INTERNET MANAGEMENT PROTOCOLS

265310
MSC/TEL/INF/BIT/EL
3 SP

AIKO PRAS
CENTRE FOR TELEMATICS AND INFORMATION TECHNOLOGY
UNIVERSITY OF TWENTE
THE NETHERLANDS

pras@ctit.utwente.nl
http://wwwhome.ctit.utwente.nl/~pras

OVERVIEW OF THE COURSE

DAY 1
INTRODUCTION

DAY 2
STRUCTURE OF MANAGEMENT INFORMATION

DAY 3
MANAGEMENT INFORMATION BASES

DAY 4
SIMPLE NETWORK MANAGEMENT PROTOCOL

DAY 5
SIMPLE NETWORK MANAGEMENT PROTOCOL - VERSION 3

DAY 6
REMOTE MONITORING, DISTRIBUTED MANAGEMENT, IMPLEMENTATIONS & STANDARDS

DAY 1

EXAMPLES OF MANAGEMENT INFORMATION

- CAMPUSNET
- SURFnet

DIFFERENT APPROACHES TO MANAGEMENT

- AD-HOC APPROACHES
- STANDARD APPROACHES

SNMP

- BASIC OPERATION
- HISTORY
- STANDARDS
- GOALS

DAY 2

STRUCTURE OF MANAGEMENT INFORMATION

- SCALARS
- TABLES
- ROW CREATION
- TEXTUAL CONVENTIONS

DAY 3

MANAGEMENT INFORMATION BASES

MIBS

MIB-II

- SNMPv2 MIB
- IF MIB
- IP MIB

HOST RESOURCES MIB

DAY 4

SIMPLE NETWORK MANAGEMENT PROTOCOL

- SNMPV1
- PROXIES
- MESSAGE ENCODING
- SNMPV2

DAY 5

SNMPv3

- DESIGN DECISIONS
- ARCHITECTURE
- SNMP MESSAGE STRUCTURE
- SECURE COMMUNICATION
  - USER SECURITY MODEL (USM)
- ACCESS CONTROL
  - VIEW BASED ACCESS CONTROL MODEL (VACM)
- IMPLEMENTATIONS
- RFCS

DAY 6

REMOTE MONITORING, DISTRIBUTED MANAGEMENT, IMPLEMENTATIONS & STANDARDS

- RMON
- DISMAN
- AGENTX
- PUBLIC DOMAIN IMPLEMENTATIONS
- COMMERCIAL IMPLEMENTATIONS
- STANDARDS

PURPOSE OF THIS COURSE

THE PURPOSE OF THIS COURSE IS TO GET AN UNDERSTANDING OF:

- WHICH MANAGEMENT INFORMATION EXISTS WITHIN THE INTERNET
- WHO DEFINES THIS INFORMATION
- THE SYNTAX IN WHICH THIS INFORMATION IS DESCRIBED
- THE PROTOCOLS THAT ALLOW THIS INFORMATION TO BE MONITORED / MODIFIED
- THE STRUCTURE OF MANAGEMENT IMPLEMENTATIONS

EXAMPLES OF MANAGEMENT INFORMATION

- UT AND CAMPUSNET
- SURFNET

NETWORK MANAGEMENT: AD-HOC APPROACHES

LOCAL SYSTEM - EXAMPLES:

- PING
- TRACEROUTE
- NETSTAT
- ...

REMOTE SYSTEM

- TELNET / RLOGIN (COMMAND LINE INTERFACE - CLI)
- WEB INTERFACE

EXAMPLES WEB INTERFACE:

- PING
- TRACEROUTE
- WHOIS
- NTOP
- ...

NETWORK MANAGEMENT: STANDARDS

INTERNET

- INTERNET ENGINEERING TASK FORCE (IETF)
- OPERATIONS AND MANAGEMENT AREA
- SNMP

ISO

- ISO-IEC/JTC 1/WG 4
- OSI
- CMIP-CMIS

ITU-T

- SG IV
- TMN

OTHERS

- DMTF
- TM FORUM
- OMG
- IEEE
- ...

CHARACTERISTICS

IETF

- MANAGEMENT SHOULD BE SIMPLE
- VARIABLE ORIENTED APPROACH
- MANAGEMENT INFORMATION EXCHANGES MAY BE UNRELIABLE

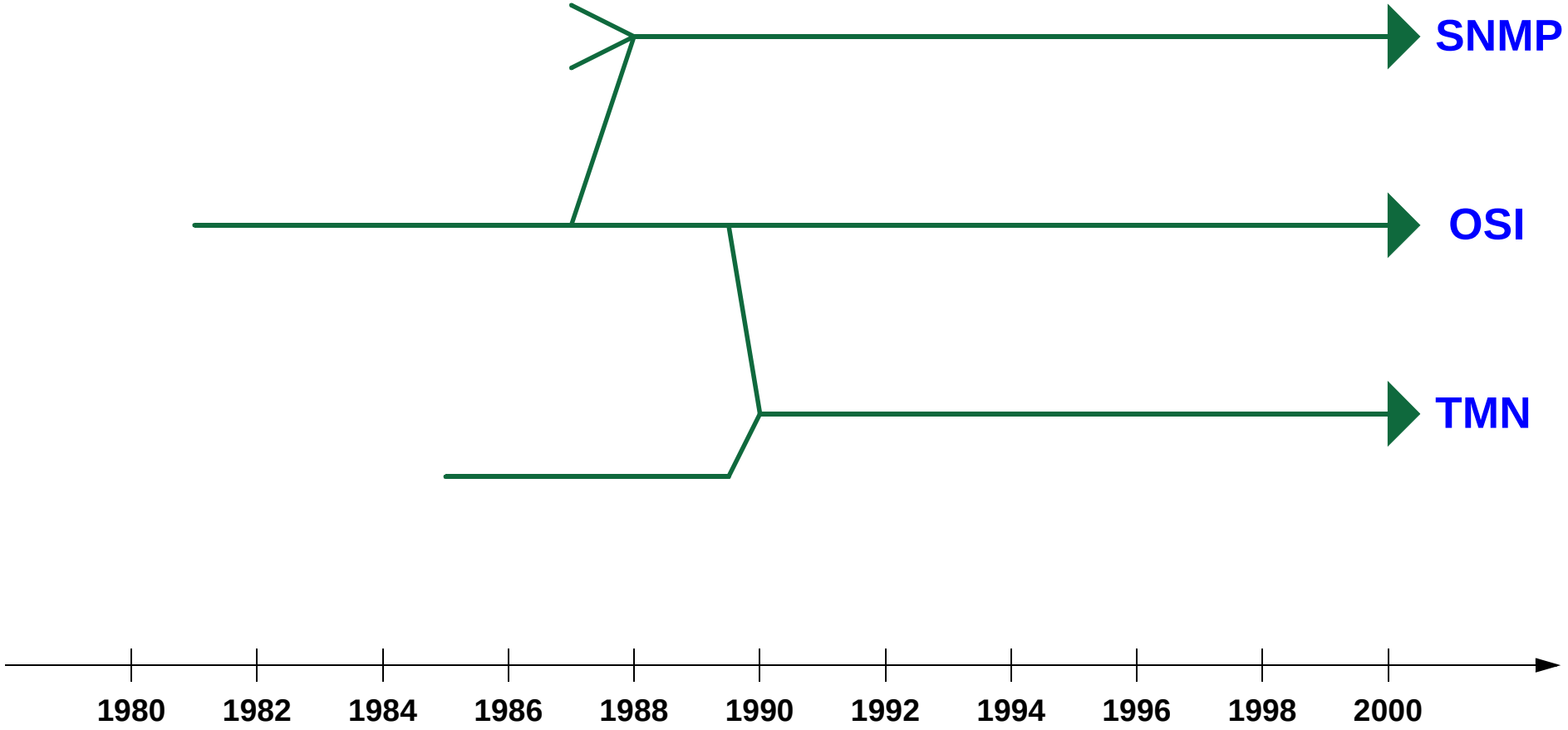
ISO

- MANAGEMENT SHOULD BE POWERFUL
- OBJECT ORIENTED APPROACH
- MANAGEMENT INFORMATION MUST BE EXCHANGED IN A RELIABLE FASHION

TMN

- DEFINES ONLY A MANAGEMENT *ARCHITECTURE*
- THE ACTUAL PROTOCOLS ARE THOSE OF OSI
- OUT-OF-BAND MANAGEMENT

HISTORY


SNMP
OSI
TMN
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000

MARKET SHARE

NUMBER OF AGENTS?

NUMBER OF MANAGERS?

MONEY?

FOR SPECIFIC ENVIRONMENTS?

- IBM MAIN-FRAMES
- PRIVATE DATA LANs
- PRIVATE DATA WANs
- PABX
- PUBLIC VOICE NETWORKS
- PUBLIC DATA NETWORKS

WHY DID SNMP SUCCEED?

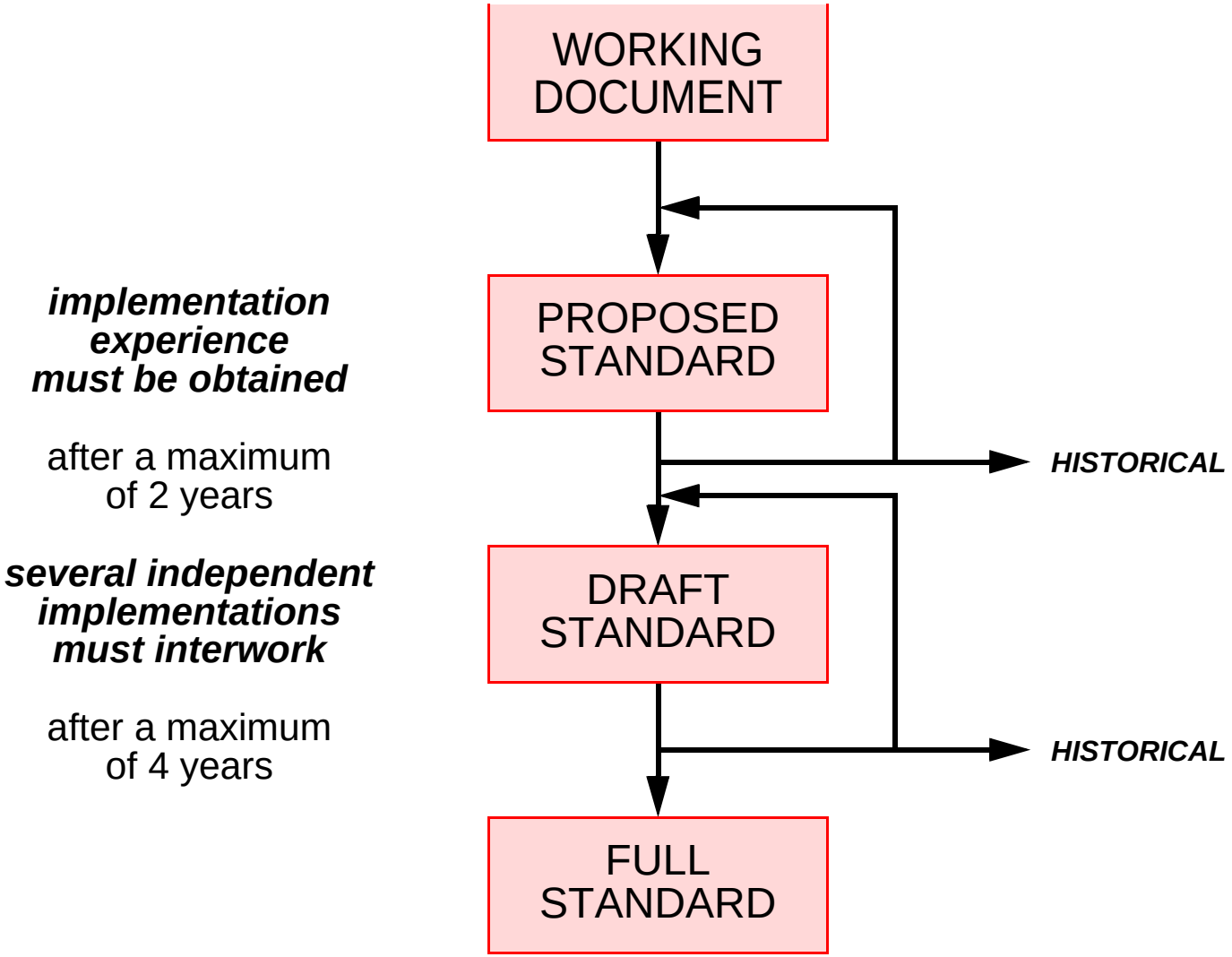
- STANDARDS CAN BE OBTAINED FOR FREE

- STANDARDS ARE AVAILABLE FROM FTP & WWW SERVERS IN AN ELECTRONIC FORM

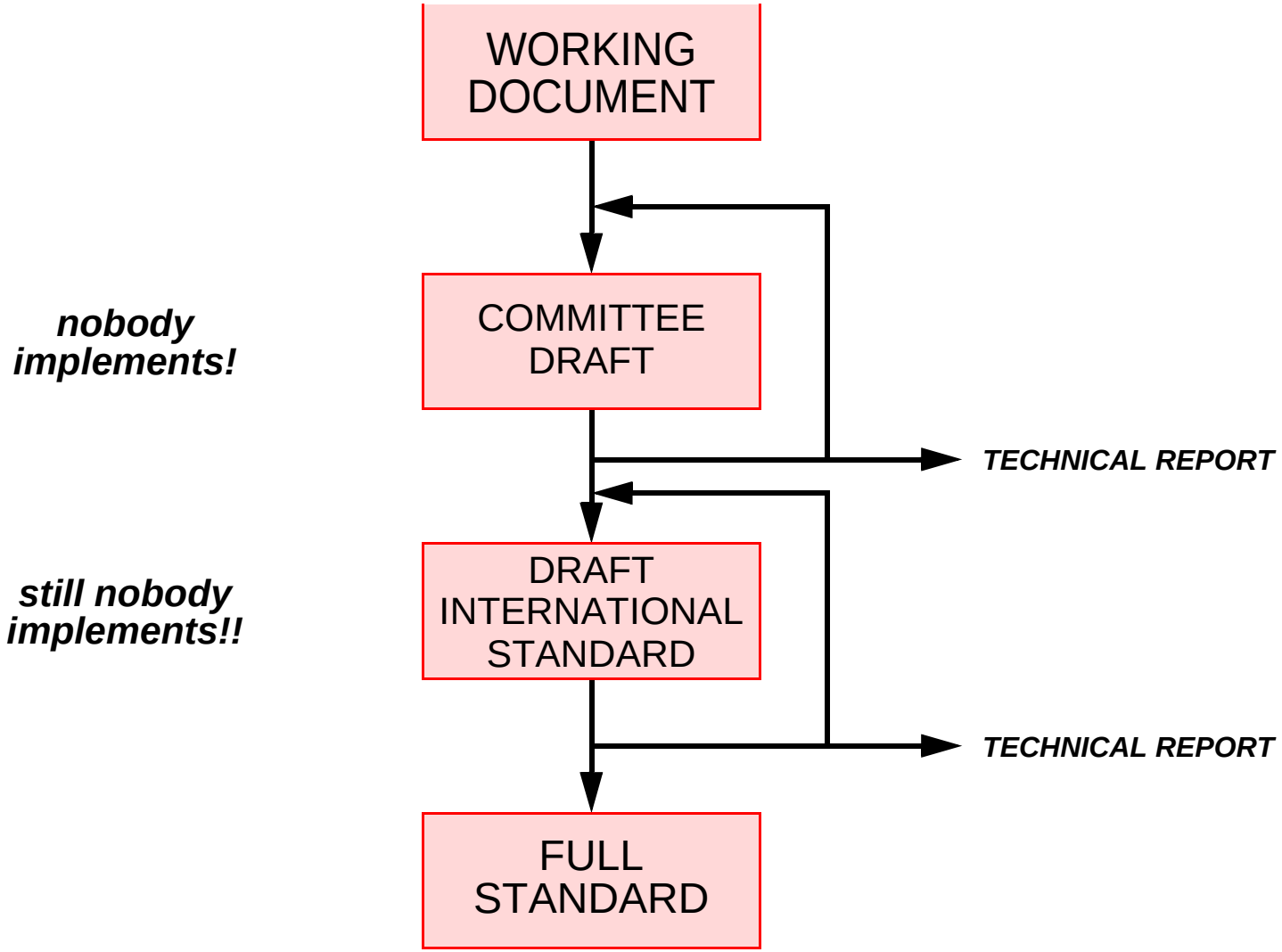
- RAPID DEVELOPMENT OF STANDARDS

- PROTOTYPES MUST DEMONSTRATE THE NEED FOR, AND THE FEASIBILITY OF STANDARDS

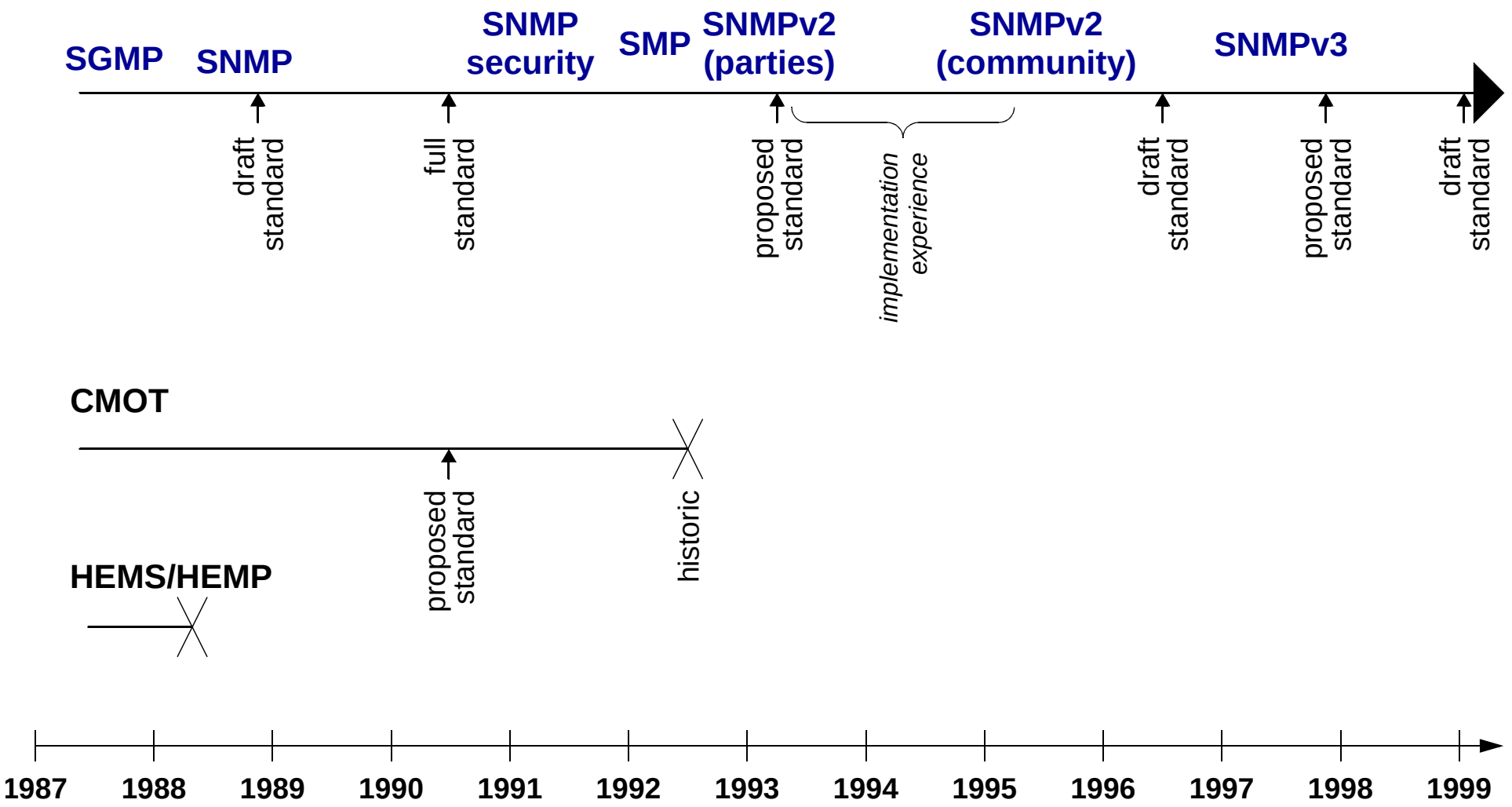
IETF STANDARDIZATION



ISO STANDARDIZATION

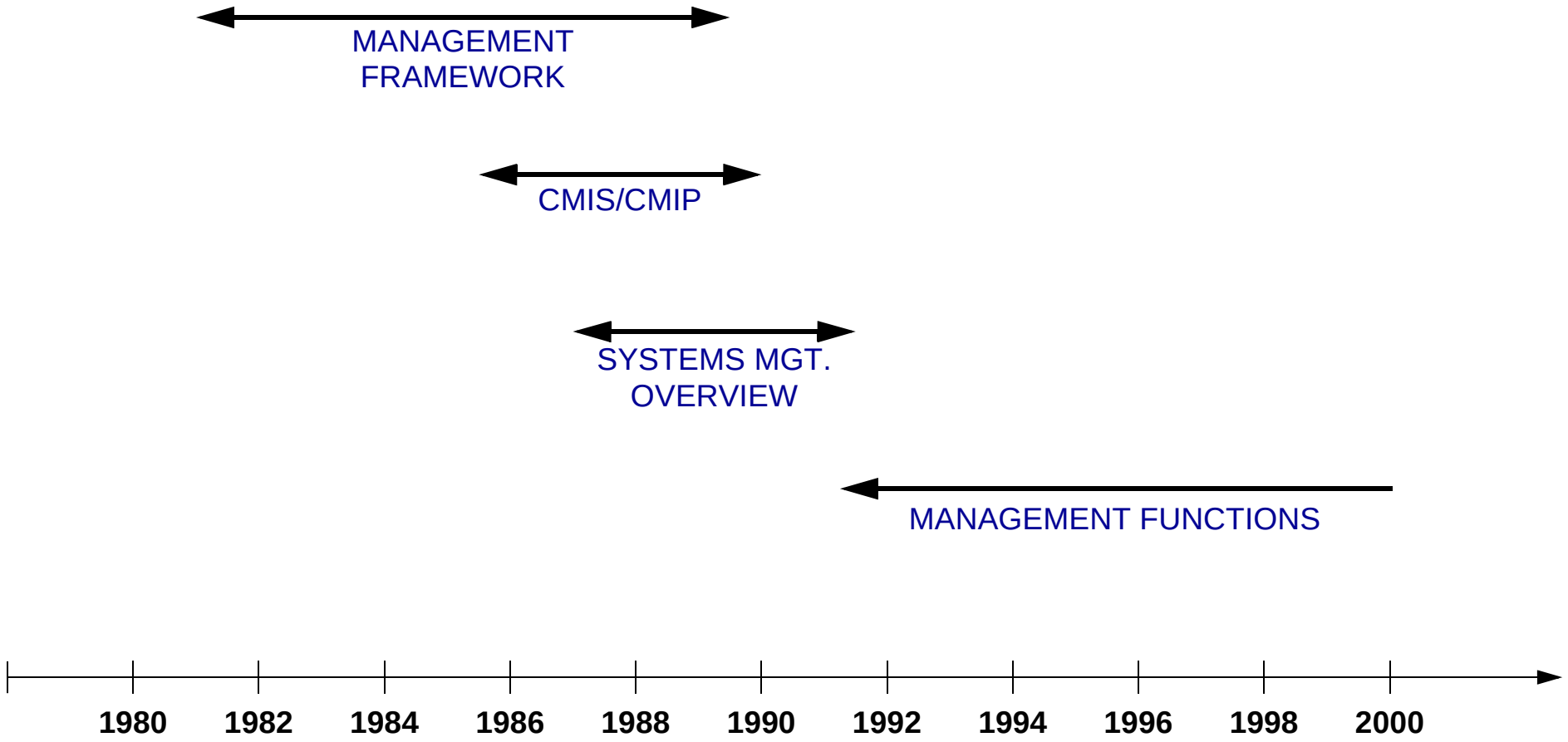


SNMP HISTORY



OSI HISTORY

SC21/WG4 ESTABLISHED



TMN HISTORY

WORK ON TMN
STARTED BY SGIV

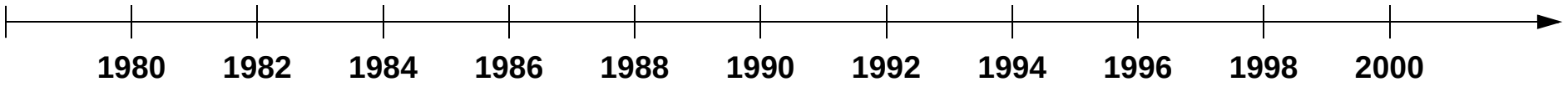
M30

IDEAS FROM OSI MGT.
WORK STARTED ON DERIVED STANDARDS
RESPONSIBILITY MODEL BECAME ANNEX

M3010

RESPONSIBILITY MODEL PART OF MAIN TEXT

M3010 (rev.)



SNMP GOALS

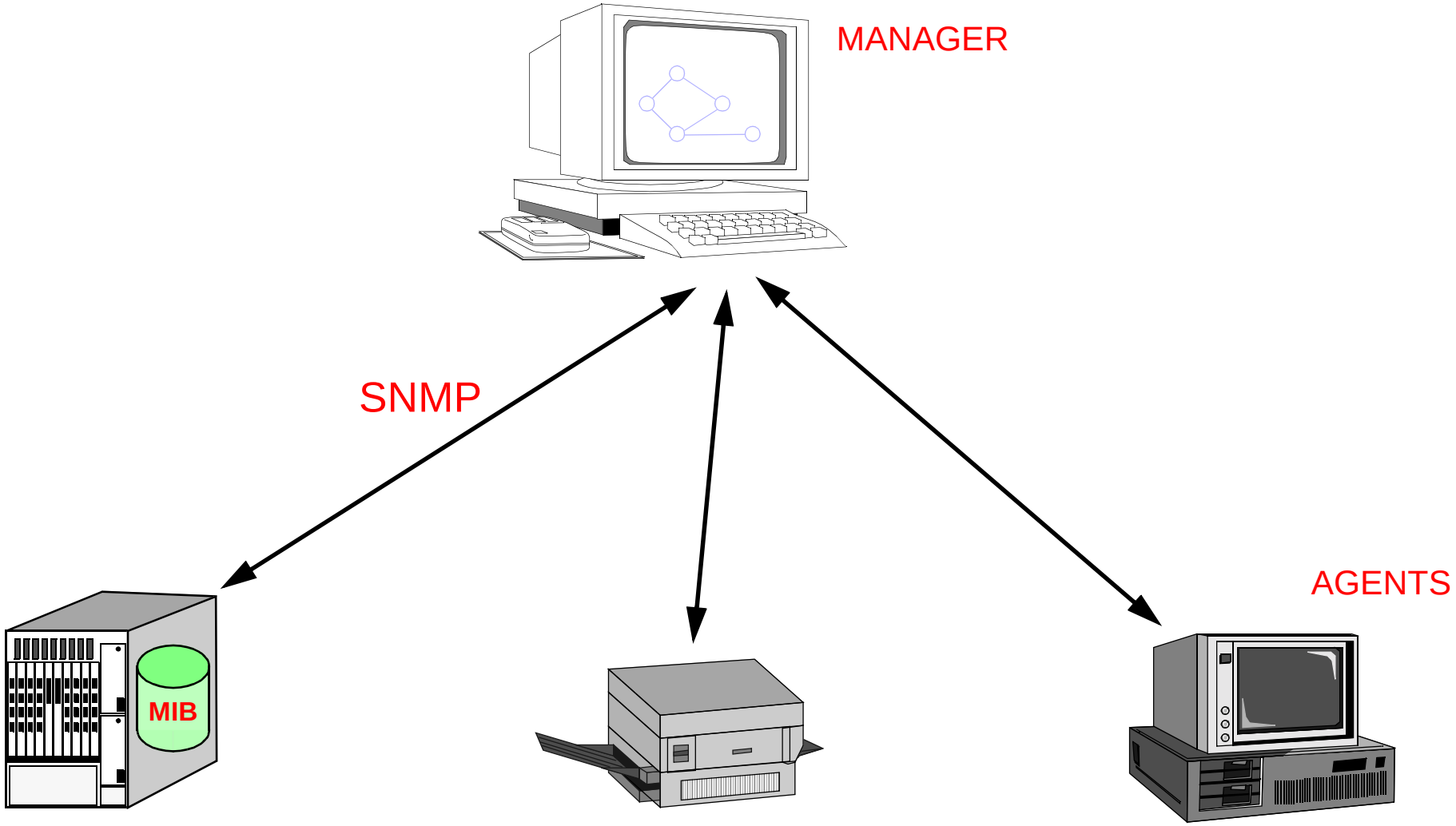
UBIQUITY
- PCs AND CRAYs

INCLUSION OF MANAGEMENT SHOULD BE INEXPENSIVE
- SMALL CODE
- LIMITED FUNCTIONALITY

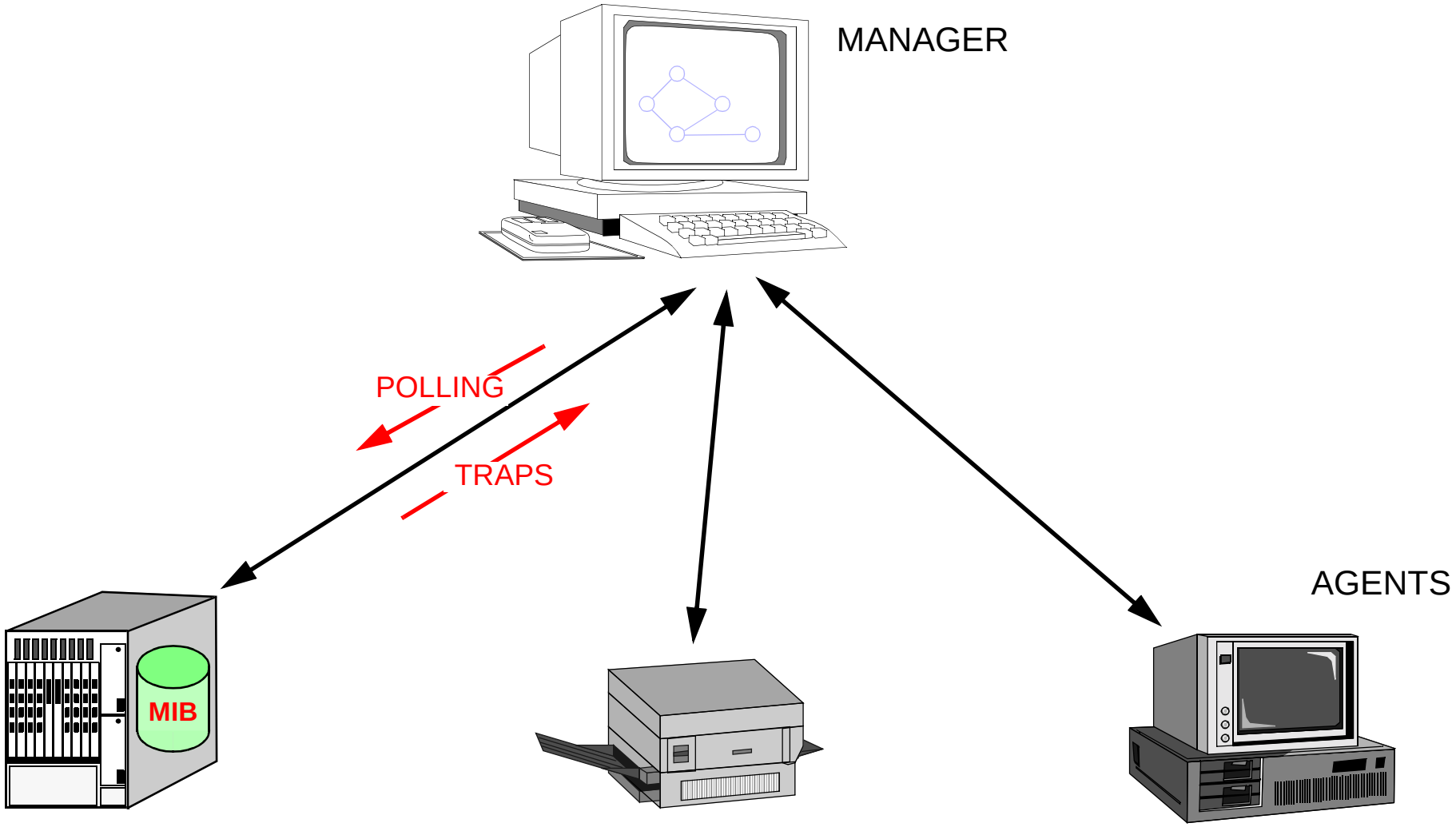
MANAGEMENT EXTENSIONS SHOULD BE POSSIBLE
- NEW MIBs

MANAGEMENT SHOULD BE ROBUST
- CONNECTIONLESS TRANSPORT

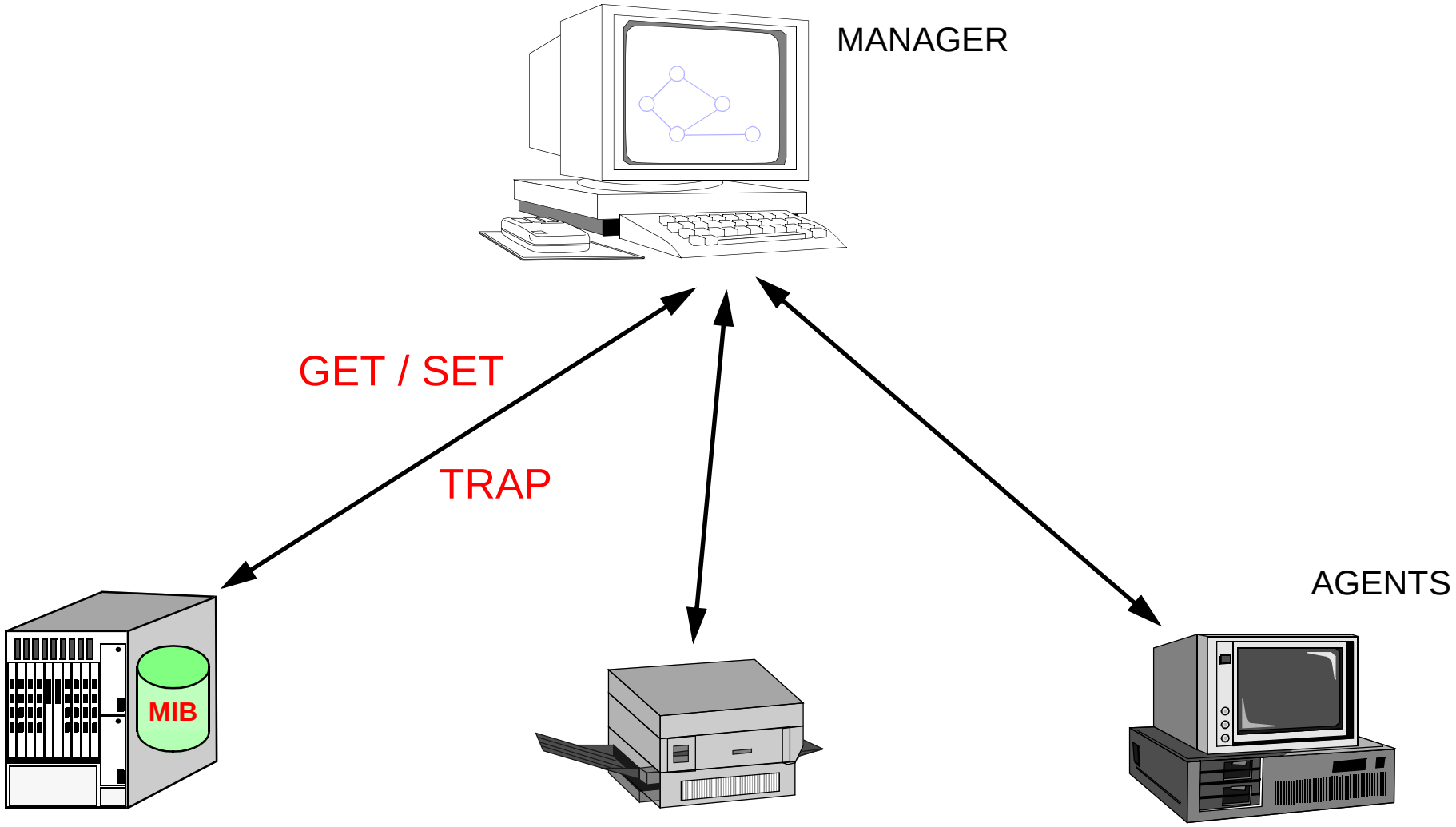
PRINCIPLE OPERATION



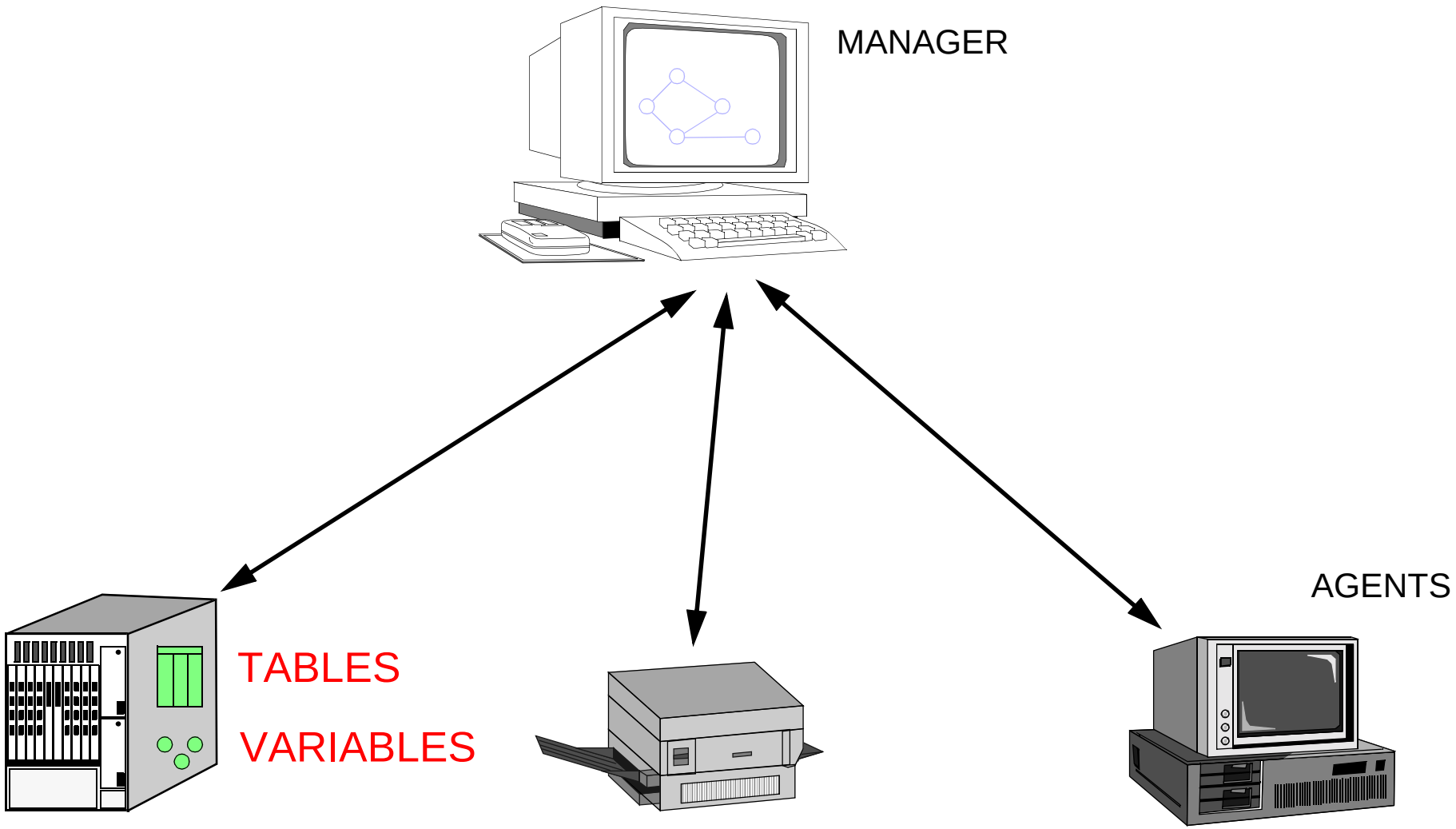
PRINCIPLE OPERATION



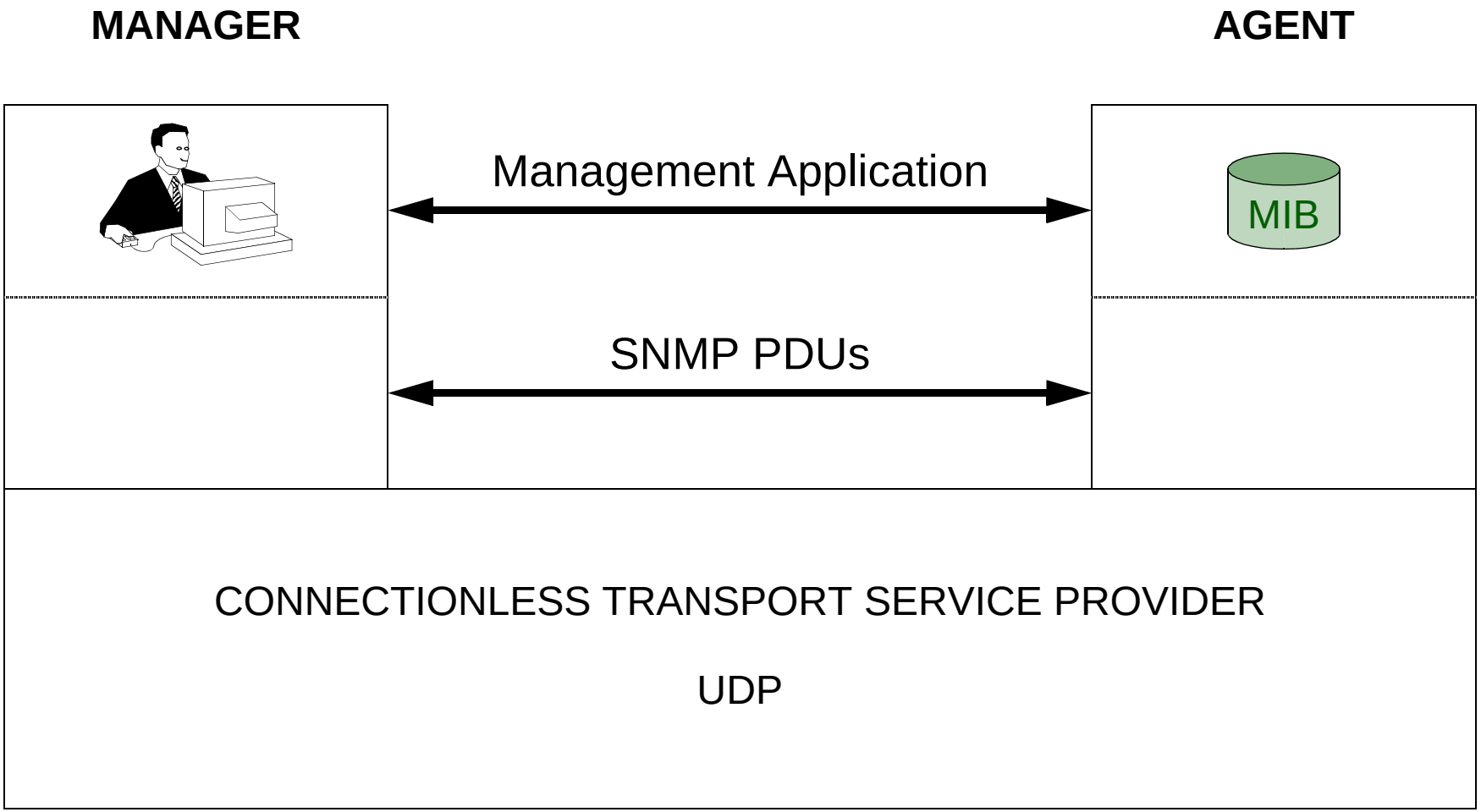
PRINCIPLE OPERATION



PRINCIPLE OPERATION



SNMP STRUCTURE



STANDARDS

SMI

- STRUCTURE OF MANAGEMENT INFORMATION
- RFC 1155

MIB-II

- MANAGEMENT INFORMATION BASE
- RFC 1213
- A LARGE NUMBER OF ADDITIONAL MIBs EXIST

SNMP

- SIMPLE NETWORK MANAGEMENT PROTOCOL
- RFC 1157
- NAME IS USED IN A MORE GENERAL SENSE

NEWER VERSIONS: SNMPv2 & SNMPv3

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)

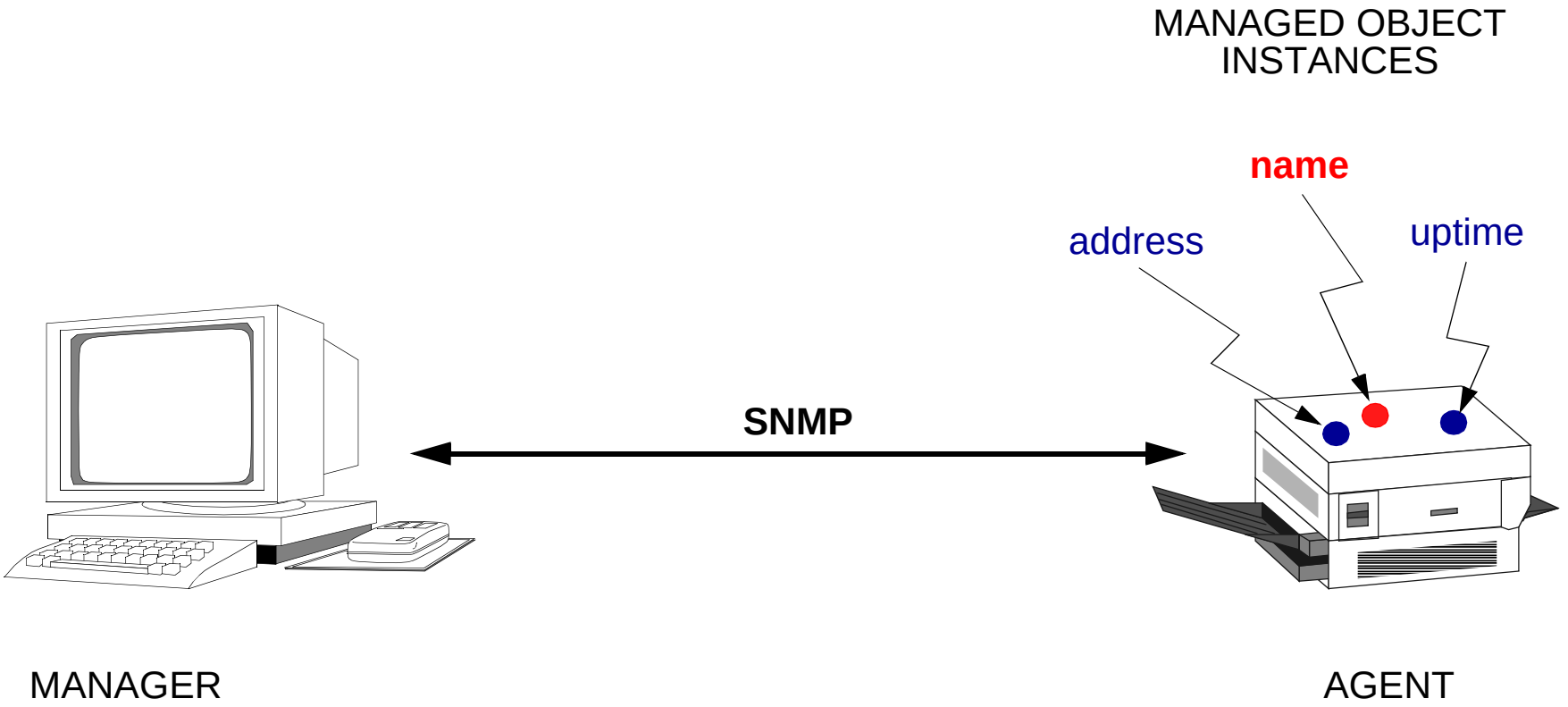
THE SNMP PROTOCOL CAN ONLY EXCHANGE
(A LIST OF) SCALARS

DEFINED IN TERMS OF ASN.1 CONSTRUCTS

SMI: DATA TYPES FOR SCALARS

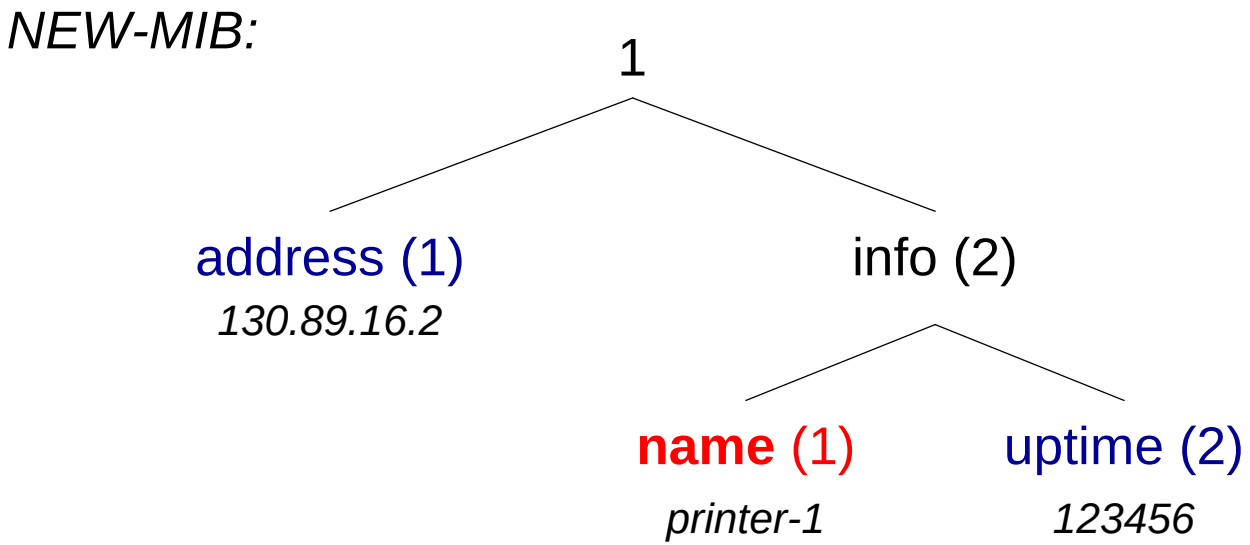
	SMIv1	SMIv2
SIMPLE TYPES:	INTEGER	INTEGER
	OCTET STRING	OCTET STRING
	OBJECT IDENTIFIER	OBJECT IDENTIFIER
	-	Integer32
APPLICATION-WIDE TYPES:	-	Unsigned32
	Gauge	Gauge32
	Counter	Counter32
	-	Counter64
	TimeTicks	TimeTicks
	IpAddress	IpAddress
	Opaque	Opaque
	NetworkAddress	-
PSEUDO TYPES:	-	BITS

