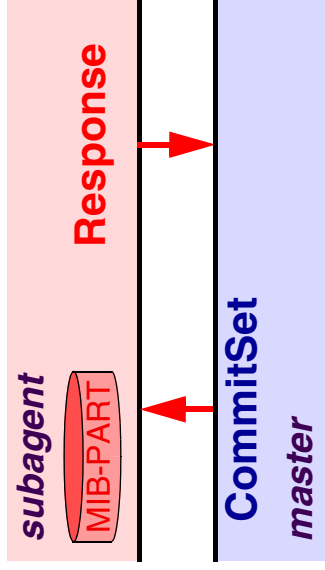
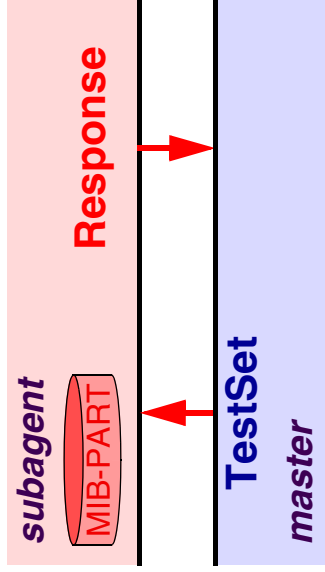
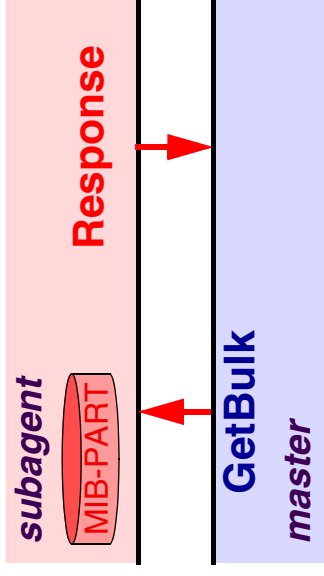
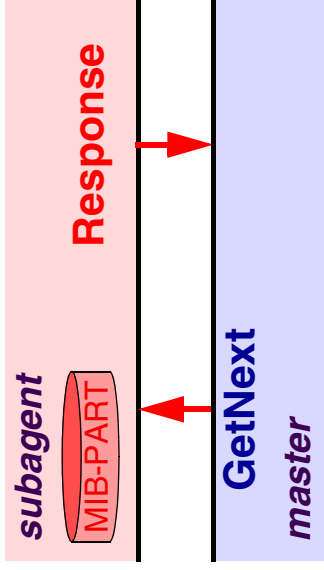
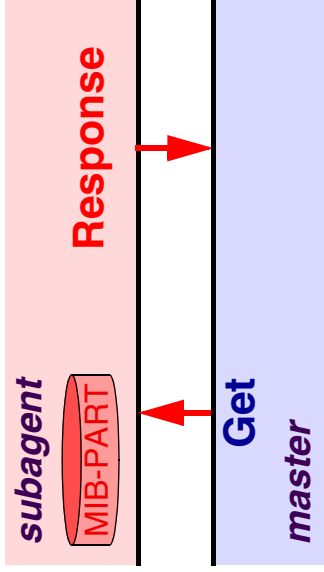
- SUBAGENTS IMPLEMENTING SEPARATE MIB MODULES
- SUBAGENTS IMPLEMENTING ROWS IN "SIMPLE TABLES"
- SUBAGENTS SHARING TABLES ALONG NON-ROW BORDERS

#### NON-GOALS

- SUBAGENTS SHARING "COMPLEX TABLES"
- SUBAGENT TO SUBAGENT COMMUNICATION



# AGENTX - NORMAL PDUS





## EXAMPLE: PDU FORMAT OF GetNext

VERSION	TYPE	FLAGS	RESERVED
SESSION ID			
TRANSACTION ID			
PACKET ID			
PAYLOAD LENGTH			
CONTEXT (OPTIONAL)			
OBJECT 1 START OF RANGE			
OBJECT 1 END OF RANGE			
OBJECT N START OF RANGE			
OBJECT 1 END OF RANGE			

LENGTH	PREFIX	INCLUDE	RESERVED
FIRST SUB IDENTIFIER			
LAST SUB IDENTIFIER			



## AGENTX - ADMINISTRATIVE PDUS

Open  
Close

AddAgentCaps  
RemoveAgentCaps

Register  
Unregister

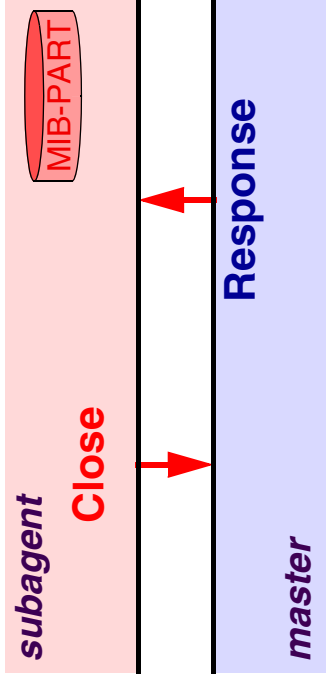
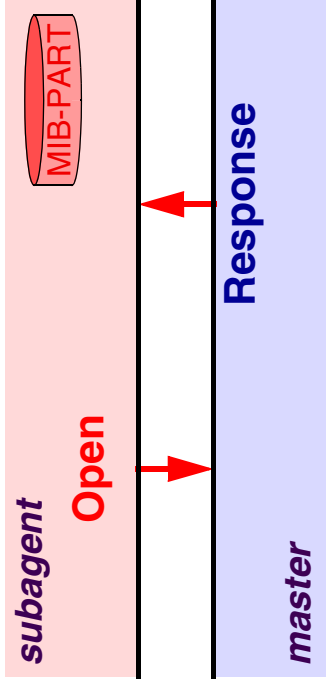
IndexAllocate  
IndexDeallocate