EXAMPLE OF SCALAR OBJECTS



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

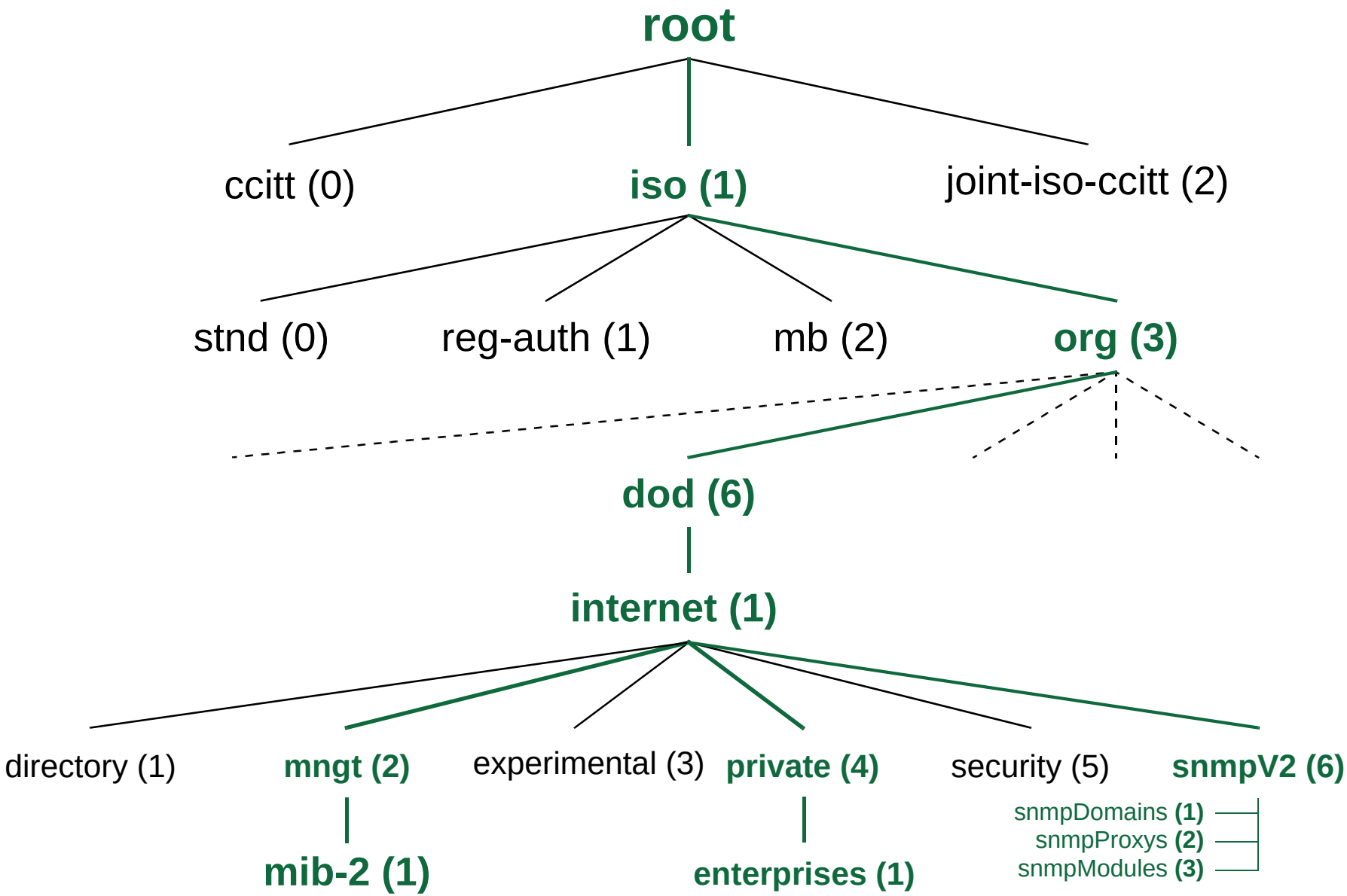
Object ID = 1.2.1
Object Instance = 1.2.1.0
Value of Instance = *printer-1*

- 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:

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

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

STATUS
- current
- deprecated
- obsolete

DESCRIPTION ""

OBJECT TYPE DEFINITION - EXAMPLE

```
-- Definition of address

address OBJECT-TYPE
SYNTAX          IpAddress
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION     "The Internet address of this system"
::= {NEW-MIB  1}
```

DEFINITION OF NON-LEAF 'OBJECTS'

Name **OBJECT IDENTIFIER ::= {...}**

EXAMPLE:
info **OBJECT IDENTIFIER ::= {**NEW-MIB 2**}**

ALTERNATIVE CONSTRUCT: OBJECT IDENTITY

EXAMPLE:

```
info        OBJECT-IDENTITY
STATUS      current
DESCRIPTION "The node under which future scalar objects
             should be registered"
::= {NEW-MIB 2}
```

DEFINITION OF A MIB

NEW-MIB **DEFINITIONS ::=**
BEGIN

import statement(s)
module identity definition

definition of all node and leaf objects

definition of implementation requirements

END

MODULE IDENTITY - EXAMPLE

```
newMibModule MODULE-IDENTITY
        LAST-UPDATED "200104041200Z"
        ORGANIZATION "UT-TSS"
        CONTACT-INFO "
                TSS Group
                University of Twente
                POBox 217
                7500 AE Enschede
                The Netherlands
                Email: simpleweb@simpleweb.org "
        DESCRIPTION
                "Experimental MIB for demo purposes"
        ::= { enterprises ut(785) 7 }
```

IMPORT STATEMENT - EXAMPLE

IMPORTS

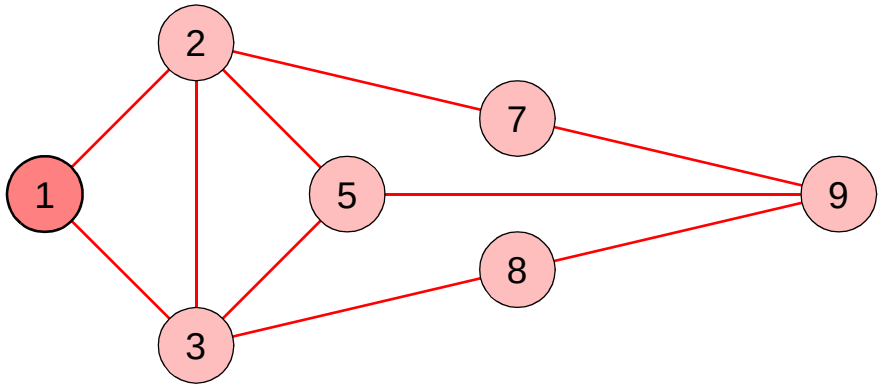
MODULE-IDENTITY, OBJECT-TYPE,
TimeTicks, enterprises

FROM SNMPv2-SMI;

TABLES

EXAMPLE: ROUTING TABLE

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



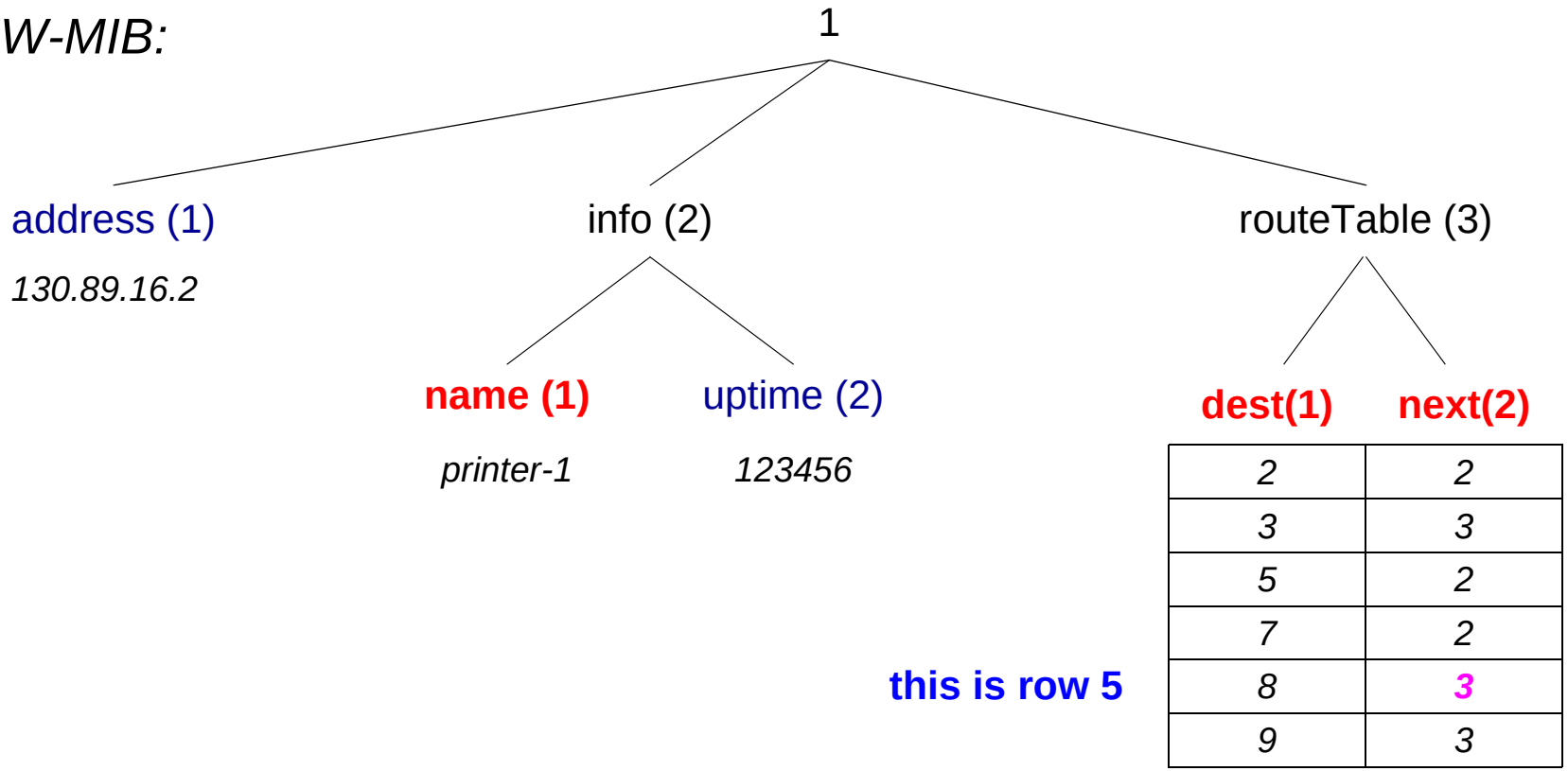
TO RETRIEVE INDIVIDUAL TABLE ENTRIES

EACH ENTRY SHOULD GET A NAME

NAMING OF TABLE ENTRIES - I

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

NEW-MIB:



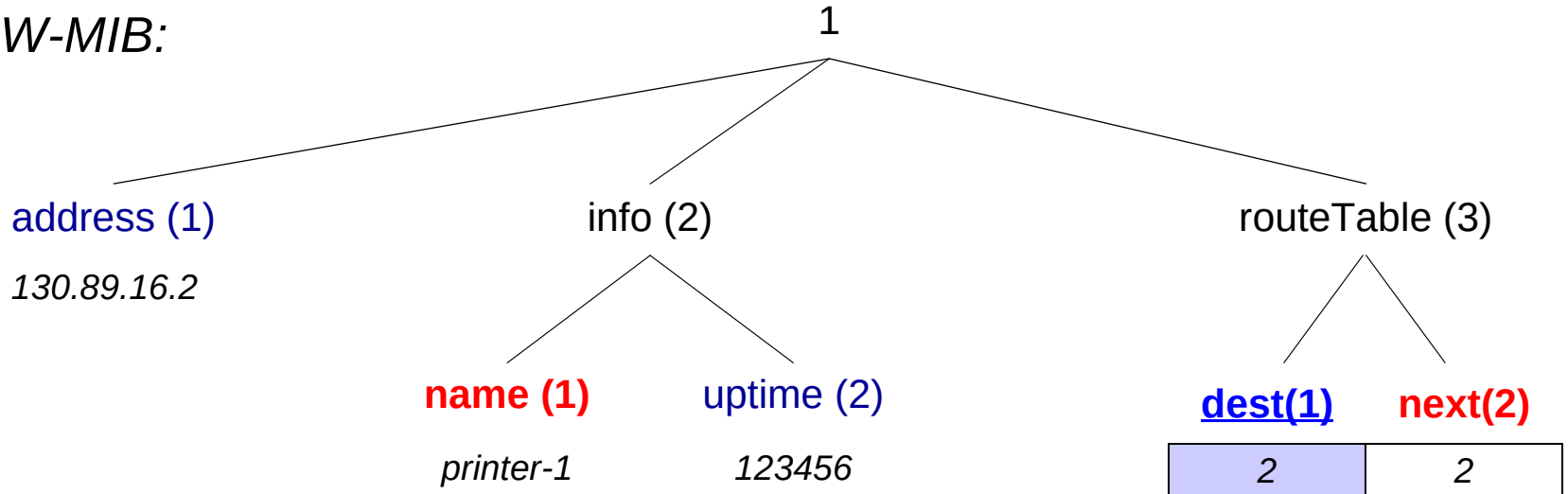
dest(1)	next(2)
2	2
3	3
5	2
7	2
8	3
9	3

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

NEW-MIB:



dest(1)	next(2)
2	2
3	3
5	**2**
7	2
8	3
9	3

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

TABLE INDEXING

GENERAL SCHEME

X.C.I

OID of Table Column number Index value

EXAMPLES:

OID of Table = 1.3

1.3.1.5 => *5*
1.3.2.5 => *2*
1.3.1.9 => *9*
1.3.2.9 => *3*
1.3.2.7 => *2*
1.3.1.1 => *entry does not exist*
1.3.2.1 => *entry does not exist*

TABLE INDEXING - NON-INTEGER INDEX

AN INDEX NEED NOT BE AN INTEGER

routeTable (3)

dest (1)	next (2)
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

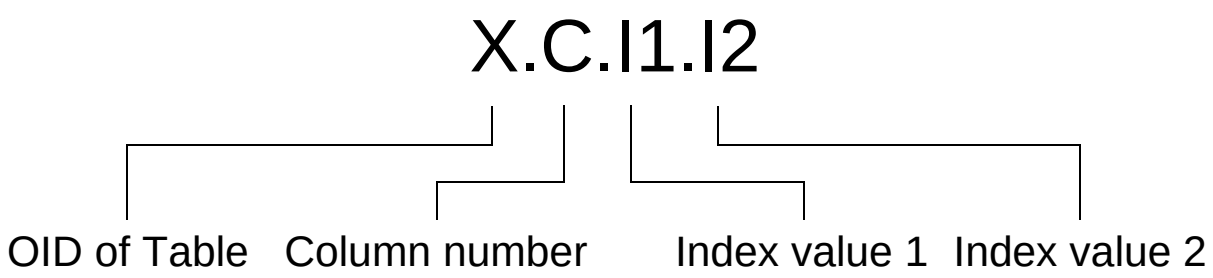


TABLE INDEXING - MULTIPLE INDEX FIELDS: EXAMPLE

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*

TABLE DEFINITION

-- Definition of the route table

```
routeTable      OBJECT-TYPE
SYNTAX          SEQUENCE OF RouteEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "This entity's routing table"
::= {NEW-MIB  3}

routeEntry      OBJECT-TYPE
SYNTAX          RouteEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     "A route to a particular destination"
INDEX           {dest, policy}
::= {routeTable  1}
```

TABLE DEFINITION (cont. 1)

```
RouteEntry::=
SEQUENCE{
        dest  ipAddress,
        policyINTEGER,
        next  ipAddress
        }
```

TABLE DEFINITION (cont. 2)

```
dest OBJECT-TYPE
SYNTAX      ipAddress
ACCESS      read-only
STATUS      current
DESCRIPTION"The address of a particular destination"
::= {route-entry  1}

policy OBJECT-TYPE
SYNTAX      INTEGER {
             costs(1)  -- lowest delay
             reliability(2) }  -- highest reliability
ACCESS      read-only
STATUS      current
DESCRIPTION"The routing policy to reach that destination"
::= {route-entry  2}

next OBJECT-TYPE
SYNTAX      ipAddress
ACCESS      read-write
STATUS      current
DESCRIPTION"The internet address of the next hop"
::= {route-entry  3}
```

DEFINITION OF NEW TYPES

TEXTUAL CONVENTIONS

TO REFINE SEMANTICS OF EXISTING TYPES

EXAMPLE:

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

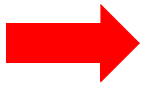
TEXTUAL CONVENTIONS

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

- Inet-Address...

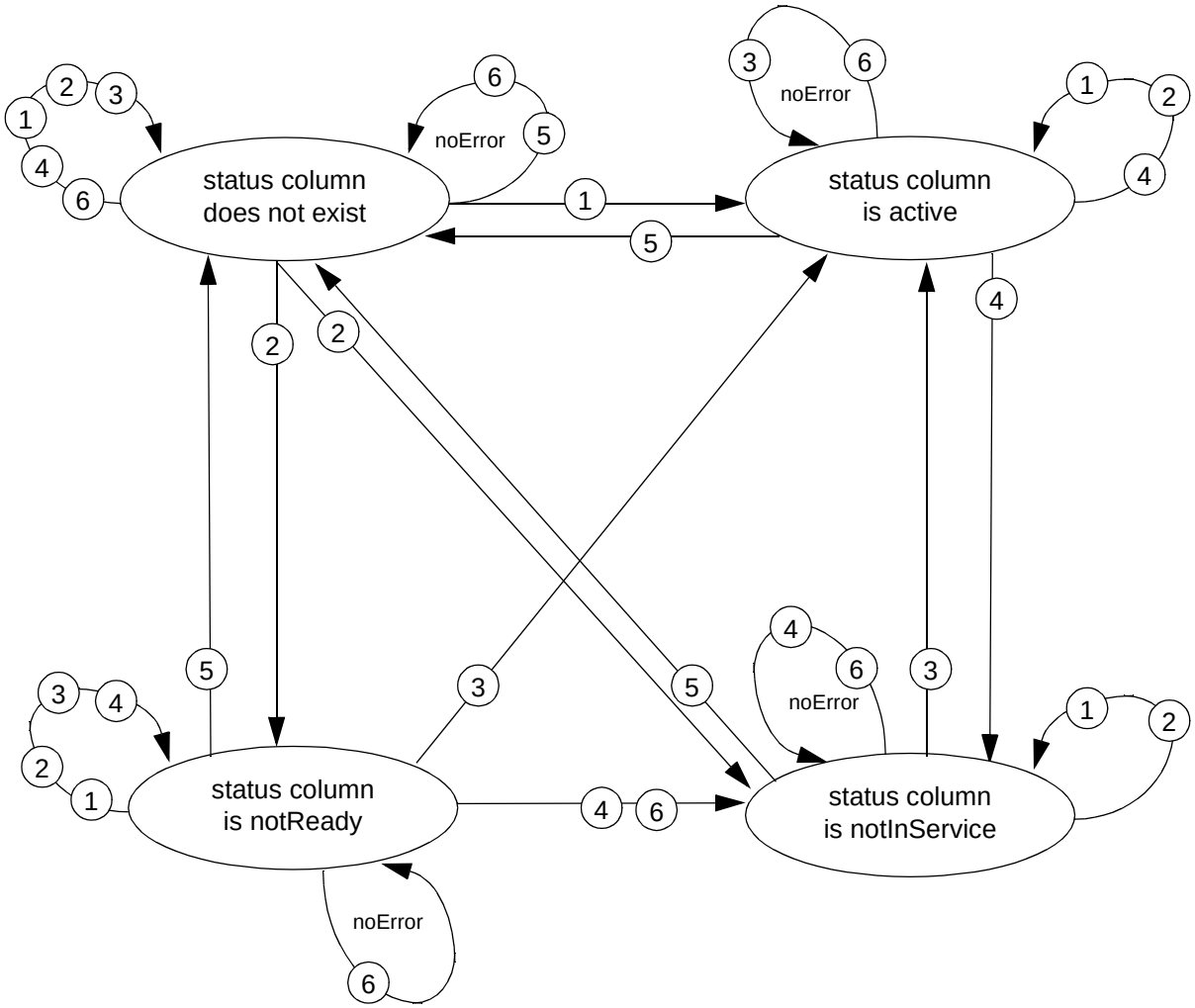
ROW-STATUS TEXTUAL CONVENTION

USED TO CHANGE TABLE ROWS



TO:	VIA:	STATUS:
130.89.16.4	130.89.1.1	ACTIVE
130.89.17.6	130.89.1.1	NOT READY
130.89.18.2	130.89.1.4	ACTIVE
130.89.18.7	130.89.1.4	ACTIVE

ROW-STATUS - STATE DIAGRAM



1	set status column to createAndGo
2	set status column to createAndWait
3	set status column to active
4	set status column to notInService
5	set status column to destroy
6	set any other column to some value

4 6 — 4 or 6

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}
```

DEFINITION OF IMPLEMENTATION REQUIREMENTS

THE MODULE-COMPLIANCE CONSTRUCT
DEFINES IMPLEMENTATION REQUIREMENTS FOR AGENTS

---

newMibCompliance **MODULE-COMPLIANCE**
STATUS ...
DESCRIPTION ...

MODULE 1	**MODULE** ... **MANDATORY-GROUPS** ... **GROUP** ... **OBJECT** ...
MODULE n	

::= { ... }

OBJECT GROUP CONSTRUCT

TO DEFINE A SET OF RELATED OBJECT TYPES

EXAMPLE:

```
newMibScalarGroup OBJECT-GROUP
   OBJECTS { address, name, uptime }
   STATUS current
   DESCRIPTION "The collection of scalar objects."
::= { demoGroups 1 }
```

SMIng

PROBLEMS WITH SMIvx

- 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, ...

SMIng

TO RESOLVE THESE PROBLEMS
A NEW SMI IS BEING DEFINED

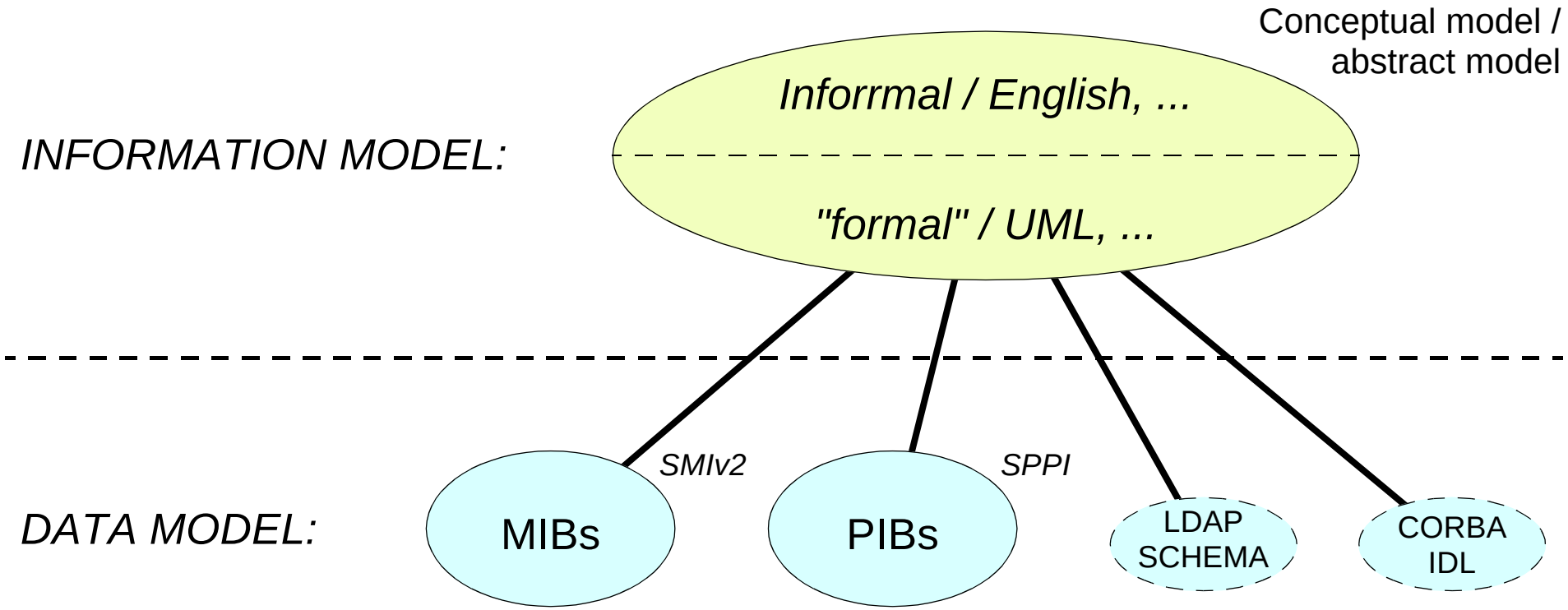
SMI next generation (ng)

STARTED AS IRTF NMRG ACTIVITY

IS NOW IETF WG

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

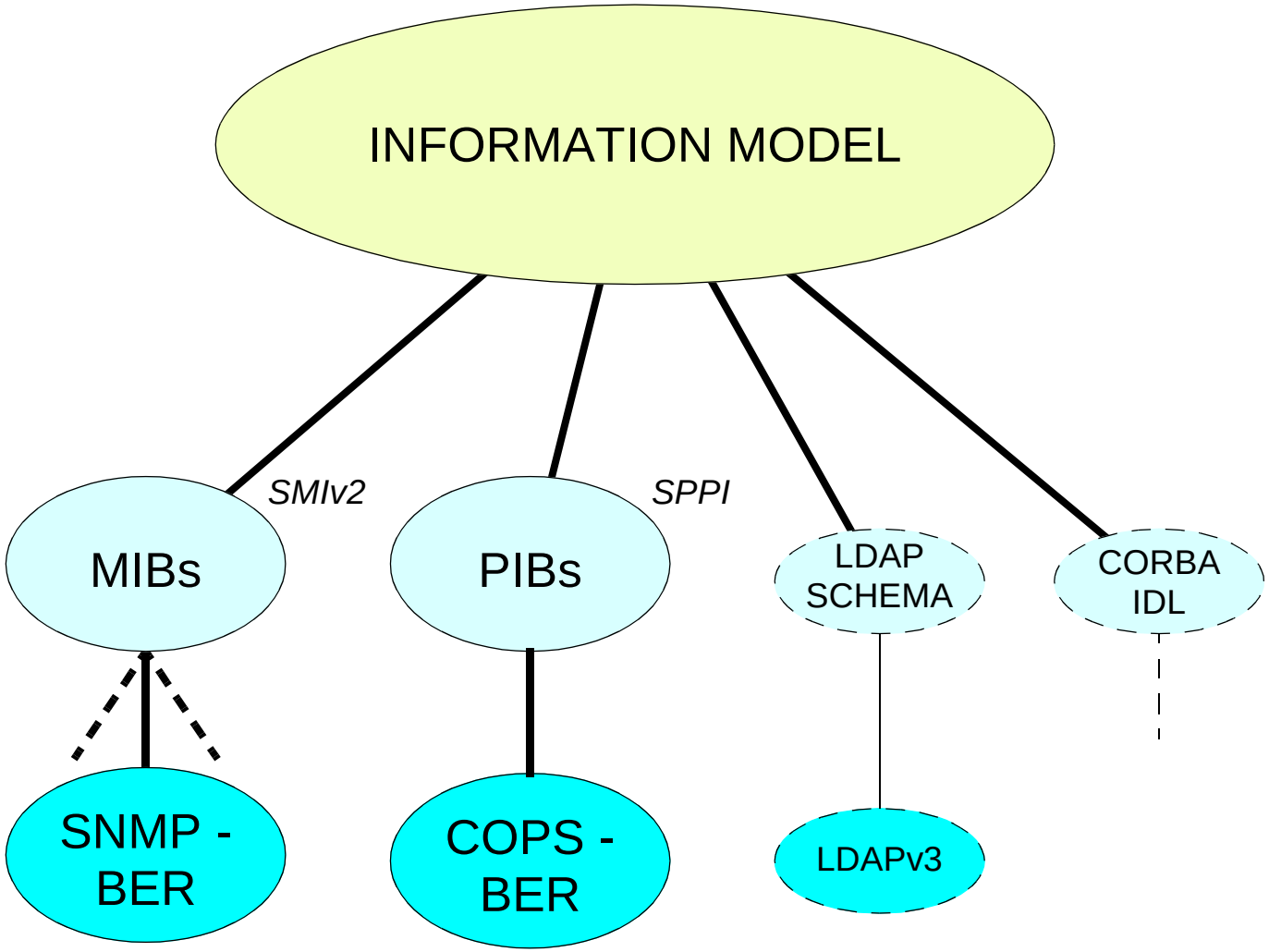
DATA VERSUS INFORMATION MODEL



DATA MODEL & "TRANSFER" PROTOCOL

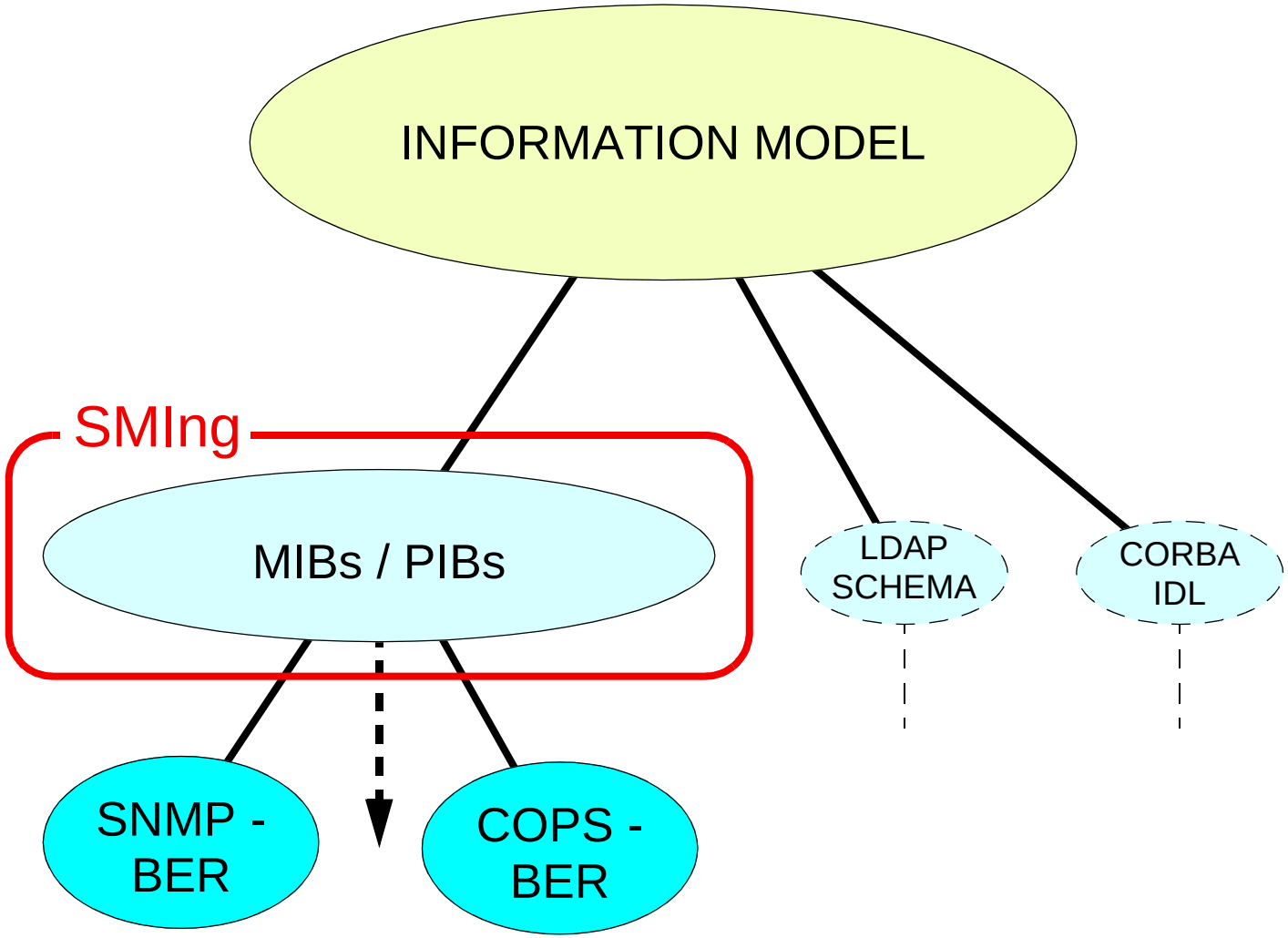
DATA MODEL:

"TRANSFER" PROTOCOL:



ROLE OF SMIng

DATA MODEL:



MIBs

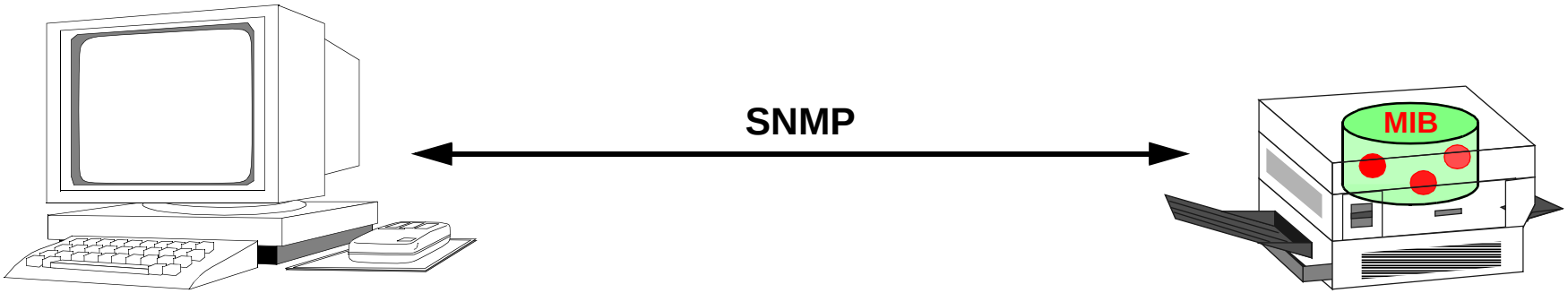
MANAGEMENT INFORMATION BASES

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

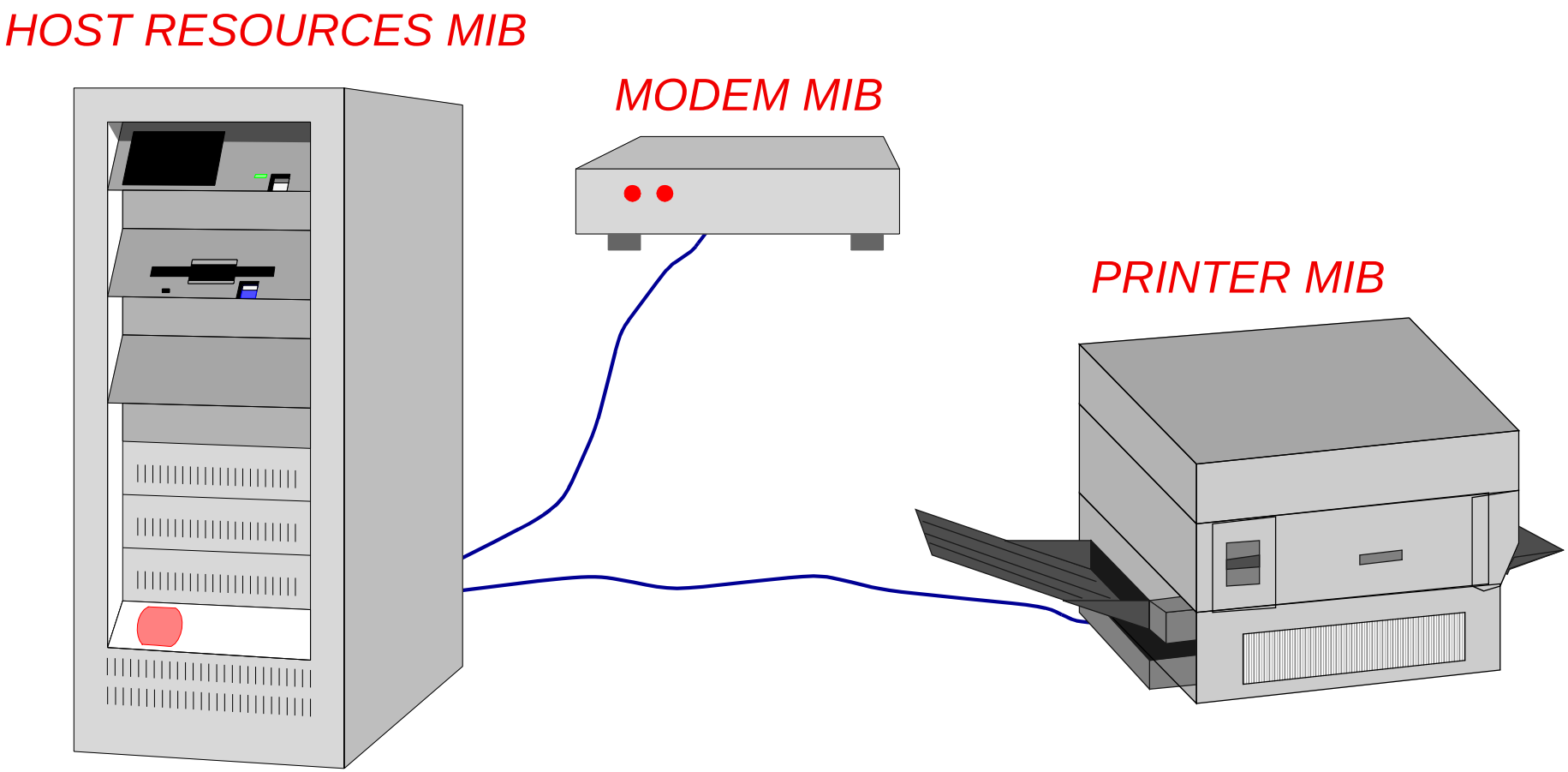
MODULARITY

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

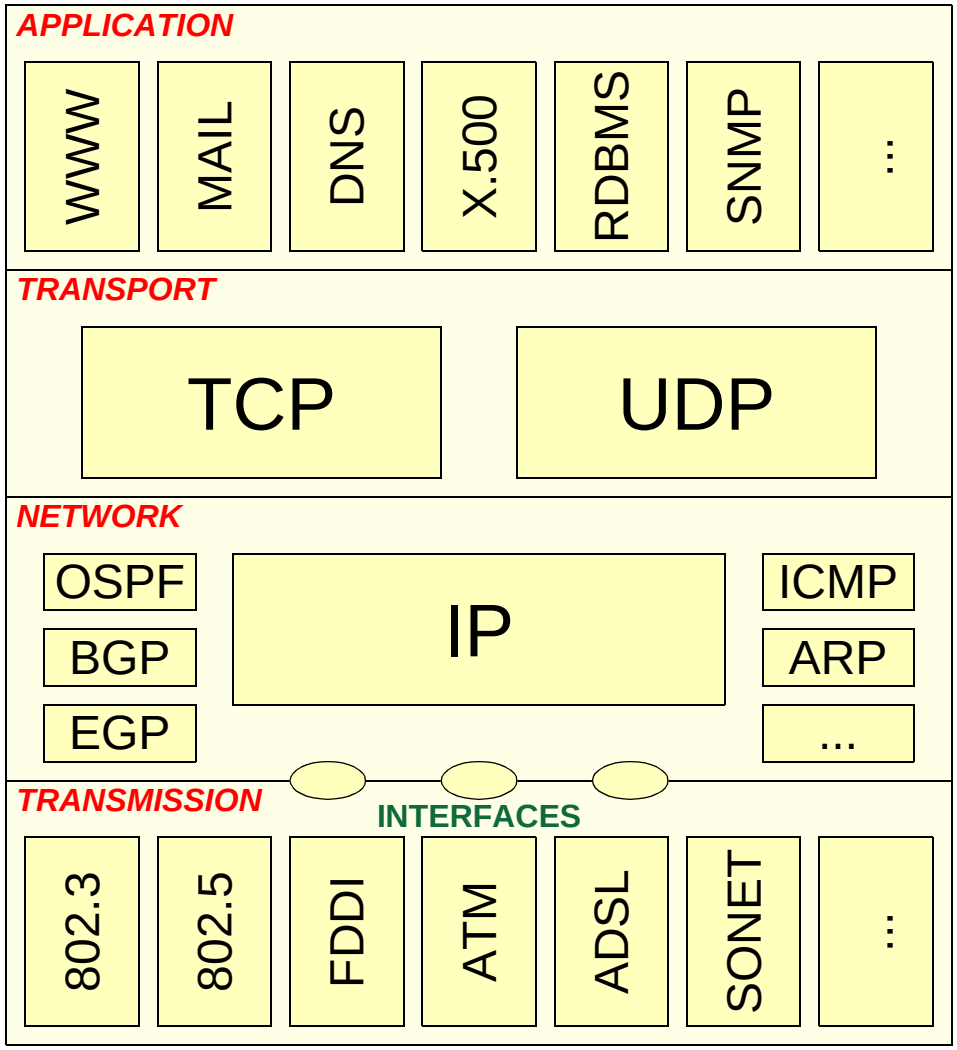
MODULES

- 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

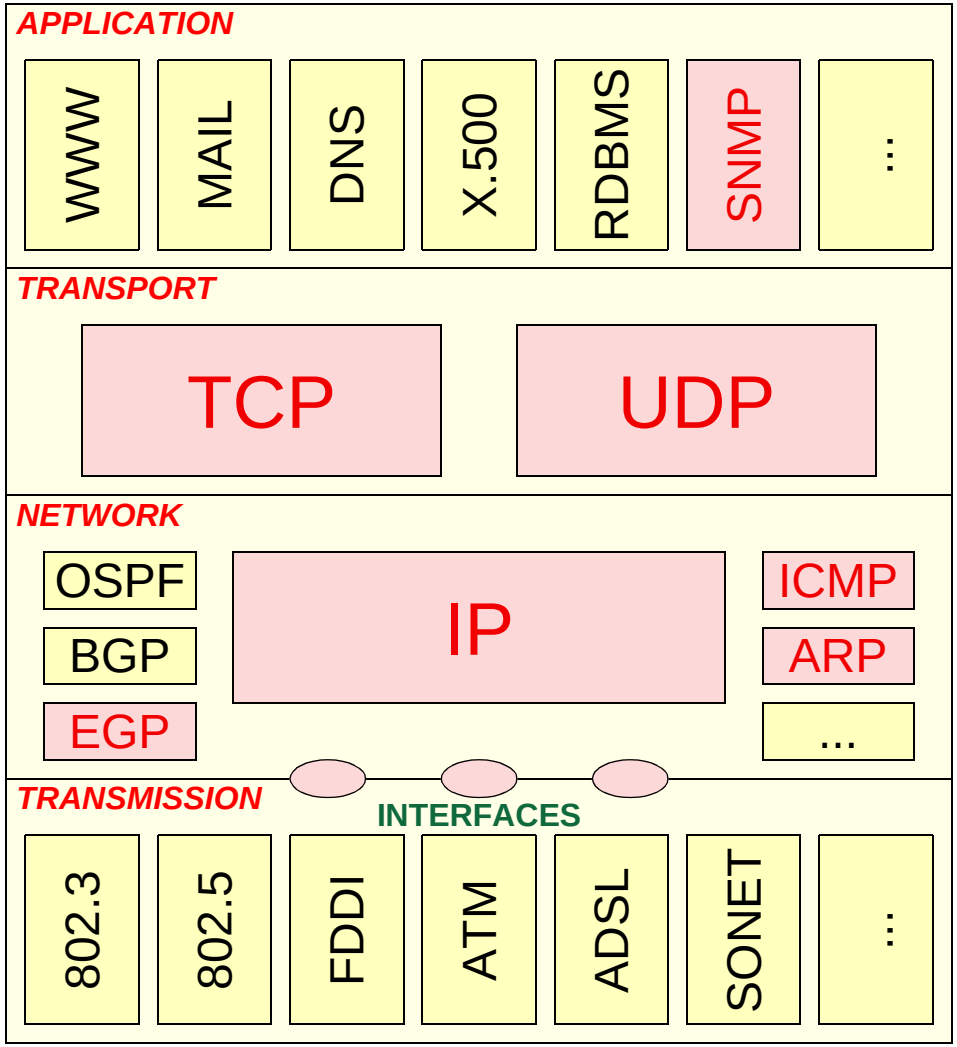
HARDWARE MIBS



PROTOCOL MIBS



PROTOCOL MIBS - EXAMPLE: MIB-II



HARDWARE SPECIFIC MIBs

Title	RFC	STATUS
Host Resources MIB	2790	D
Entity MIB	2737	P
Job Monitoring MIB	2707	I
Printer	1759	P
Modem	1696	P
Parallel printer-like Hardware	1660	D
RS-232-like Hardware	1659	D
Character Stream Devices	1658	D
UPS	1628	P

TRANSMISSION MIBs

Title	RFC	STATUS
UNI/NNI Multilink Frame Relay function	3020	P
Frame Relay/ATM PVC Service Interworking Function	2955	P
Frame Relay Service	2954	P
Inverted Stack Table Extension to the Interfaces Group	2864	P
Interfaces Group	2863	D
Fabric Element in Fibre Channel Standard	2837	P
NBMA Next Hop Resolution Protocol (NHRP)	2677	P
Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions	2674	P
Radio Frequency MIB for MCNS/DOCSIS compliant RF interfaces	2670	P
Cable Device MIB for DOCSIS compliant Cable Modems and Cable Modem Termination Systems	2669	P
IEEE 802.3 Medium Attachment Units (MAUs)	2668	P
Object Identifiers for Identifying Ethernet Chip Sets	2666	I

Title	RFC	STATUS
Ethernet-like Interface Types	2665	P
ADSL Lines	2662	P
SONET/SDH Interface Type	2558	P
ATM Management	2515	P
Textual Conventions and OBJECT-IDENTITIES for ATM Management	2514	P
DS3/E3 Interface Type	2496	P
DS1, E1, DS2 and E2 Interface Types	2495	P
DS0 and DS0 Bundle Interface Type	2494	P
Classical IP and ARP Over ATM (IPOA)	2320	P
IEEE 802.12 Repeater Devices	2266	P
Dial Control	2128	P
ISDN	2127	P
Frame Relay DTEs	2115	D
IEEE 802.3 Repeater Devices	2108	P
Data Link Switching	2024	P
IEEE 802.12 Interfaces	2020	P
IEEE 802.5 Station Source Routing	1749	P

Title	RFC	STATUS
IEEE 802.5	1748	D
SMDS	1694	D
Source Routing Bridges	1525	P
FDDI	1512	P
Bridges	1493	D
Bridge Network Control Protocol of PPP	1474	P
IP Network Control Protocol of PPP	1473	P
Security Protocols of PPP	1472	P
Link Control Protocol of PPP	1471	P
Multiprotocol Interconnect over X.25	1461	P
X.25 Packet Layer	1382	P
X.25 LAPB	1381	P

NETWORK LAYER MIBs

Title	RFC	STATUS
IPv6 MIB for The Multicast Listener Discovery Protocol	3019	P
Protocol Independent Multicast MIB for IPv4	2934	E
Internet Group Management Protocol MIB	2933	P
IPv4 Multicast Routing MIB	2932	P
Textual Conventions for Internet Network Addresses	2851	P
Definitions of MO for the Virtual Router Redundancy Protocol	2787	P
IP Tunnel MIB	2667	P
MIB for IPv6: ICMPv6 Group	2466	P
MIB for IPv6: Textual Conventions and General Group	2465	P
Definitions of MO for Multicast over UNI 3.0/3.1 based ATM Networks	2417	P
Integrated Services - Guaranteed Service Ext.	2214	P
Integrated Services	2213	P
RSVP	2206	P
IP Forwarding Table	2096	P

Title	RFC	STATUS
IP MIB	2011	P
IP Mobility Support	2006	P
OSPF Version 2	1850	D
RIP Version 2 MIB Extension	1724	D
BGP Version 4	1657	D
Identification MIB	1414	P
BGP Version 3	1269	P
MIB-II	1213	S

TRANSPORT LAYER MIBs

Title	RFC	STATUS
Real-Time Transport Protocol	2959	P
IP Version 6 MIB for the User Datagram Protocol	2454	P
IP Version 6 MIB for the Transmission Control Protocol	2452	P
User Datagram Protocol (UDP)	2013	P
Transmission Control Protocol (TCP)	2012	P

APPLICATION LAYER MIBs

Title	RFC	STATUS
MIB for the PINT Services Architecture	3055	P
Mail Monitoring MIB	2789	P
Network Services Monitoring	2788	P
RADIUS Accounting Server MIB	2621	I
RADIUS Accounting Client MIB	2620	I
RADIUS Authentication Server MIB	2619	P
RADIUS Authentication Client MIB	2618	P
Directory Server Monitoring MIB	2605	P
Definitions of Managed Objects for WWW Services	2594	P
Application Management MIB	2564	P
Definitions of System-Level Managed Objects for Applications	2287	P
SNMPv2 MIB	1907	P
RDBMS MIB	1697	P
DNS Resolver MIB Extensions	1612	P
DNS Server MIB Extensions	1611	P

REMOTE MONITORING AND MEASUREMENT

Title	RFC	STATUS
Remote Network Monitoring (RMON) MIB	2819	S
Traffic Flow Measurement: Meter MIB	2720	P
RMON MIB Extensions for Switched Networks Version 1.0	2613	P
RMON Version 2	2021	P
Token Ring extensions to RMON	1513	P

DISTRIBUTED MANAGEMENT

Title	RFC	STATUS
Notification Log MIB	3014	P
Expression MIB	2982	P
Event MIB	2981	P
Remote Ping, Traceroute, and Lookup Operations	2925	P
Delegation of Management Scripts	2592	P
Scheduling Management Operations	2591	P

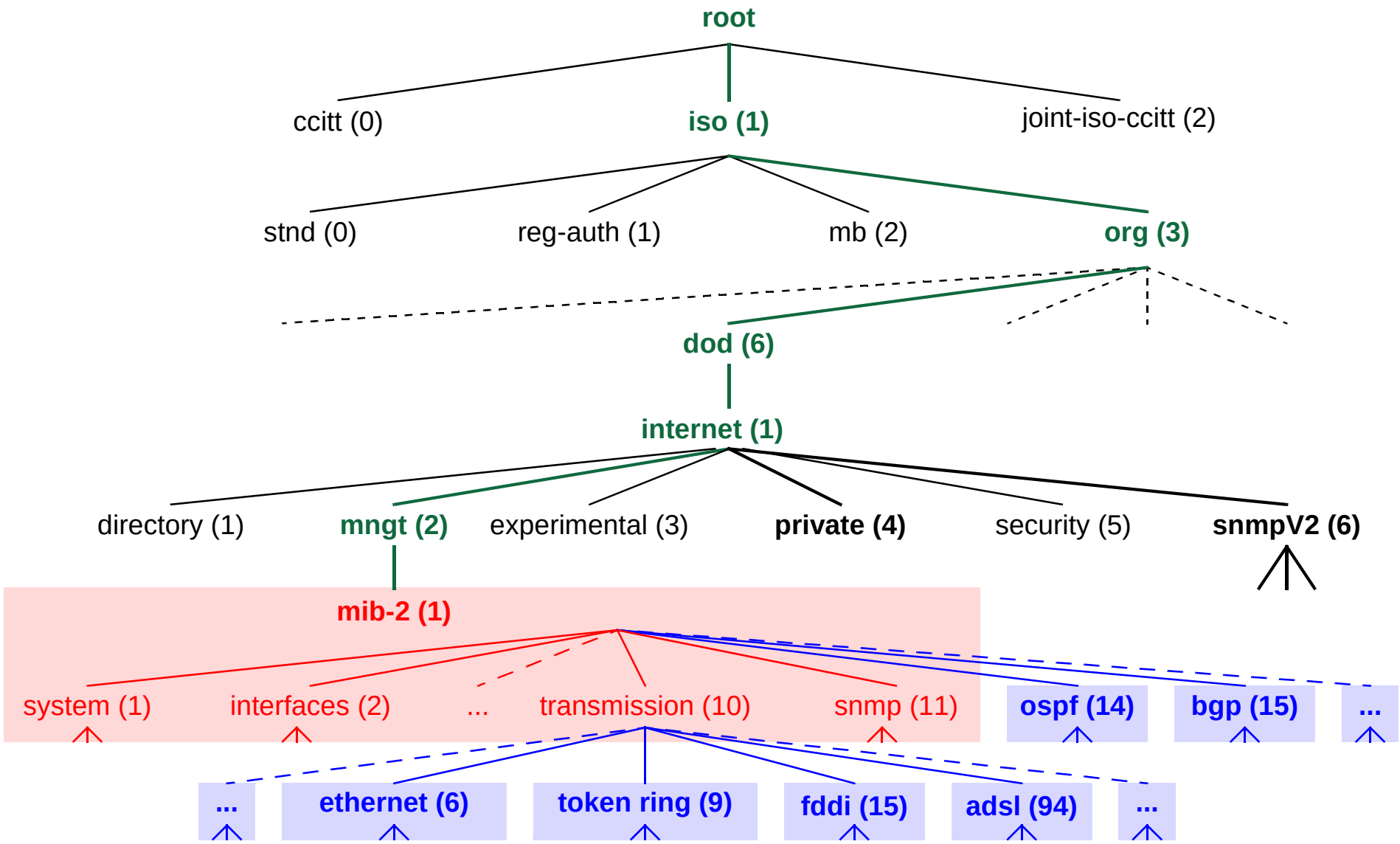
VENDOR SPECIFIC MIBs

Title	RFC	STATUS
APPN/HPR in IP Networks	2584	P
TN3270E Response Time Collection	2562	P
TN3270E	2561	P
Extended Border Node	2457	P
APPN TRAPS	2456	P
APPN	2455	P
HPN	2238	P
DLUR	2232	P
APPC	2051	P
TCP/IPX Connection	1792	E
SNA Data Link Control (SDLC)	1747	P
Appletalk	1742	P
SNA NAUs	1666	P
DECNET Phase IV	1559	D

MISCELLANY

Title	RFC	STATUS
Common Open Policy Service (COPS) Protocol Clients	2940	P
Physical Topology	2922	I
Service Level Agreements Performance Monitoring	2758	E
Definitions of Managed Objects for Extensible SNMP Agents	2742	P
Collection and Storage of Accounting Information for CO Networks	2513	P
Accounting Information for ATM Networks	2512	P
Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals	2493	P
Techniques for managing asynchronously generated alerts	1224	E

NAMING OF MIBs



RELATION BETWEEN MIBs - 1

	MIB-II	HOST	REPEATER	BRIDGE	RMON
INTERFACE STATISTICS	✓				
IP, TCP & UDP STATISTICS	✓				
SNMP STATISTICS	✓				
HOST JOB COUNTS		✓			
HOST FILE SYSTEM INFORMATION		✓			
LINK TESTING			✓	✓	
NETWORK TRAFFIC STATISTICS			✓	✓	✓
TABLE WITH ALL MAC ADDRESSES			✓		✓
STATISTICS PER HOST			✓		✓

RELATION BETWEEN MIBs - 2

	MIB-II	HOST	REPEATER	BRIDGE	RMON
HISTORICAL STATISTICS					✓
SPANNING TREE PERFORMANCE				✓	
WIDE AREA LINK PERFORMANCE				✓	
TRESHOLDS FOR ANY VARIABLE					✓
CONFIGURABLE STATISTICS					✓
TRAFFIC MATRIX WITH ALL NODES					✓
‘HOST TOP N’ INFORMATION					✓
PACKET / PROTOCOL ANALYSIS					✓
DISTRIBUTED LOGGING					✓

MIB-II

DEFINES THE VARIABLES TO MANAGE THE TCP/IP PROTOCOL STACK

170 VARIABLES

RFC 1213
SMIv1

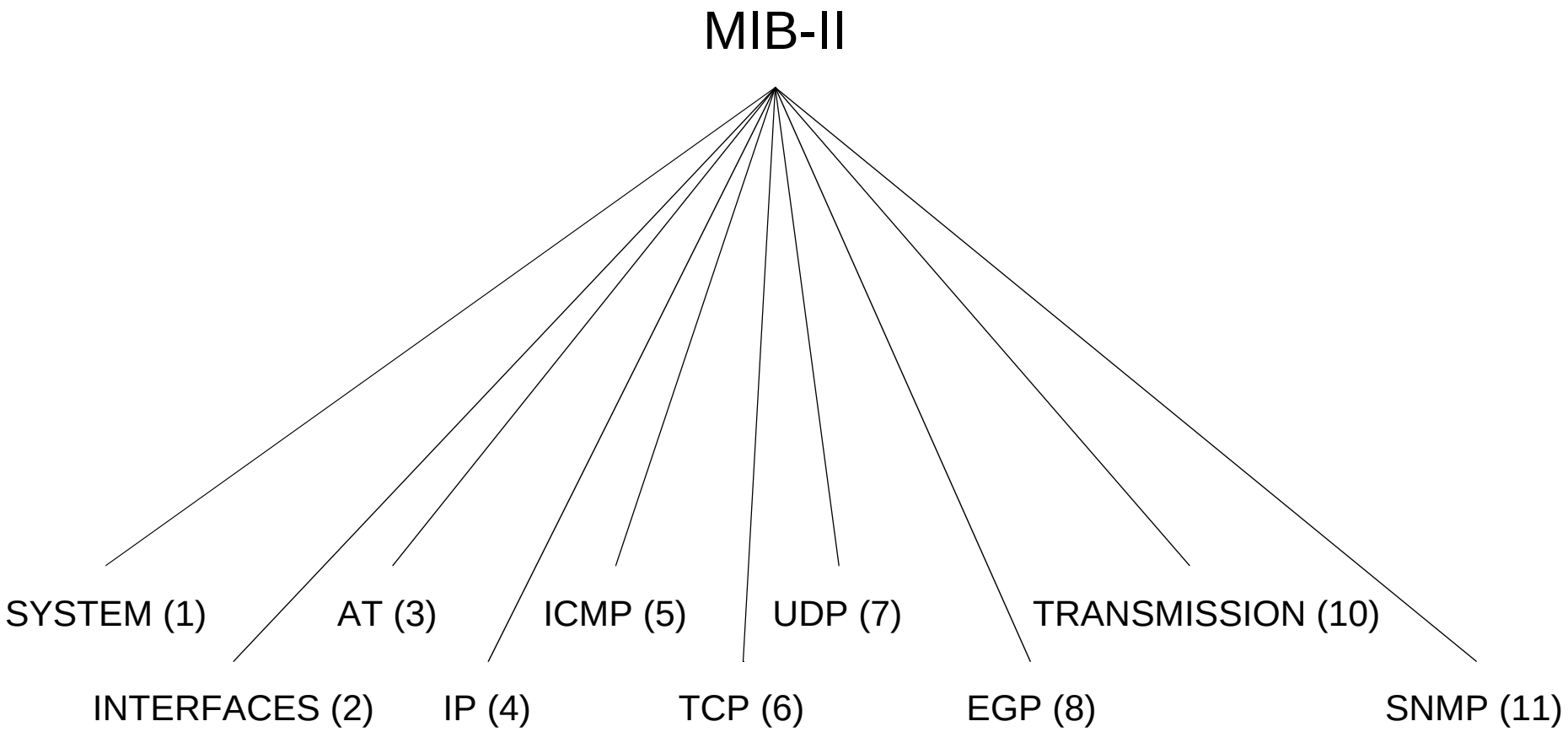
ENHANCEMENT OF MIB-I

RFC 1156

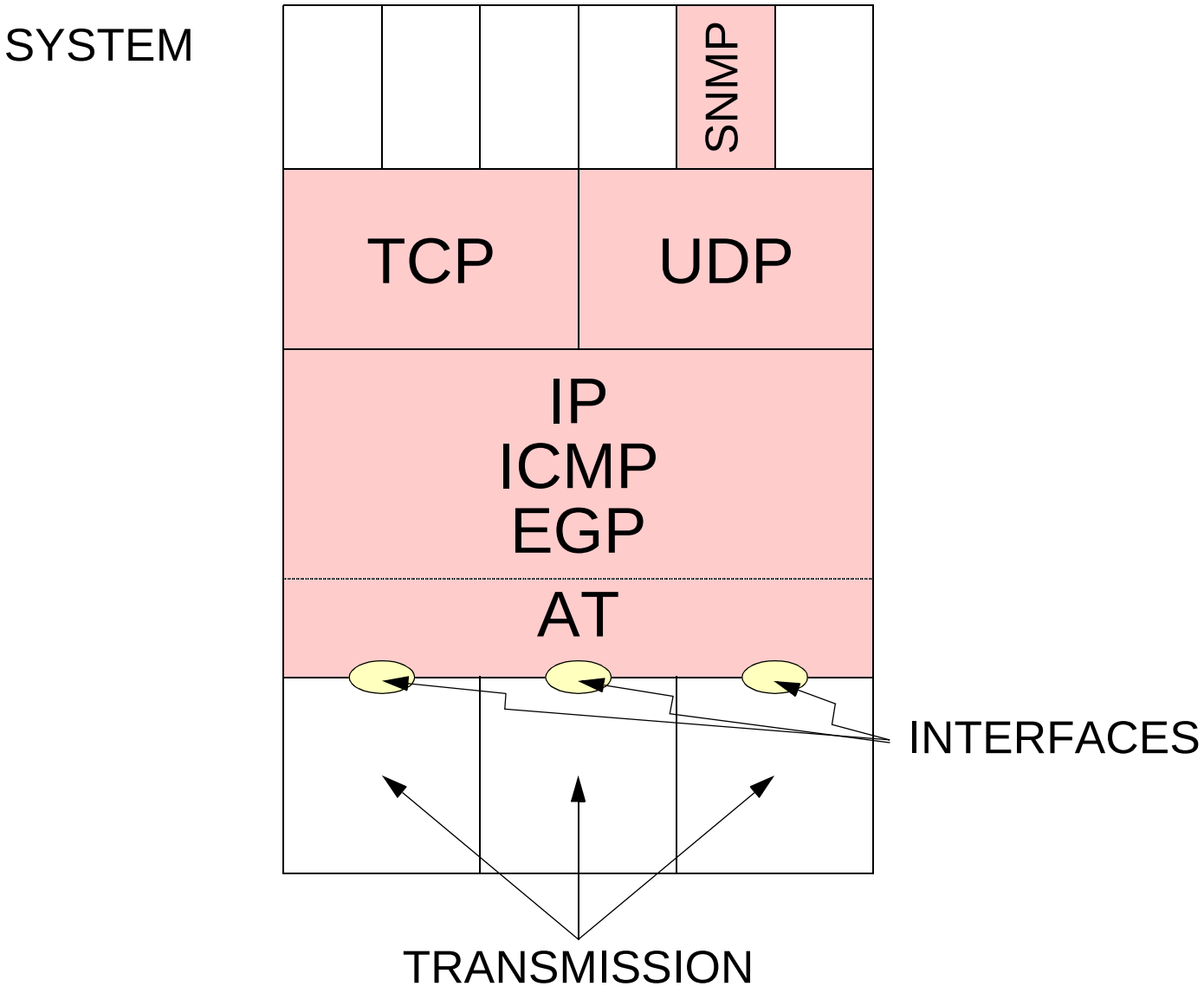
DESIGN CRITERIA

- 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
SYSTEM (1)
INTERFACES (2)
AT (3)
IP (4)
ICMP (5)
TCP (6)
UDP (7)
EGP (8)
TRANSMISSION (10)
SNMP (11)

MIB-II GROUPS IN A PROTOCOL STACK



NEW VERSIONS

SYSTEM GROUP ⇨ SNMPv2 MIB (RFC 1907)

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)

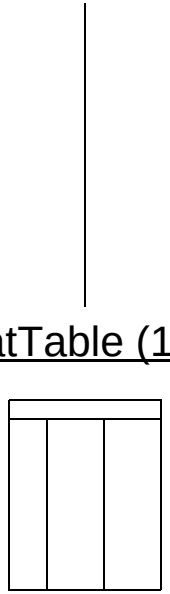
EGP GROUP ⇨ OUTDATED (BGP)

TRANSMISSION GROUP ⇨ IS PLACEHOLDER

SNMP GROUP ⇨ SNMPv2 MIB (RFC 1907)

ADDRESS TRANSLATION GROUP

at (3)



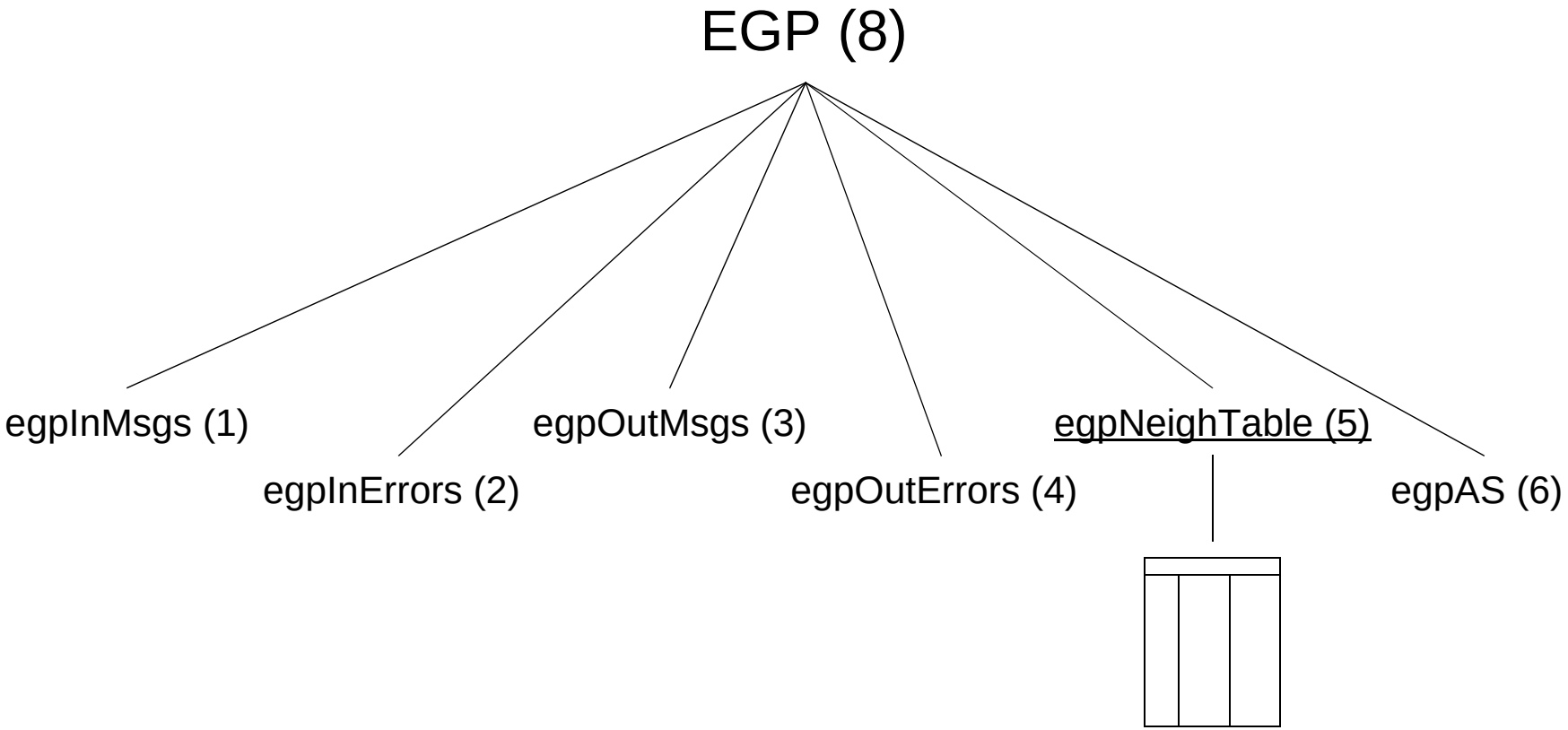
atTable (1)

- DEPRECATED STATUS

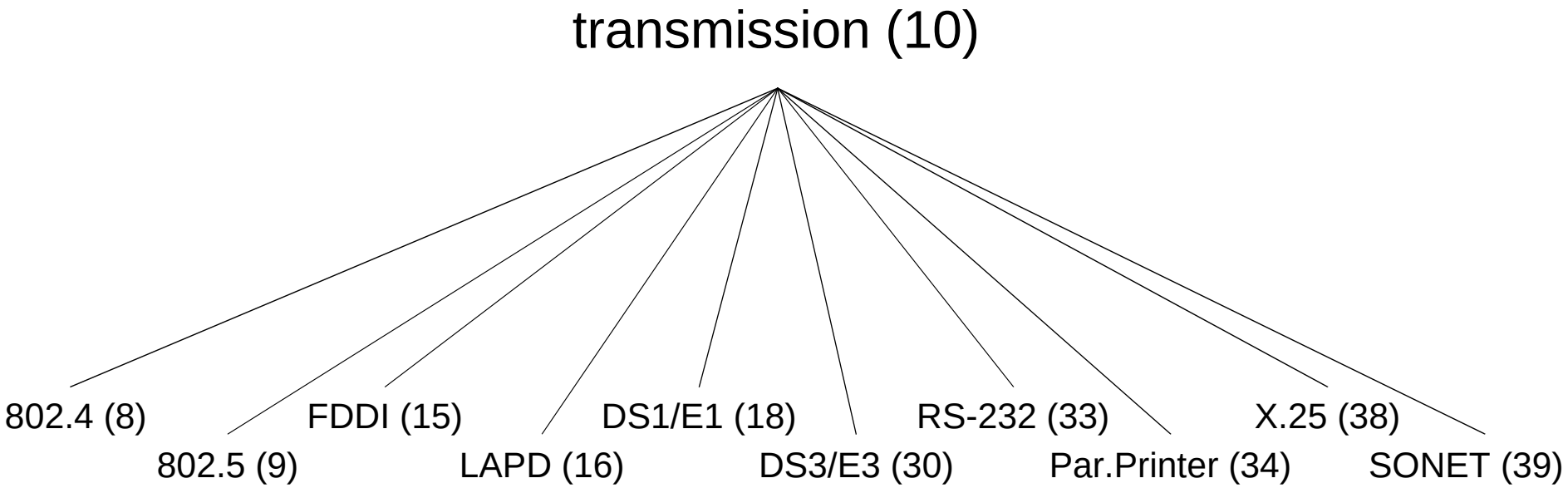
atTable

atIfIndex	atPhysAddress	atNetAddress
1		aa.bb.cc.dd
2		ee.ff.gg.hh
n		ww.xx.yy.zz

EGP GROUP


EGP (8)
egpInMsgs (1)
egpInErrors (2)
egpOutMsgs (3)
egpOutErrors (4)
egpNeighTable (5)
egpAS (6)

TRANSMISSION GROUP

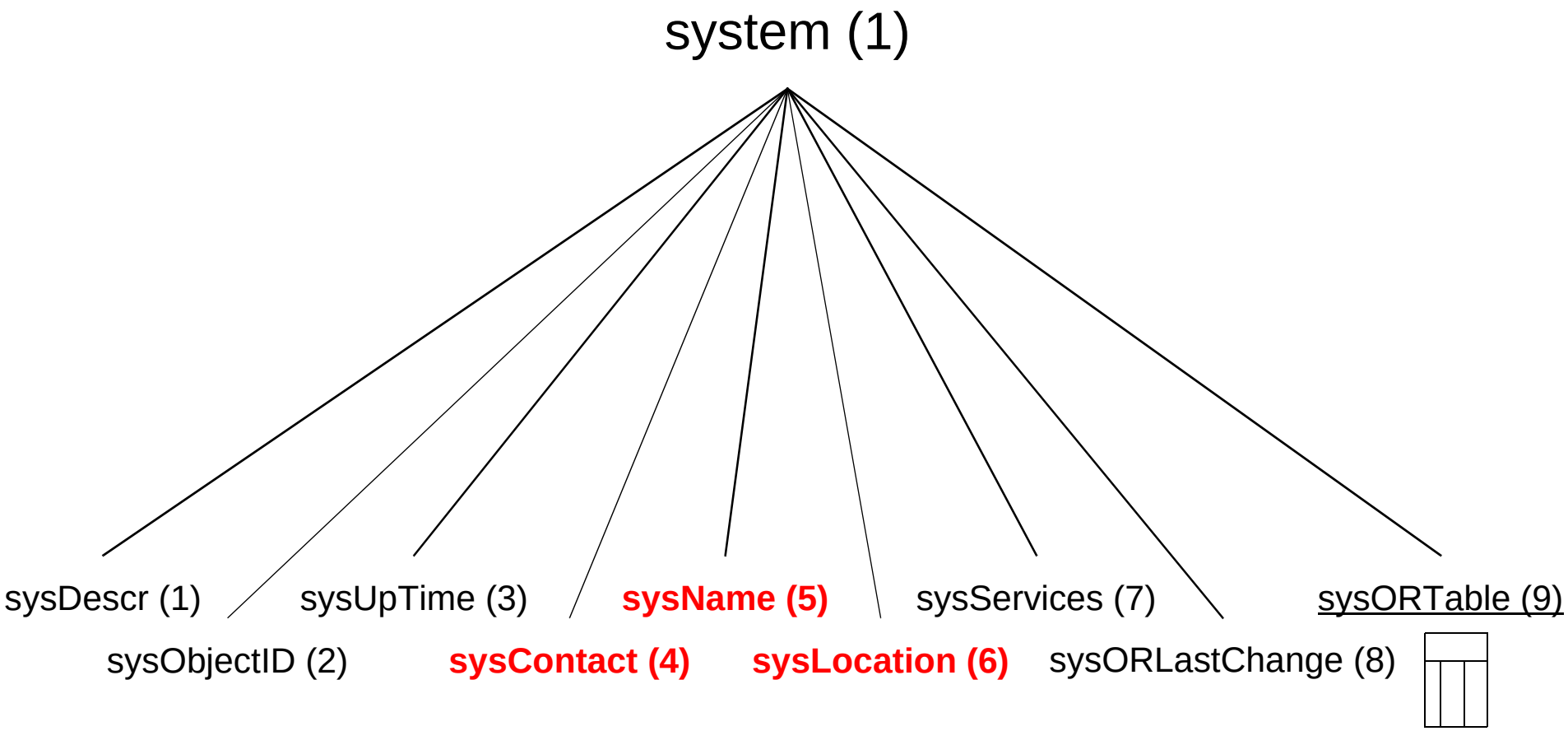


SNMPv2 MIB

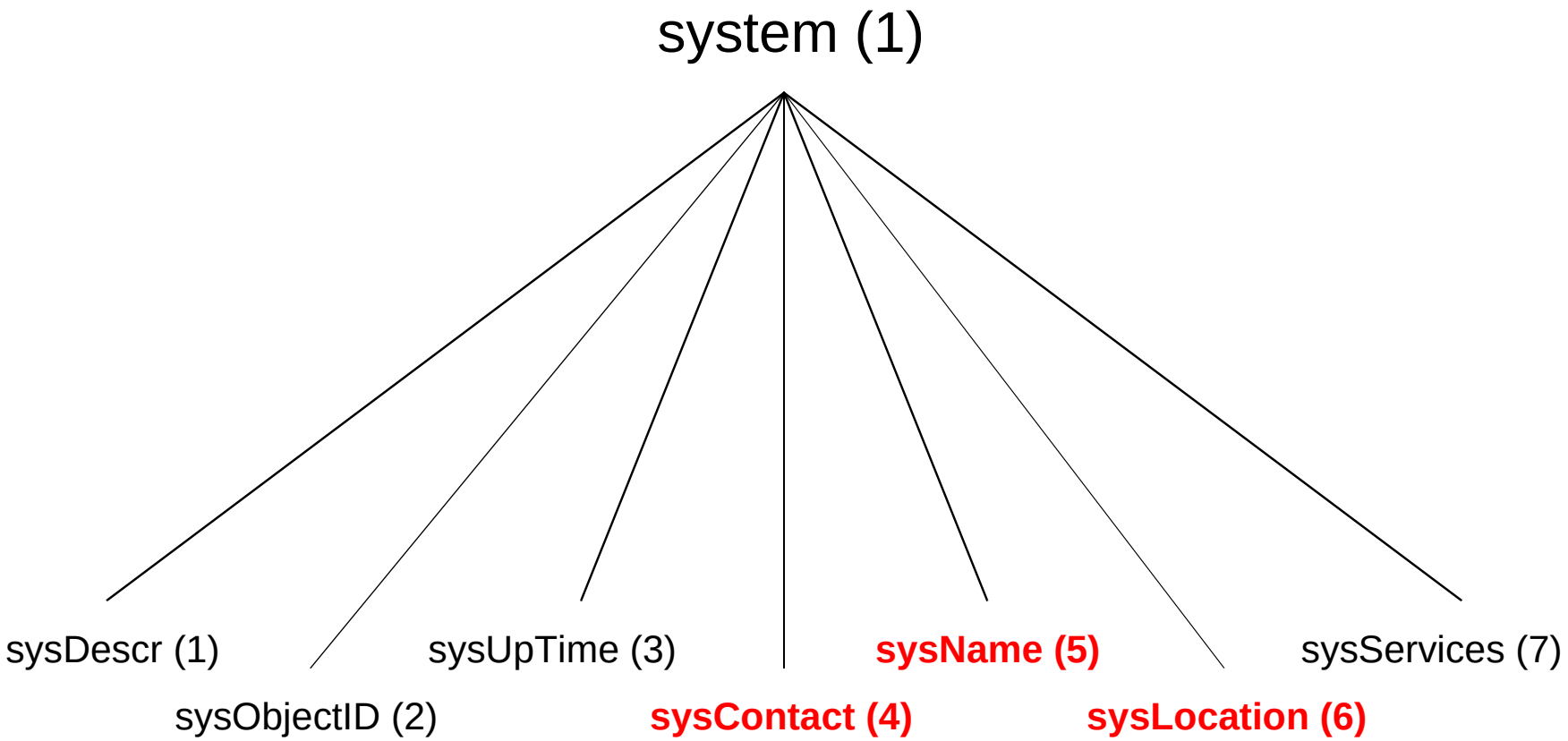
RFC 1907
PROPOSED STANDARD

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

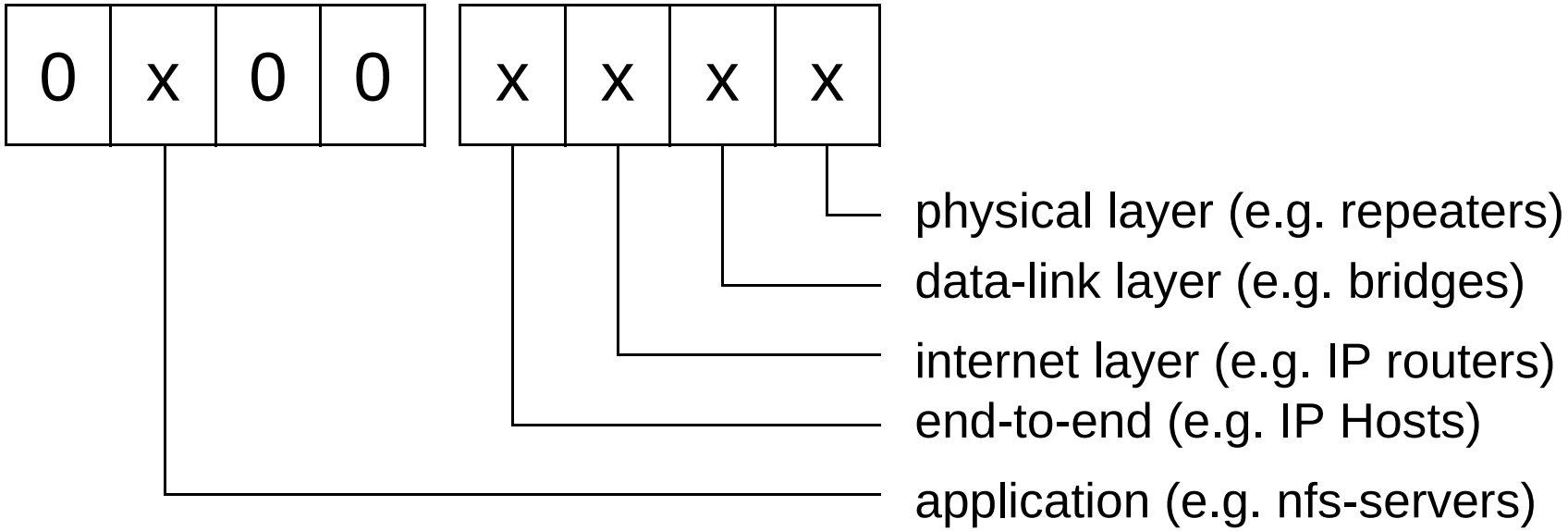
SYSTEM GROUP



ORIGINAL SYSTEM GROUP (MIB-II)



sysServices



EXAMPLE

sysDescr:	"**Cisco Gateway**"
sysObjectID:	**1.3.6.1.4.1.9.1.1**
sysUpTime:	**37153422** *(4 days, 7 h, 12 min, 14.22 s)*
sysContact:	"**helpdesk@cs.utwente.nl**"
sysName:	"**utic01.cs.utwente.nl**"
sysLocation:	"**near logica meeting room**"
sysServices:	**6** *(bridge and router functions)*

sysORTable - EXAMPLE

sysORIndex	sysORID	sysORDescr	sysORUpTime
1	IF-MIB!ifMIB	The MIB module to describe generic objects for network interface sub-layers	82
2	SNMPv2-MIB!snmpMIB	The MIB module for SNMPv2 entities	82
3	TCP-MIB!tcpMIB	The MIB module for managing TCP implementations	82
4	IP-MIB!ip	The MIB module for managing IP and ICMP implementations	85
5	UDP-MIB!udpMIB	The MIB module for managing UDP implementations	85

SNMP GROUP

7 READ-ONLY COUNTERS
(MIB-II HAD 29 COUNTERS!)

- THE NUMBER OF RECEIVED SNMP MESSAGES

- THE NUMBER OF RECEIVED SNMP MESSAGES,
FOR A VERSION NOT SUPPORTED

- THE NUMBER OF RECEIVED SNMP MESSAGES,
WITH AN INCORRECT COMMUNITY NAME / OPERATION NOT ALLOWED

- THE NUMBER OF RECEIVED SNMP MESSAGES,
WHICH COULD NOT BE PARSED,
OR WHICH WERE DROPPED

1 READ-WRITE VARIABLE

- TO ENABLE / DISABLE AUTHENTICATION TRAPS

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:

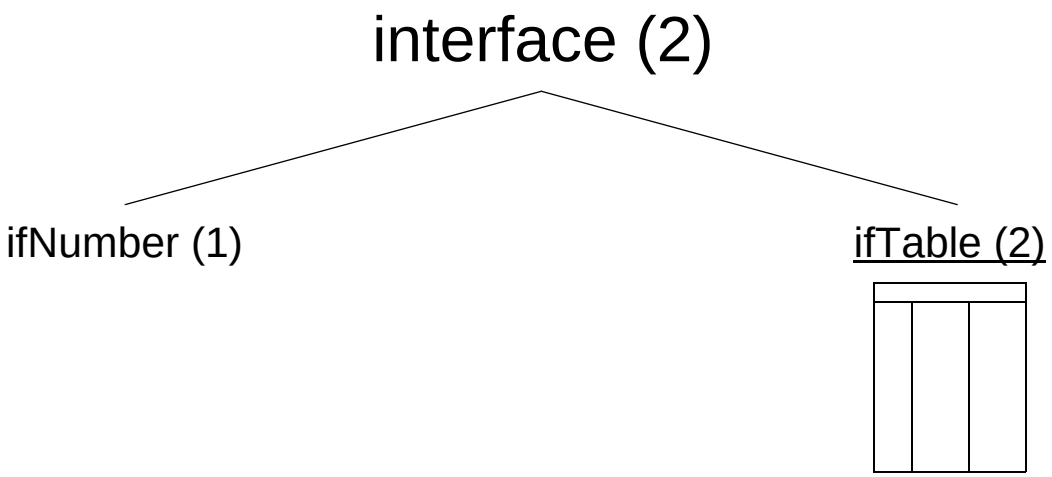
- ifStackTable
- ifTable
- ifXTable

SEE ALSO: INVERTED STACK TABLE

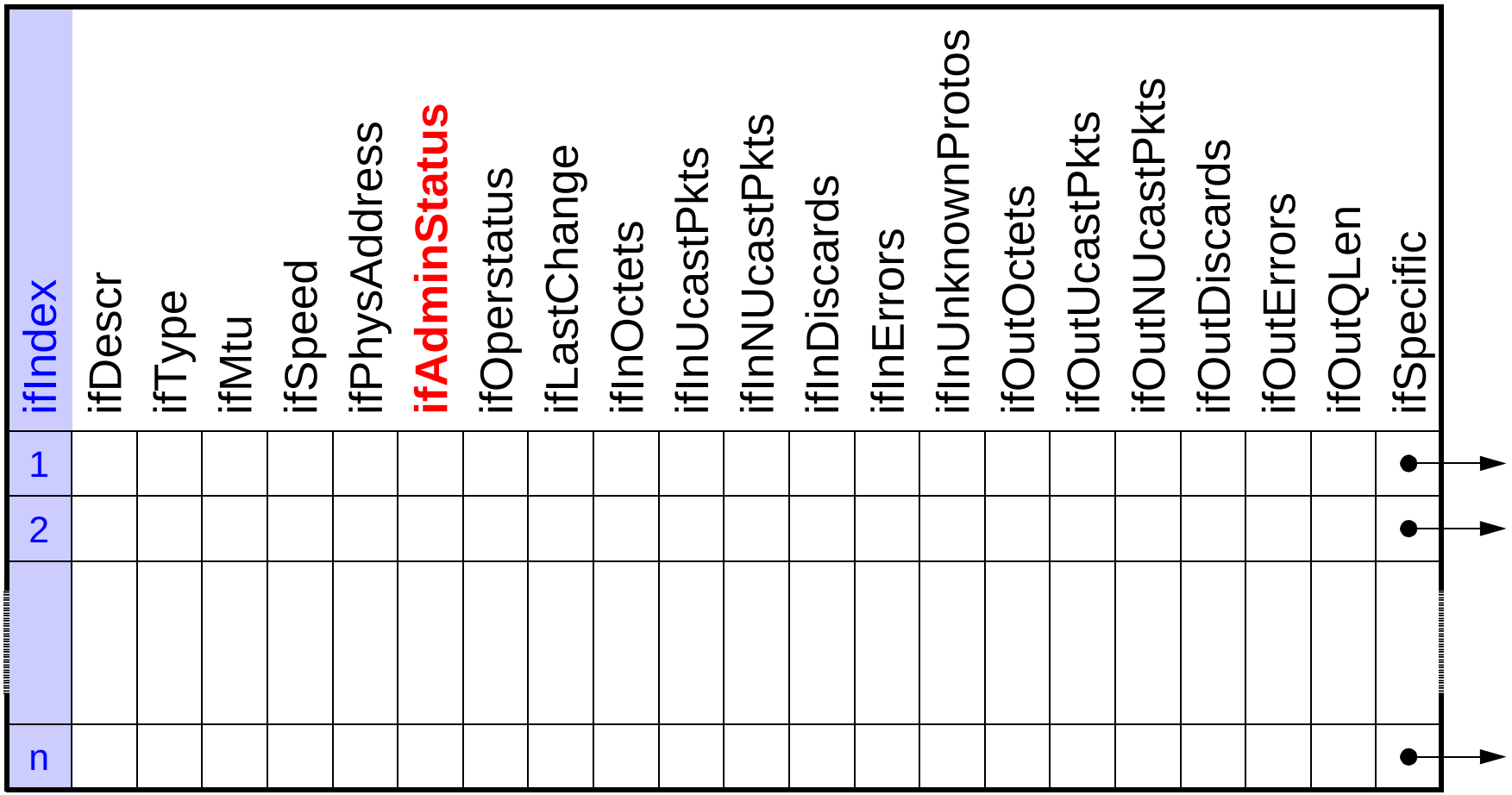
RFC 2864
DRAFT STANDARD

IF GROUP OF MIB-II

- PROVIDES DETAILED INFORMATION OF "GENERALIZED" INTERFACES
- CONTAINS POINTERS TO MEDIA SPECIFIC MIBS
- INTERFACES CAN BE PHYSICAL OR VIRTUAL
- DEFINES 1 COUNTER AND 1 TABLE



ifTable OF MIB-II



ifType and ifStatus

- ifType

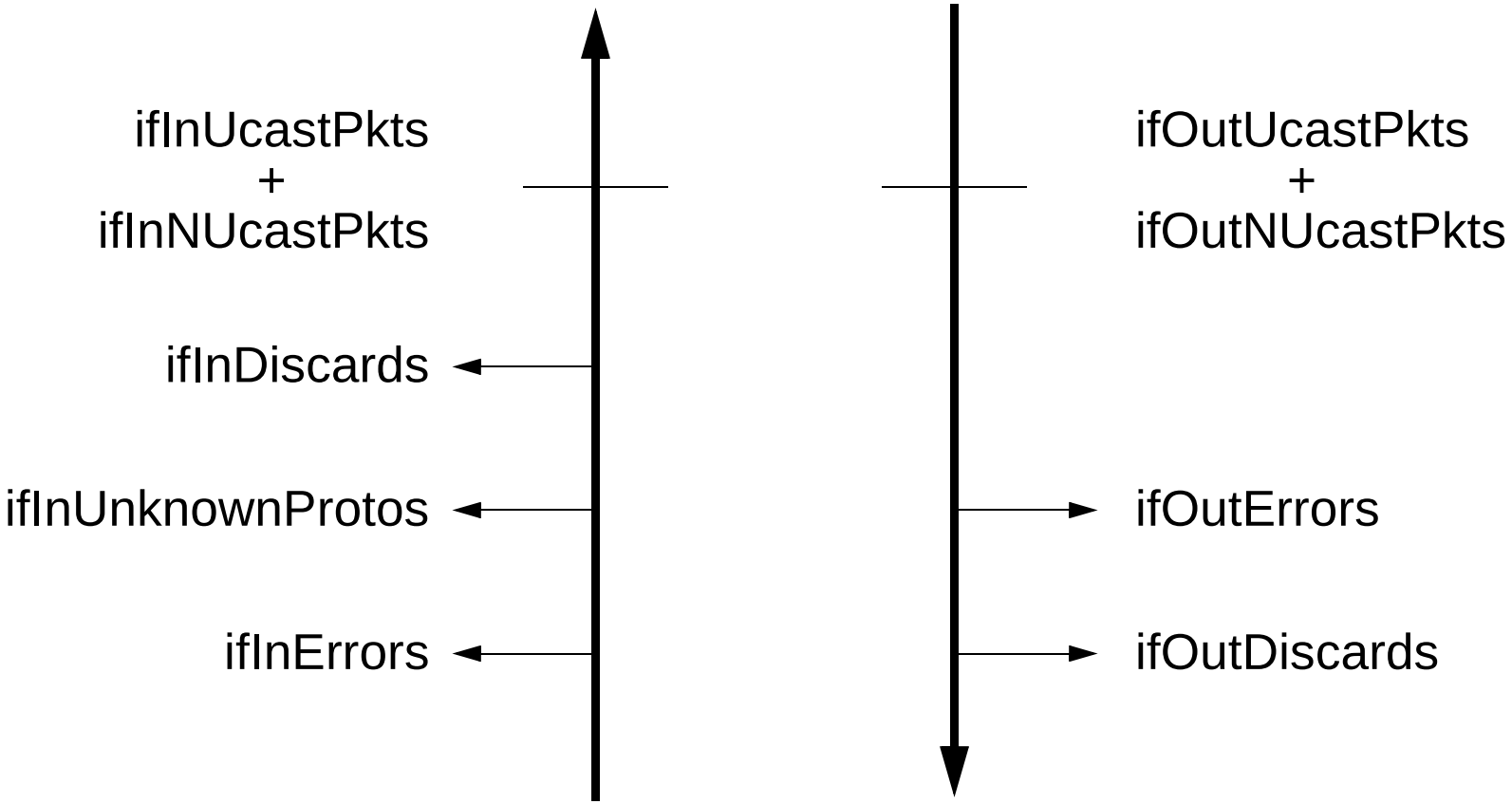
EXAMPLES:

1	Undefined	16	LAPB
6	Ethernet	20	ISDN Basic
7	IEEE 802.3	21	ISDN Primary
8	IEEE 802.4	23	PPP
9	IEEE 802.5	24	Loopback
10	IEEE 802.6	28	SLIP
15	FDDI	32	Frame Relay

- ifAdminStatus / ifOperStatus

1 = up
2 = down
3 = testing

IF PACKET COUNT



PROBLEMS WITH ifTable OF MIB-II

LAYERED INTERFACES DIFFICULT TO DESCRIBE
- WHAT SHOULD BE THE VALUE OF ifType?

NO DISTINCTION BETWEEN VIRTUAL & PHYSICAL INTERFACES

NO SUPPORT FOR DYNAMIC ADDITION/REMOVAL OF INTERFACES

COUNTERS ARE 32 BITS (SMIv1)
- WRAP AROUND 10 MBIT ETHERNET: 57 MINUTES
- WRAP AROUND 1 GBIT ETHERNET: LESS THAN 1 MINUTE!

GAUGES ARE 32 BITS
- ifSpeed

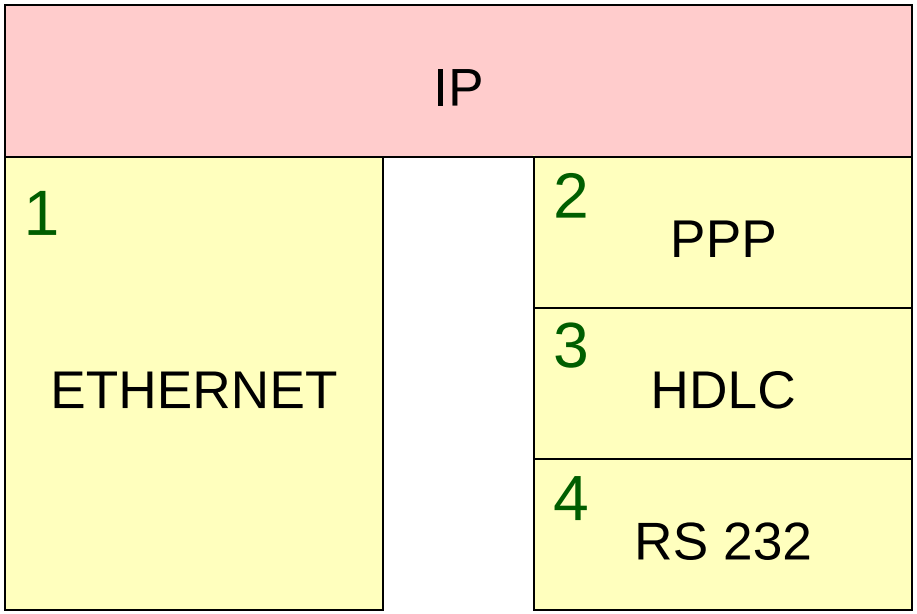
INTERFACE TYPES "HARD CODED" IN STANDARD

NO DIFFERENCE BETWEEN MULTICAST & BROADCAST

TOO MANY TRAPS

INTERFACE STACKS

LAYERING OF INTERFACES NOW DEFINED BY ifStackTable



ifStackTable:

higher	lower	status
0	1	active
1	0	active
0	2	active
2	3	active
3	4	active
4	0	active

CHANGES / ADDITIONS TO THE ifTable

NEW interfaceIndex TEXTUAL CONVENTION

- INTERFACE NUMBERING NEED NOT BE CONSEQUETIVE

NEW IANAifType

- INTERFACE TYPES CAN BE DEFINED EXTERNALLY

HIGH CAPACITY COUNTERS

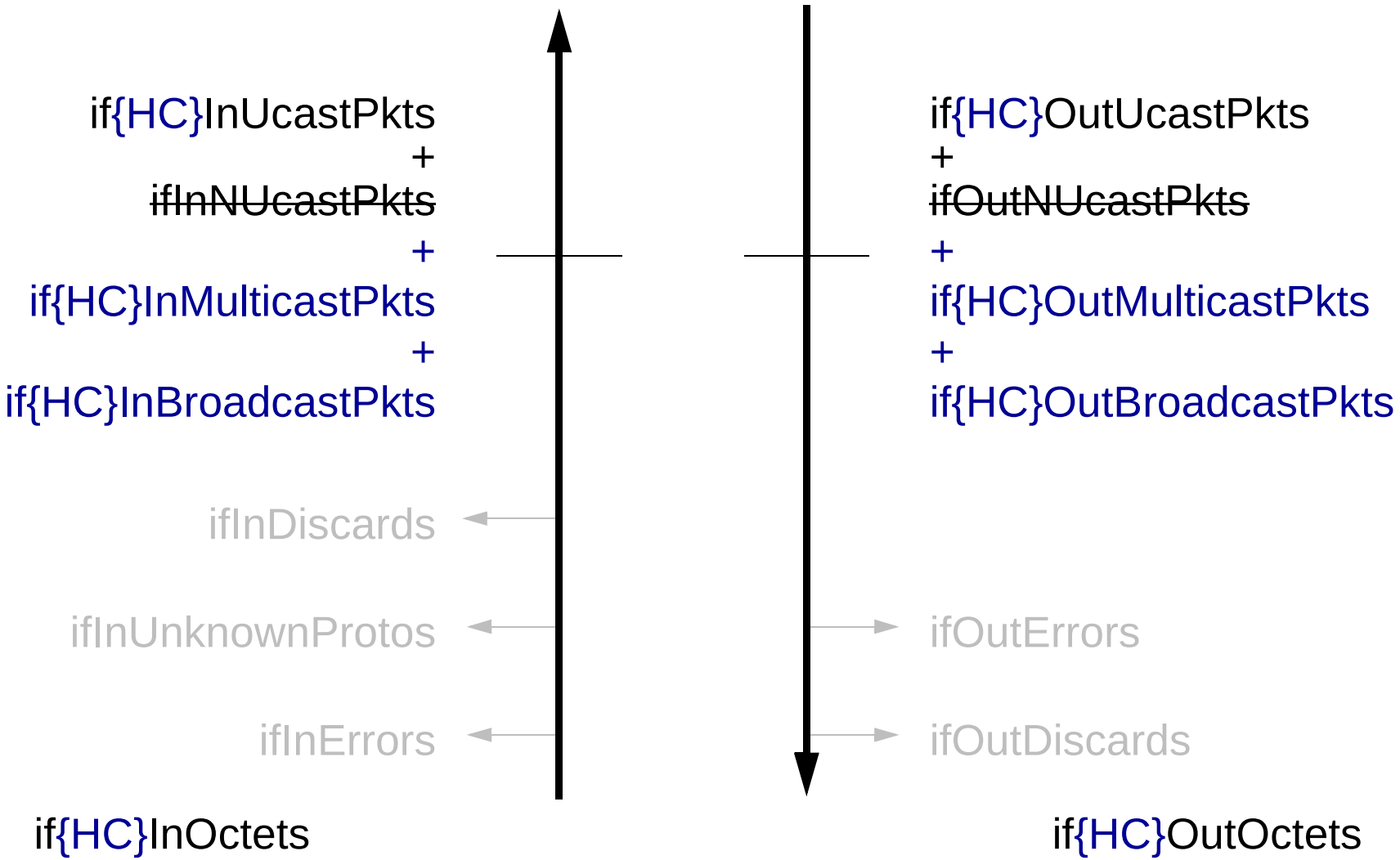
NEW ifHighSpeed OBJECT

- IN 1.000.000 BITS/S

SEVERAL OTHER NEW OBJECTS

- ifName
- ifAlias
- ifLinkUpDownTrapEnable
- ifConnectorPresent
- ifPromiscuousMode
- ifCounterDiscontinuityTime

ifXTable: HIGH CAPACITY COUNTERS



OTHER ADDITIONS

RECEIVE ADDRESS TABLE

- DEFINES THE ADDRESSES FOR WHICH EACH INTERFACE WILL RECEIVE PACKETS / FRAMES

linkUp / linkDown NOTIFICATION TYPES

COMPLIANCE GROUPS

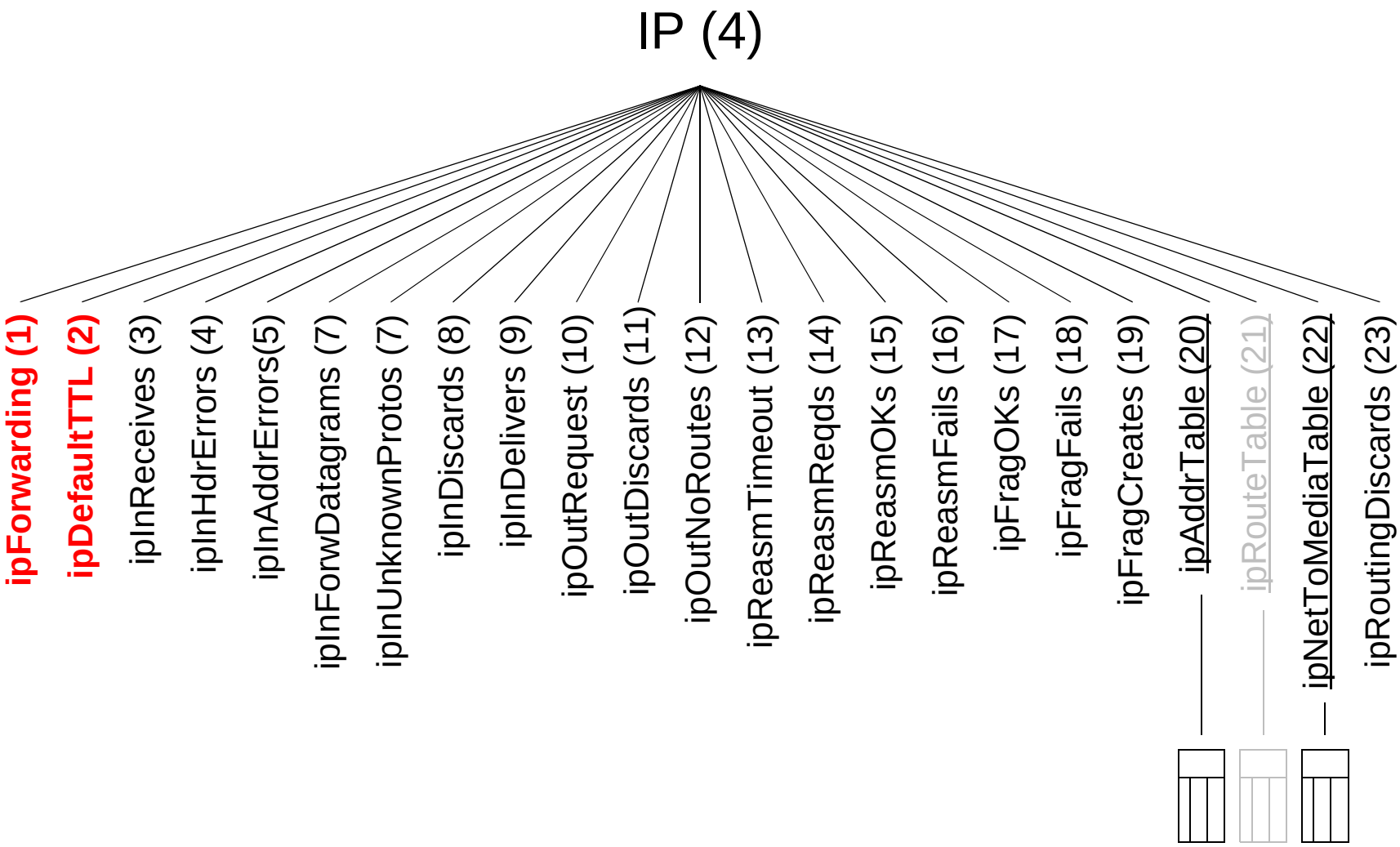
...

IP MIB

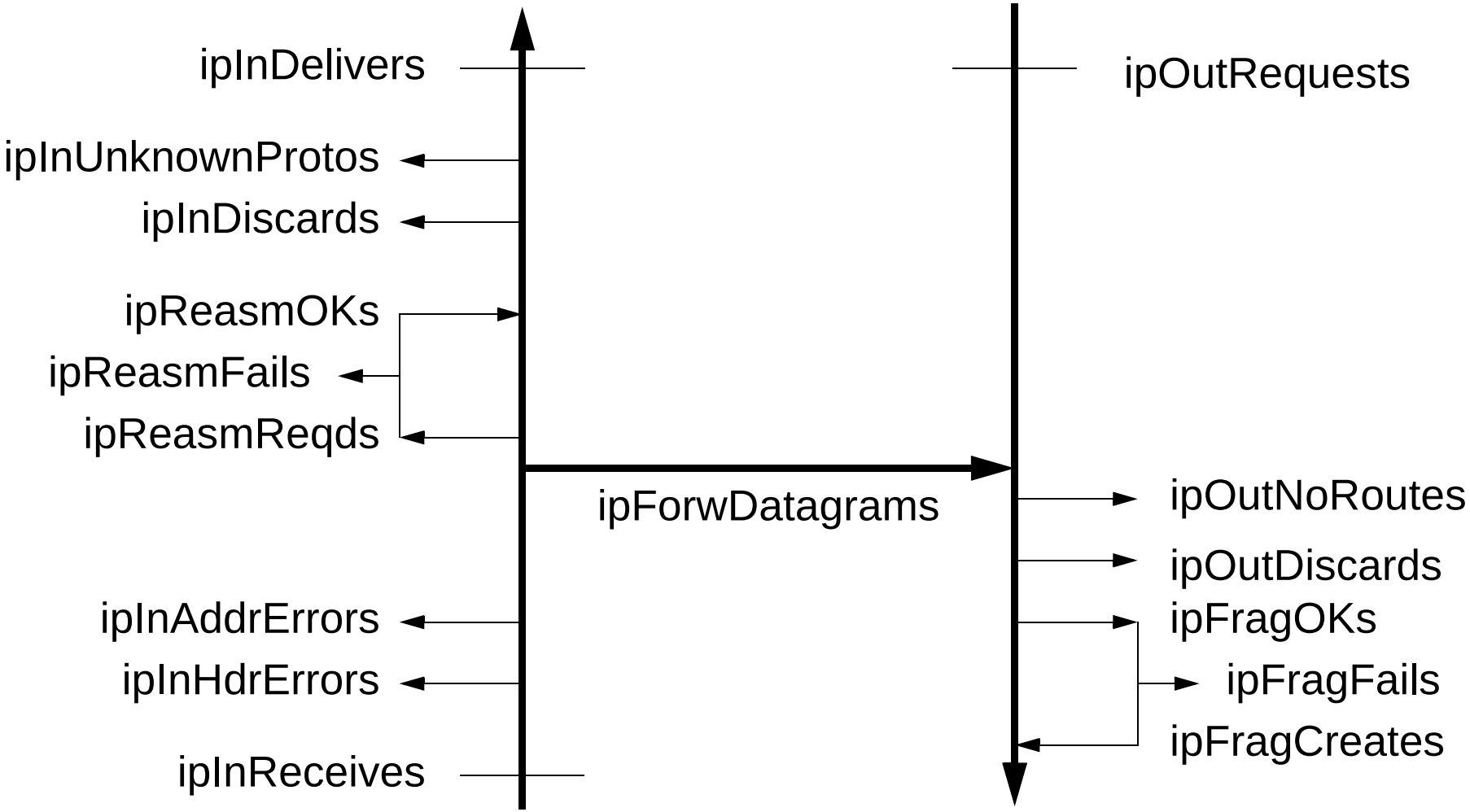
RFC 2011

- IP GROUP
- ICMP GROUP
- IP MIB Conformance

IP GROUP


IP (4)
ipForwarding (1)
ipDefaultTTL (2)
ipInReceives (3)
ipInHdrErrors (4)
ipInAddrErrors(5)
ipInForwDatagrams (7)
ipInUnknownProtos (7)
ipInDiscards (8)