Ping

Response



## OPEN & CLOSE



TO ESTABLISH A SESSION

A UNIQUE `sessionID` IS ASSIGNED

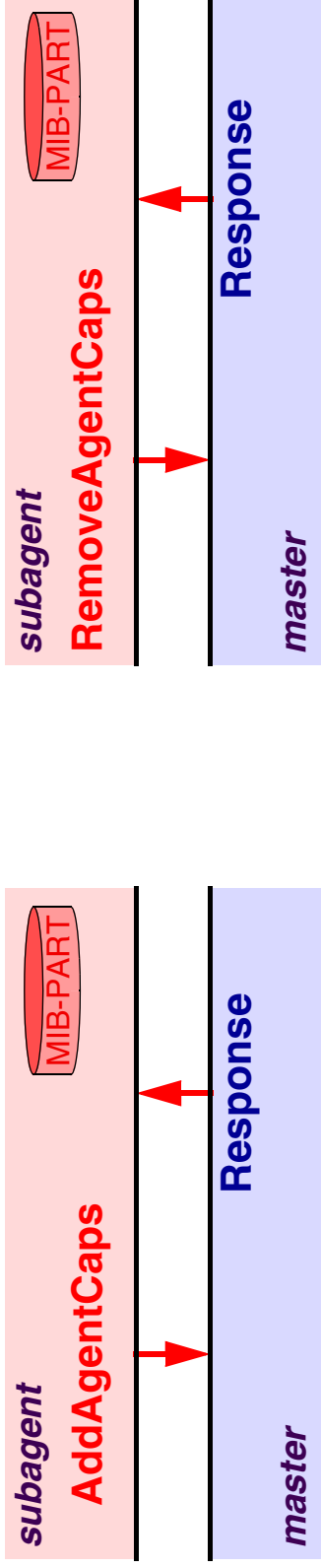
SUBAGENT SPECIFIES DEFAULT TIME-OUT

RESPONSES FROM MASTER ALWAYS INCLUDE `sysUpTime`

SESSION CAN BE CLOSED BY MASTER OR SUBAGENT



## AGENT CAPABILITIES



TO INFORM THE MASTER OF THE AGENT'S CAPABILITIES

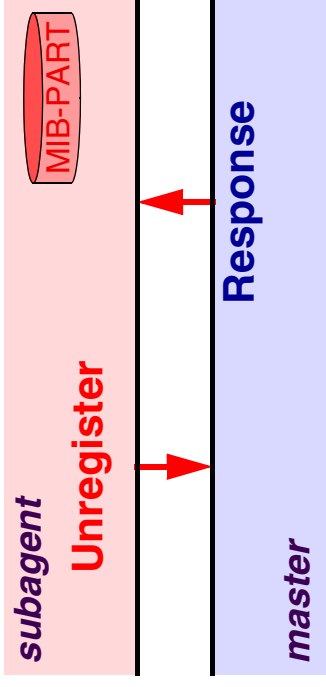
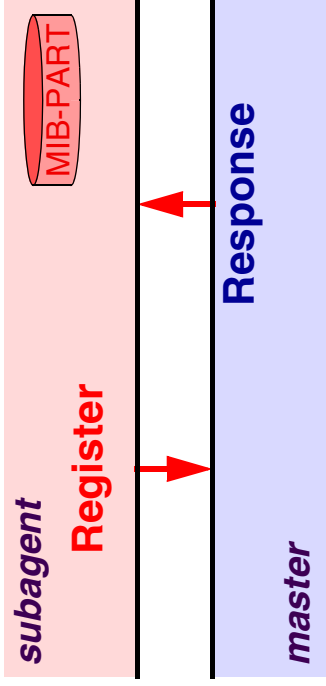
CAPABILITIES ARE DEFINED AS:

- AN OBJECT ID
- A HUMAN READABLE STRING

THE CAPABILITIES ARE STORED IN THE `sysORTable`



## REGISTRATION



### CHOICE BETWEEN:

- TOP REGISTRATION
- RANGE REGISTRATION

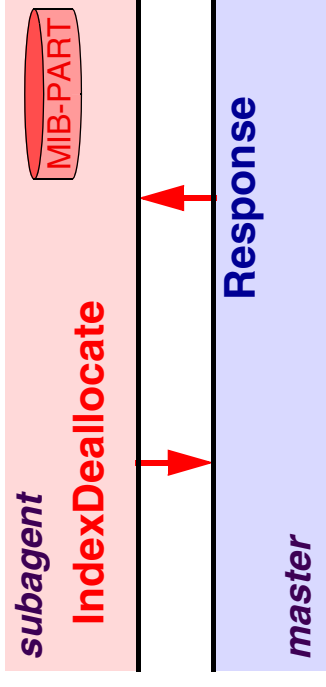
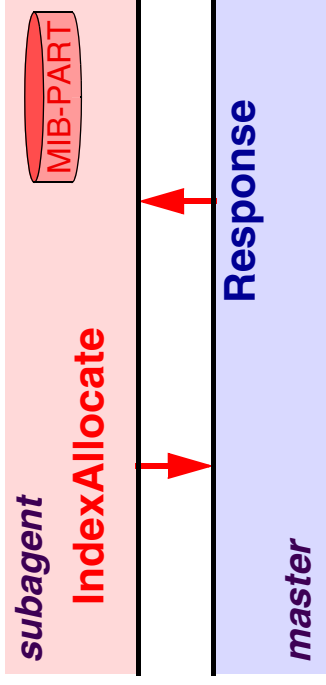
### PRIORITY CAN BE SPECIFIED

- TO DETERMINE THE AUTHORITATIVE SUBAGENT

### TIME-OUT CAN BE SPECIFIED



## INDEX ALLOCATION



TO ALLOCATE ONE OR MORE TABLE ROWS

SUBAGENT REQUESTS ALLOCATION OF:

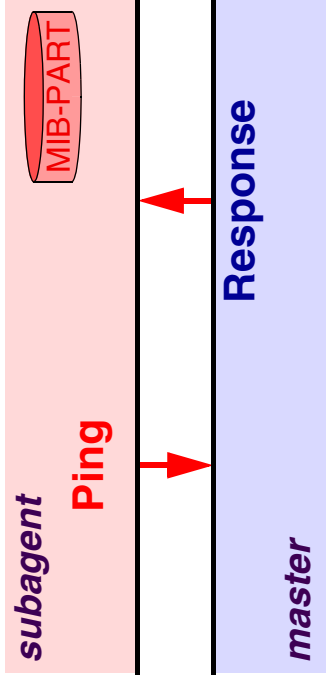
- A SPECIFIC INDEX VALUE
- AN INDEX VALUE THAT IS NOT CURRENTLY ALLOCATED
- AN INDEX VALUE THAT HAS NEVER BEEN ALLOCATED