ipInDelivers (9)
ipOutRequest (10)
ipOutDiscards (11)
ipOutNoRoutes (12)
ipReasmTimeout (13)
ipReasmReqds (14)
ipReasmOKs (15)
ipReasmFails (16)
ipFragOKs (17)
ipFragFails (18)
ipFragCreates (19)
ipAddrTable (20)
ipRouteTable (21)
ipNetToMediaTable (22)
ipRoutingDiscards (23)

IP PACKET COUNT



ipAddrTable

ipAdEntAddr	ipAdEntIfIndex	ipAdEntNetMask	ipAdEntBcastAddr	ipAdEntReasmMaxSize
192.89.16.4	1	255.255.255.0	1	65535
192.89.16.8				

ipNetToMediaTable

ipNetToMedia IfIndex	ipNetToMedia PhysAddress	ipNetToMedia NetAddress	ipNetToMedia Type
1	08:00:20:00:25:66	129.14.16.4	3 (dynamic)
2			

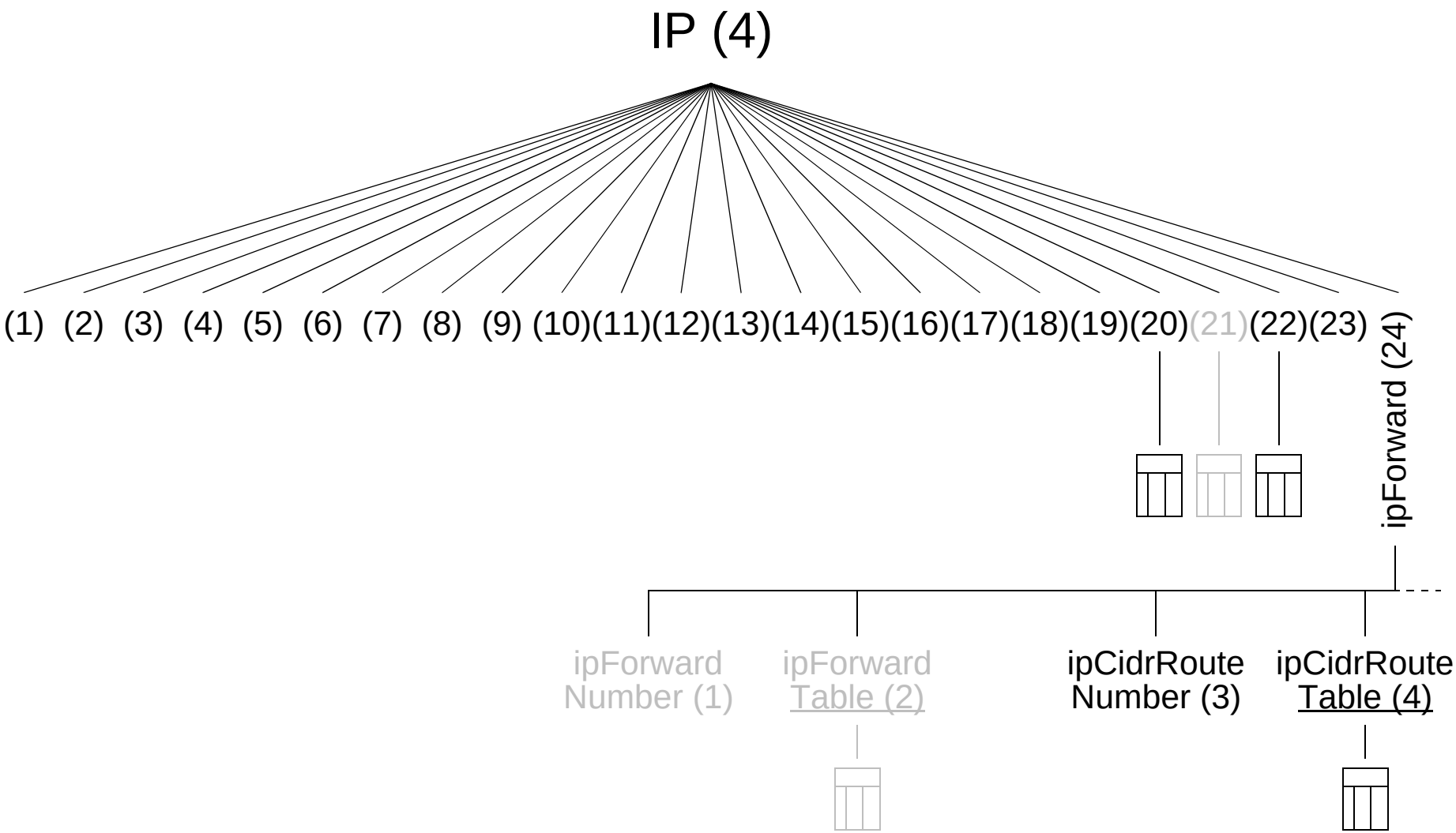
ipRouteTable of MIB-II

ipRouteDest	ipRouteNextHop	ipRouteIfIndex	ipRouteMask	ipRouteType	ipRouteMetric1	ipRouteMetric2	ipRouteMetric3	ipRouteMetric4	ipRouteMetric5	ipRouteAge	ipRouteProto	ipRouteInfo
129.14.16.4	129.16.1.7	1	255.255.0.0	3							ospf	
192.89.16.8												

NOT INCLUDED IN IP-MIB

IP Forwarding MIB

RFC 2096



ipCidrRouteTable

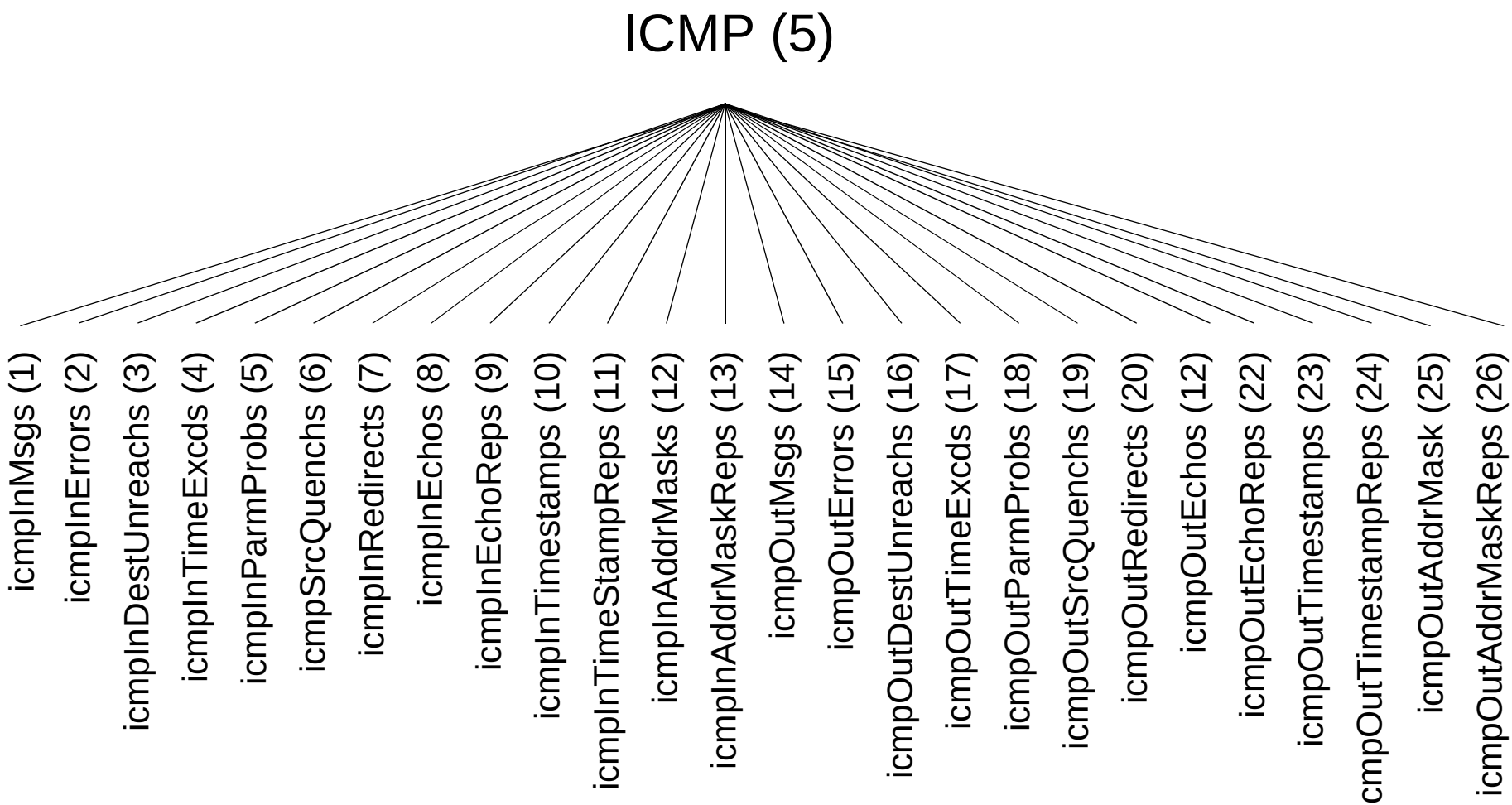
ipCidrRouteDest	ipCidrRouteMask	ipCidrRouteTos	ipCidrRouteNextHop	ipCidrRouteIfIndex	ipCidrRouteType	ipCidrRouteProto	ipCidrRouteAge	ipCidrRouteInfo	ipCidrRouteNextHopAS	ipCidrRouteMetric1	ipCidrRouteMetric2	ipCidrRouteMetric3	ipCidrRouteMetric4	ipCidrRouteMetric5	ipCidrRouteStatus
33.14.16.4	255.255.255.0	0	33.1.1.7	1	4	ospf			16.1.1.1						

ipForwardTable

DEFINED BY RFC 1354 / OBSOLETED BY RFC 2096

ipForwardDest	ipForwardMask	ipForwardPolicy	ipForwardNextHop	ipForwardIfIndex	ipForwardType	ipForwardProto	ipForwardAge	ipForwardInfo	ipForwardNextHopAS	ipForwardMetric1	ipForwardMetric2	ipForwardMetric3	ipForwardMetric4	ipForwardMetric5
33.14.16.4	255.255.255.0	0	33.1.1.7	1	4	ospf			16.1.1.1					

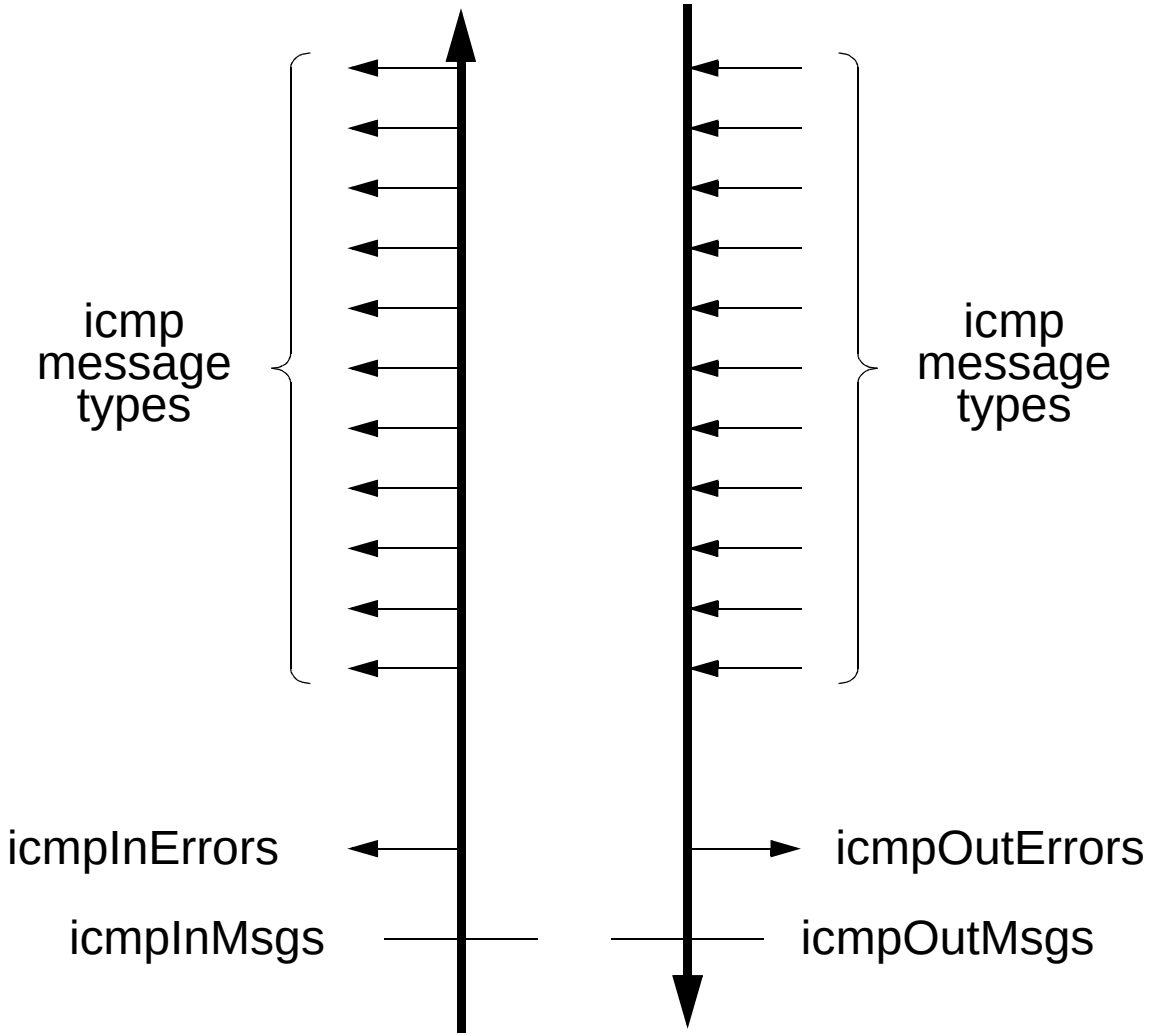
ICMP GROUP


ICMP (5)
icmpInMsgs (1)
icmpInErrors (2)
icmpInDestUnreachs (3)
icmpInTimeExcds (4)
icmpInParmProbs (5)
icmpSrcQuenchs (6)
icmpInRedirects (7)
icmpInEchos (8)
icmpInEchoReps (9)
icmpInTimestamps (10)
icmpInTimeStampReps (11)
icmpInAddrMasks (12)
icmpInAddrMaskReps (13)
icmpOutMsgs (14)
icmpOutErrors (15)
icmpOutDestUnreachs (16)
icmpOutTimeExcds (17)
icmpOutParmProbs (18)
icmpOutSrcQuenchs (19)
icmpOutRedirects (20)
icmpOutEchos (12)
icmpOutEchoReps (22)
icmpOutTimestamps (23)
icmpOutTimestampReps (24)
icmpOutAddrMask (25)
icmpOutAddrMaskReps (26)

ICMP PACKET COUNT

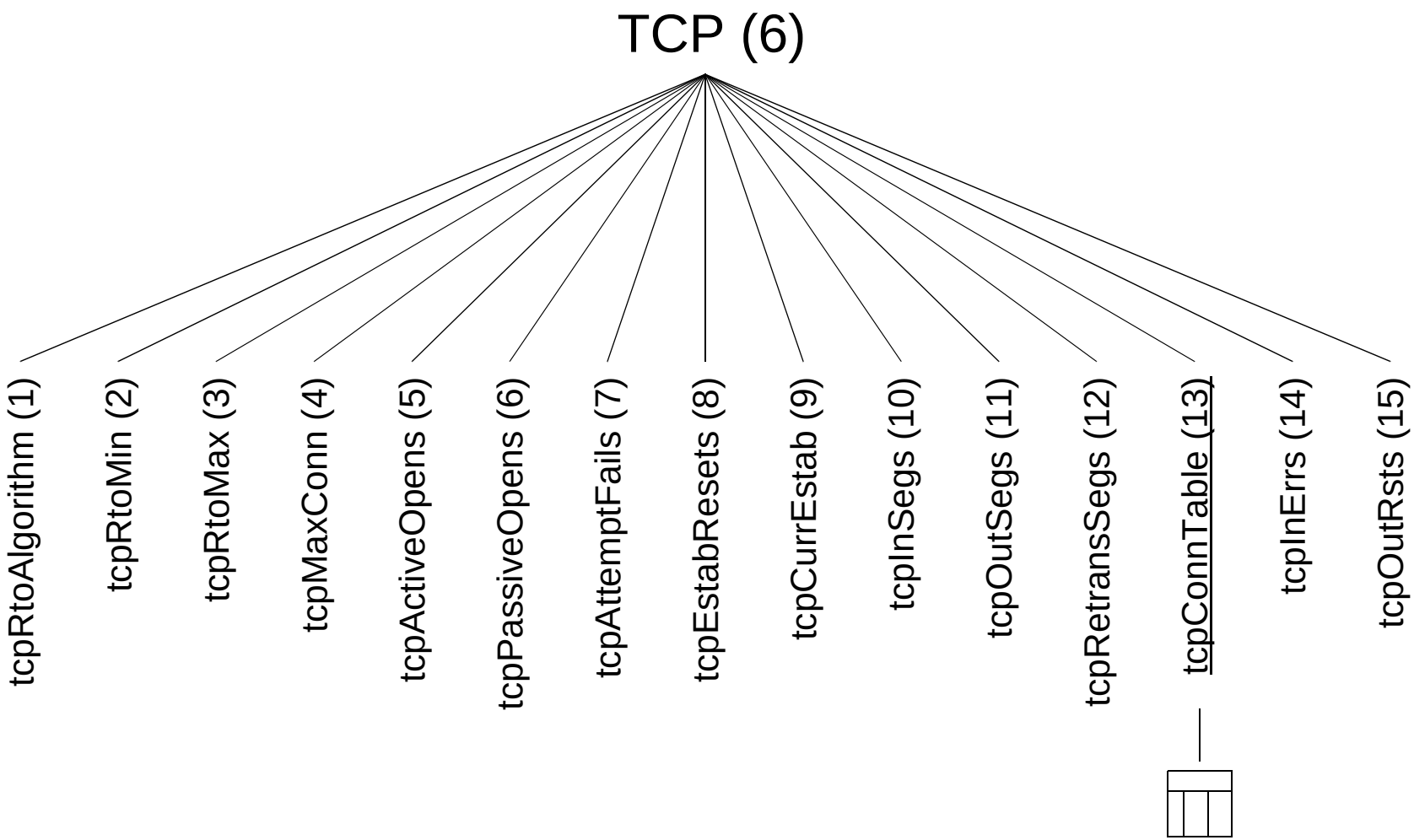
COUNTS FOR EACH ICMP MESSAGE TYPE HOW OFTEN IT HAS BEEN:

- TRANSMITTED
- RECEIVED



TCP MIB

RFC 2012 - PROPOSED STANDARD
OBJECT DEFINITIONS IDENTICAL TO THOSE OF MIB-II

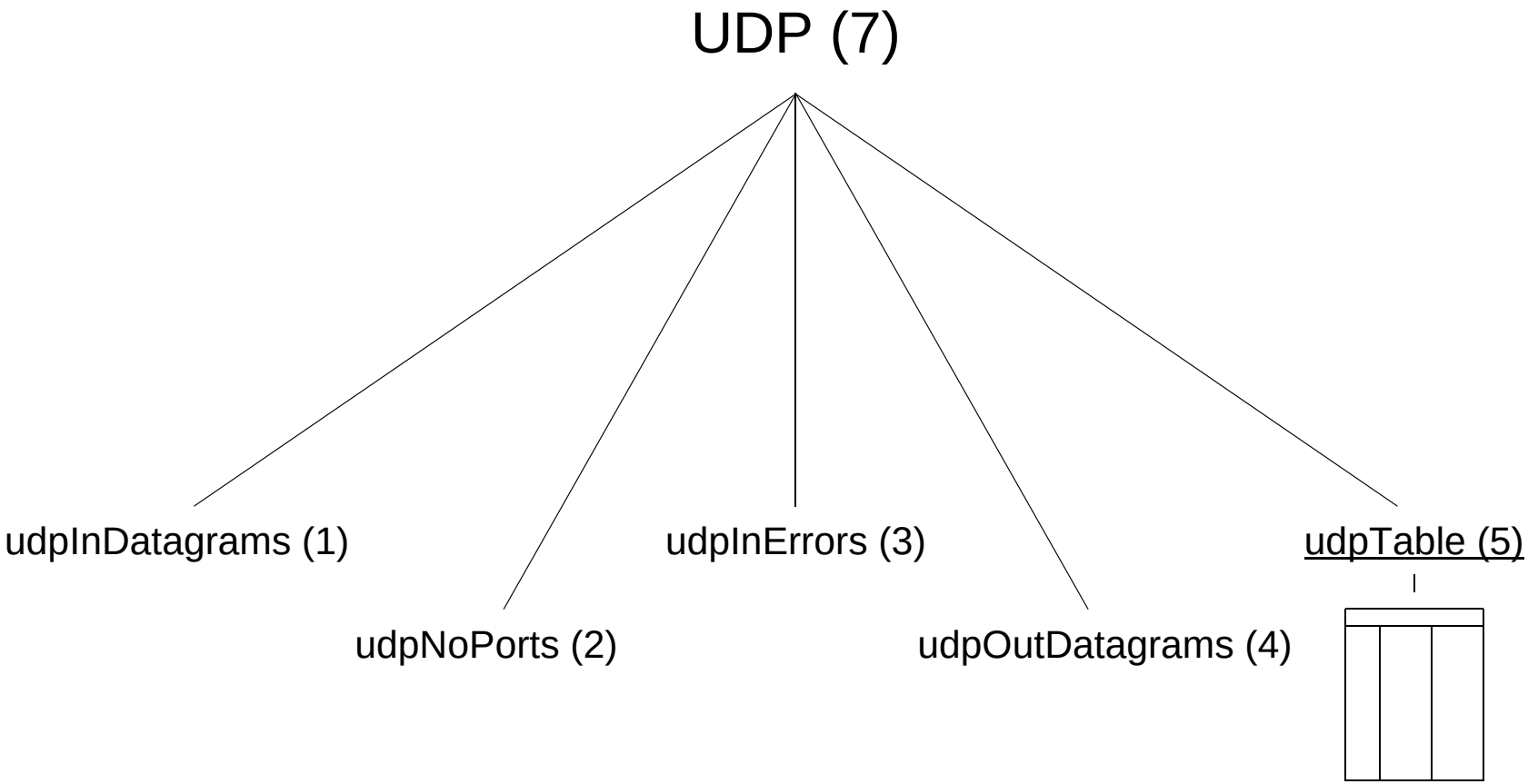


tcpConnTable

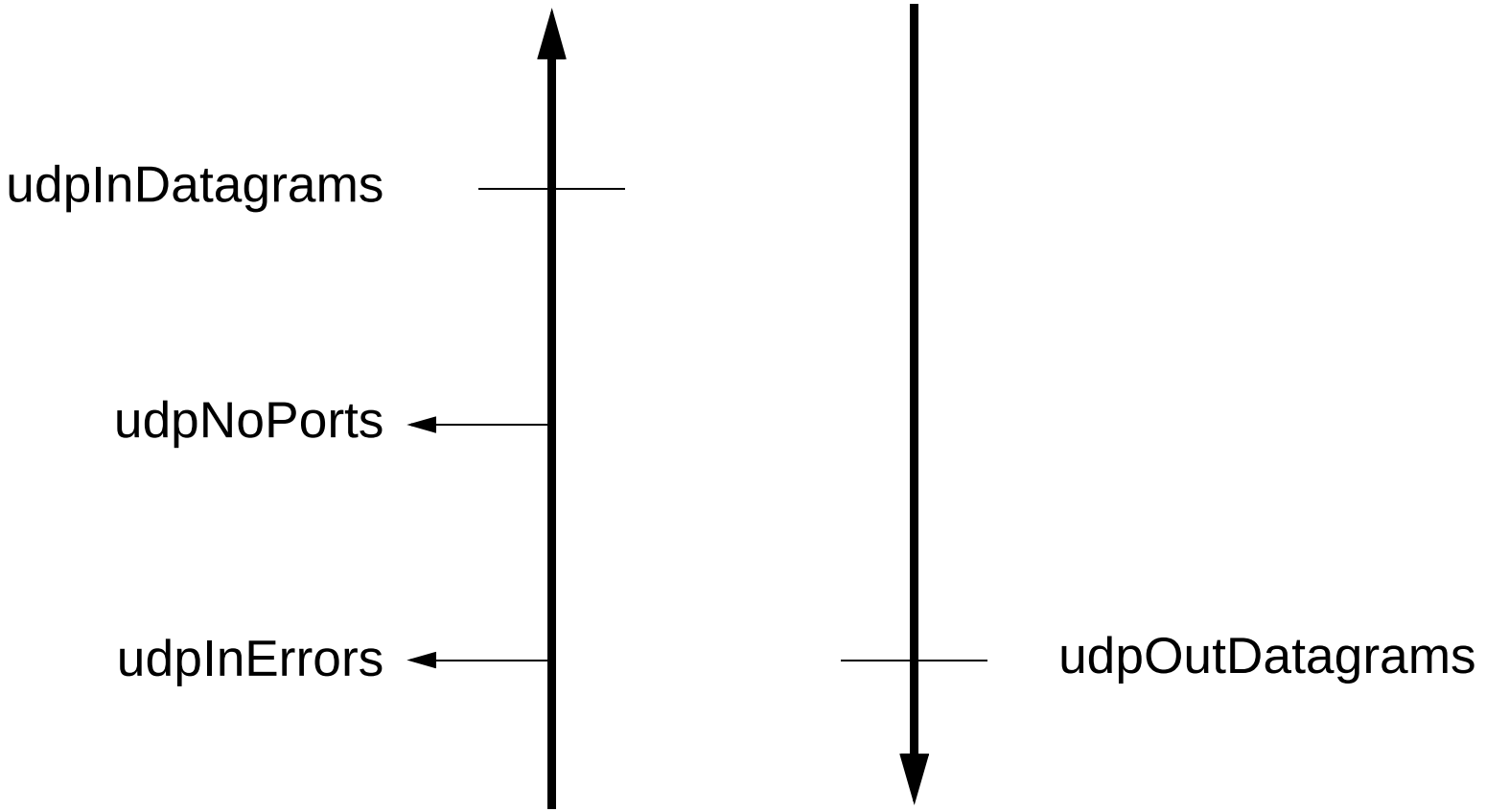
tcpConn State	tcpConn LocalAddress	tcpConn LocalPort	tcpConn RemoteAddress	tcpConn RemotePort
listen	0.0.0.0	23	0.0.0.0	0

UDP-MIB

RFC 2013 - PROPOSED STANDARD
OBJECT DEFINITIONS IDENTICAL TO THOSE OF MIB-II



UDP PACKET COUNT



udpTable

udpLocalAddress	udpLocalPort
129.16.4.12	161

HOST RESOURCES MIB

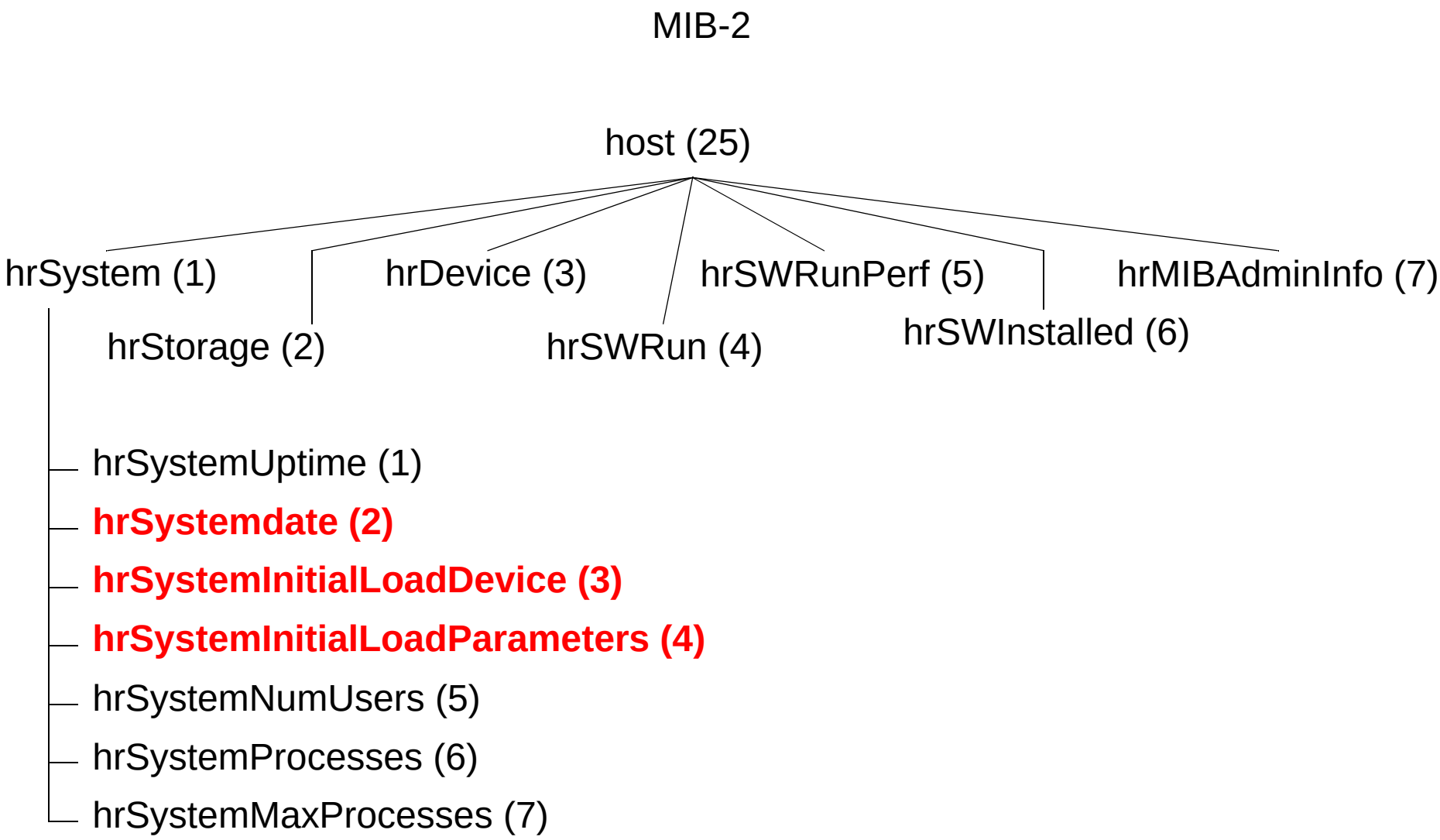
RFC 2790

PROVIDES INFORMATION RELATED TO:

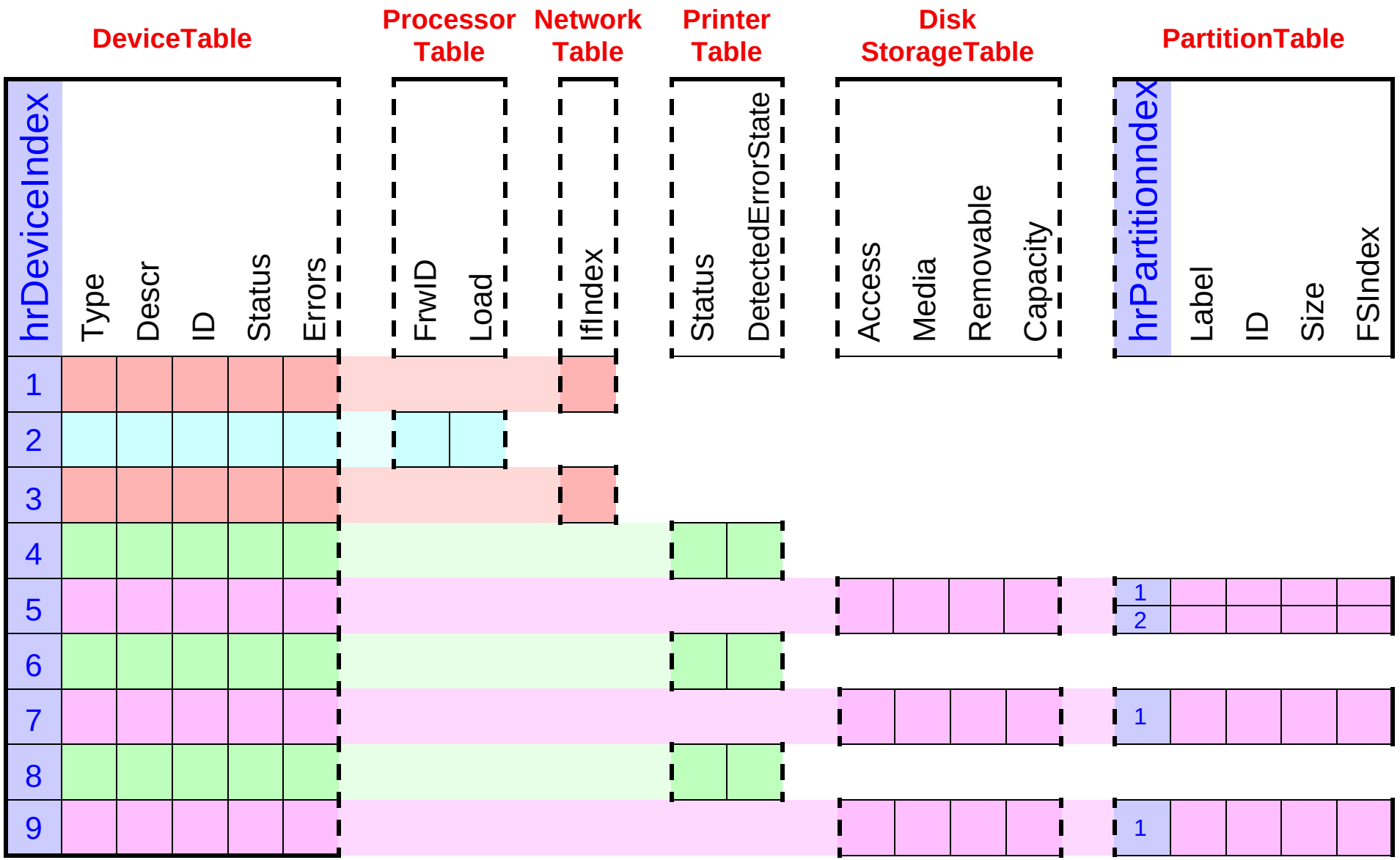
- PROCESSOR
- DISKS
- STORAGE
- NETWORK INTERFACES
- PRINTERS
- PROGRAMS
- ...

ORGANIZED IN 11 TABLES PLUS SOME VARIABLES

MAIN VARIABLES



MAIN TABLES


DeviceTable
Processor Table
Network Table
Printer Table
Disk StorageTable
PartitionTable
hrDeviceIndex
Type
Descr
ID
Status
Errors
FrwID
Load
IfIndex
Status
DetectedErrorState
Access
Media
Removable
Capacity
hrPartitionndex
Label
ID
Size
FSIndex
1
2
3
4
5
6
7
8
9

FILE SYSTEM TABLE

hrFSTable

hrFSIndex	FSMountpoint	FSRemoteMountPoint	FSType	FSAccess	FSBootable	FSStorageIndex	**FSLastFullBackupDate**	**FSLastpartialBackupDate**

SOFTWARE TABLES

SWRunTable

- Index
- Name
- ID
- Path
- Parameters
- Type
- **Status**

SWRunPerfTable

- CPU
- Mem

SWInstalledTable

- Name
- ID
- Type
- Date

STORAGE VARIABLE & TABLE

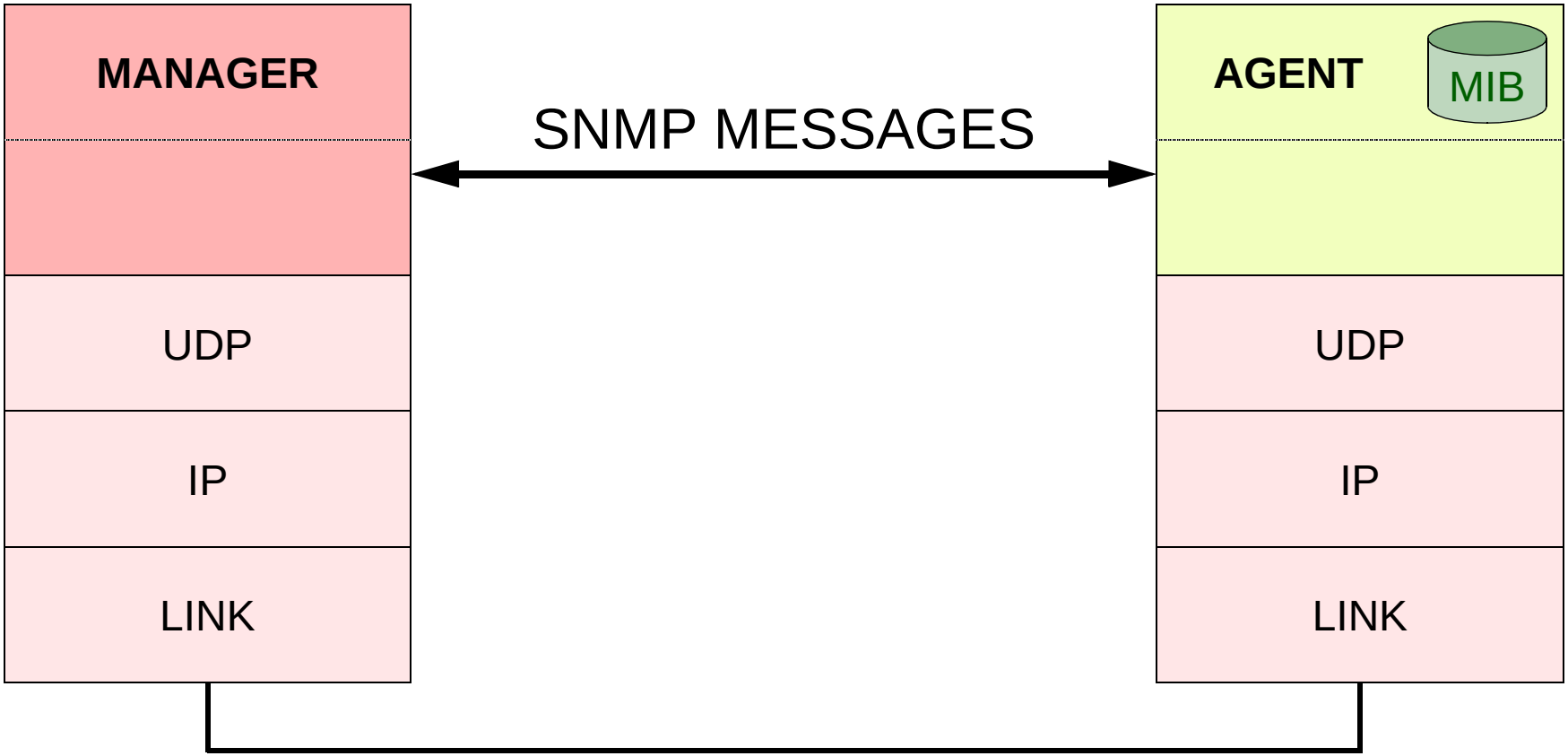
VARIABLE:

- MemorySize

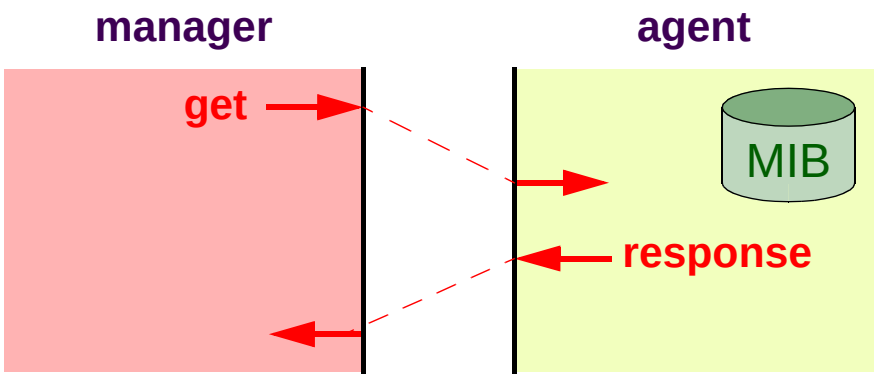
TABLE:

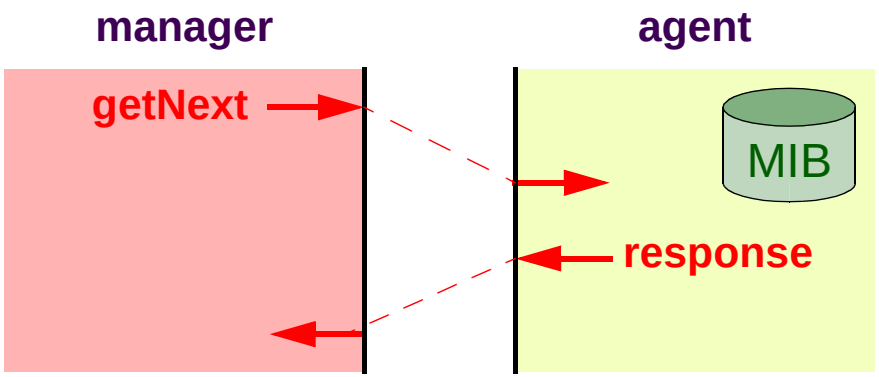
- Index
- Type
- Descr
- AllocationUnits
- **Size**
- Used
- AllocationFailures

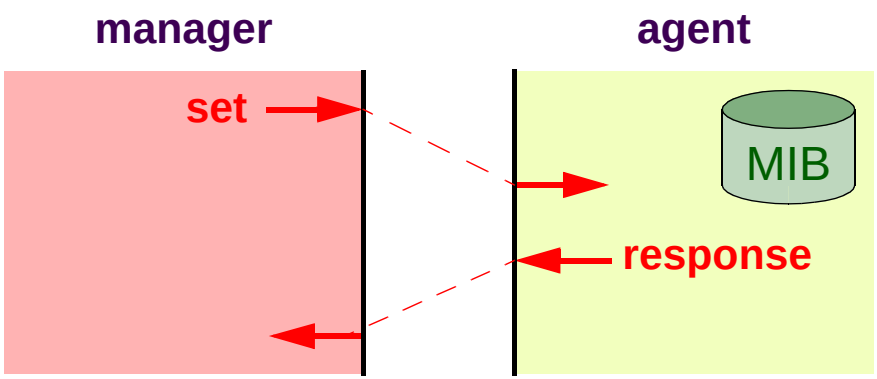
SNMP PROTOCOL

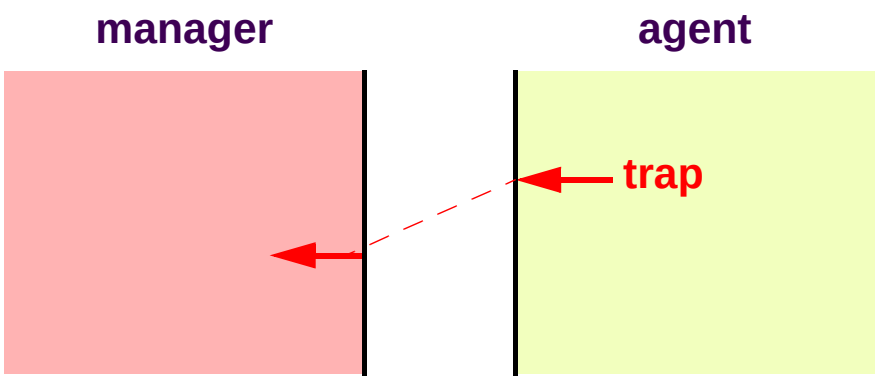


OVERVIEW OF PDUs



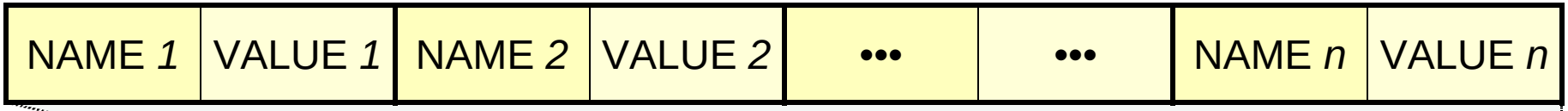






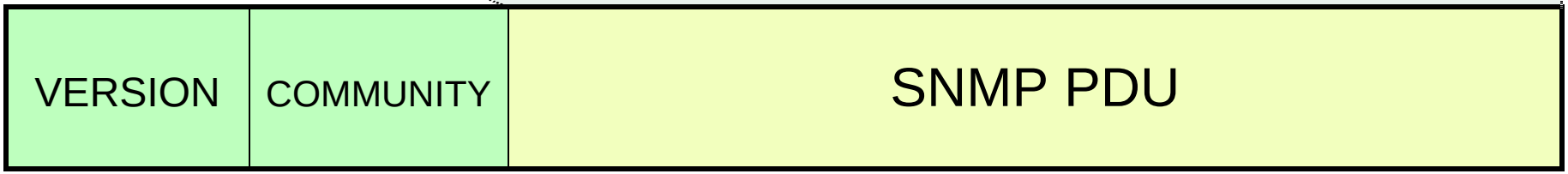
MESSAGE & PDU STRUCTURE

variable bindings:

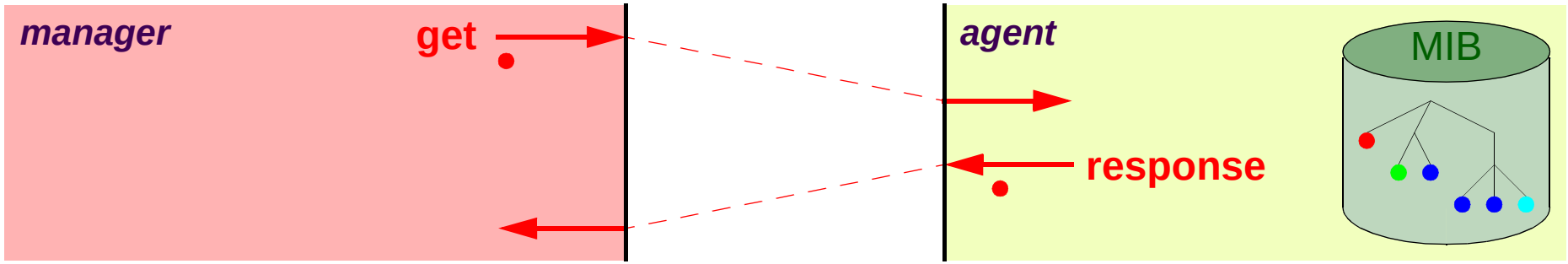


SNMP PDU:

SNMP message:



GET

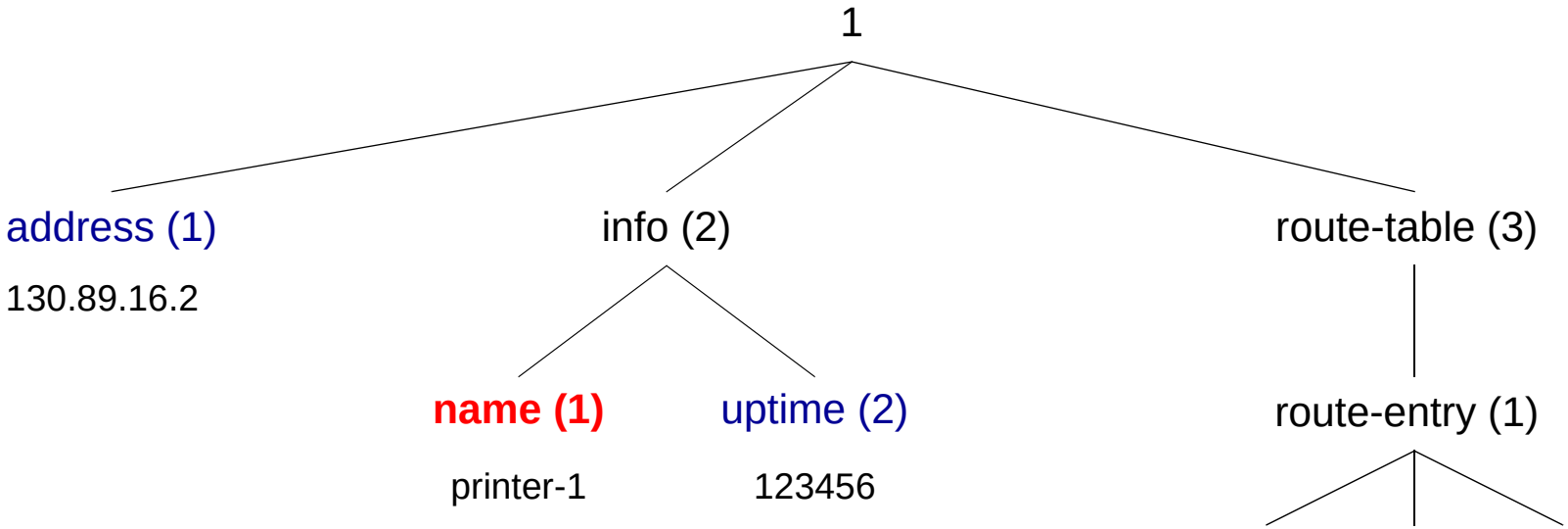


TO REQUEST THE VALUE OF 1 OR MORE VARIABLES

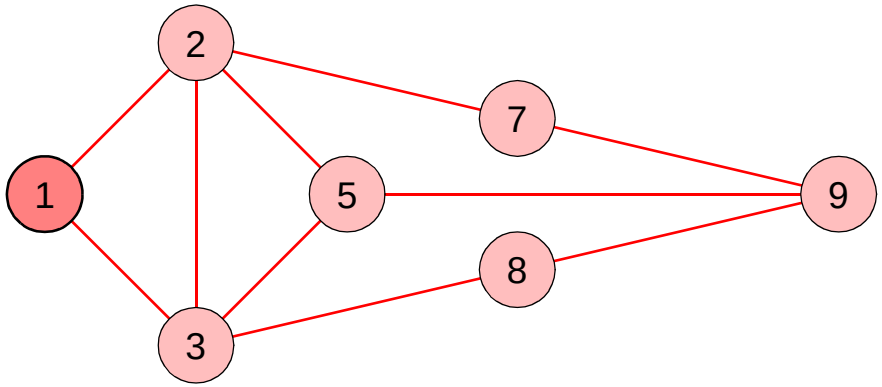
POSSIBLE ERRORS:

- noSuchName ⇨ Object does not exist / Object is not a leaf
- tooBig ⇨ Result does not fit in response PDU
- genErr ⇨ All other causes

EXAMPLE MIB



2	1	2
3	1	3
5	1	2
5	2	3
7	1	2
8	1	3
9	1	2



GET EXAMPLES

get(1.1.0)
response(1.1.0 => *130.89.16.2*)

get(1.2.0)
response(error-status = noSuchName)

get(1.1)
response(error-status = noSuchName)

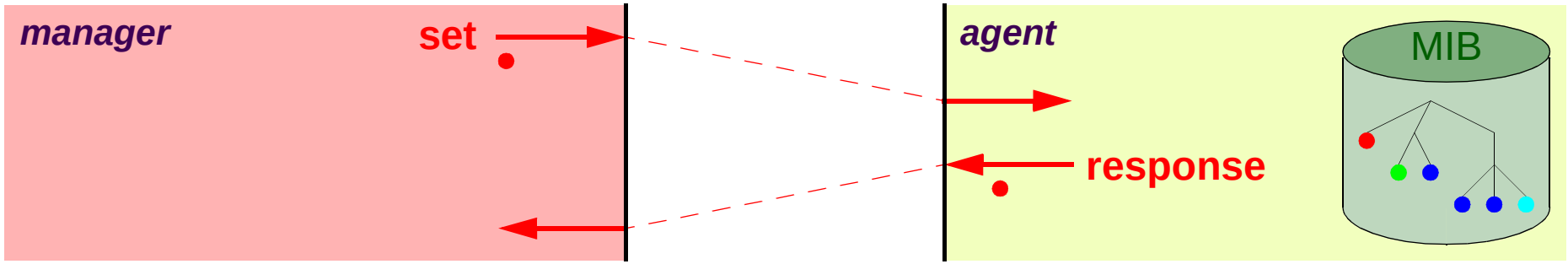
get(1.1.0; 1.2.2.0)
response(1.1.0 => *130.89.16.2*; 1.2.2.0 => *123456*)

get(1.3.1.3.5.1)
response(1.3.1.3.5.1 => *2*)

get(1.3.1.1.5.1)
response(1.3.1.1.5.1 => *5*)

get(1.3.1.1.5.1, 1.3.1.2.5.1, 1.3.1.3.5.1)
response(1.3.1.1.5.1 => *5*, 1.3.1.2.5.1 => *1*, 1.3.1.3.5.1 => *2*)

SET



TO ASSIGN A VALUE TO AN EXISTING OBJECT INSTANCE

TO CREATE NEW INSTANCES
- TABLE ROWS

THE SET REQUEST IS ATOMIC

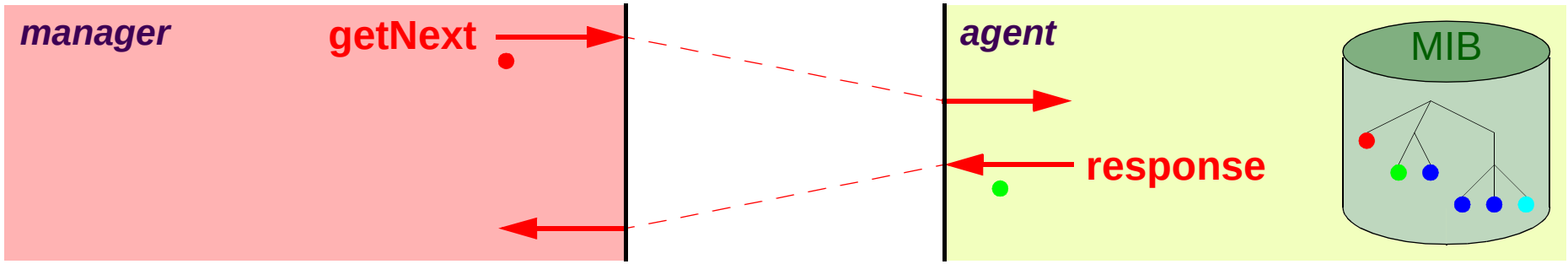
POSSIBLE ERRORS:
- noSuchName
- badValue
- tooBig
- genErr

SET EXAMPLES

set(1.2.1.0 => *my-printer*)
response(noError; 1.2.1.0 => *my-printer*)

set(1.2.1.0 => *my-printer*, 1.2.2.0 => *0*)
response(error-status = noSuchName; error-index = 2)

GET-NEXT



RETRIEVES THE INSTANCE NAME AND VALUE OF THE **NEXT** MIB ELEMENT

TO DISCOVER MIB STRUCTURES

TO RETRIEVE TABLE ROWS

POSSIBLE ERRORS:

- noSuchName (= END OF MIB)