MASTER AGENT MAINTAINS DATABASE

AFTER ALLOCATION REGISTRATION IS STILL NEEDED



# PING



TO MONITOR IF THE MASTER AGENT IS STILL ABLE  
TO RECEIVE AND SEND AGENTX PDUS



## **DISTRIBUTED MANAGEMENT**

THREE APPROACHES ARE BEING DEFINED

MIB BASED

- EXPRESSION MIB
- EVENT MIB
- NOTIFICATION LOG MIB

SCRIPT BASED

- SCRIPT MIB
- SCHEDULE MIB

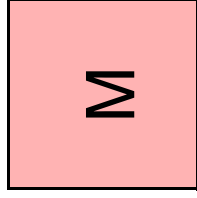
REMOTE OPERATIONS BASED

- REMOTE OPERATIONS MIB

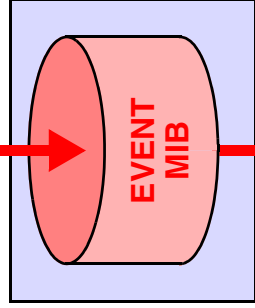


# EXPRESSION AND EVENT MIB

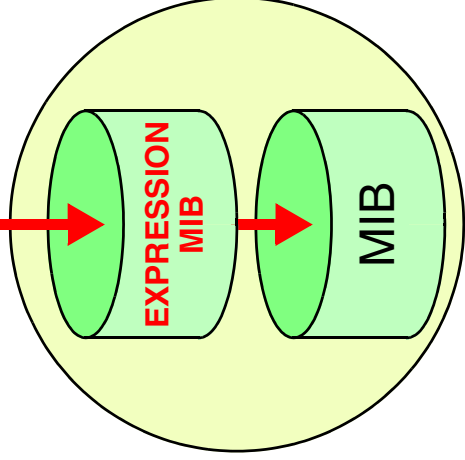
**TOP LEVEL  
MANAGER**



**INTERMEDIATE LEVEL  
MANAGER**



**AGENT**





## EXPRESSION AND EVENT MIB: CHARACTERISTICS

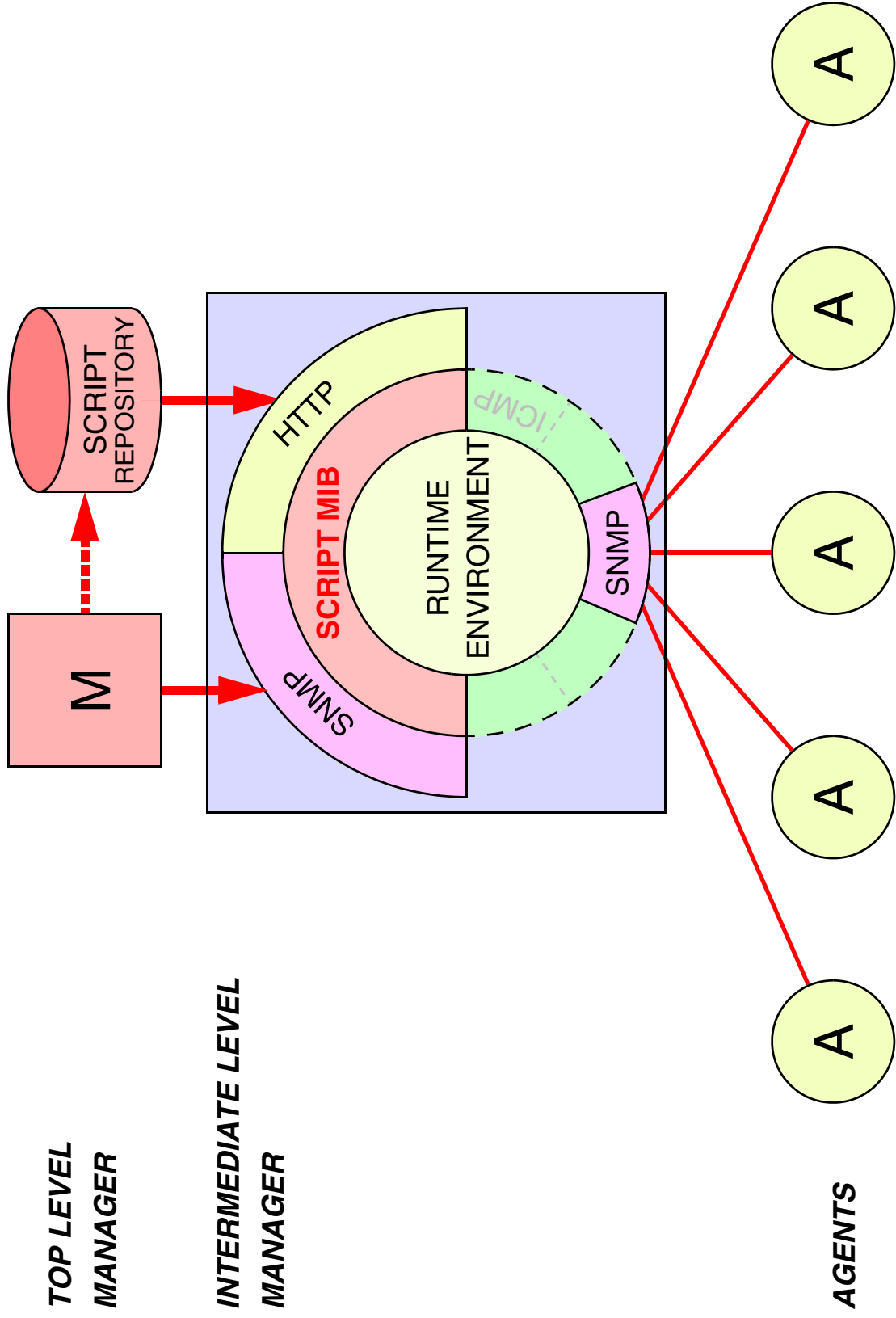
- STANDARD MIB APPROACH
  - RESEMBLES THE OLD SNMPv2 M2M MIB
- EXPRESSION MIB:**
- INPUT ARE (WILDCARDED) VARIABLES OF A (LOCAL) MIB
  - OPERATES ON ABSOLUTE AS WELL AS DELTA VALUES
    - RICH SET OF EXPRESSIONS
    - THE OUTPUT IS STORED IN THE VALUE TABLE
  - THIS TABLE MAY SERVE AS INPUT FOR OTHER EXPRESSIONS