- tooBig
- genErr

GET-NEXT EXAMPLES

getNext(1.1.0)
response(1.2.1.0 => *printer-1*)

getNext(1.2.1.0)
response(1.2.2.0 => *123456*)

getNext(1)
response(1.1.0 => *130.89.16.2*)

getNext(1.3.1.3.5.1)
response(1.3.1.3.5.2 => *3*)

getNext(1.3.1.1; 1.3.1.2; 1.3.1.3)
response(1.3.1.1.2.1 => *2*; 1.3.1.2.2.1 => *1*; 1.3.1.3.2.1 => *2*)

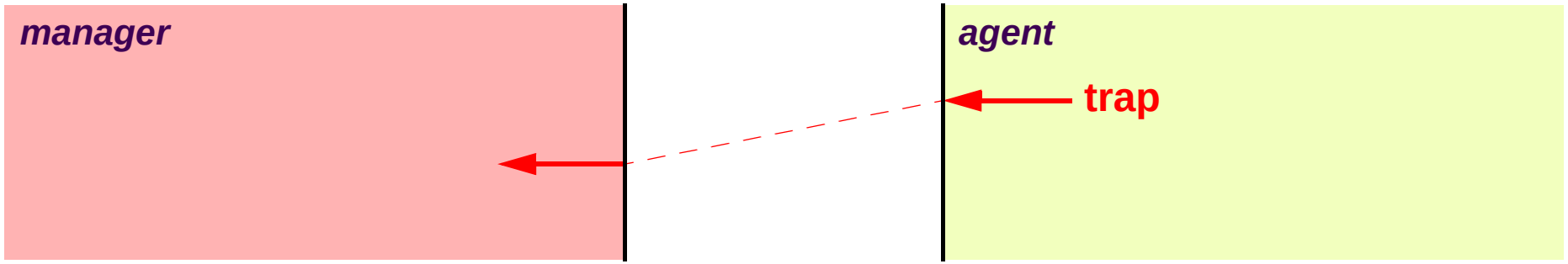
getNext(1.3.1.1.2.1; 1.3.1.2.2.1; 1.3.1.3.2.1)
response(1.3.1.1.3.1 => *3*; 1.3.1.2.3.1 => *1*; 1.3.1.3.3.1 => *3*)

LEXICOGRAPHICAL ORDERING

THE MIB CAN BE CONSIDERED AS AN ORDERED LIST

INSTANCE ID	INSTANCE VALUE
1.1.0	130.89.16.2
1.2.1.0	printer-1
1.2.2.0	123456
1.3.1.1.2.1	2
1.3.1.1.3.1	3
1.3.1.1.5.1	5
•••	•••
1.3.1.1.9.1	9
1.3.1.2.2.1	1
1.3.1.2.3.1	1
•••	•••
1.3.1.2.9.1	1
1.3.1.3.2.1	2
1.3.1.3.3.1	3
1.3.1.3.5.1	2
1.3.1.3.5.2	3
1.3.1.3.7.1	2
•••	•••

TRAP



TO SIGNAL AN EVENT

TRAP RECEPTION IS NOT CONFIRMED
(THUS UNRELIABLE)

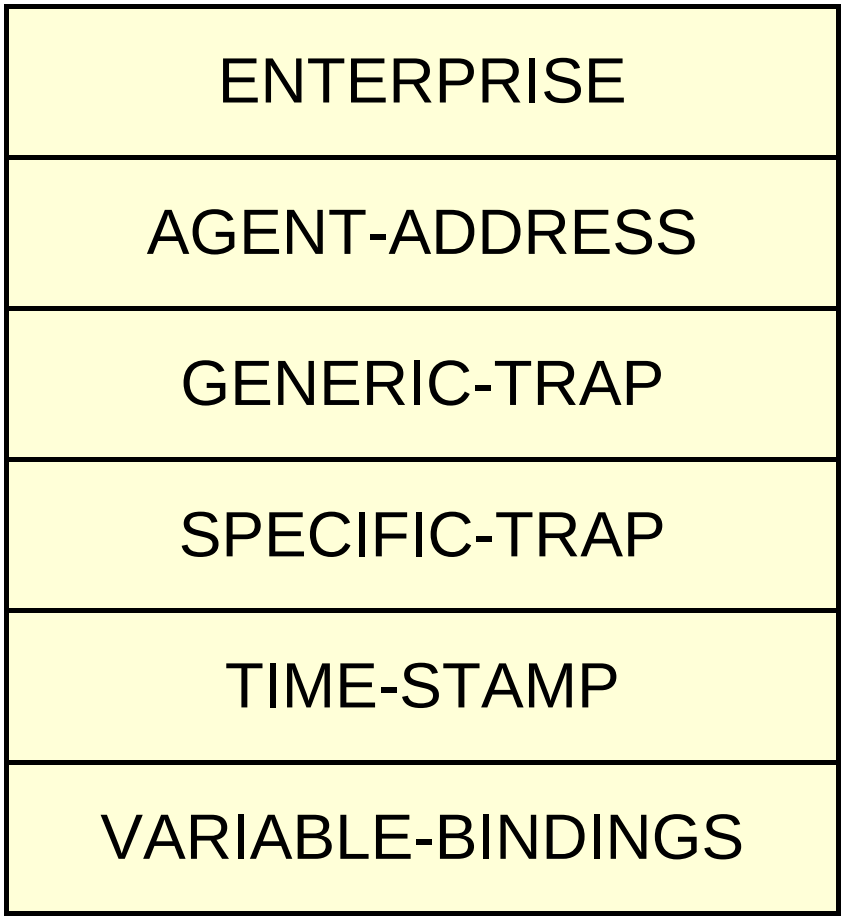
POLLING REMAINS NECESSARY

AGENTS MAY BE CONFIGURED SUCH THAT:
- NO TRAPS WILL BE TRANSMITTED
- TRAPS WILL BE TRANSMITTED TO CERTAIN MANAGERS

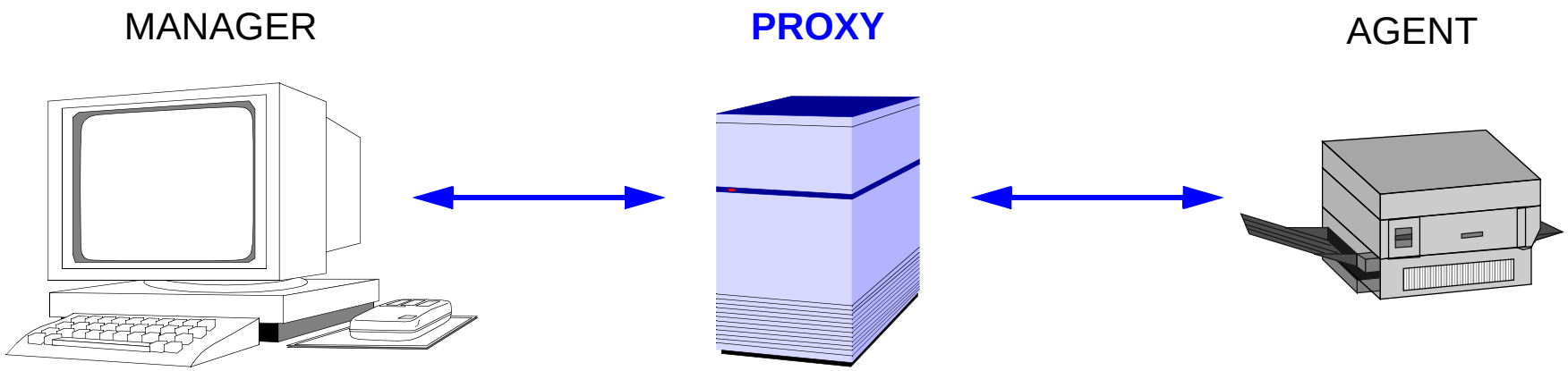
DEFINED TRAPS

- COLDSTART
- WARMSTART
- LINKDOWN
- LINKUP
- AUTHENTICATION FAILURE
- EGPNEIGHBOURLOSS
- ENTERPRISESPECIFICTRAP

TRAP - PDU FORMAT



PROXY MANAGEMENT

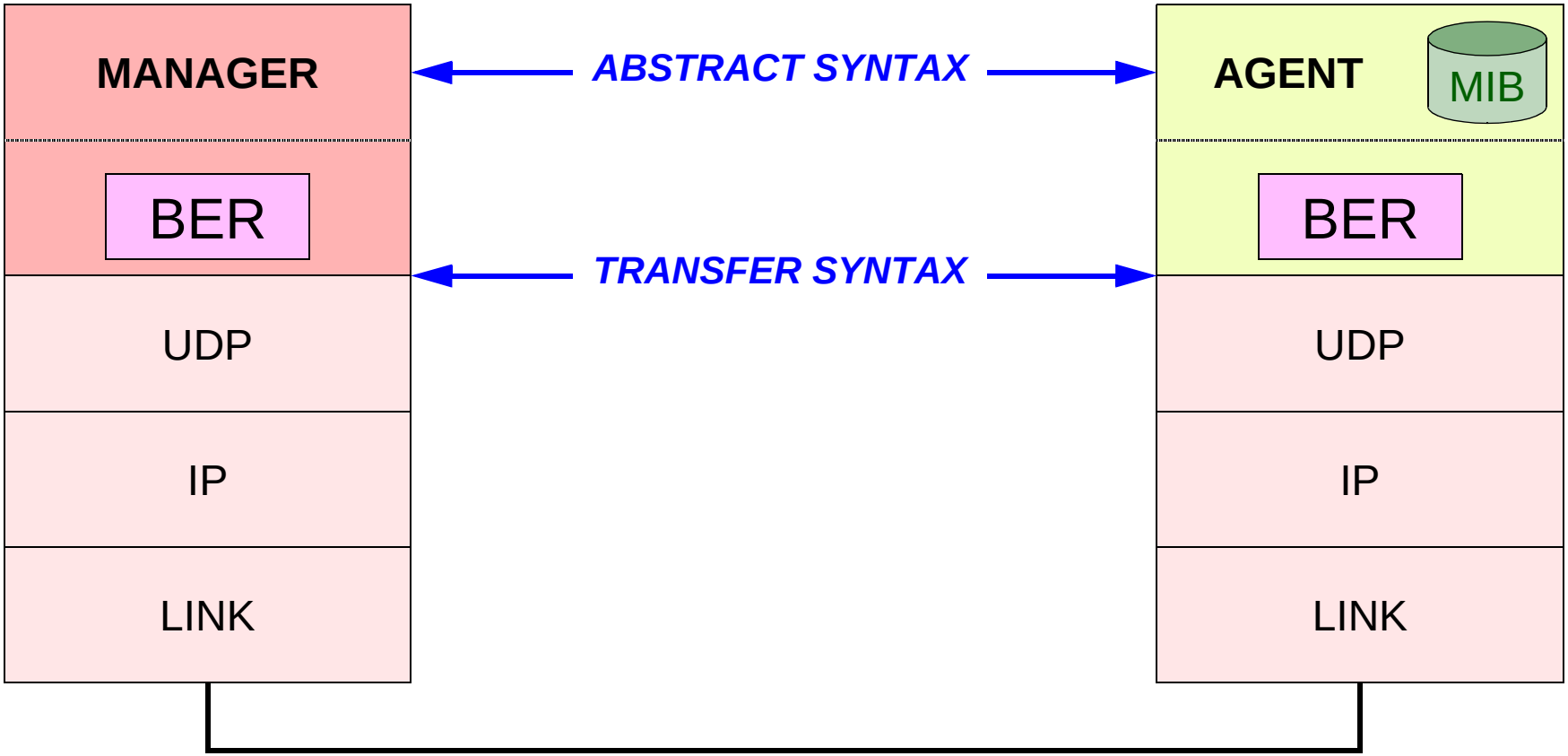


TERM HAS TRADITIONALLY BEEN USED FOR DEVICES THAT :

- TRANSLATE BETWEEN DIFFERENT TRANSPORT DOMAINS
- TRANSLATE BETWEEN DIFFERENT SNMP VERSIONS
- TRANSLATE BETWEEN SNMP AND OTHER MANAGEMENT PROTOCOLS
- AGGREGATE LOW LEVEL MANAGEMENT INFO INTO HIGH LEVEL INFO
- ETC.

NOWADAYS THE TERM DENOTES A DEVICE
THAT FORWARDS SNMP MESSAGES,
BUT DOESN'T LOOK AT THE INDIVIDUAL OBJECTS

SNMP MESSAGE ENCODING



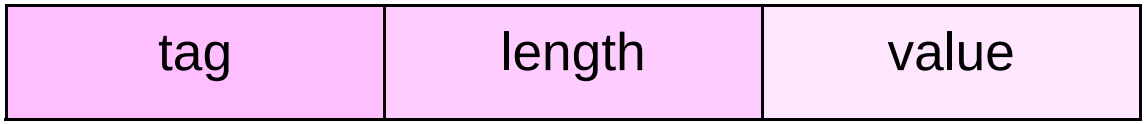
THE DESCRIPTION OF MIBS AND MESSAGE FORMATS
IS BASED ON THE ASN.1 SYNTAX

THE MAPPING FROM AN ABSTRACT SYNTAX UPON A TRANSFER SYNTAX
IS DEFINED BY THE BASIC ENCODING RULES (BER)

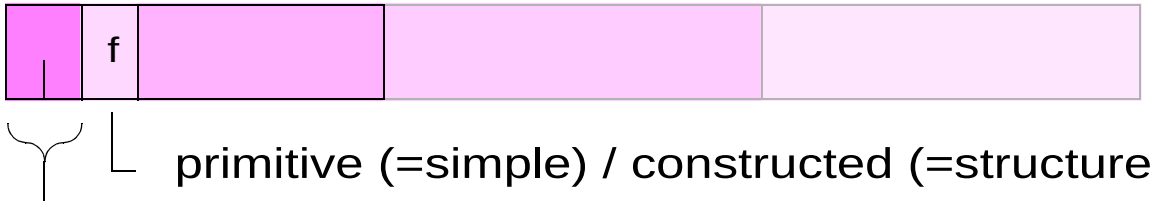
BASIC ENCODING RULES

EACH ASN.1 VALUE IS ENCODED AS AN OCTET STRING

THIS ENCODING RESULTS INTO A SEQUENCE OF
TAG, LENGTH, VALUE
STRUCTURES



TAG FIELD



Universal tags

BIT PATTERN	ASN.1 TYPE
00 0 0 0010	INTEGER
00 0 0 0100	OCTET STRING
00 0 0 0110	OBJECT IDENTIFIER

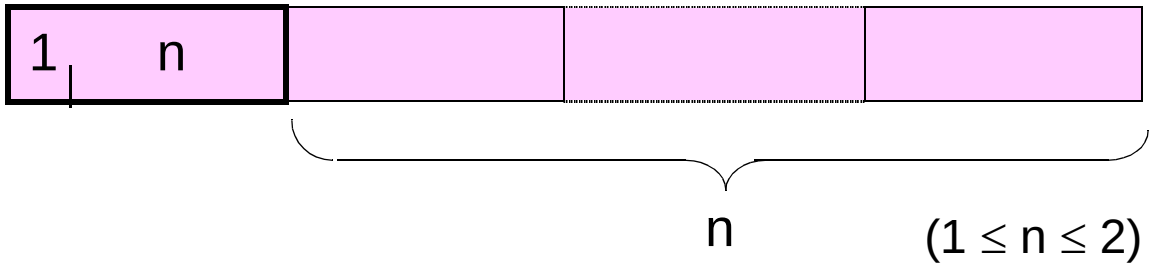
Application-wide tags

BIT PATTERN	APPLICATION TYPE
01 0 0 0000	IpAddress
01 0 0 0001	Counter32
01 0 0 0010	Gauge32
01 0 0 0010	Unsigned32
01 0 0 0011	TimeTicks
01 0 0 0100	Opaque
01 0 0 0110	Counter64

LENGTH FIELD

SHORT FORM:

LONG FORM:



SNMPv2

OVERVIEW:

LIMITATIONS OF SNMPv1

HISTORY OF SNMPv2
- HIERARCHIES
- SECURITY

SNMPv2 PROTOCOL OPERATIONS

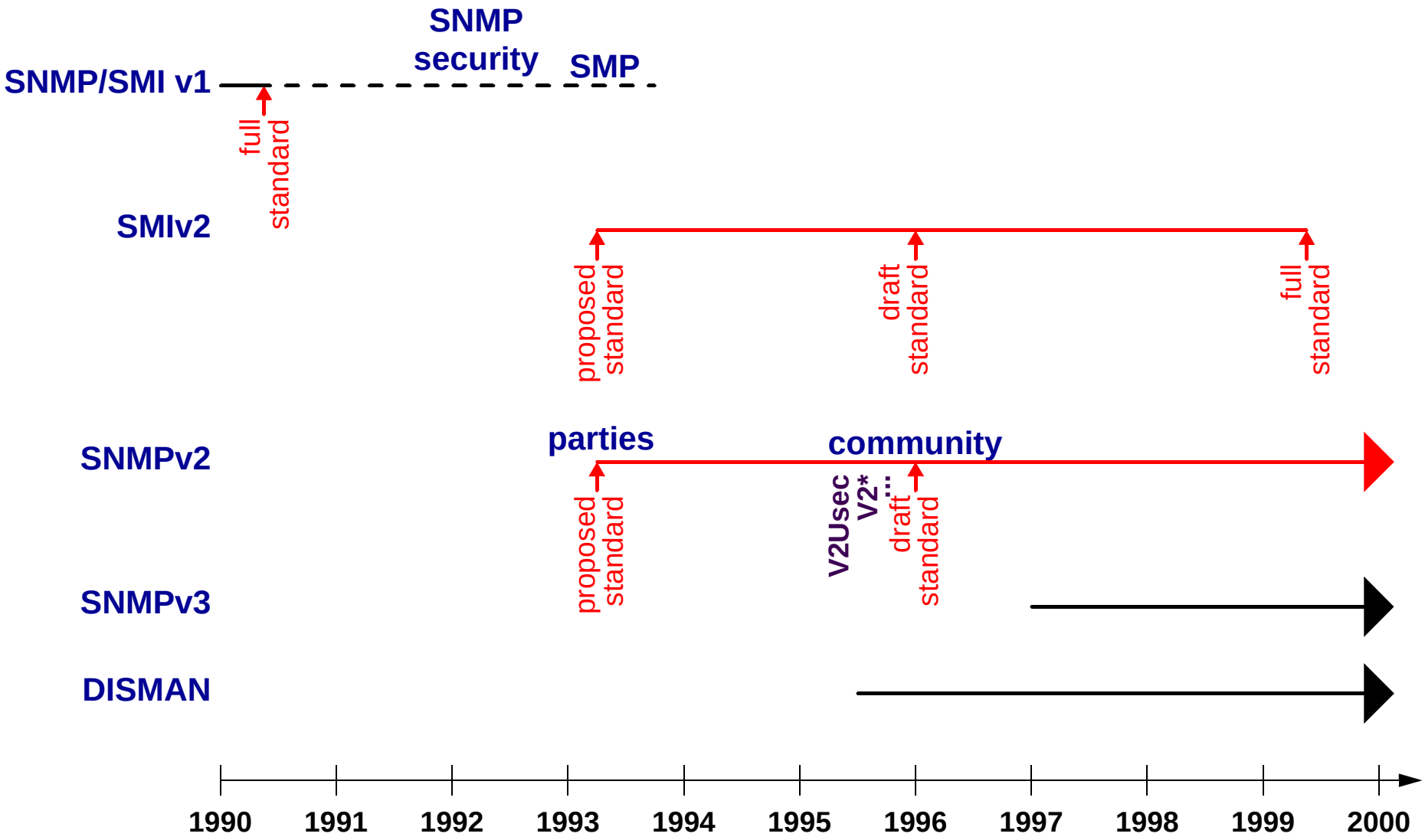
TRANSPORT INDEPENDENCE

RFCs

LIMITATIONS OF SNMPv1

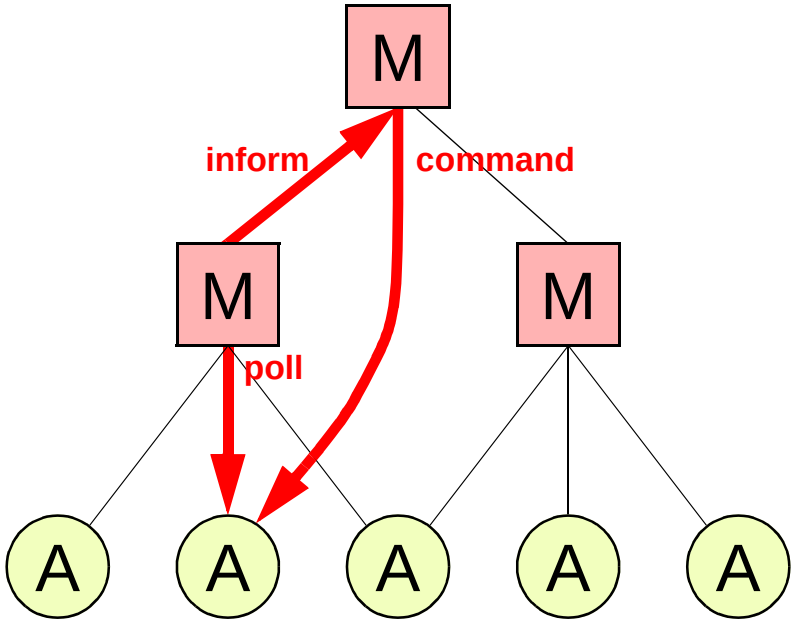
- UNDOCUMENTED RULES
- LIMITED ERROR CODES
- LIMITED DATA TYPES
- LIMITED NOTIFICATIONS
- LIMITED PERFORMANCE
- TRANSPORT DEPENDENCE
- LACK OF HIERARCHIES
- LACK OF SECURITY

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:

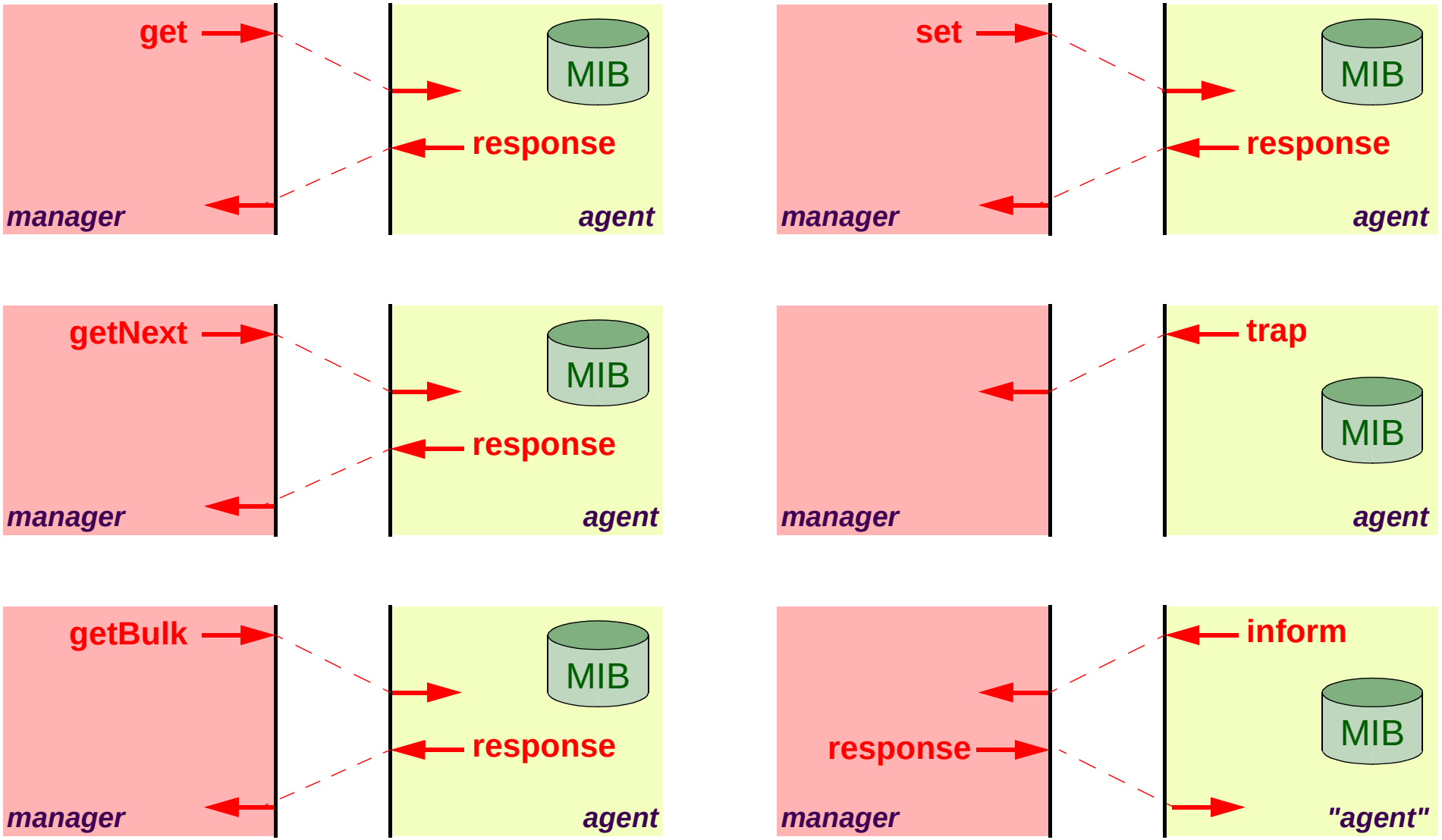
GENERAL AGREEMENT THAT PARTY BASED MODEL WAS TOO COMPLEX!
MANY NEW PROPOSALS APPEARED:

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

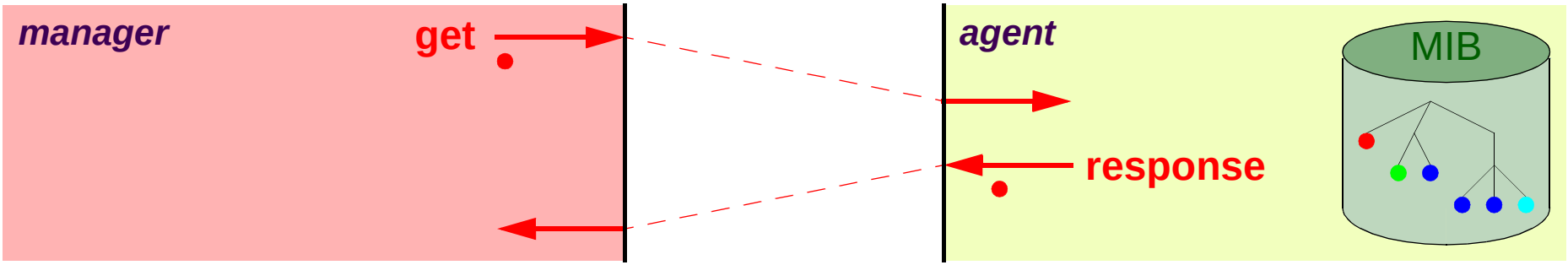
1997:

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 EXAMPLES

get(1)
response(error-status => noError, 1.2 => noSuchObject)

get(1.1)
response(error-status => noError, 1.2.0 => noSuchInstance)

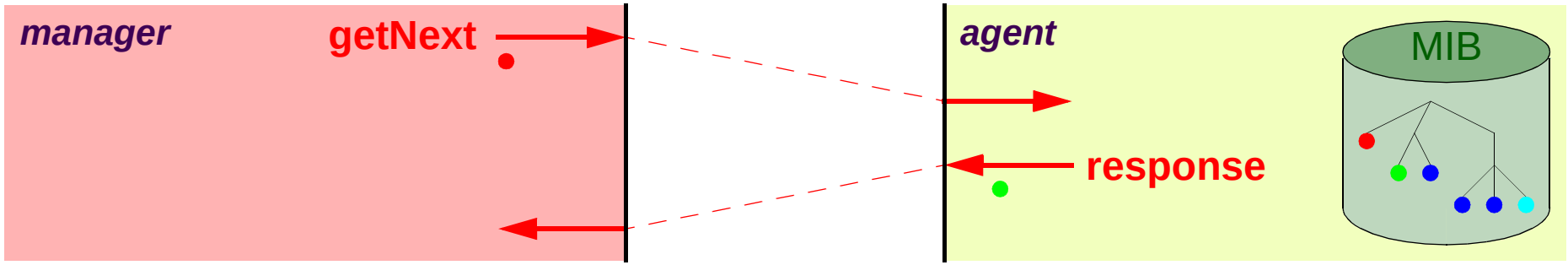
get(1.1.9)
response(error-status => noError, 1.2.0 => noSuchInstance)

get(1.2)
response(error-status => noError, 1.4.0 => noSuchObject)

get(1.4.0)
response(error-status => noError, 1.4.0 => noSuchObject)

get(1.1.0, 1.4.0)
response(error-status => noError, 1.1.0 => *130.89.16.2,* 1.4.0 => noSuchObject)

GET-NEXT



SIMILAR TO SNMPv1, EXCEPT FOR "EXCEPTIONS"

POSSIBLE EXCEPTIONS:

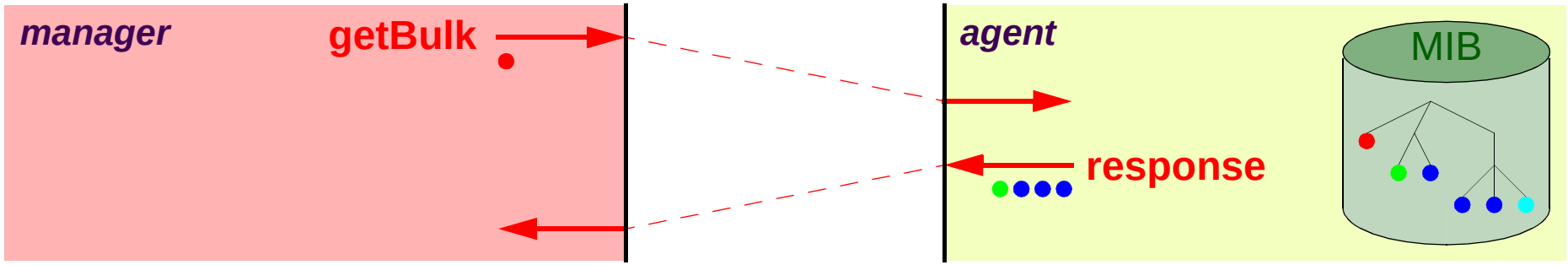
- endOfMibView

EXAMPLE

getNext(1.4.0)

response(error-status => noError, 1.4.0 => endOfMibView)

GET-BULK

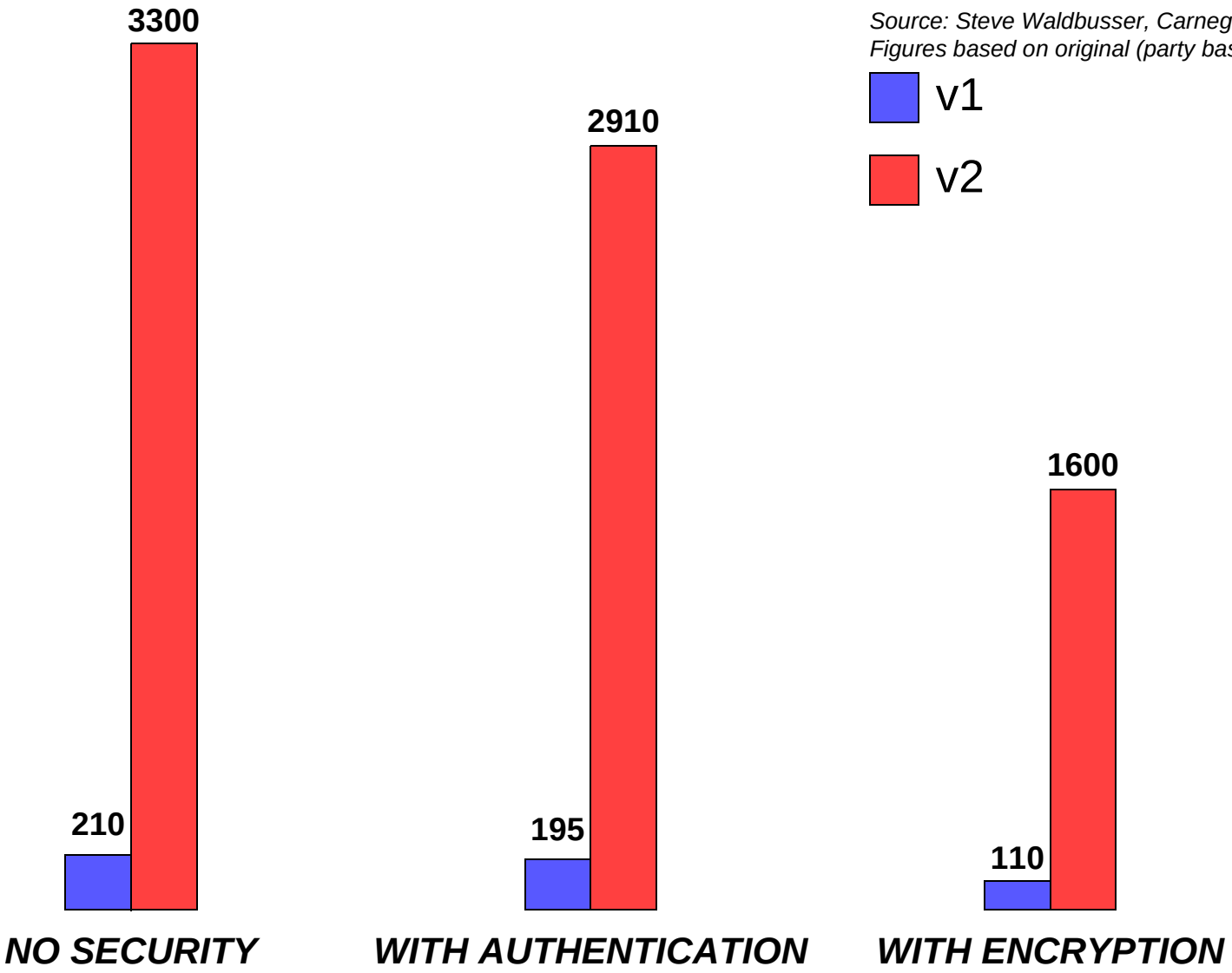


NEW IN SNMPv2

TO RETRIEVE A LARGE NUMBER OF VARBINDS

IMPROVES PERFORMANCE!

GETBULK PERFORMANCE



Source: Steve Waldbusser, Carnegie-Mellon University
Figures based on original (party based) SNMPv2

GET-BULK

getBulk REQUEST HAS TWO ADDITIONAL PARAMETERS:

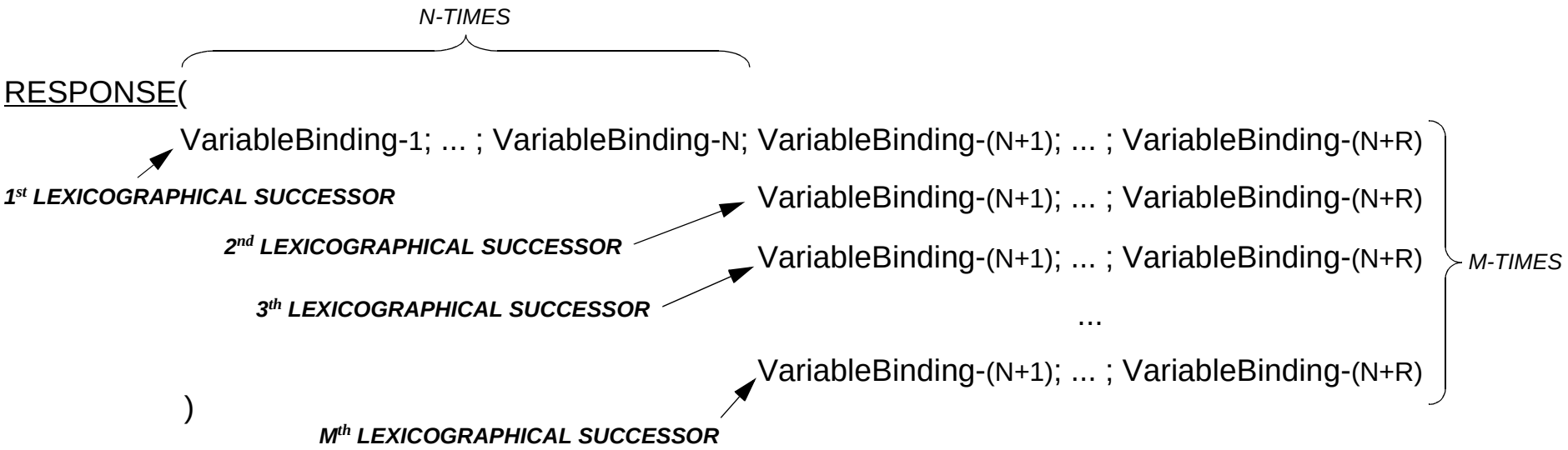
- non-repeators
- max-repetitions

- 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

GET-BULK

REQUEST(non-repeaters = N; max-repetitions = M;
VariableBinding-1; ... ; VariableBinding-N; VariableBinding-(N+1); ... ; VariableBinding-(N+R)
)



GET-BULK EXAMPLE

getBulk(max-repetitions = 4; 1.1)

response(
1.1.0 => *130.89.16.2*
1.2.1.0 => *printer-1*
1.2.2.0 => *123456*
1.3.1.1.2.1 => *2*)

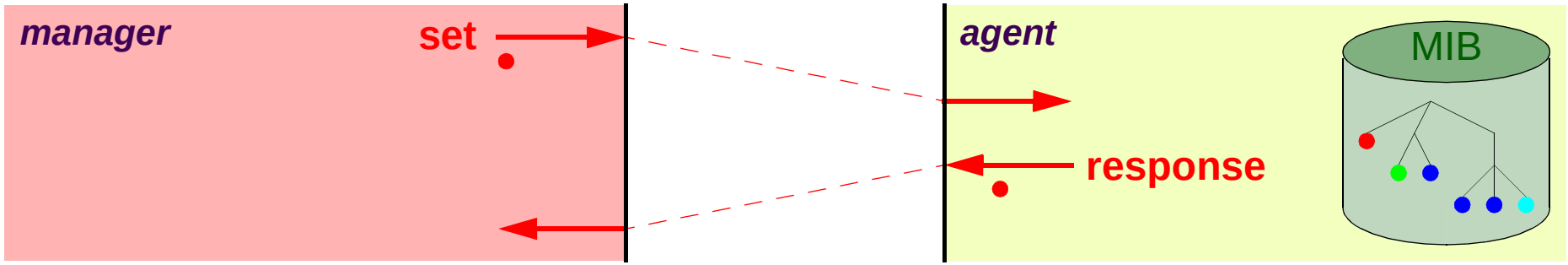
getBulk(max-repetitions = 3; 1.3.1.1; 1.3.1.2; 1.3.1.3)

response(

1.3.1.1.2.1 => *2*;	1.3.1.2.2.1 => *1*;	1.3.1.3.2.1 => *2*
1.3.1.1.3.1 => *3*;	1.3.1.2.3.1 => *1*;	1.3.1.3.3.1 => *3*
1.3.1.1.5.1 => *5*;	1.3.1.2.5.1 => *1*;	1.3.1.3.5.1 => *2*

)

SET



SIMILAR TO SNMPv1

CONCEPTUAL TWO PHASE COMMIT:

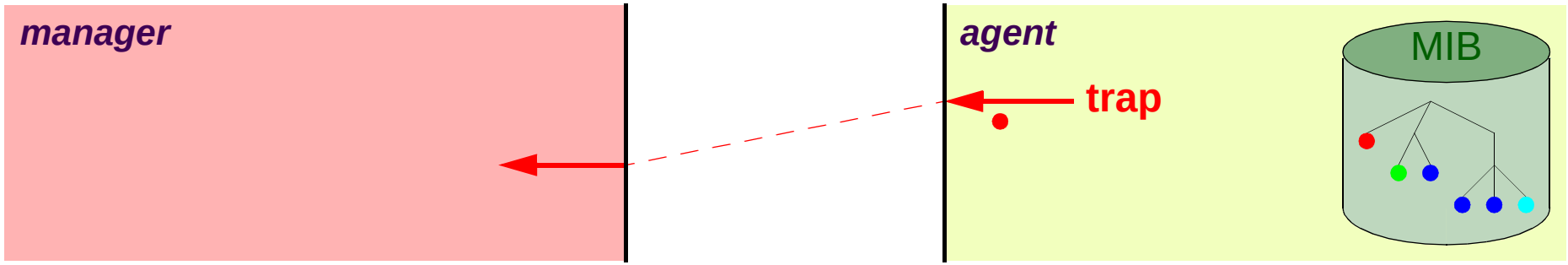
- PHASE 1: PERFORM VARIOUS CHECKS
- PHASE 2: PERFORM THE ACTUAL SET

MANY NEW ERROR CODES ARE DEFINED

NEW ERROR CODES FOR SETS

	SNMPv1	SNMPv2
PHASE 1:	badValue	wrongValue
	badValue	wrongEncoding
	badValue	wrongType
	badValue	wrongLength
	badValue	inconsistentValue
	noSuchName	noAccess
	noSuchName	notWritable
	noSuchName	noCreation
	noSuchName	inconsistentName
	genErr	resourceUnavailable
	genErr	genErr
PHASE 2:	genErr	CommitFailed
	genErr	undoFailed

TRAP



SNMPv1:

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

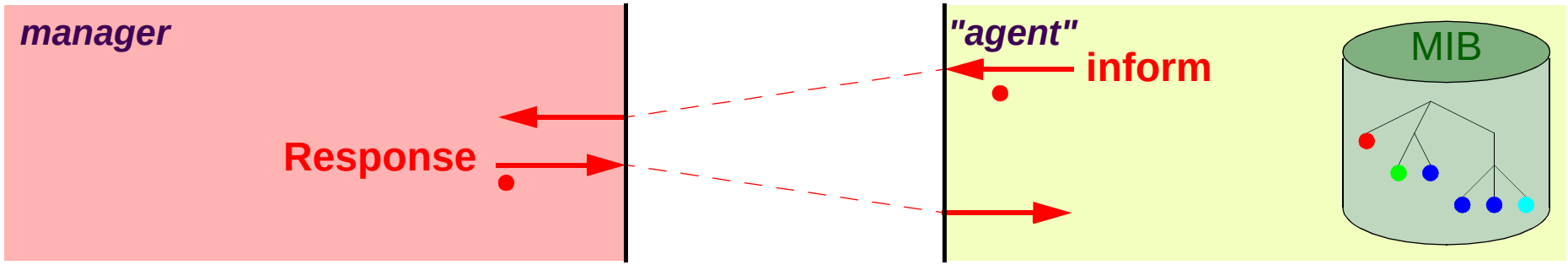
SNMPv2:

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

EXAMPLE OF NOTIFICATION TYPE MACRO

```
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}
```

INFORM



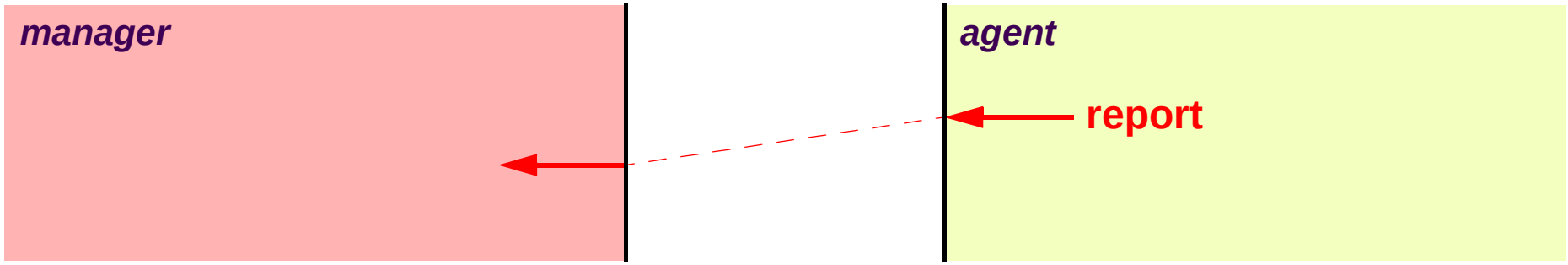
CONFIRMED TRAP

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

TRANSPORT DEPENDANCE

SNMPv1:

- UDP

SNMPv2:

- UDP
- CLNS (OSI)
- DDP (APPLETALK)
- IPX

SNMPv2 RFCs

COMMUNICATION MODEL

- DRAFT STANDARD
- RFC 1905, RFC1906

SECURITY MODEL - SNMPv2C:

- COMMUNITY BASED SNMP
- SAME ‘SECURITY MECHANISMS’ AS SNMPv1
- EXPERIMENTAL STATUS
- RFC 1901

SECURITY MODEL - SNMPv2U:

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

INFORMATION MODEL:

- STANDARD
- RFC2578, RFC2579, RFC2580

SNMPv2 - SUMMARY

IMPROVED COMMUNICATION MODEL

- TRAPS HAVE SAME FORMAT AS OTHER PDUS
- GET-BULK PDU
- ADDITIONAL ERROR CODES FOR SETS

TWO SECURITY MODELS

- SNMPv2C: COMMUNITY BASED
- SNMPv2U: USER BASED

INDEPENDENCE OF UNDERLYING TRANSPORT

- MIB-II SPLIT INTO MODULES

SECURITY AND HIERARCHIES TO SNMPv3 & DISMAN

IMPROVED INFORMATION MODEL (SMIv2)

- ADDITIONAL DATA TYPES
- TEXTUAL CONVENTIONS

E.G. ROW STATUS

- NOTIFICATIONS

SNMPv3

OVERVIEW:

DESIGN DECISIONS

ARCHITECTURE

SNMP MESSAGE STRUCTURE

SECURE COMMUNICATION
- USER SECURITY MODEL (USM)

ACCESS CONTROL
- VIEW BASED ACCESS CONTROL MODEL (VACM)

IMPLEMENTATIONS

RFCs

DESIGN DECISIONS

ADDRESS THE NEED FOR SECURY 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

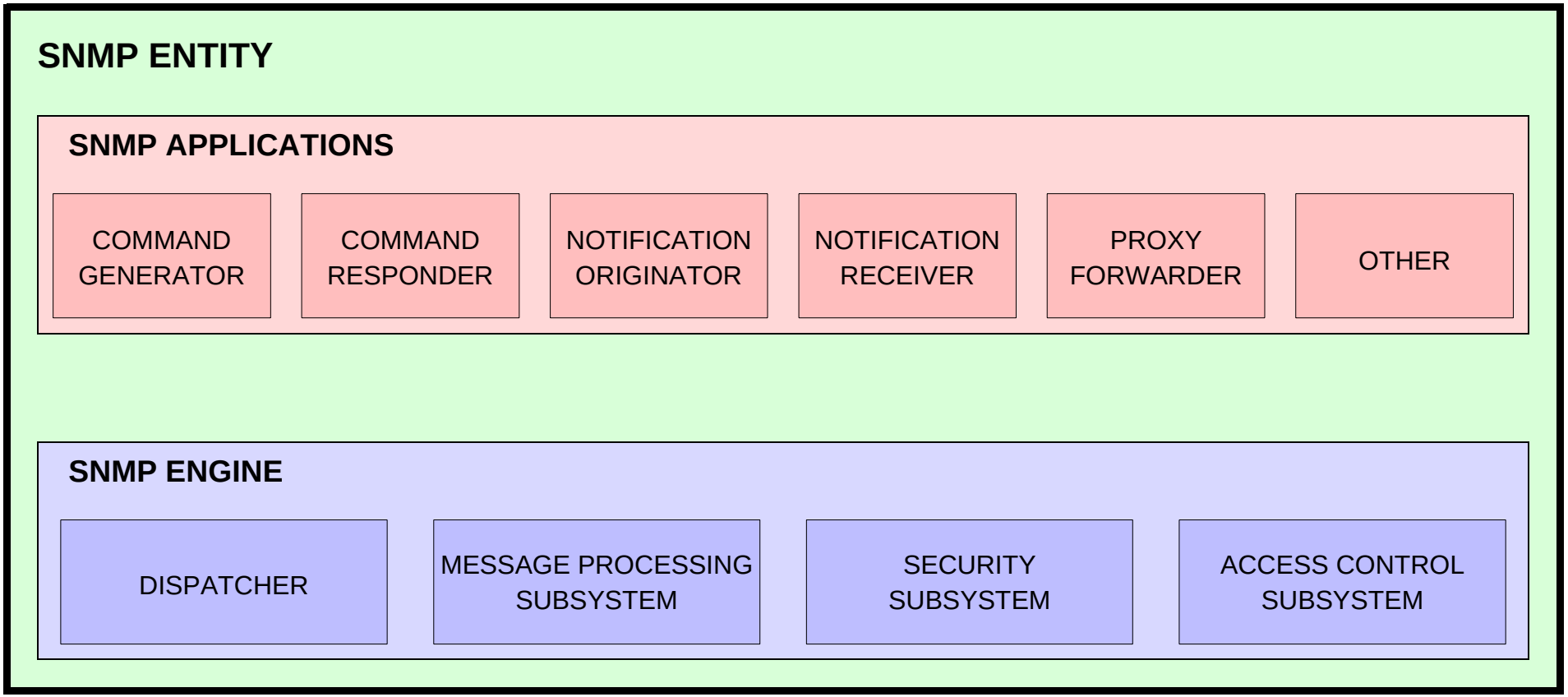
KEEP SNMP AS SIMPLE AS POSSIBLE

ALLOW FOR MINIMAL IMPLEMENTATIONS

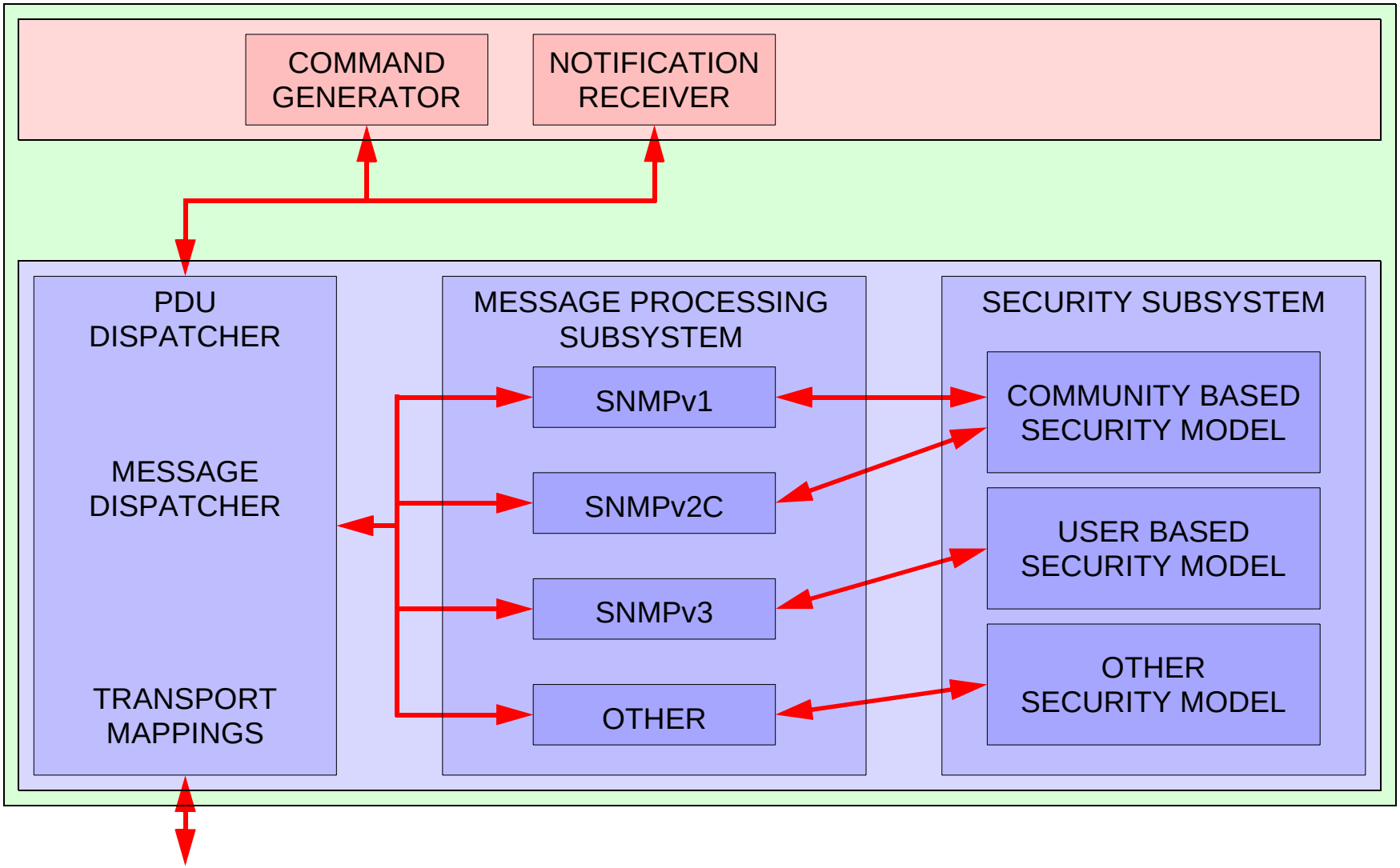
SUPPORT ALSO THE MORE COMPLEX FEATURES,
WHICH ARE REQUIRED IN LARGE NETWORKS

RE-USE EXISTING SPECIFICATIONS, WHENEVER POSSIBLE

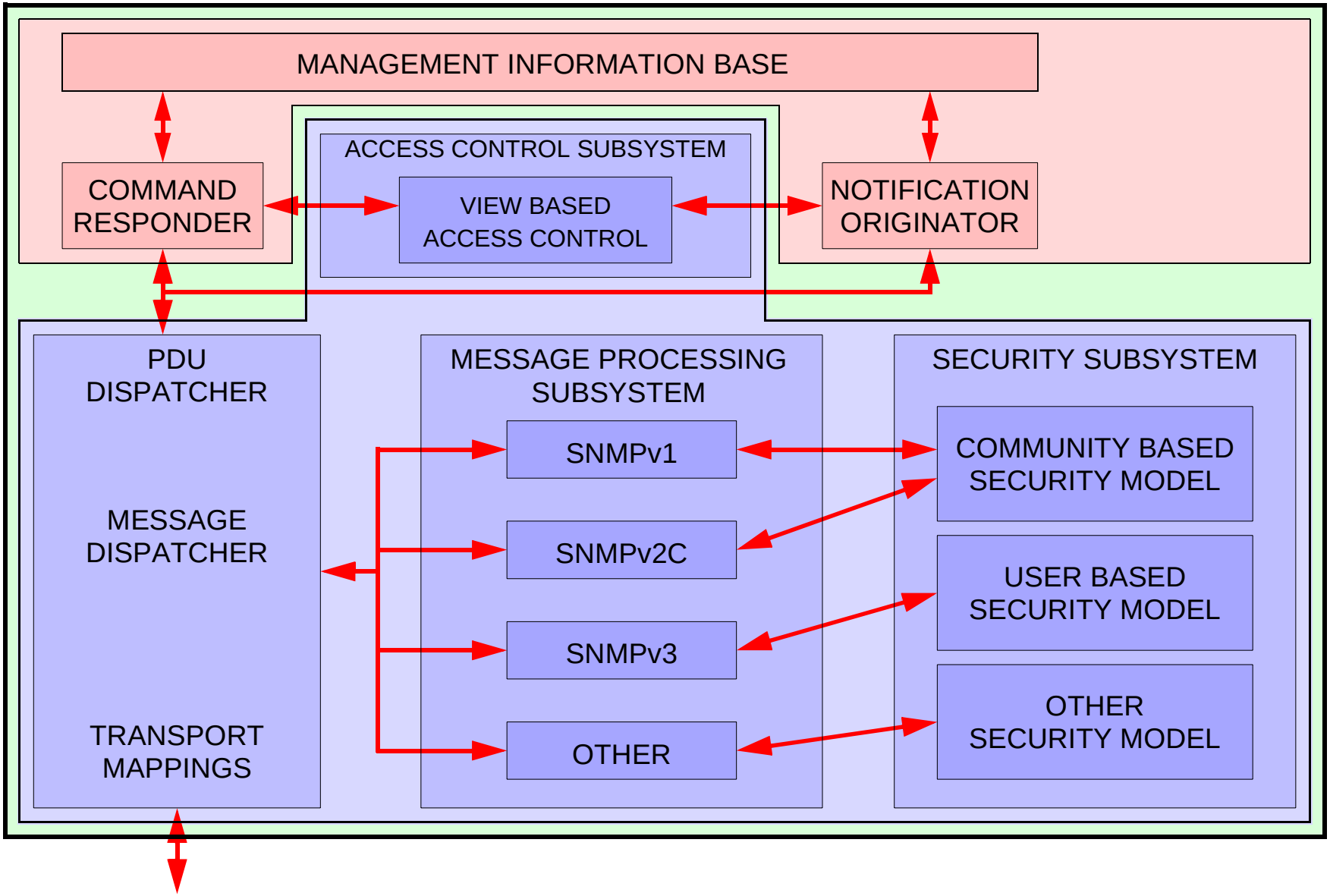
SNMPv3 ARCHITECTURE



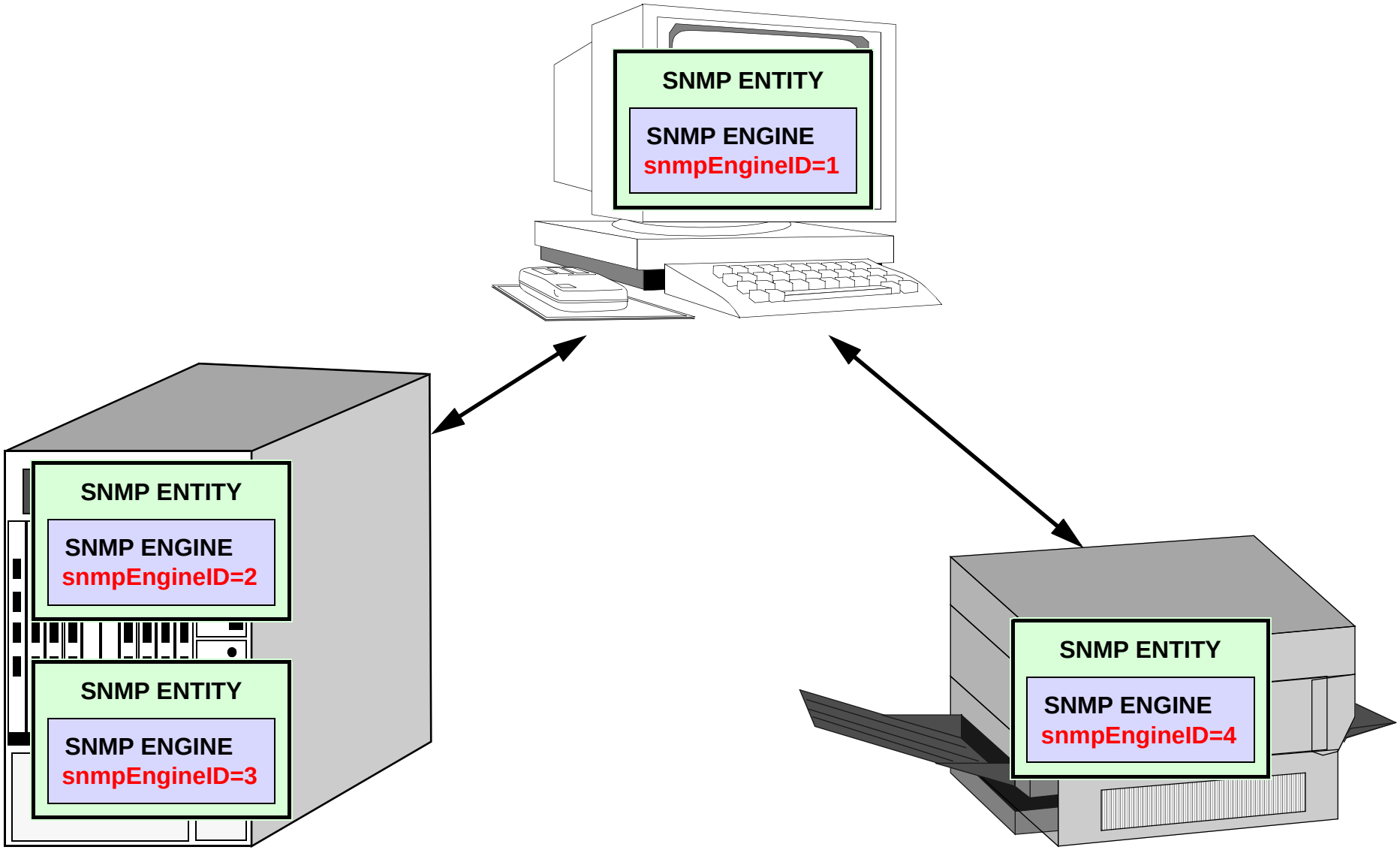
SNMPv3 ARCHITECTURE: MANAGER



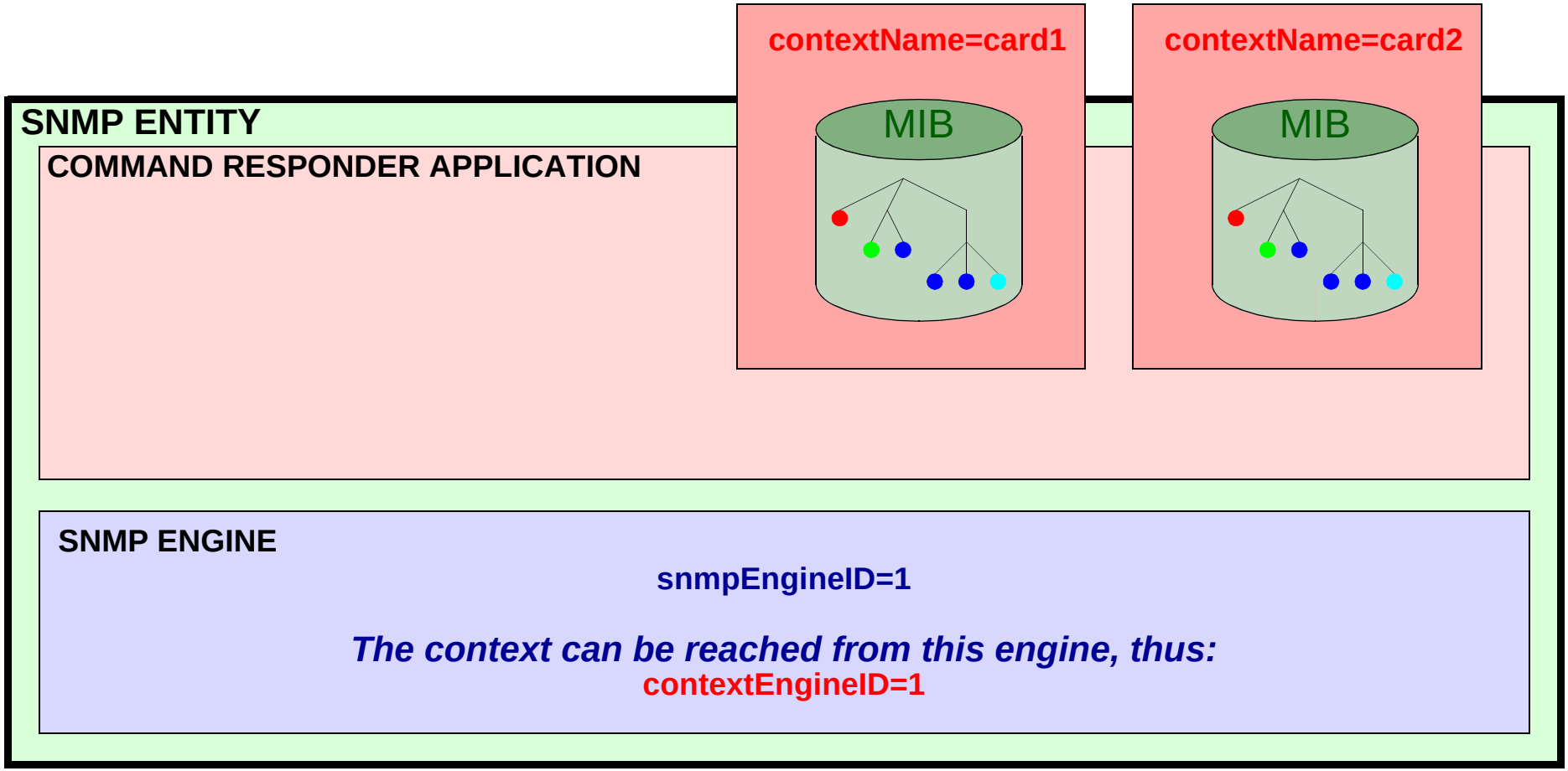
SNMPv3 ARCHITECTURE: AGENT



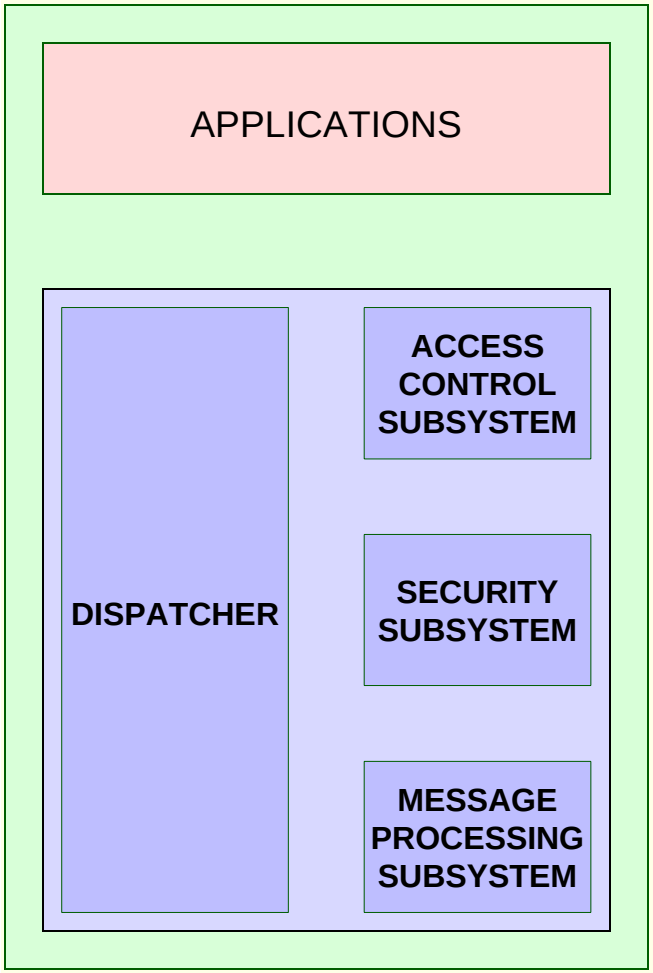
CONCEPTS: snmpEngineID

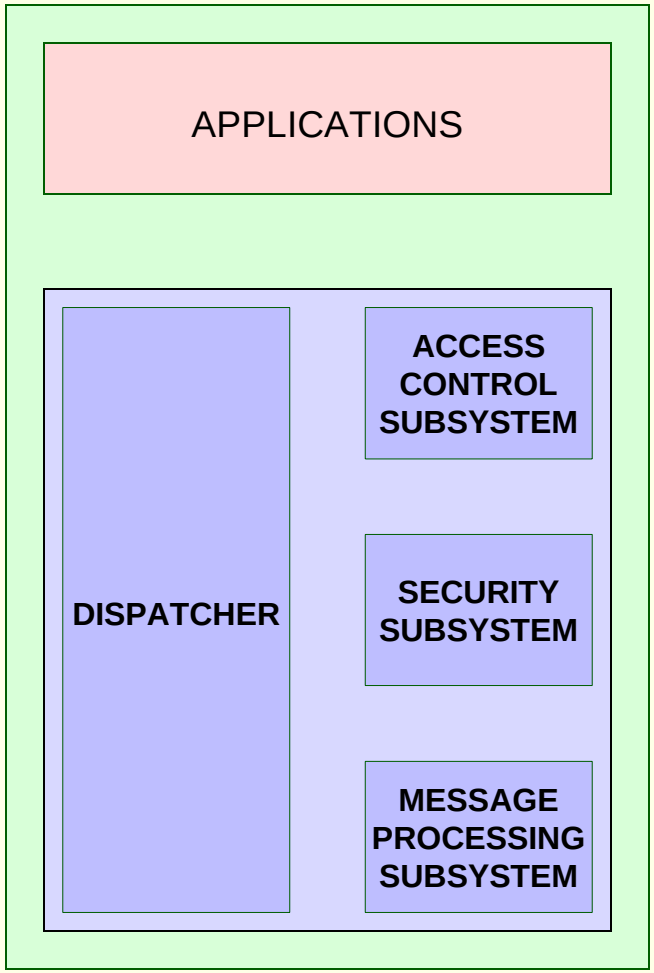


CONCEPTS: Context



PRIMITIVES BETWEEN MODULES

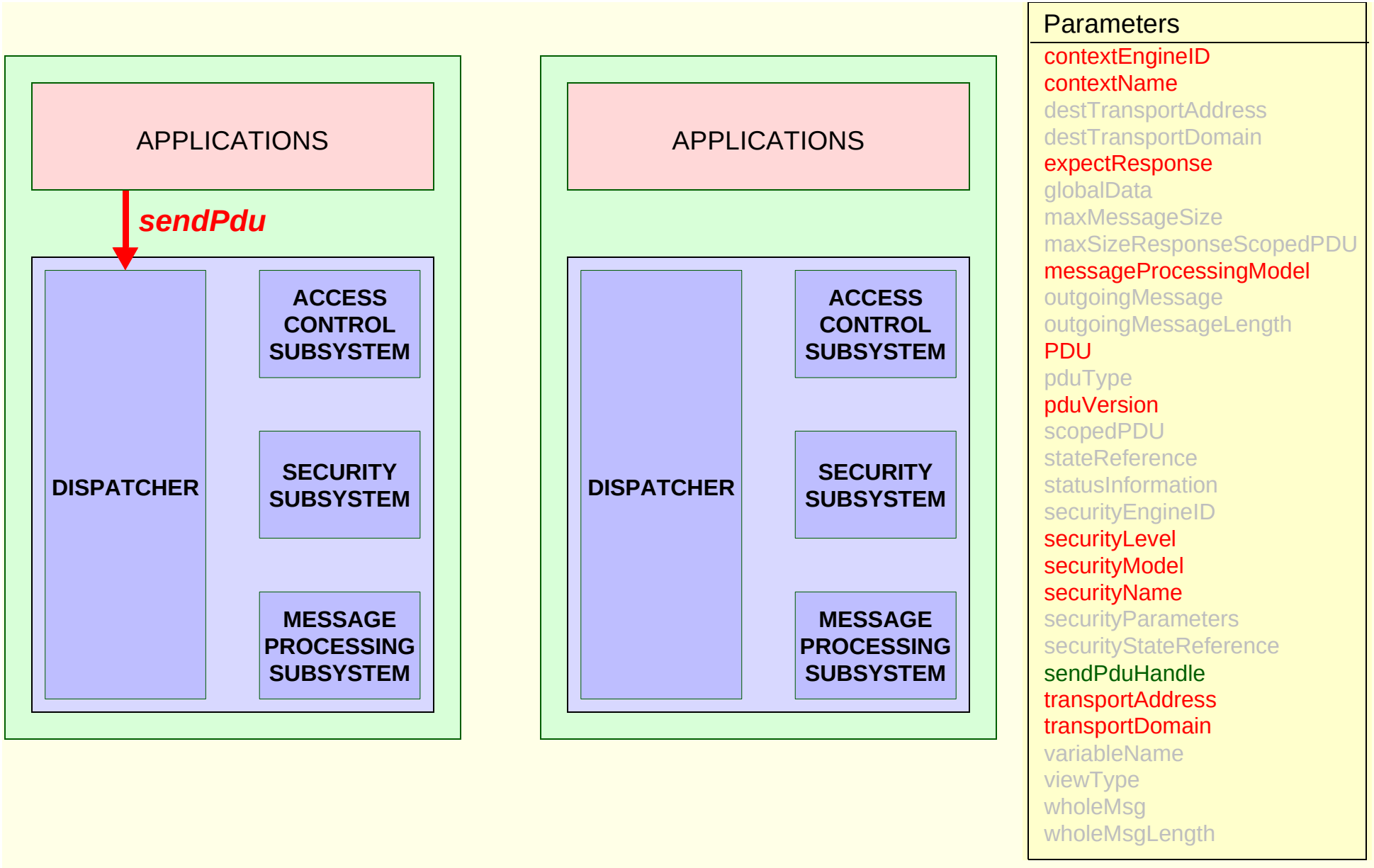




Parameters

contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

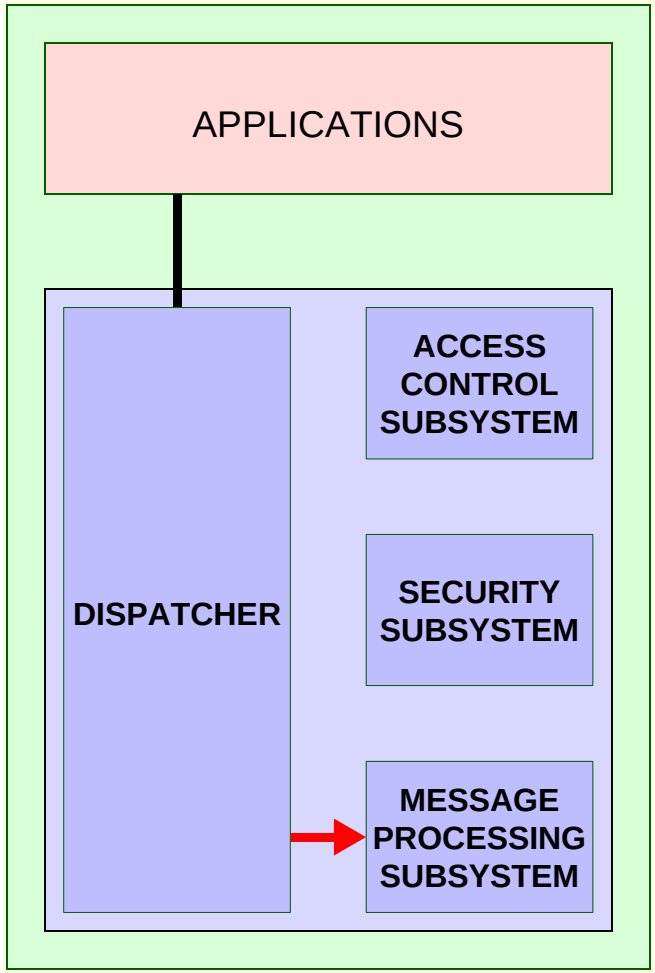
sendPdu

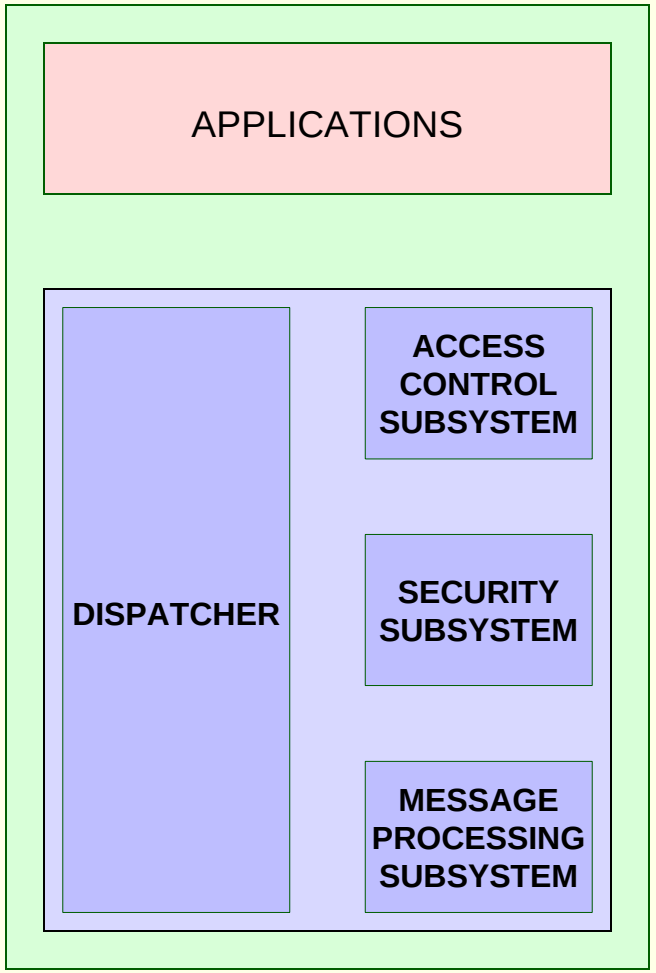


Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

prepareOutgoingMessage

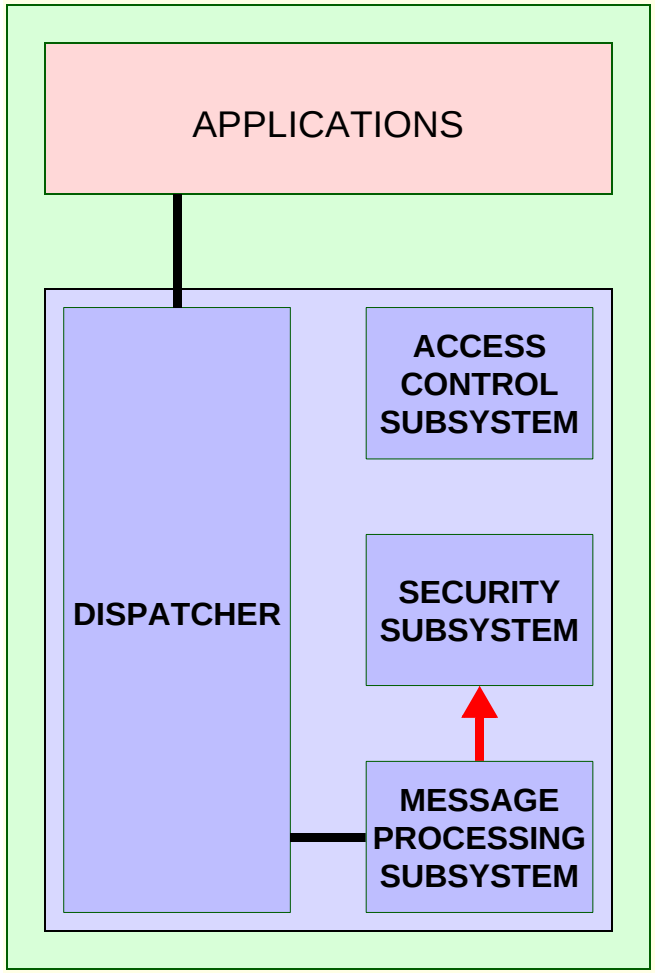


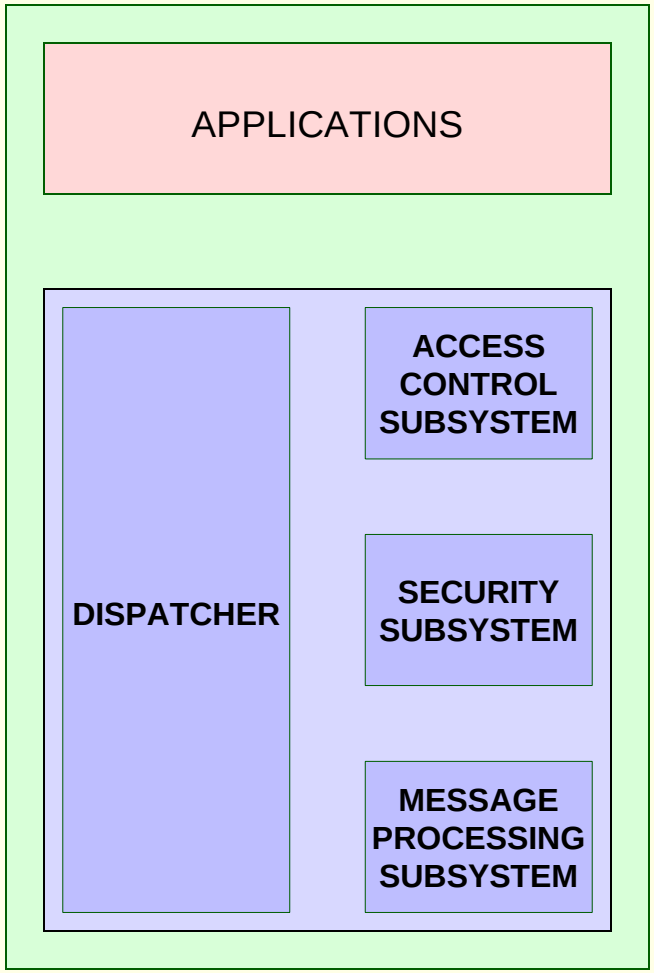


prepareOutgoingMessage

Parameters
contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

generateRequestMsg



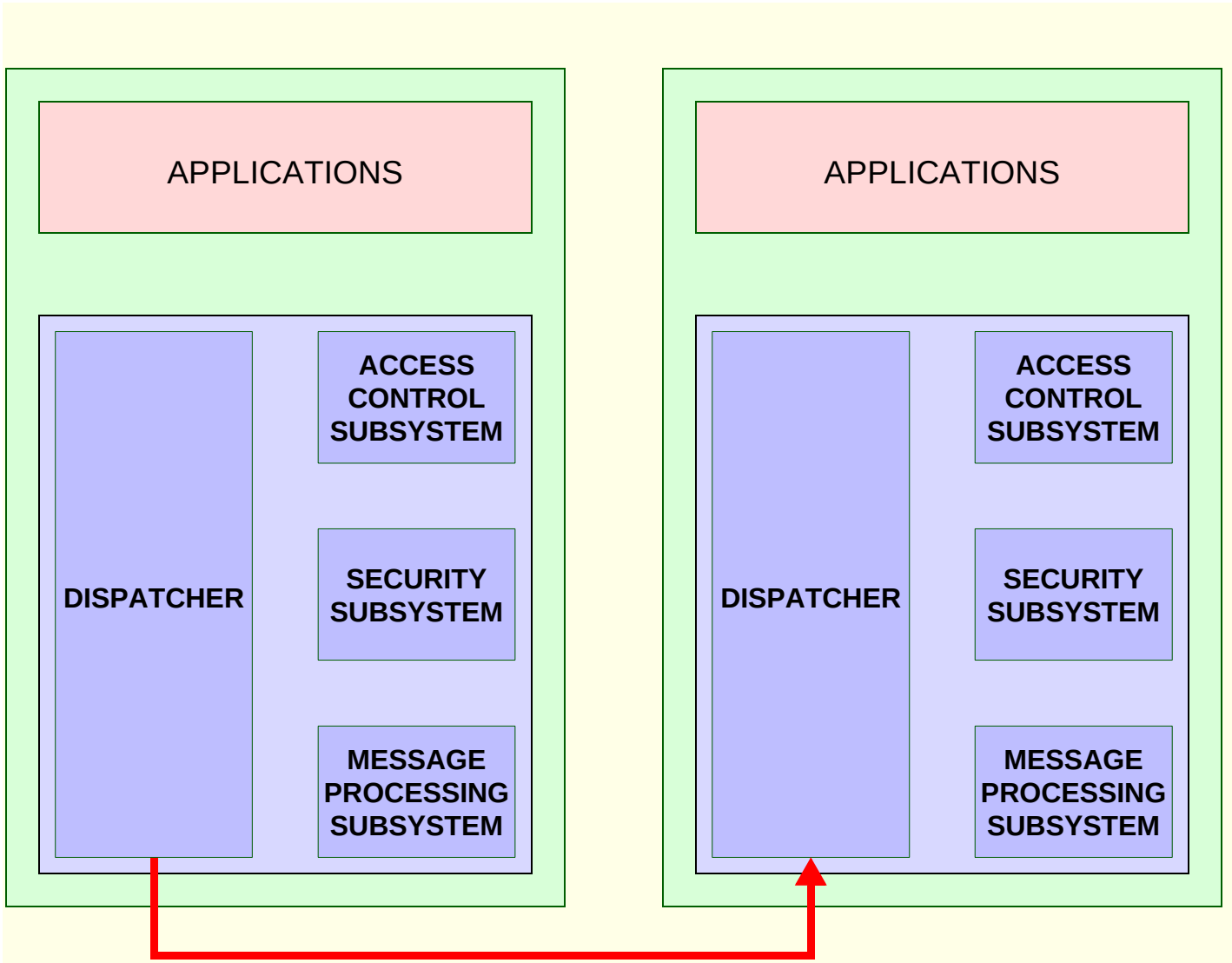


generateRequestMsg

Parameters

contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

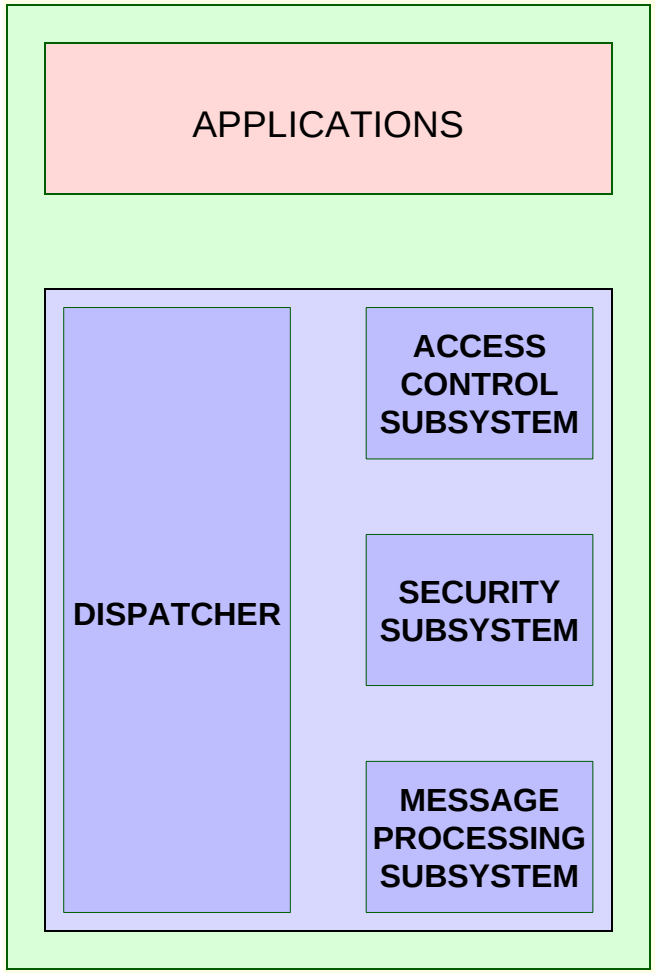
send / receive

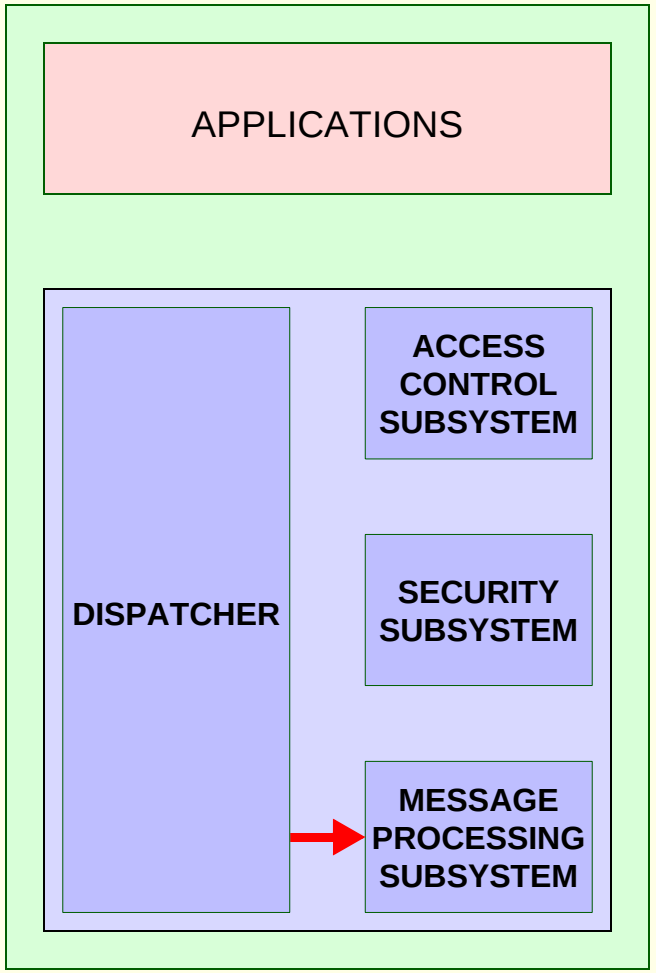


Parameters

contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

prepareDataElements

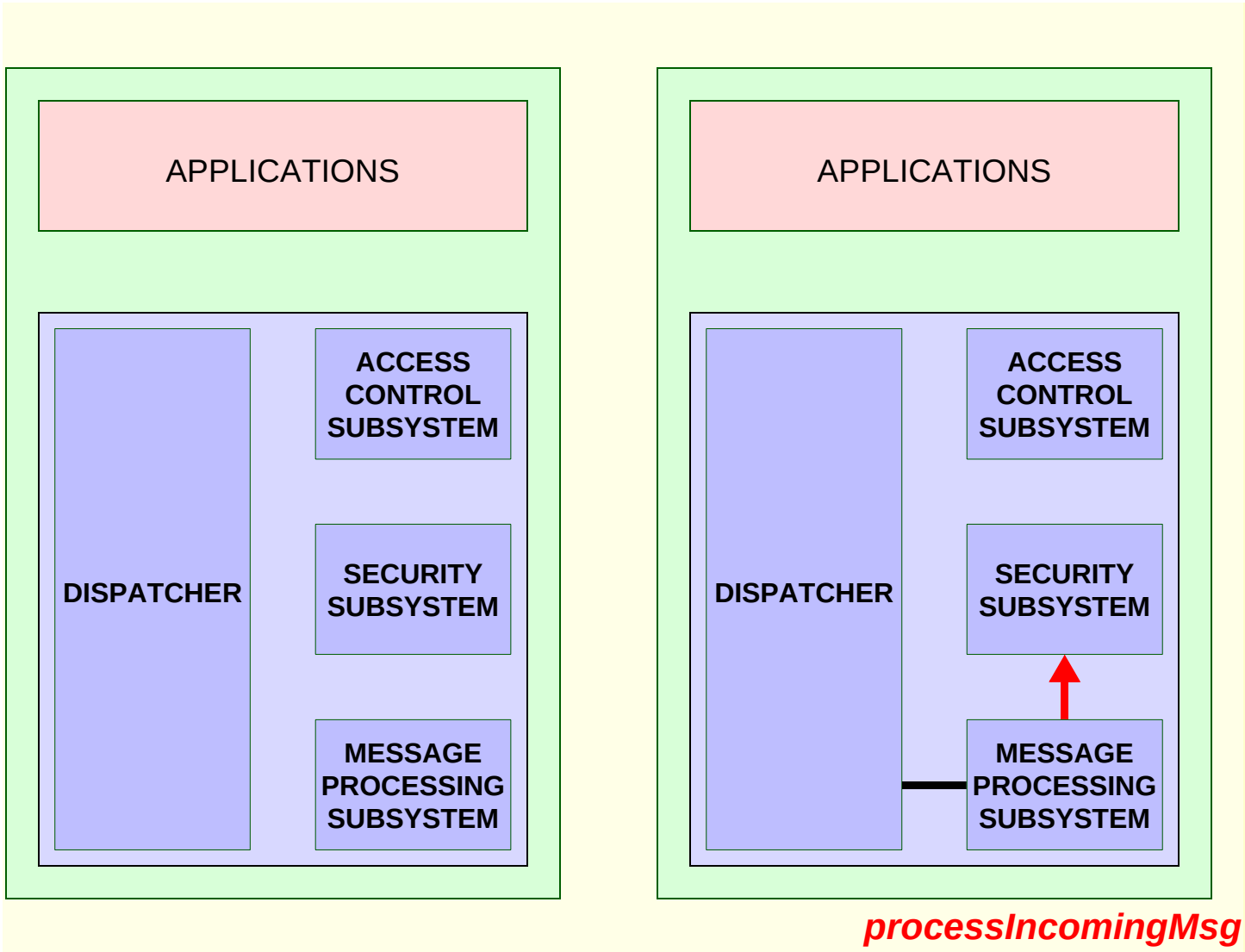




prepareDataElements

Parameters
contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

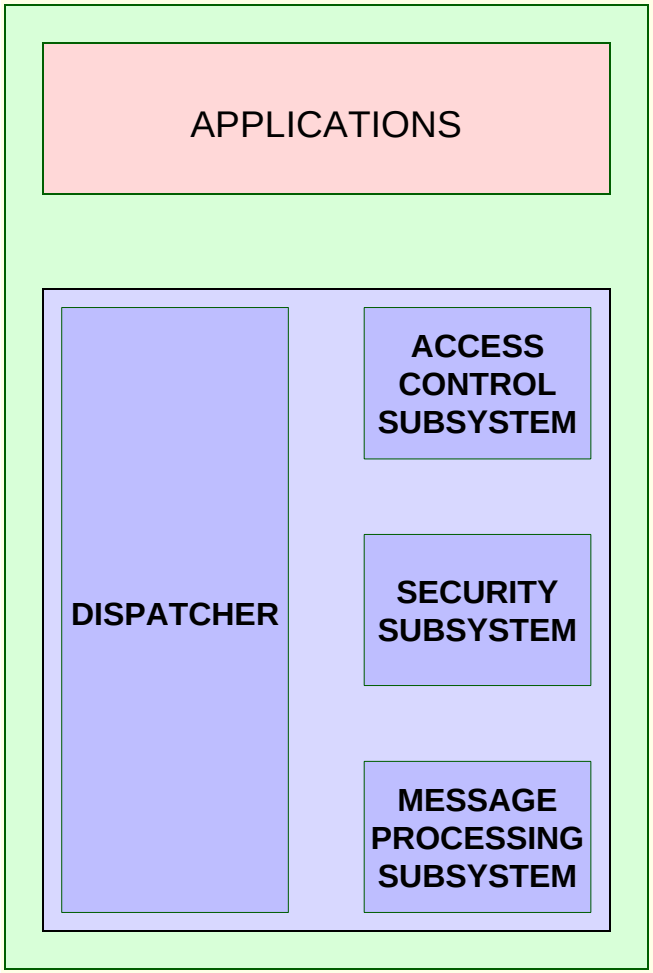
processIncomingMsg

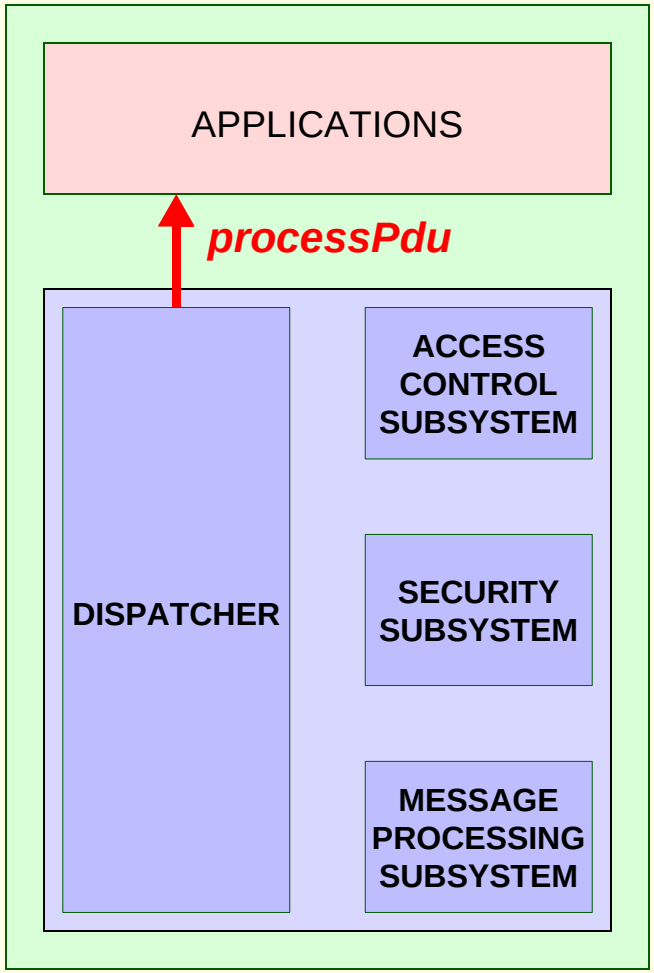


Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

processPdu

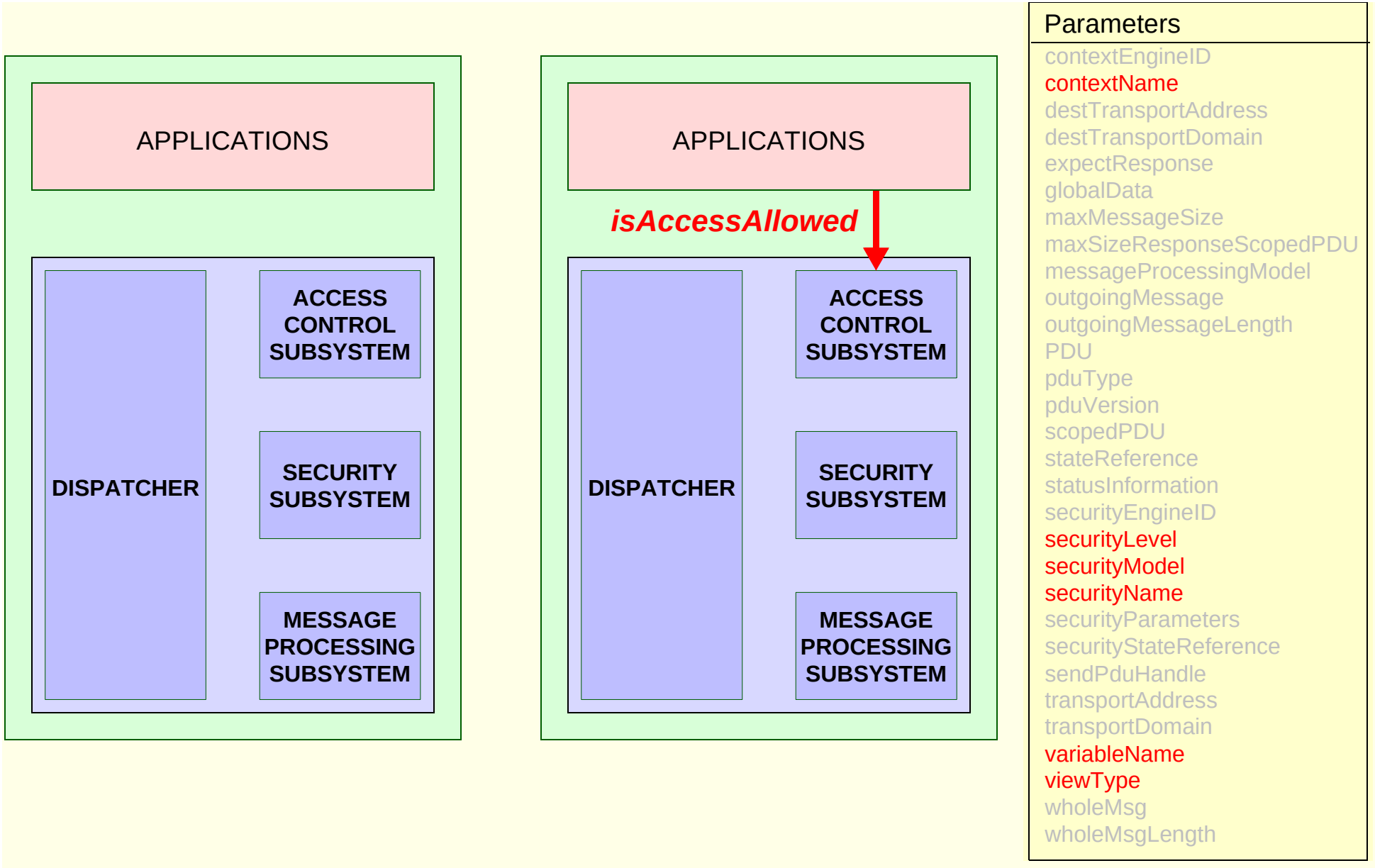




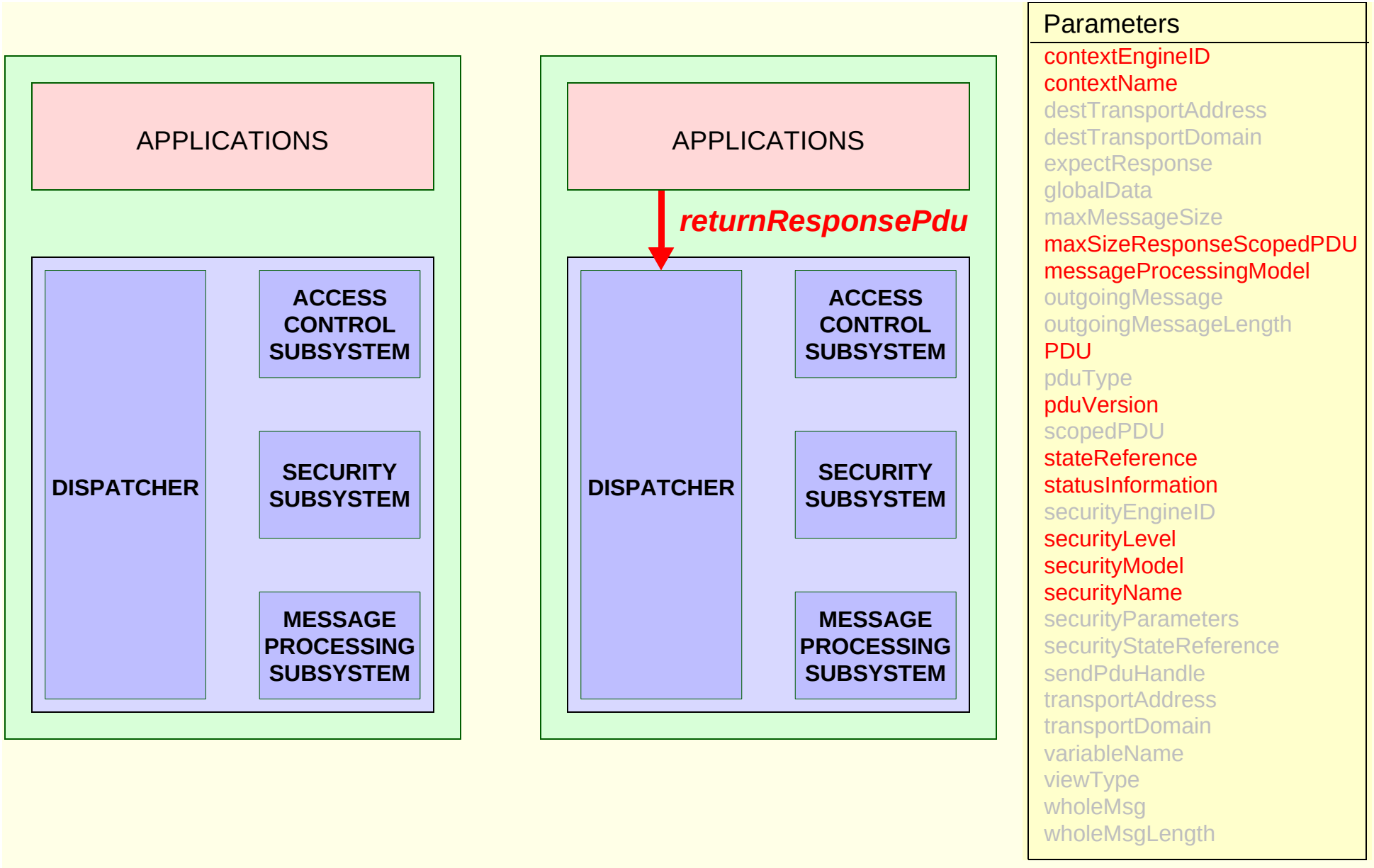
Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

isAccessAllowed



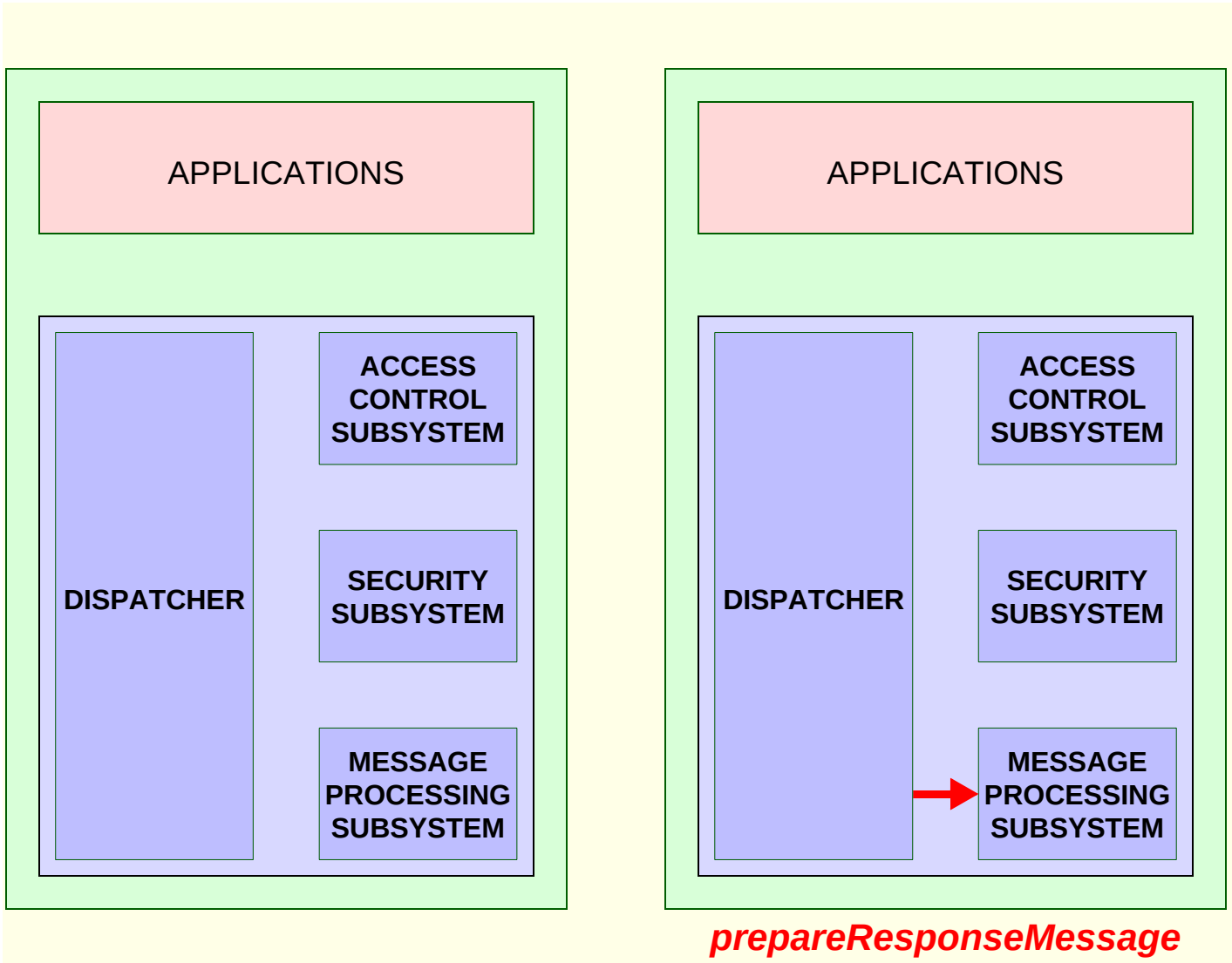
returnResponsePdu



Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

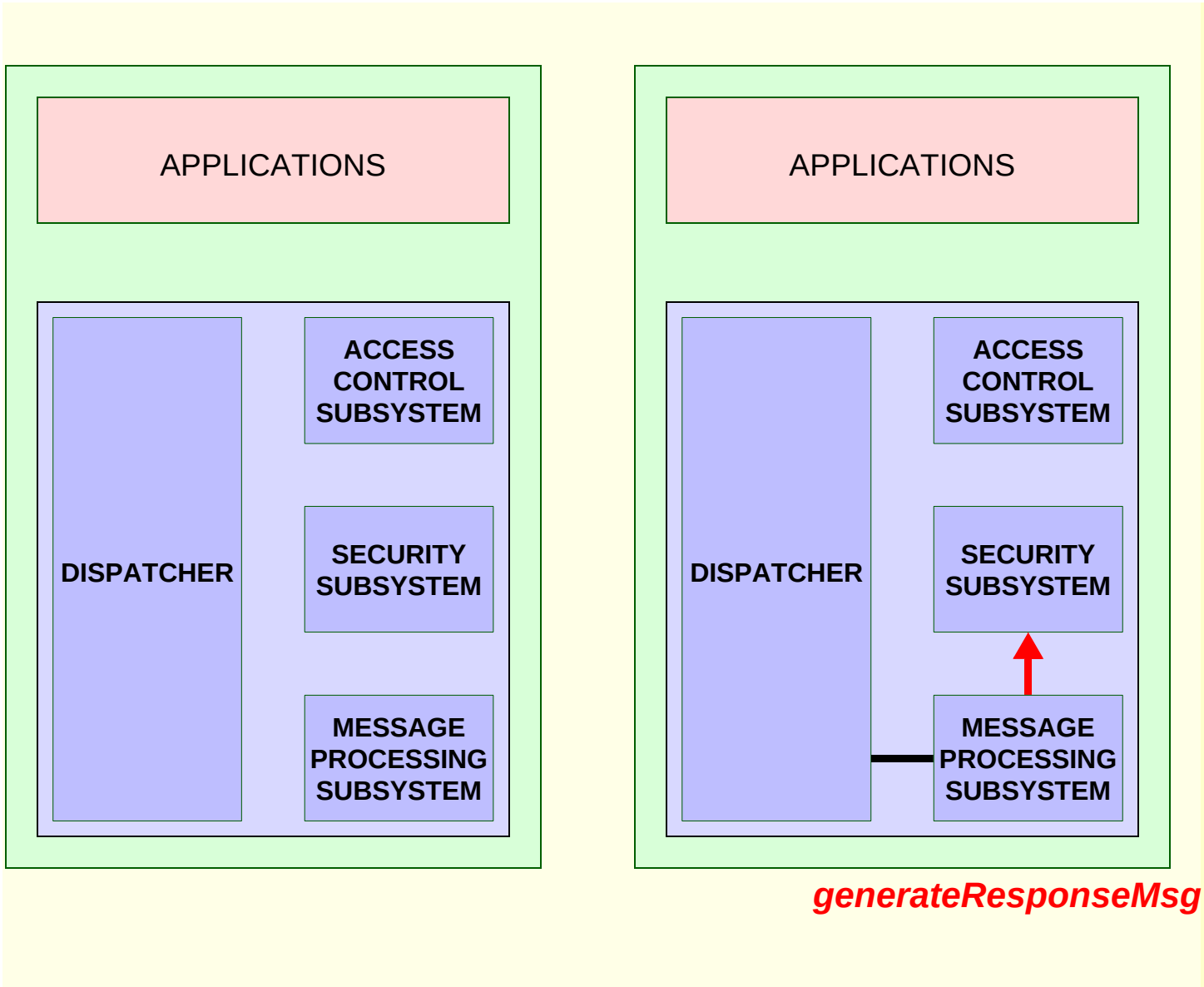
prepareResponseMessage



Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

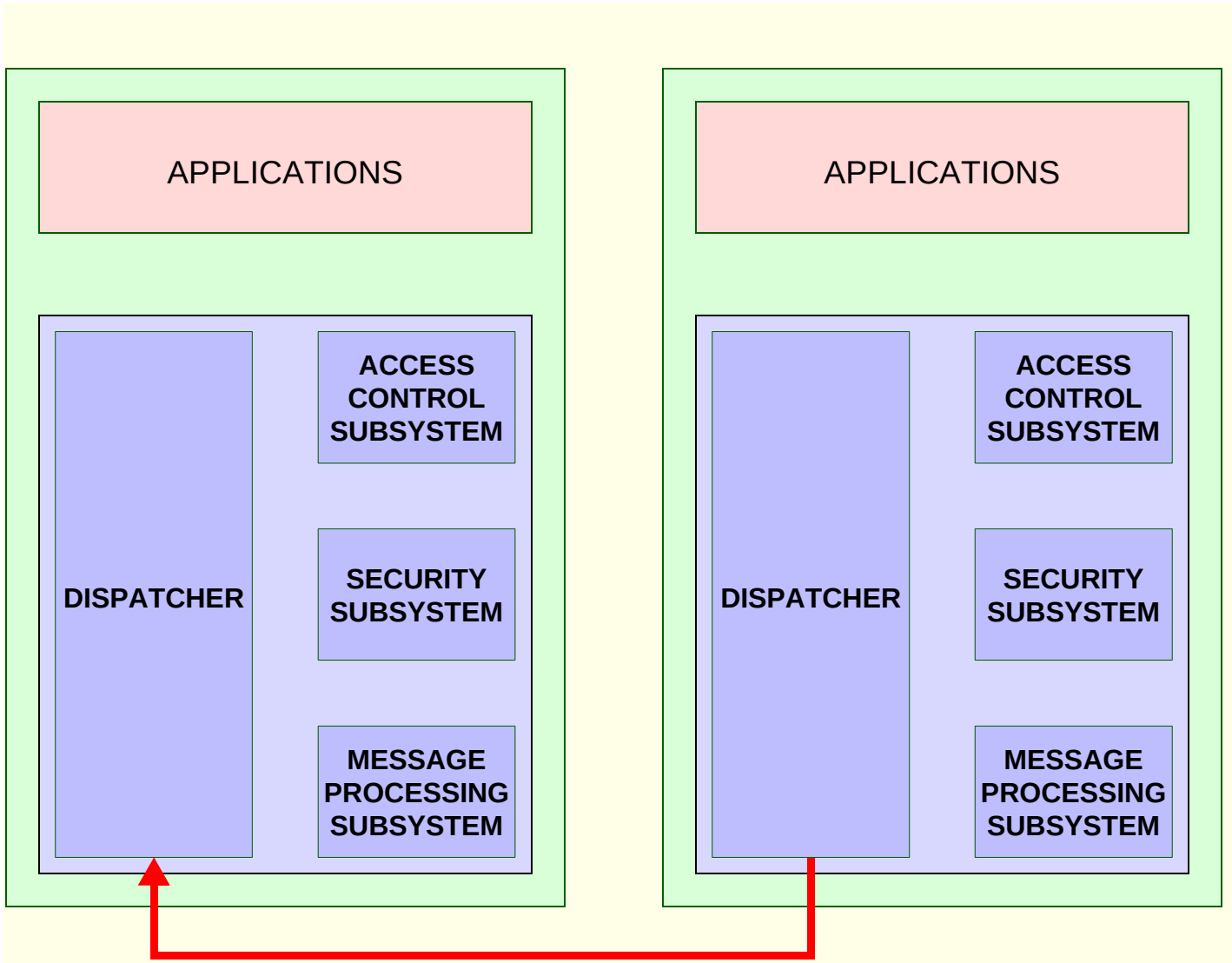
generateResponseMsg



Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

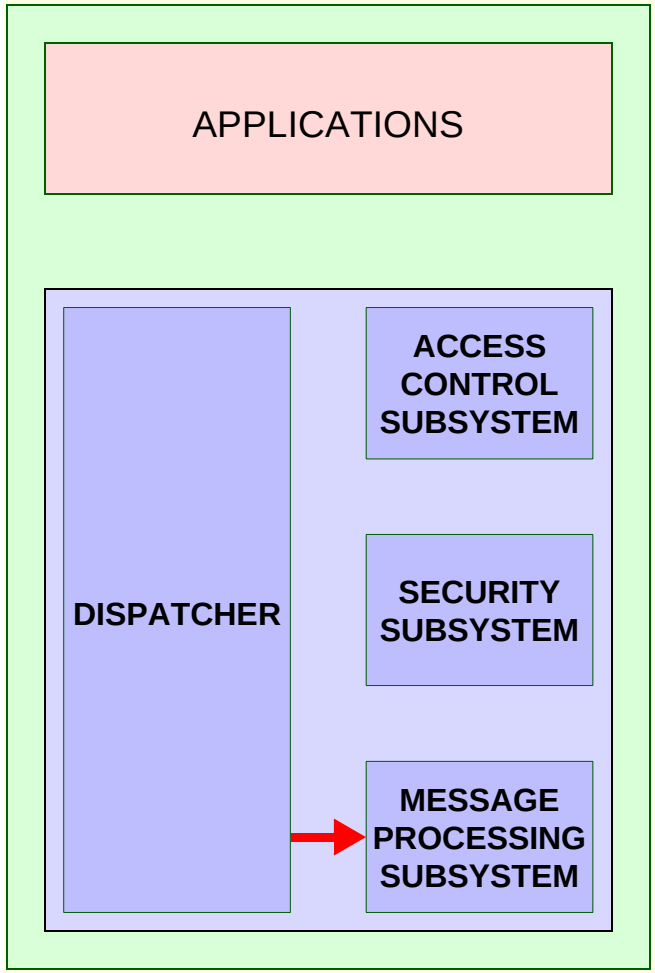
send / receive

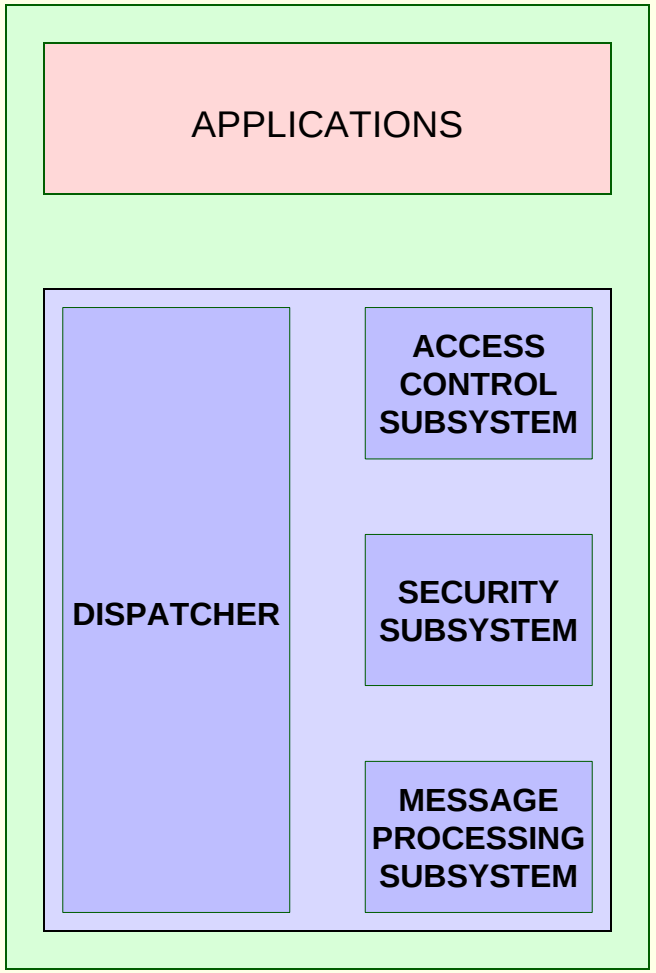


Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

prepareDataElements

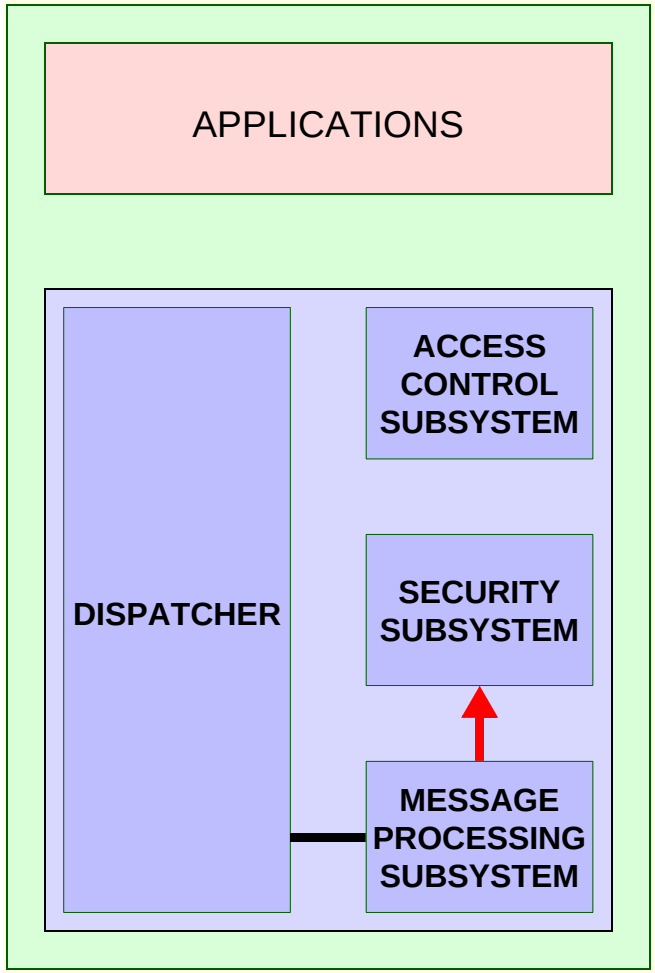


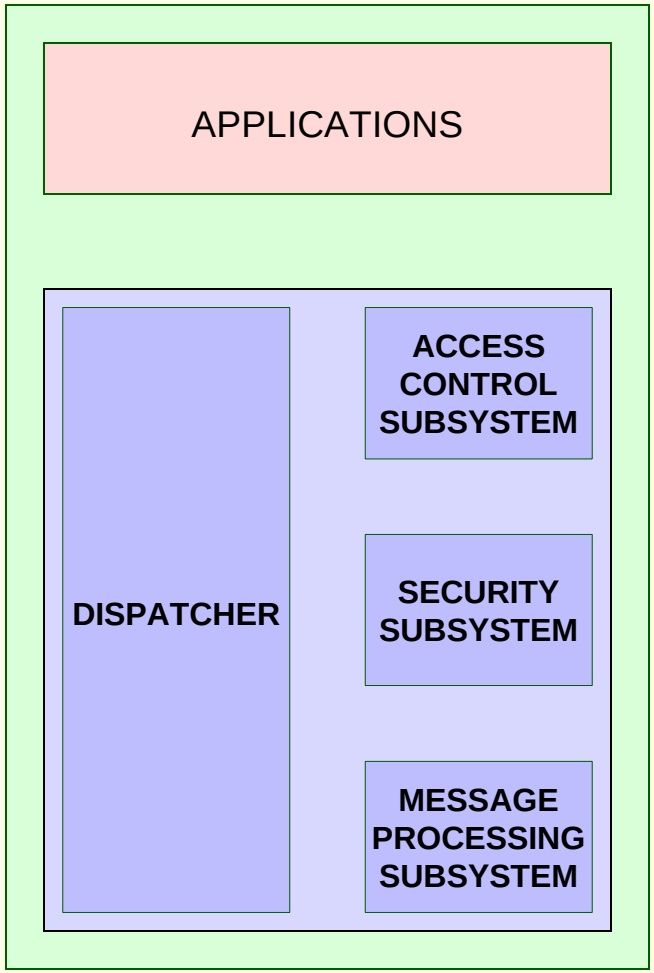


prepareDataElements

Parameters
contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

processIncomingMsg



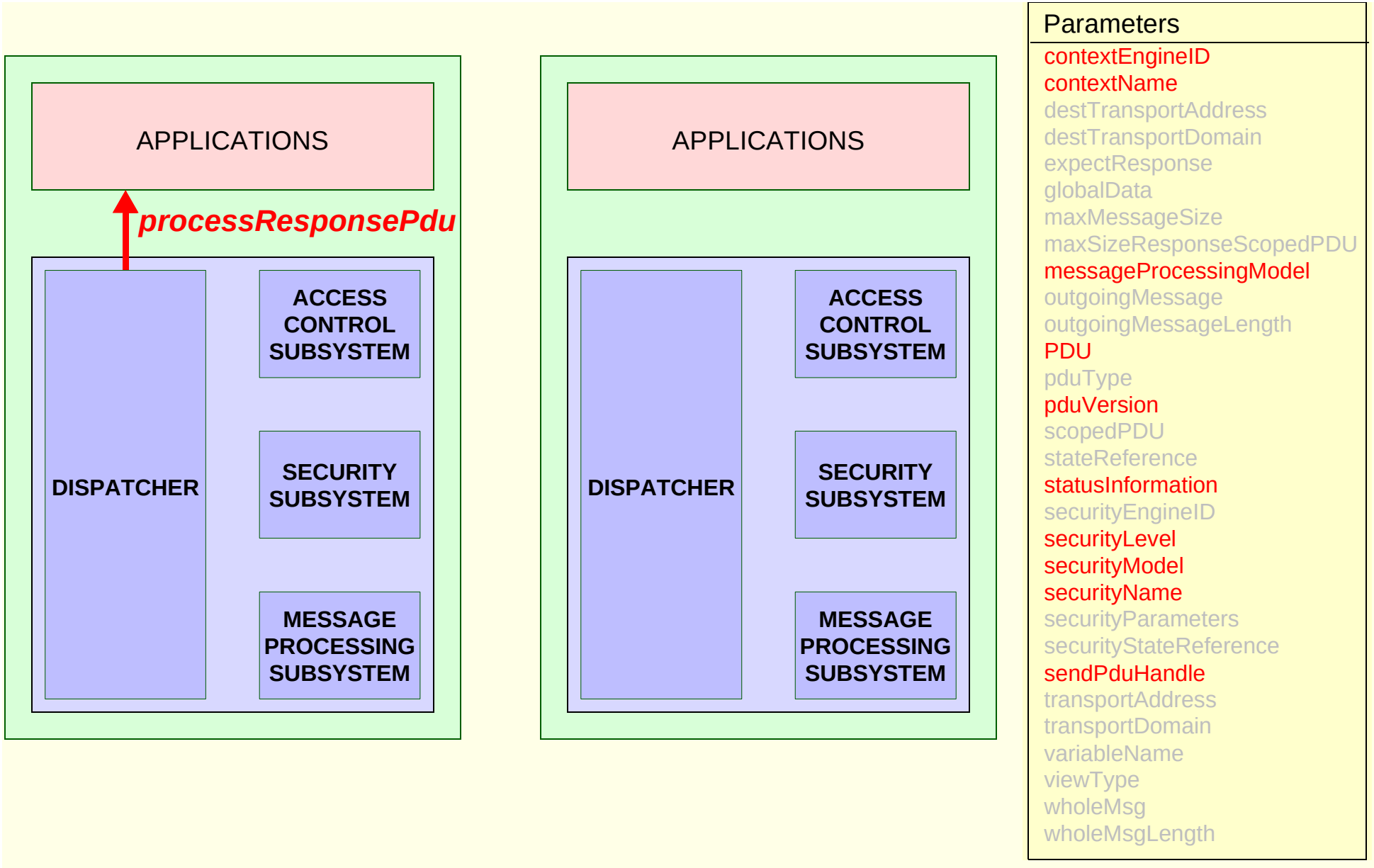


processIncomingMsg

Parameters

contextEngineID
contextName
destTransportAddress
destTransportDomain
expectResponse
globalData
maxMessageSize
maxSizeResponseScopedPDU
messageProcessingModel
outgoingMessage
outgoingMessageLength
PDU
pduType
pduVersion
scopedPDU
stateReference
statusInformation
securityEngineID
securityLevel
securityModel
securityName
securityParameters
securityStateReference
sendPduHandle
transportAddress
transportDomain
variableName
viewType
wholeMsg
wholeMsgLength

processResponsePdu



Parameters

- contextEngineID
- contextName
- destTransportAddress
- destTransportDomain
- expectResponse
- globalData
- maxMessageSize
- maxSizeResponseScopedPDU
- messageProcessingModel
- outgoingMessage
- outgoingMessageLength