**EVENT MIB:**

- INPUT ARE VARIABLES OF A (REMOTE) MIB
- TRIGGERS ON CHANGES, OR THRESHOLD CROSSING
- GENERATES A NOTIFICATION OR SET OPERATION



# SCRIPT MIB



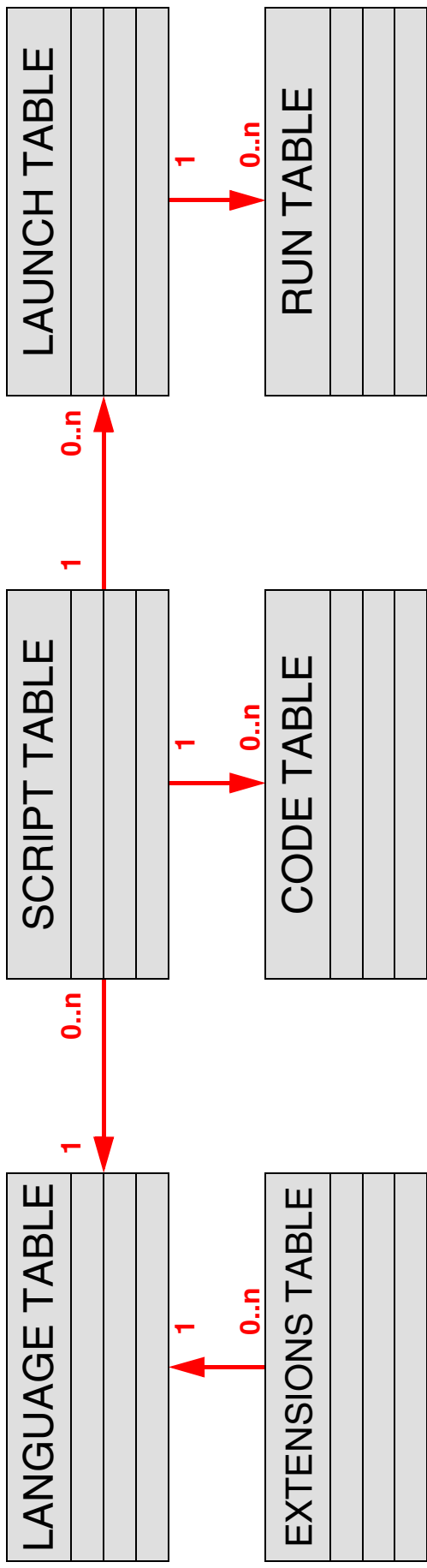


## **SCRIPT MIB: CHARACTERISTICS**

- FUNCTIONALITY CAN BE DEFINED AT RUN-TIME
  - POWERFUL AUTONOMOUS ACTIONS
- MAY BE EASIER TO OPERATE FOR THE TOP-LEVEL MANAGER
  - PROTECTION MECHANISMS NECESSARY
  - DIFFERENT SCRIPT LANGUAGES



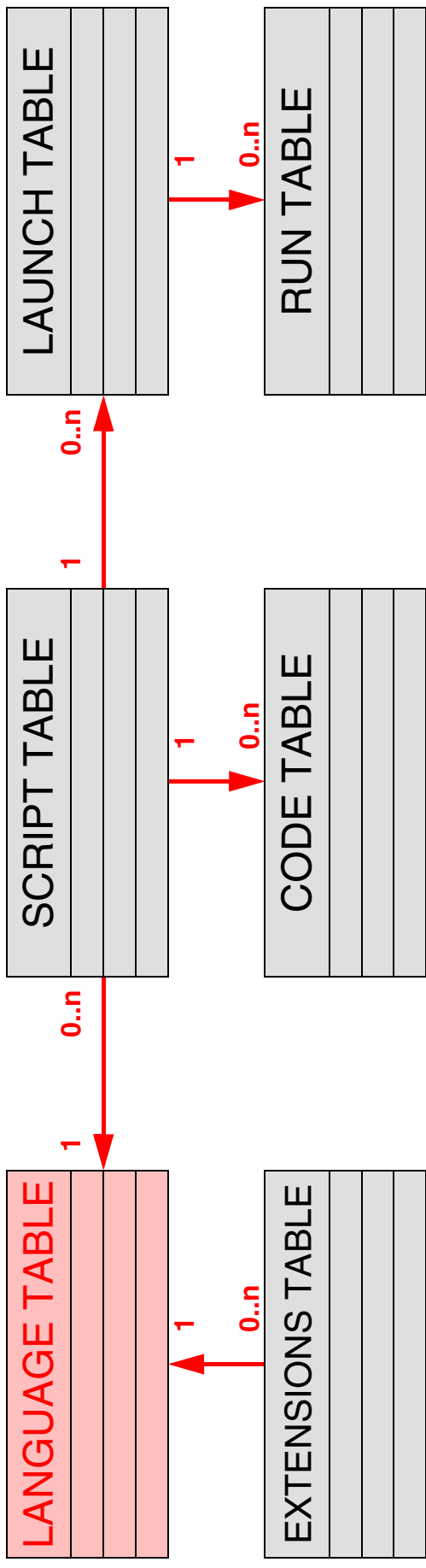
# SCRIPT MIB: STRUCTURE



CONSISTS OF 6 TABLES



## SCRIPT MIB: LANGUAGE TABLE



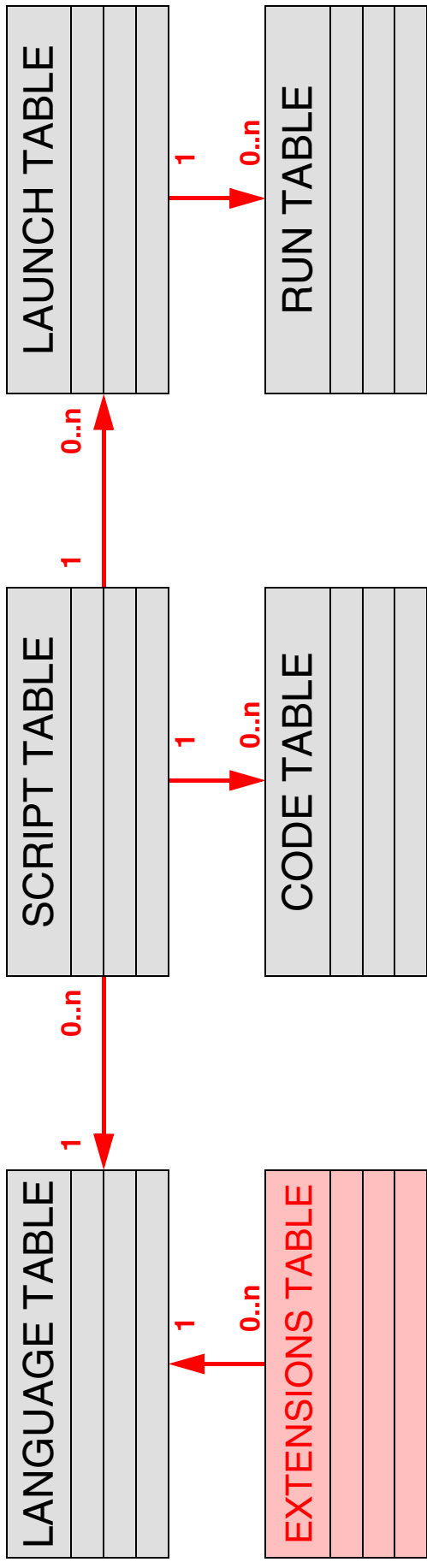
DEFINES THE LANGUAGES THIS SYSTEM SUPPORTS

- AN OID TO INDICATE THE LANGUAGE
  - THE VERSION
- AN OID TO INDICATE THE VENDOR
  - THE REVISION
  - A DESCRIPTION

TABLE IS READ ONLY



## SCRIPT MIB: EXTENSIONS TABLE



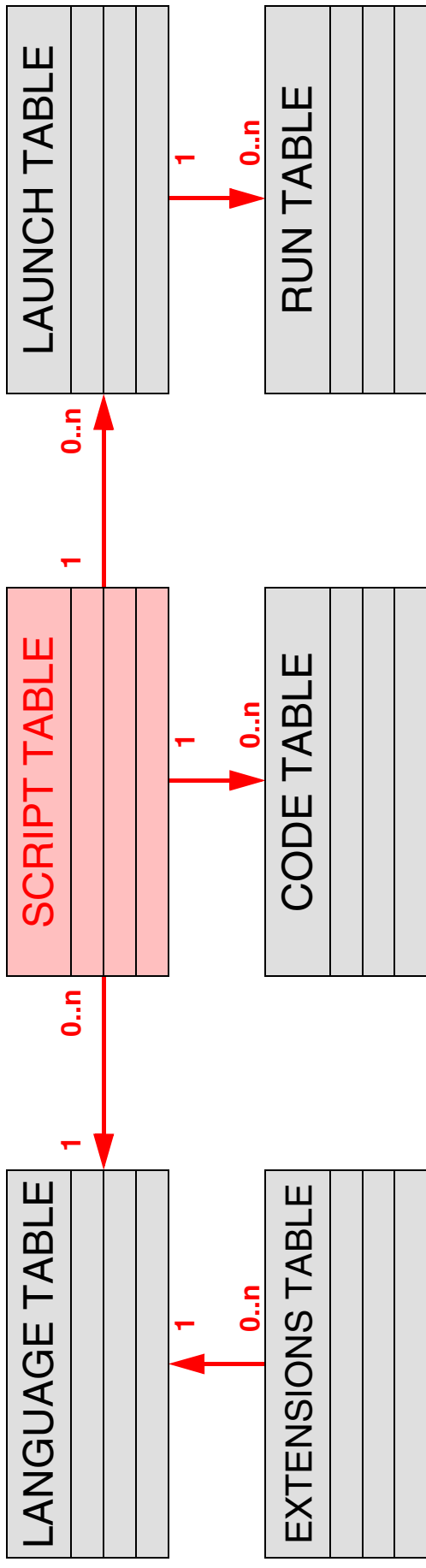
## DEFINES THE EXTENSIONS FOR EACH LANGUAGE

- AN OID TO INDICATE THE EXTENSION
  - THE VERSION
- AN OID TO INDICATE THE VENDOR
  - THE REVISION
  - A DESCRIPTION

TABLE IS READ ONLY



## SCRIPT MIB: SCRIPT TABLE



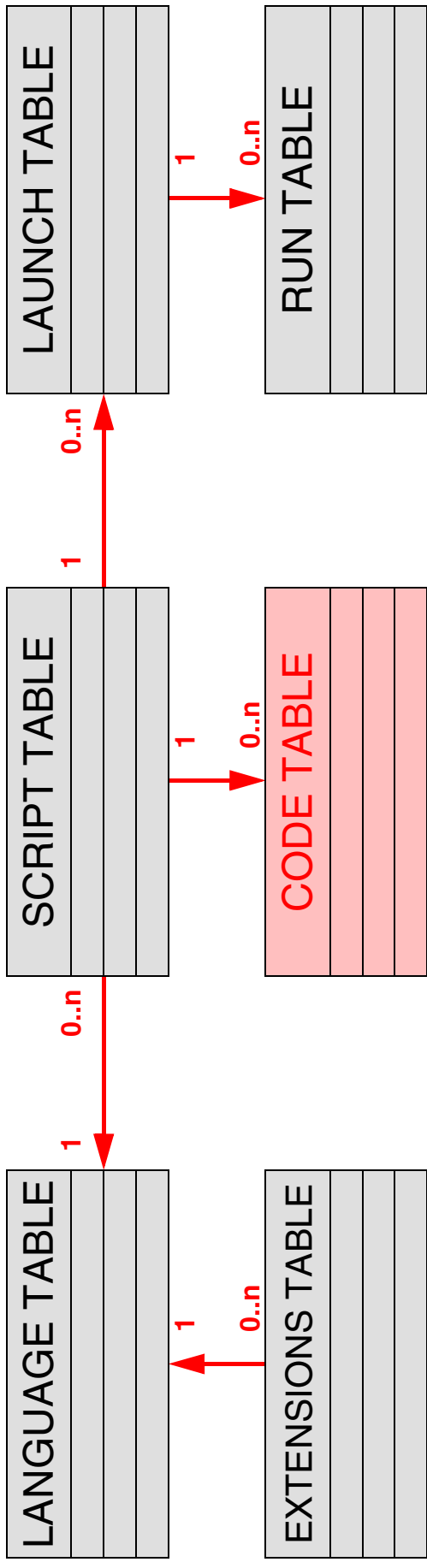
## LISTS ALL SCRIPTS KNOWN TO THE SYSTEM

### ALLOWS TO:

- DOWNLOAD SCRIPTS FROM A URL (PULL MODEL)
- READ SCRIPTS FROM LOCAL NON-VOLATILE STORAGE
  - STORE SCRIPTS IN LOCAL NON-VOLATILE STORAGE
  - DELETE SCRIPTS FROM LOCAL NON-VOLATILE STORAGE
- LIST PERMANENT SCRIPTS (THAT CAN NOT BE CHANGED OR REMOVED)
- READ AND MODIFY THE SCRIPT STATUS (ENABLED, DISABLED, EDITING)



## SCRIPT MIB: CODE TABLE



### LISTS THE CODE OF A SCRIPT

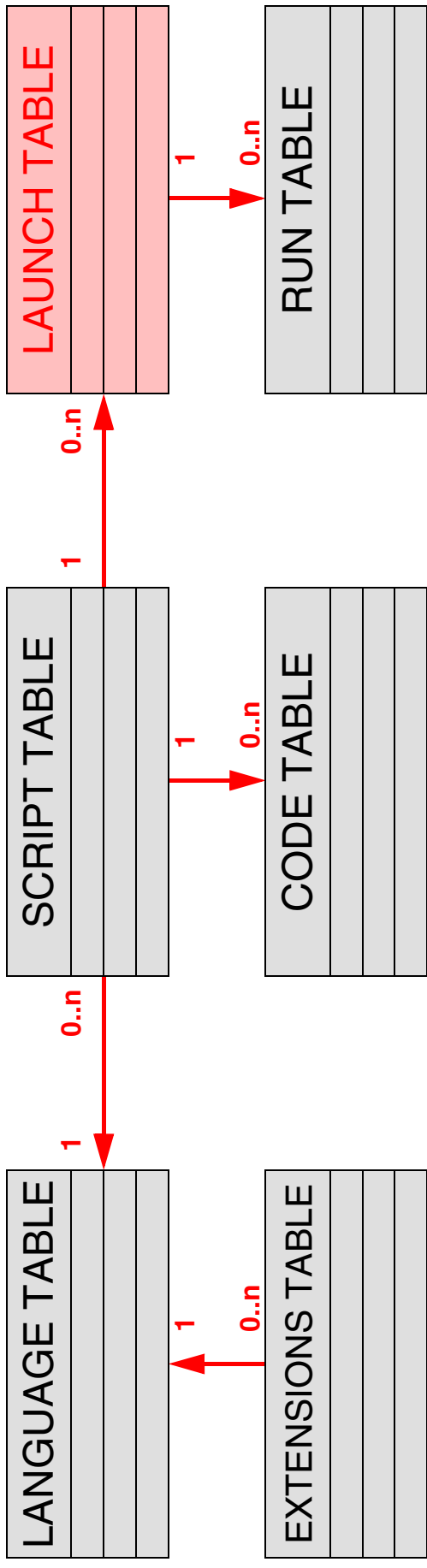
#### ALLOWS TO:

- DOWNLOAD SCRIPTS VIA SNMP (PUSH MODEL)
- MODIFY SCRIPTS VIA SNMP (EDITING)

### IMPLEMENTATION IS OPTIONAL



## SCRIPT MIB: LAUNCH TABLE

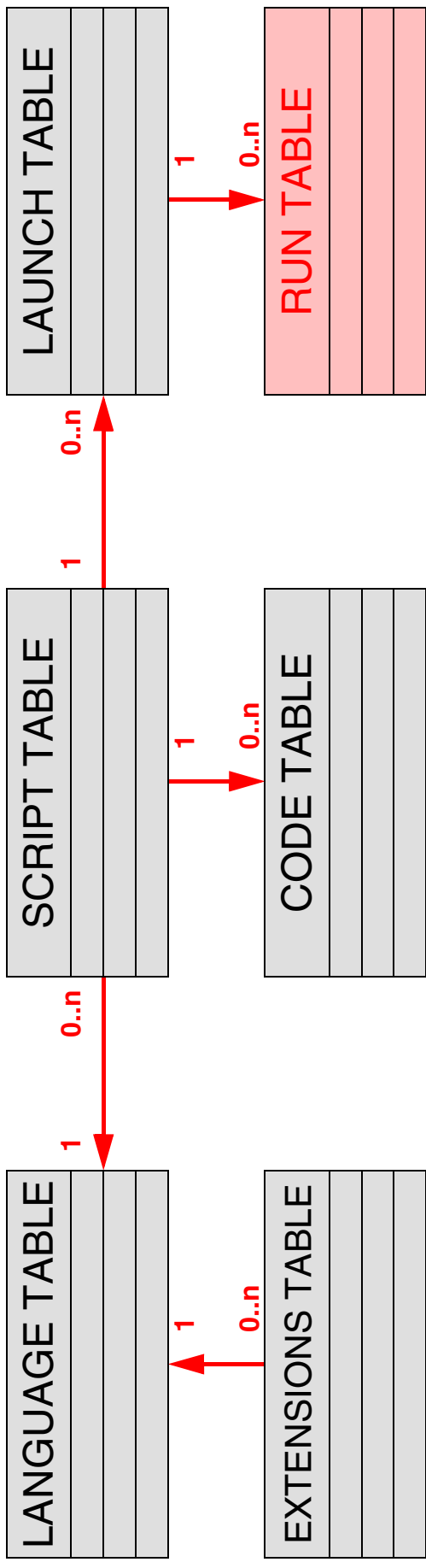


### ALLOWS TO:

- ASSOCIATE A SCRIPT WITH A 'PERSON' WHO INVOKES EXECUTION
- PROVIDE ARGUMENTS AND PARAMETERS FOR SCRIPT INVOCATION
  - INVOKE SCRIPTS WITH A SINGLE SET OPERATIONS
  - CONTROL THE NUMBER OF ACTIVE INVOCATIONS
  - CONTROL THE TOTAL NUMBER OF INVOCATIONS



## SCRIPT MIB: RUN TABLE



### ALLOWS TO:

- RETRIEVE STATUS INFORMATION FROM RUNNING SCRIPTS
- CONTROL RUNNING SCRIPTS (SUSPEND, RESUME, ABORT)
- RETRIEVE RESULTS FROM RECENTLY TERMINATED SCRIPTS
- CONTROL THE REMAINING MAXIMUM LIFETIME OF A RUNNING SCRIPT
  - CONTROL HOW LONG SCRIPT RESULTS ARE ACCESSIBLE



## **SCHEDULE MIB**

PERFORMS SET OPERATIONS

FOR EXAMPLE ON THE SCRIPT MIB

- TARGET MUST BE *Integer32*

ON A PERIODIC OR CALENDER DRIVEN BASE



## **REMOTE OPERATIONS MIB**

### **PING MIB**

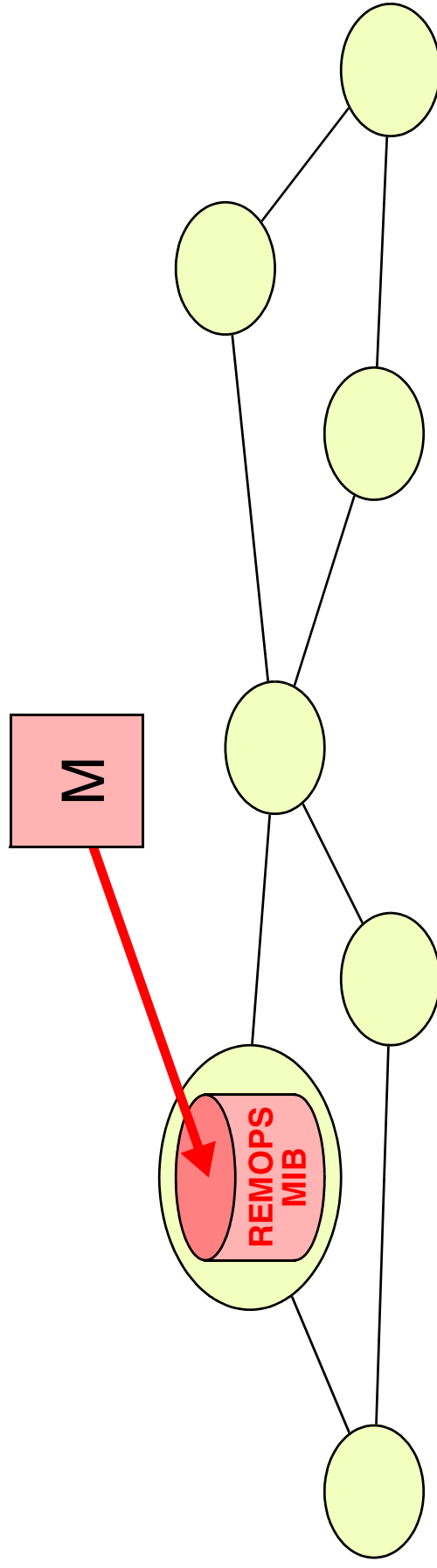
- TO PERFORM PING FROM A REMOTE HOST

### **TRACEROUTE MIB**

- TO PERFORM TRACEROUTE FROM A REMOTE HOST

### **NAME LOOKUP MIB**

- TO PERFORM NAME LOOKUP FROM A REMOTE HOST





## **IETF WGs - I**

### PROTOCOL AND DATA DEFINITION:

- SNMPv3
- SMIng
- EOS
- DISMAN

### MIBs

- ADSL
- BRIDGE
- ENTITY
- ETHERNET INTERFACES & HUB

### MEASURING

- BENCHMARK
- RMON
- IPFIX
- PSAMP
- PTOMAIN

# **IETF WGs - II**

## OPERATIONS AND DEPLOYMENT

- DNS
- IPV6
- MBONED

## AAA

- AAA
- NASREQ

## POLICY & CONFIGURATION

- POLICY
- RAP
- CONFIGURATION MANAGEMENT

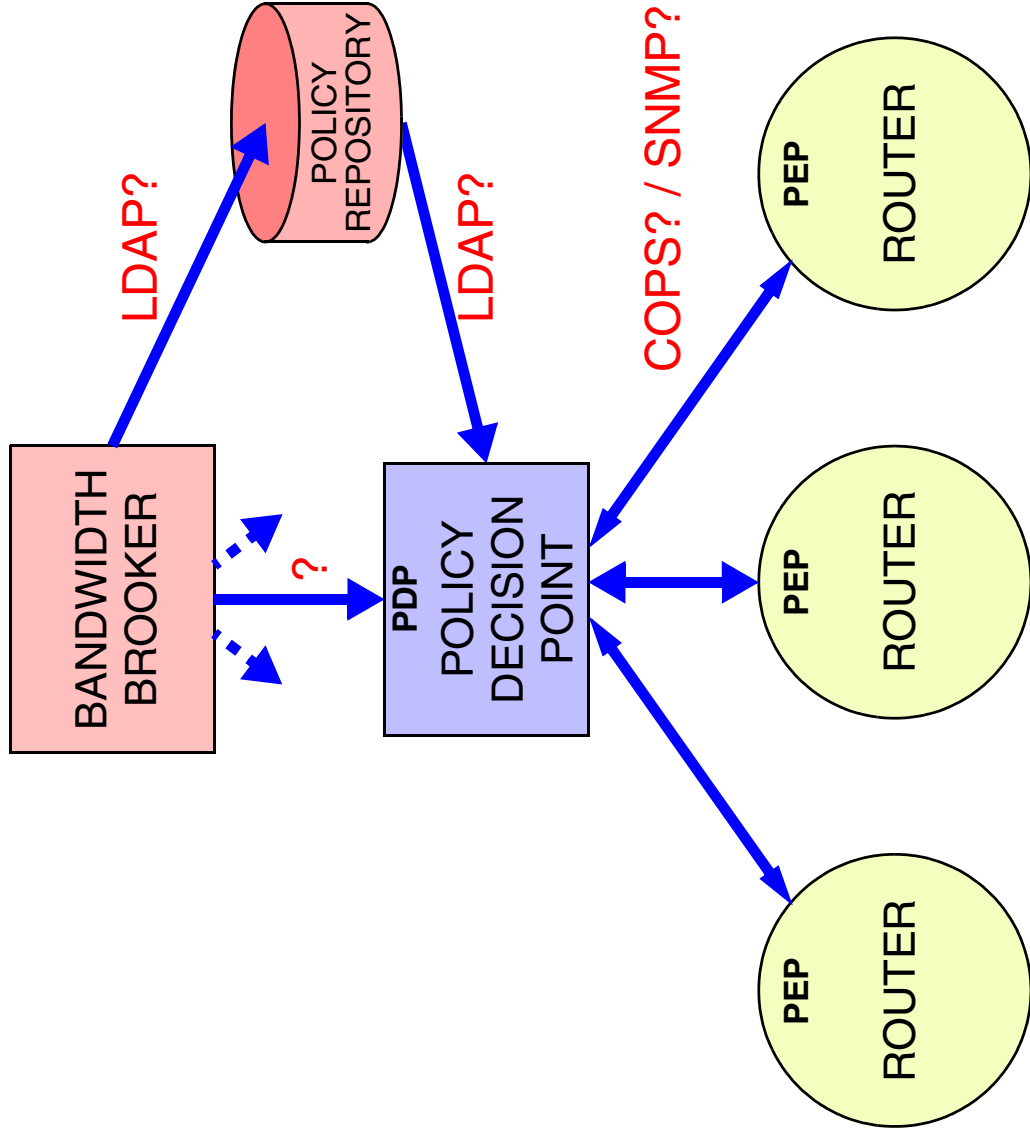
## OTHER

- MULT16



# NEW DEVELOPMENTS

## POLICY BASED MANAGEMENT





## COPS VERSUS SNMP

### COPS:

- SPECIAL CASE OF CONFIGURATION MANAGEMENT
- HIGHER LEVEL OBJECTS THAN USUAL WITH SNMP
  - POLICY INFORMATION BASE (PIB)
- SINGLE OPERATION TO ADD OR DELETE TABLE ROWS
- RELIABLE COMMUNICATION BETWEEN PDP AND PEP (BECAUSE OF TCP)
  - EACH PEP IS CONNECTED TO SINGLE PDP

### SNMP:

- INTEGRATED APPROACH TO MANAGEMENT
  - POLICIES CAN BE DEFINED WITHIN MIBS
- EACH PEP MAY BE CONNECTED TO MULTIPLE PDPs



## **EVOLUTION OF SNMP**

EOS  
2001

### **BASIC GOAL:**

**IMPROVE PERFORMANCE OF SNMP BULK DATA RETRIEVALS**

### **DIFFERENT ALTERNATIVES:**

- OLD DELTA COMPRESSION
- NEW PDUs (LIKE GET-TABLE)
- NEW BULK DATA TRANSFER MIBS

**MANY IDEAS COME FROM IRTF-NMRG**

**SNMP OVER TCP MAPPING**

**RFC3430**



## **XML BASED MANAGEMENT**

DMTF HAS ALWAYS BEEN ACTIVE IN THIS FIELD

INTERNET STANDARDIZATION ORGANIZATIONS:

- IRTF-NMRG
- XMLCONF MAILING LIST
  - IETF BOFS
- IAB WORKSHOP

VENDORS:

- JUNIPER

GOOD FOR:

- SOLVING SNMP'S DEFICIENCIES
- CONFIGURATION MANAGEMENT
- CLI INTEGRATION / CODE REUSE



## **EXAMPLE**

```
<rpc>
  <get-interface-information>
    <statistics/>
  </get-interface-information>
</rpc>

<rpc-reply>
  <interface-information>
    <InOctets>123456</InOctets>
    <InErrors>789</InErrors>
    <OutOctets>654321</OutOctets>
    <OutErrors>0</OutErrors>
  </interface-information>
</rpc-reply>
```



## **WEB SERVICES FOR MANAGEMENT**

### RECENT RESEARCH

#### OASIS

##### ADVANTAGES:

- COMMON MIDDLEWARE TECHNOLOGY
  - MANY SOFTWARE COMPONENTS
- FAST (MANAGEMENT APPLICATION) DEVELOPMENT



## **EXAMPLE**

```
<definitions name="InterfaceInformation"
...
<message name="Statistics">
</message>
<message name="StatisticsResult">
  <part name="InOctets"
    element="xsd:unsignedInt"/>
  <part name="InErrors" ...
  <part name="OutOctets" ...
  <part name="OutErrors" ...
</message>
<service name="InterfaceInfoService">
  <port ...
    {mapping on underlying protocol}
    {URI of web service}
  </port>
</service>
</definitions>
```



**MORE INFO ON WEB SERVICES FOR MANAGEMENT**

SEE PANEL 1 TOMORROW

(10:30-12:00)



## **WWW SERVERS**

- IETF

<http://www.ietf.org/>

- The SimpleWeb

<http://www.simpleweb.org/>

- The Simple Times

<http://www.simple-times.org/>

- The Smurfland NM Web Server

<http://netman.cit.buffalo.edu/>



## **BOOKS**

- W. Stallings  
SNMP, SNMPv2, SNMPv3 and RMON1 and 2  
Third edition, Addison-Wesley, 1999  
ISBN: 0-201-48534-6

- D. Zeltserman  
A Practical Guide to SNMPv3 and Network Management  
Prentice Hall, 1999  
ISBN: 0-13-021453-1

- D. Perkins, E. McGinnis  
Understanding SNMP MIBs  
Prentice Hall, 1996  
ISBN: 0-13-437708-7



## **ARTICLES**

The Simple Times: *Special issue on Agent Extensibility*  
Issue 4-2, April 1996

The Simple Times: *Special issue on SNMPv3*  
Issue 5-1, December 1997

The Simple Times: *An overview of the AgentX Protocol*  
Issue 6-1, March 1998

The Simple Times: *Special issue on SNMPv3*  
Issue 7-2, November 1999

William Stallings,  
Security Comes to SNMP: The New SNMPv3 Proposed Internet Standards  
The Protocol Journal, December 1998

William Stallings,  
SNMPv3: A Security Enhancement for SNMP,  
IEEE Communications Survey, Q4, 1998