- PDU
- pduType
- pduVersion
- scopedPDU
- stateReference
- statusInformation
- securityEngineID
- securityLevel
- securityModel
- securityName
- securityParameters
- securityStateReference
- sendPduHandle
- transportAddress
- transportDomain
- variableName
- viewType
- wholeMsg
- wholeMsgLength

MODULES OF THE SNMPv3 ARCHITECTURE

DISPATCHER AND MESSAGE PROCESSING MODULE

- SNMPv3 MESSAGE STRUCTURE
- snmpMPDMIB
- RFC 2572

APPLICATIONS

- snmpTargetMIB
- snmpNotificationMIB
- snmpProxyMIB
- RFC 2573

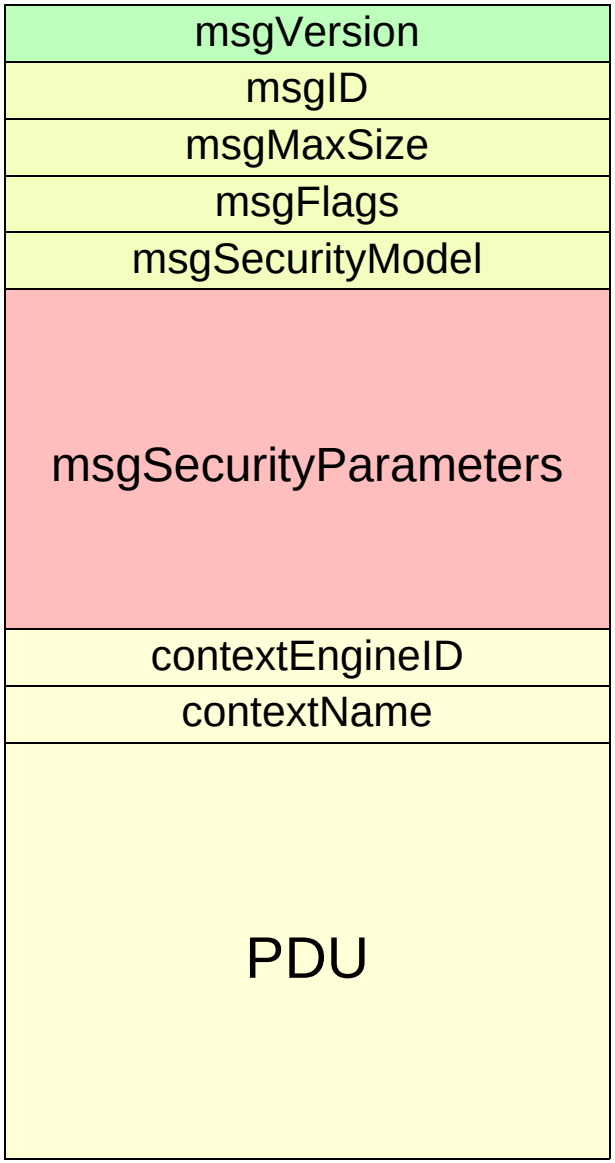
SECURITY SUBSYSTEM

- USER BASED SECURITY MODEL
- snmpUsmMIB
- RFC 2574

ACCESS CONTROL SUBSYSTEM

- VIEW BASED ACCESS CONTROL MODEL
- snmpVacmMIB
- RFC 2575

SNMPv3 MESSAGE STRUCTURE



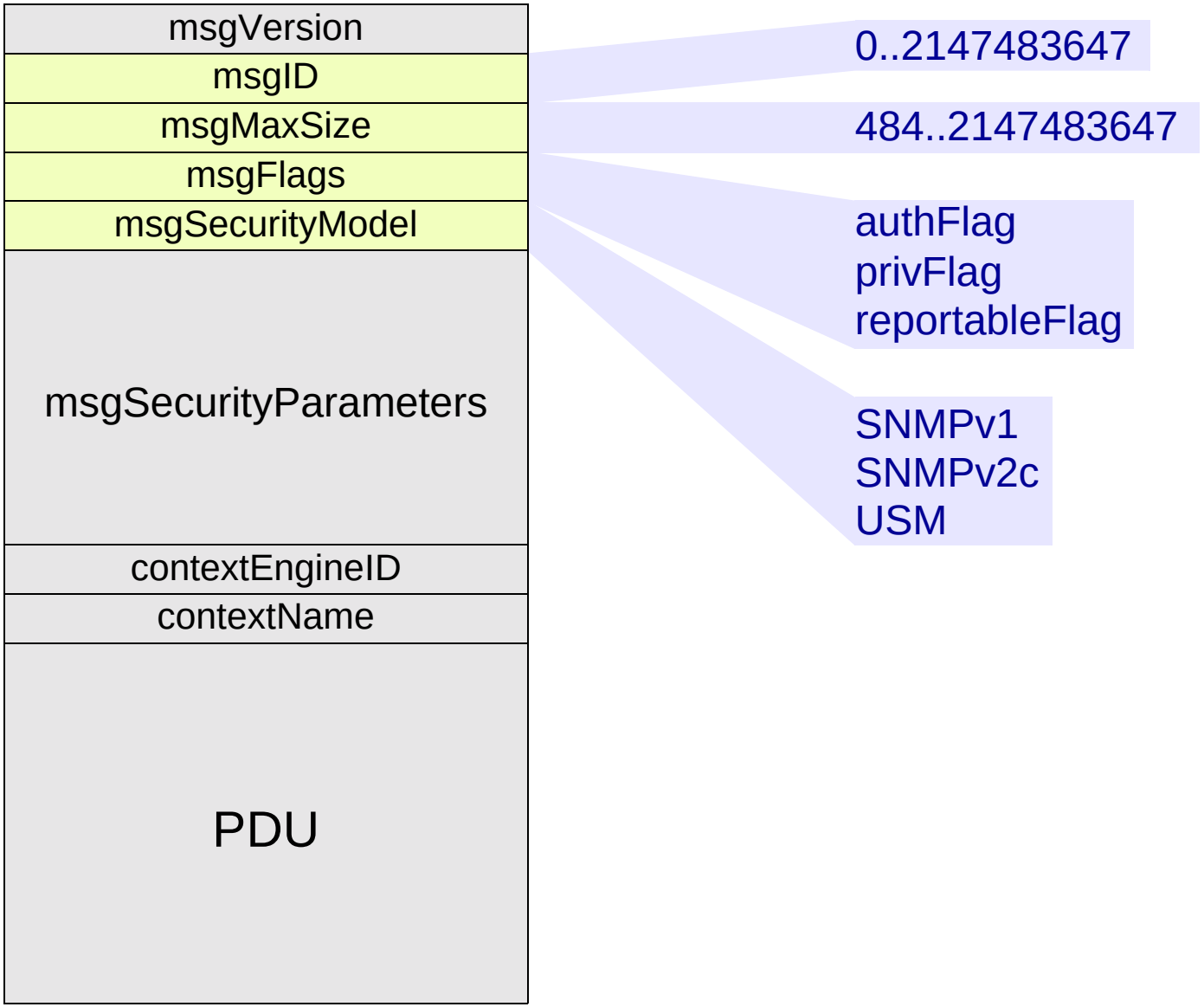
USED BY MESSAGE PROCESSING SUBSYSTEM

USED BY SNMPv3 PROCESSING MODULE

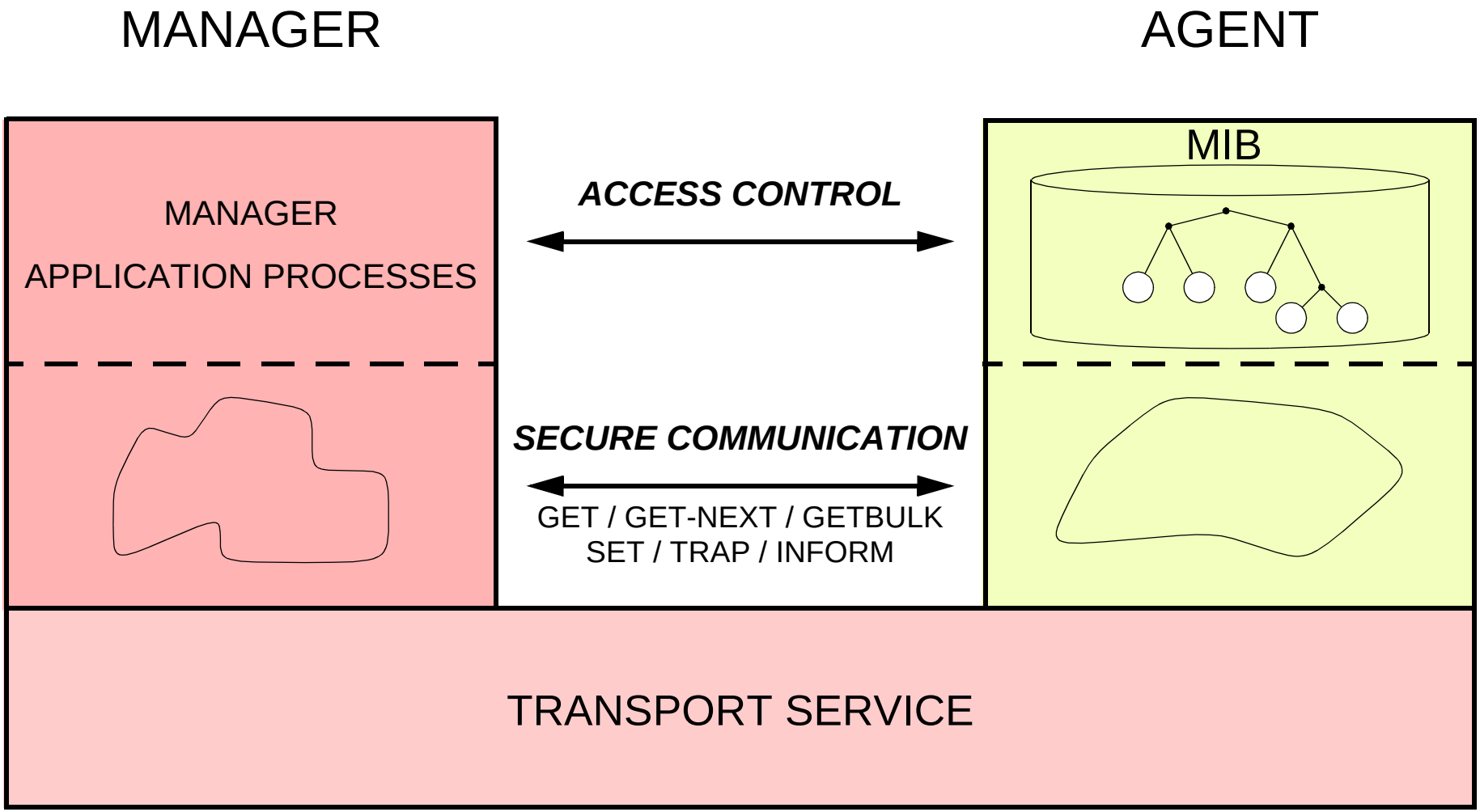
USED BY SECURITY SUBSYSTEM

USED BY ACCESS CONTROL SUBSYSTEM
AND APPLICATIONS

SNMPv3 PROCESSING MODULE PARAMETERS



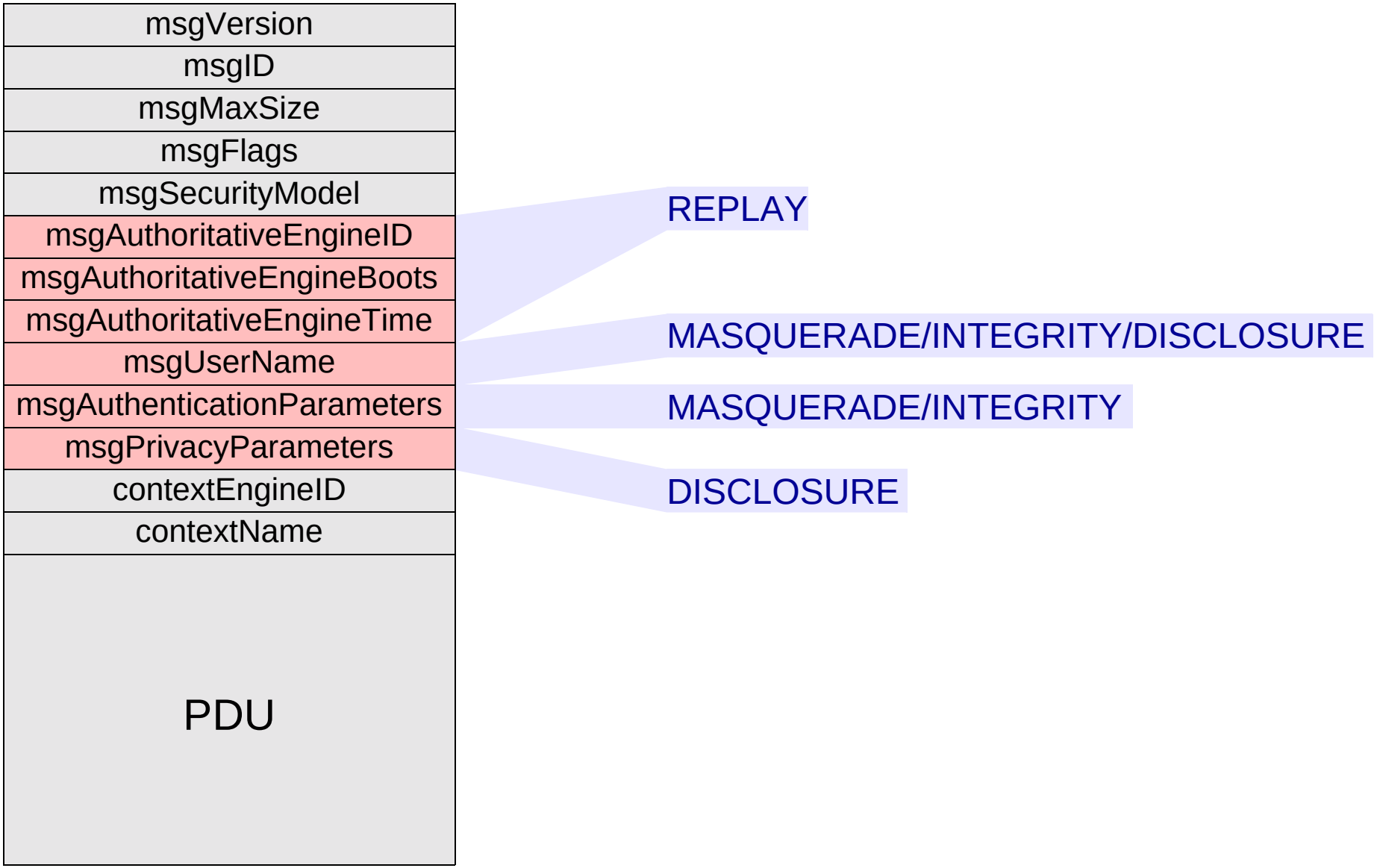
SECURE COMMUNICATION VERSUS ACCESS CONTROL



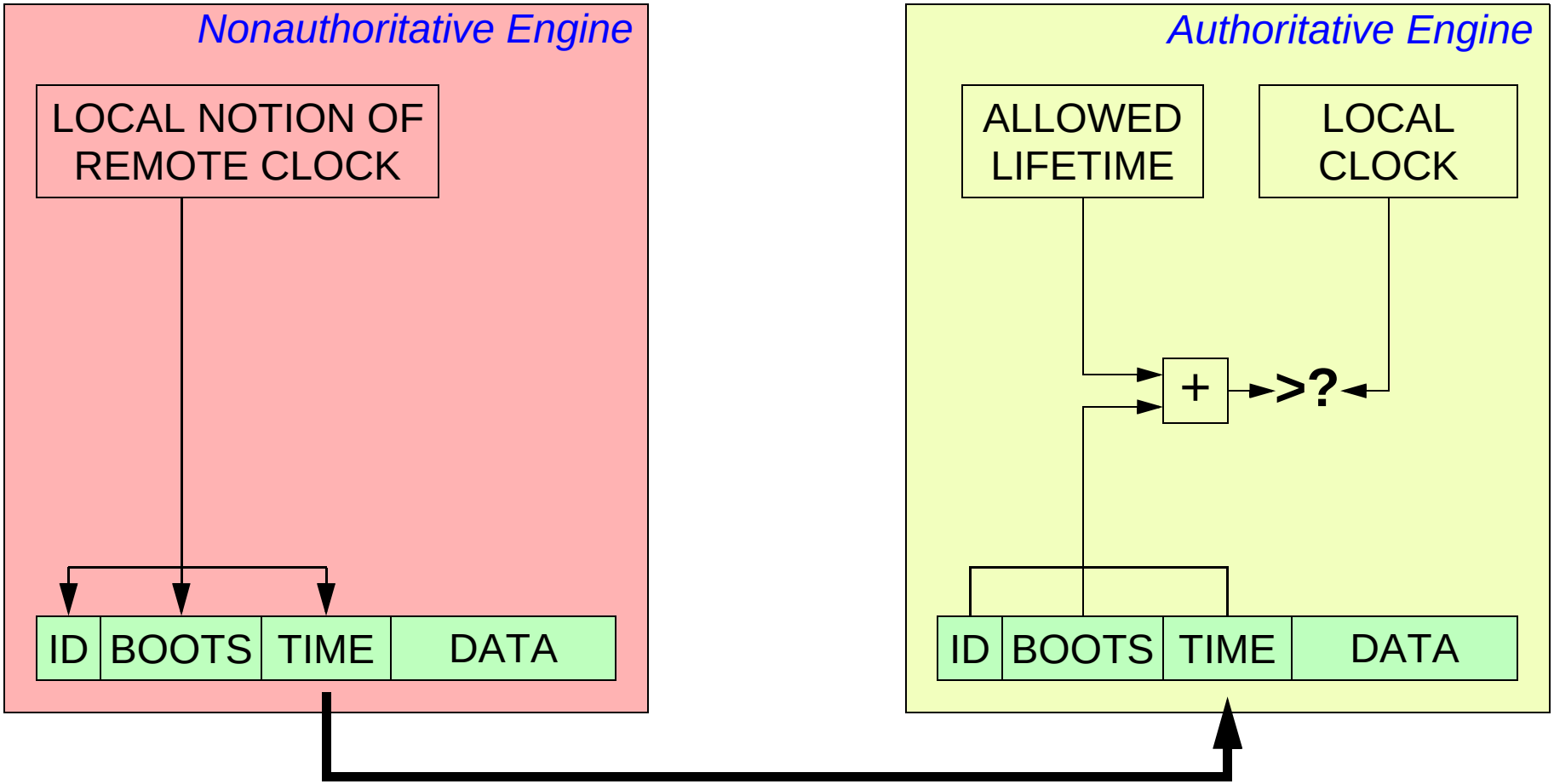
USM: SECURITY THREATS

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

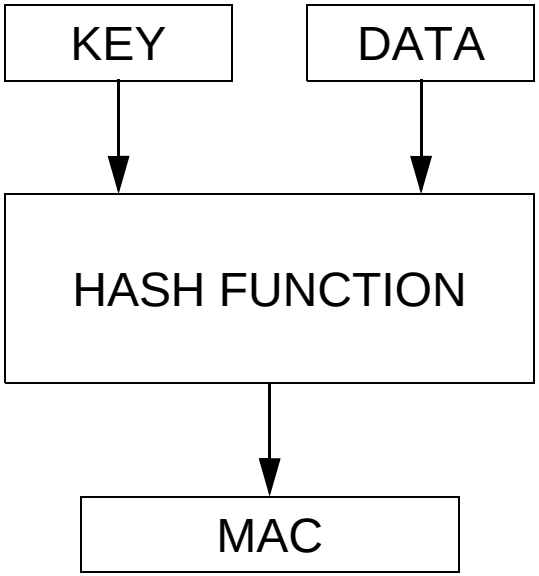
USM MESSAGE STRUCTURE



IDEA BEHIND REPLAY PROTECTION

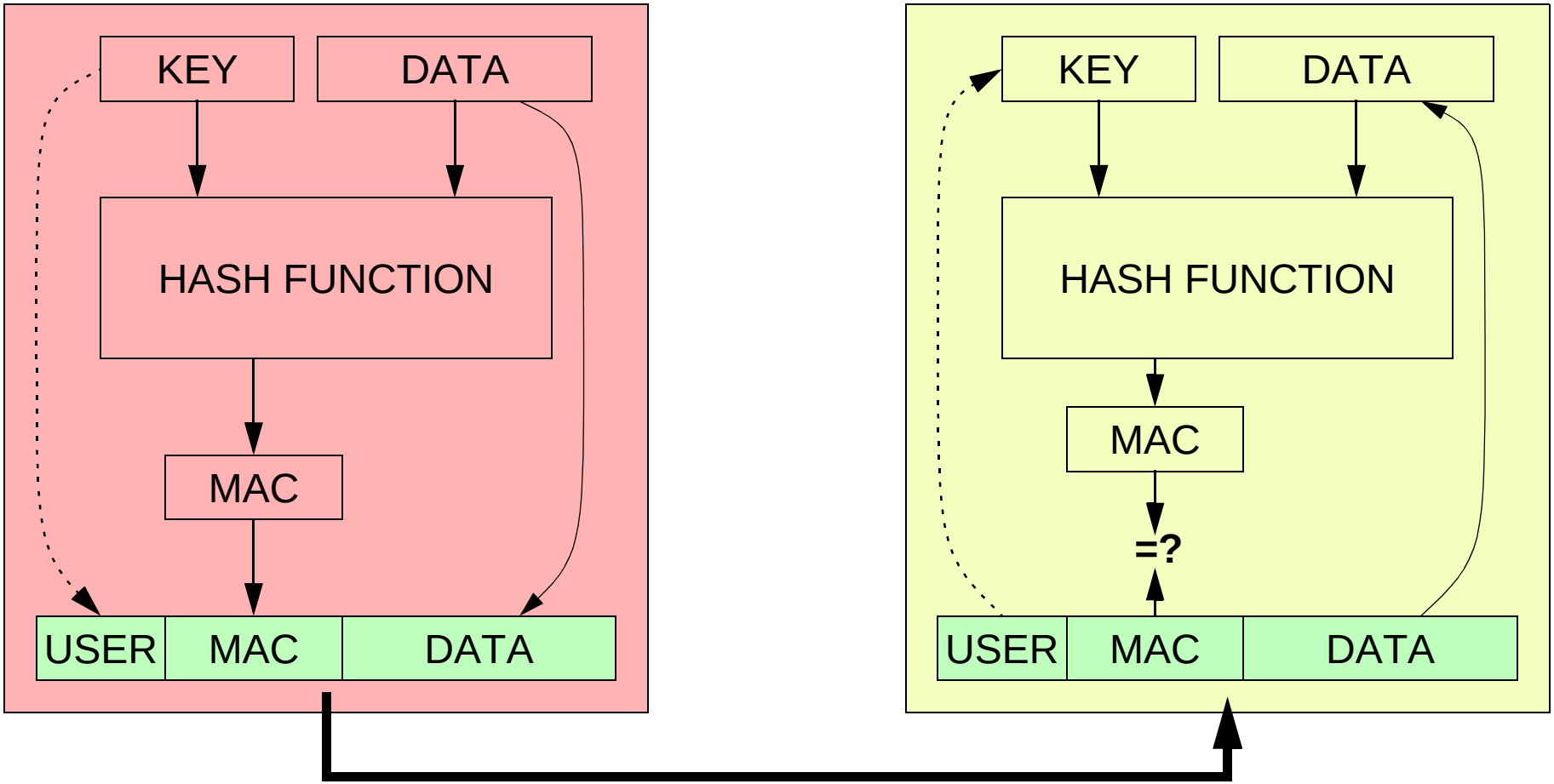


IDEA BEHIND DATA INTEGRITY AND AUTHENTICATION

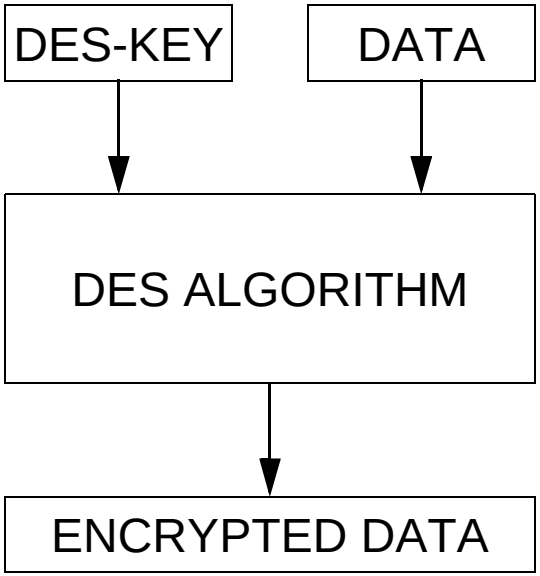


ADD THE MESSAGE AUTHENTICATION CODE (MAC) TO THE DATA
AND SEND THE RESULT

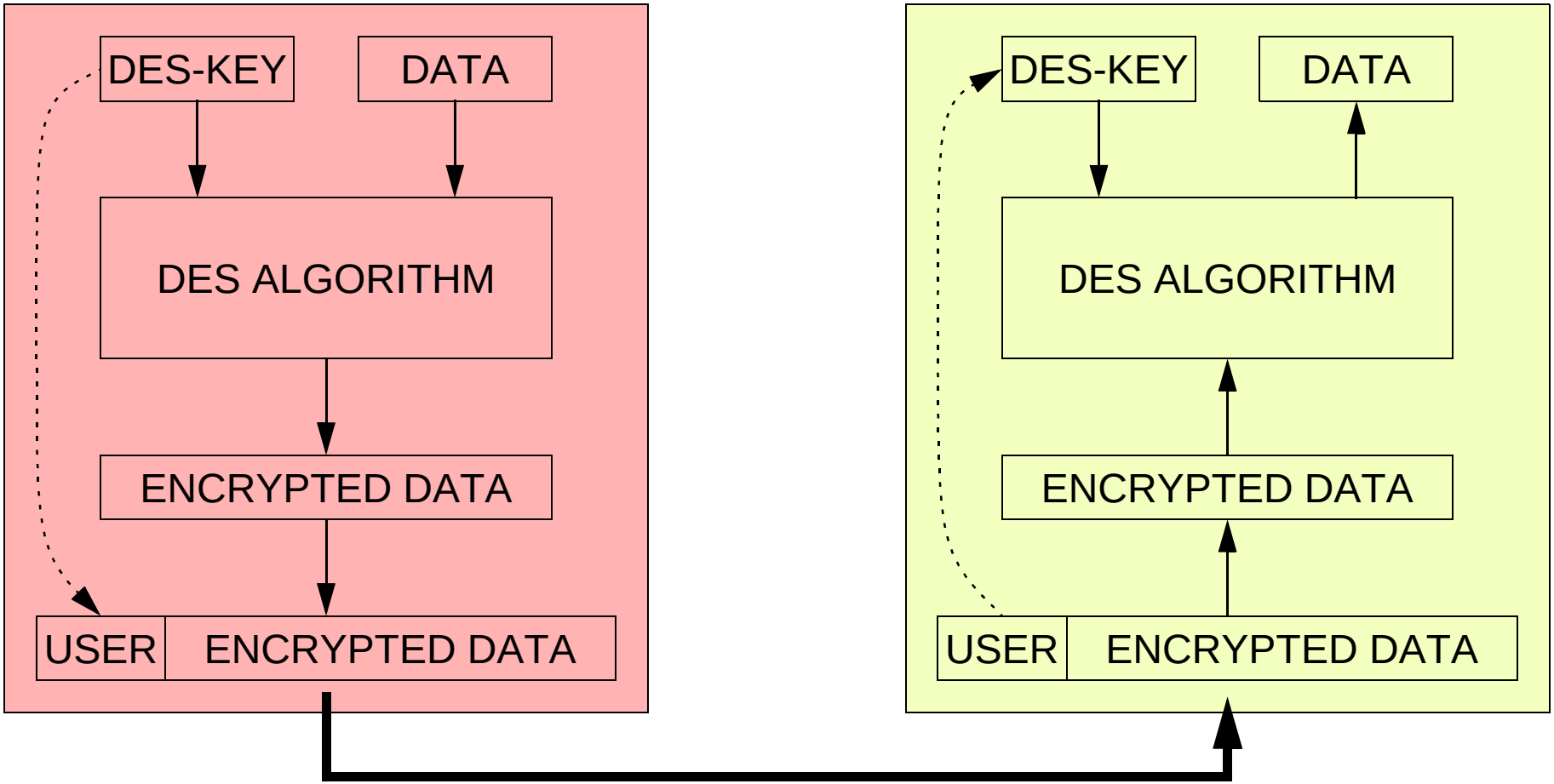
IDEA BEHIND AUTHENTICATION



IDEA BEHIND THE DATA CONFIDENTIALITY (DES)



IDEA BEHIND ENCRYPTION



VIEW BASED ACCESS CONTROL MODEL

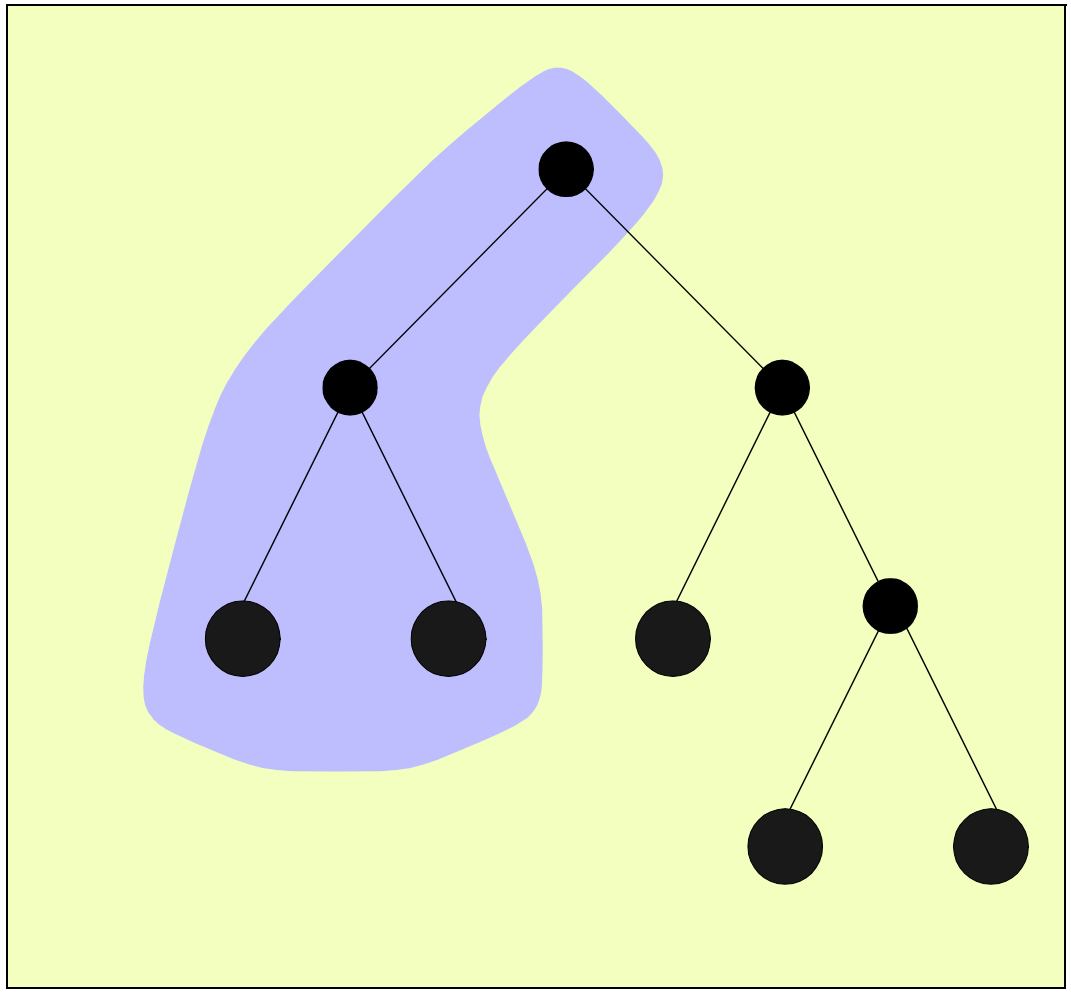
ACCESS CONTROL TABLE

MIB VIEWS

ACCESS CONTROL TABLES

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

MIB VIEWS

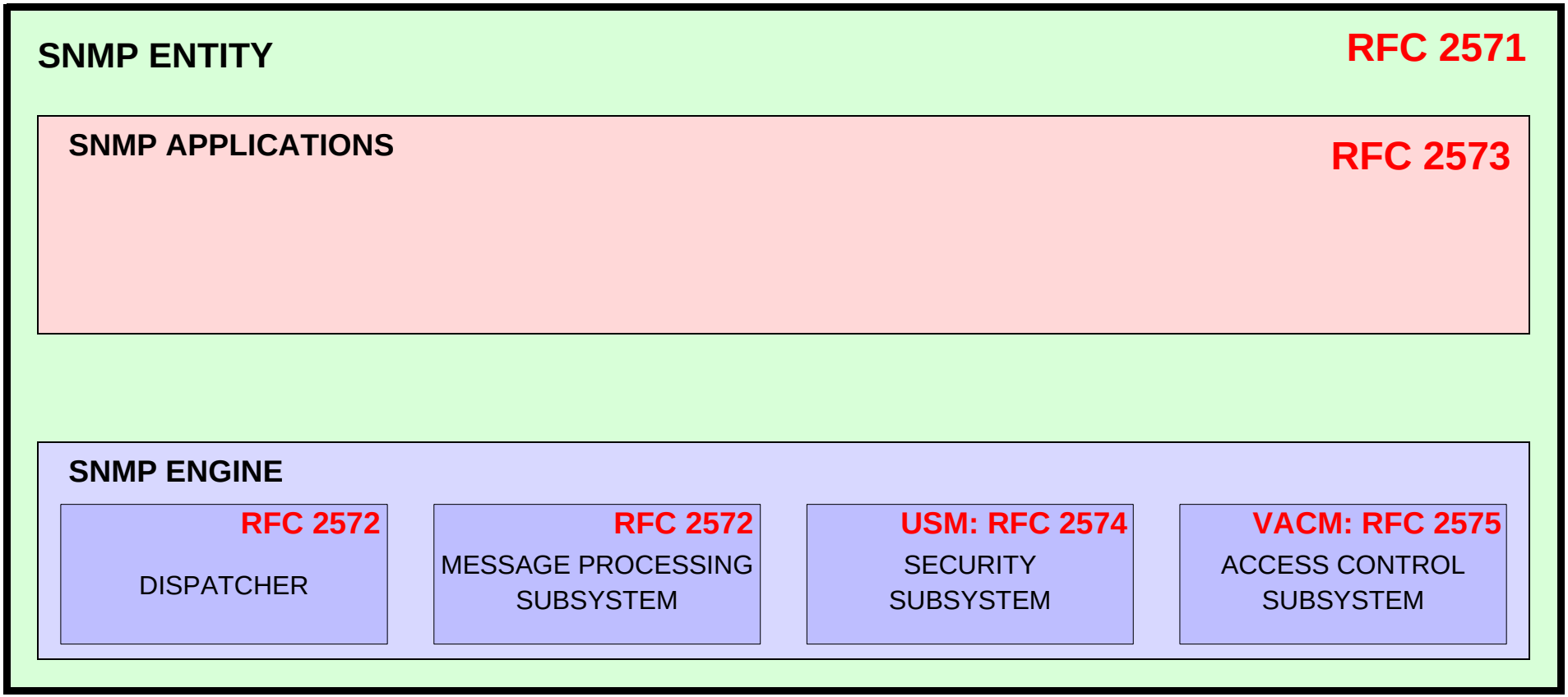


SNMPv3 IMPLEMENTATIONS

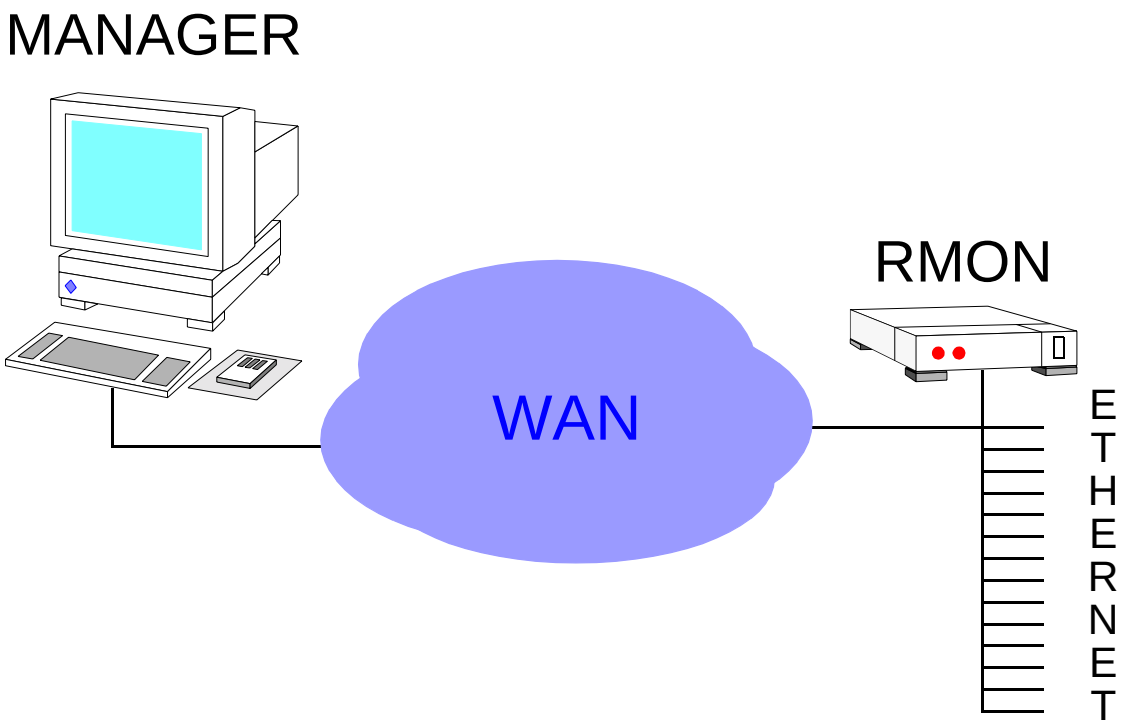
ACE*COMM
AdventNet
BMC Software
Cisco
Epilogue
Gambit communications
Halcyon
IBM
ISI
IWL
MG-SOFT
MultiPort Corporation
SimpleSoft
SNMP Research

SNMP++
TU of Braunschweig
UCD
University of Quebec

SNMPv3 RFCs


SNMP ENTITY
RFC 2571
SNMP APPLICATIONS
RFC 2573
SNMP ENGINE
RFC 2572
DISPATCHER
RFC 2572
MESSAGE PROCESSING SUBSYSTEM
USM: RFC 2574
SECURITY SUBSYSTEM
VACM: RFC 2575
ACCESS CONTROL SUBSYSTEM

REMOTE MONITORING



RMON1 (RFC 1757 - DRAFT)
TOKEN RING EXTENSIONS TO RMON (RFC 1513 - PROPOSED)
RMON2 (RFC 2021 - PROPOSED)
SMON (RFC 2613 - PROPOSED)

RMON1 GROUPS

NINE GROUPS:

- STATISTICS
- HISTORY
- HOST
- HOST TOP N
- TRAFFIC MATRIX
- ALARMS
- FILTERS
- PACKET CAPTURE
- EVENTS

STATISTICS GROUP

KEEPS STATISTICS PER ETHERNET SEGMENT

SHOWS:

- PACKETS
- OCTETS
- BROADCASTS
- MULTICASTS
- COLLISIONS
- ERRORS

KEEPS TRACK OF PACKET SIZE DISTRIBUTION:

- 65 - 127 OCTETS
- 128 - 255 OCTETS
- 256 - 511 OCTETS
- 512 - 1023 OCTETS
- 1024 - 1518 OCTETS

STATISTICS GROUP - ERRORS

	< 64 Bytes	64 to 1518	>1518 bytes
WELL-FORMED PACKETS	undersize	GOOD!	oversize
BAD FCS ERRORS	fragments	CRC or alignment errors	jabber

HISTORY GROUP

STORES INFORMATION OF STATISTICS GROUP
EXCEPT PACKET SIZE DISTRIBUTION

USES A CIRCULAR BUFFER
- BUCKETS
- SIZE MAY BE SET BY MANAGER

MANAGER MAY SET:
- THE ETHERNET SEGMENTS (INTERFACES)
- SAMPLING INTERVAL

HOST INFORMATION

- HOST
- HOST TOP N

IN / OUT:
PACKETS / OCTETS

OUT:
BROADCASTS
MULTICASTS
ERRORS

INFORMATION INDEXED BY:
- INTERFACE AND MAC ADDRESS (hostTable)
- CREATION TIME (hostTimetable)
- SORTED ON SOME VARIABLE VALUE (hostTopN)

TRAFFIC MATRIX

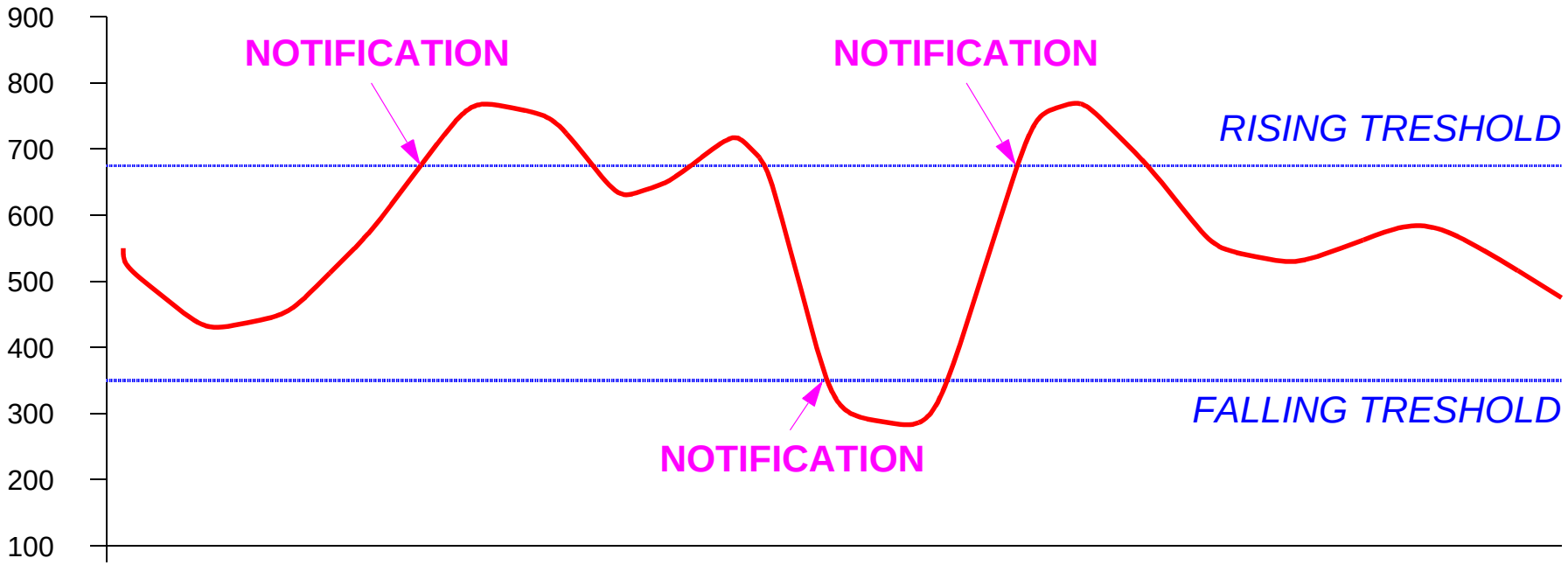
FOR EACH SOURCE & DESTINATION

- PACKETS
- OCTETS
- ERRORS

USEFUL:

- TO PROVIDE "WHAT IF" ANALYSIS
- TO DETECT INTRUDERS

ALARM GROUP

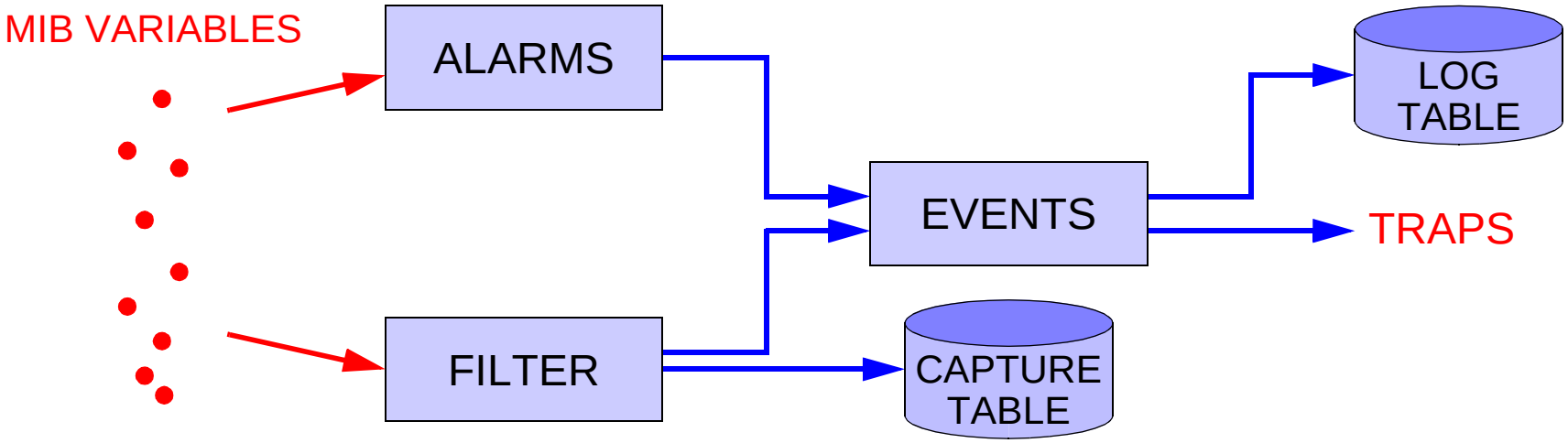


ABSOLUTE OR DELTA VALUES

TRIGGERS ON:

- RISING ALARM
- FALLING ALARM
- RISING OR FALLING ALARM

OTHER GROUPS



FILTER GROUP

- TO COUNT PACKETS THAT CARRY A SPECIFIC BIT-PATTERN

PACKET CAPTURE GROUP

- TO STORE SPECIFIC PACKETS

EVENT GROUP

- TO DEFINE THE VARIOUS EVENTS
- TO DETERMINE ON LOGGING AND / OR TRANSMISSION OF TRAPS

RMON2

TO MONITOR ALL HIGHER LAYER PROTOCOLS

EXTENDS RMON1 WITH FOLLOWING GROUPS:

- PROTOCOL DIRECTORY GROUP
- PROTOCOL DISTRIBUTION GROUP
- ADDRESS MAPPING GROUP
- NETWORK LAYER HOST GROUP
- NETWORK LAYER MATRIX GROUP
- APPLICATION LAYER HOST GROUP
- APPLICATION LAYER MATRIX GROUP
- USER HISTORY GROUP
- PROBE CONFIGURATION GROUP

DISTRIBUTED MANAGEMENT

THREE APPROACHES ARE BEING DEFINED

MIB BASED
- EXPRESSION MIB
- EVENT MIB
- NOTIFICATION LOG MIB

SCRIPT BASED
- SCRIPT MIB
- SCHEDULE MIB

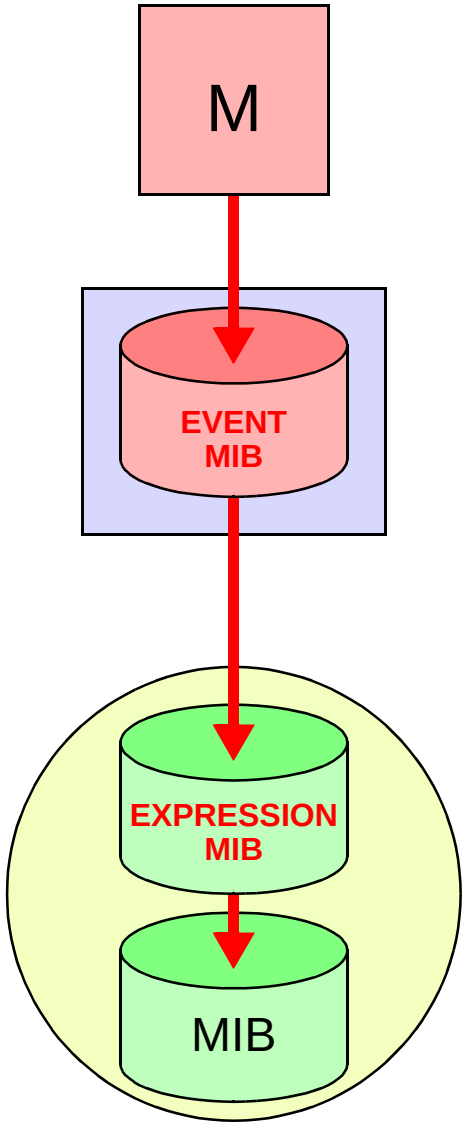
REMOTE OPERATIONS BASED
- REMOTE OPERATIONS MIB

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:

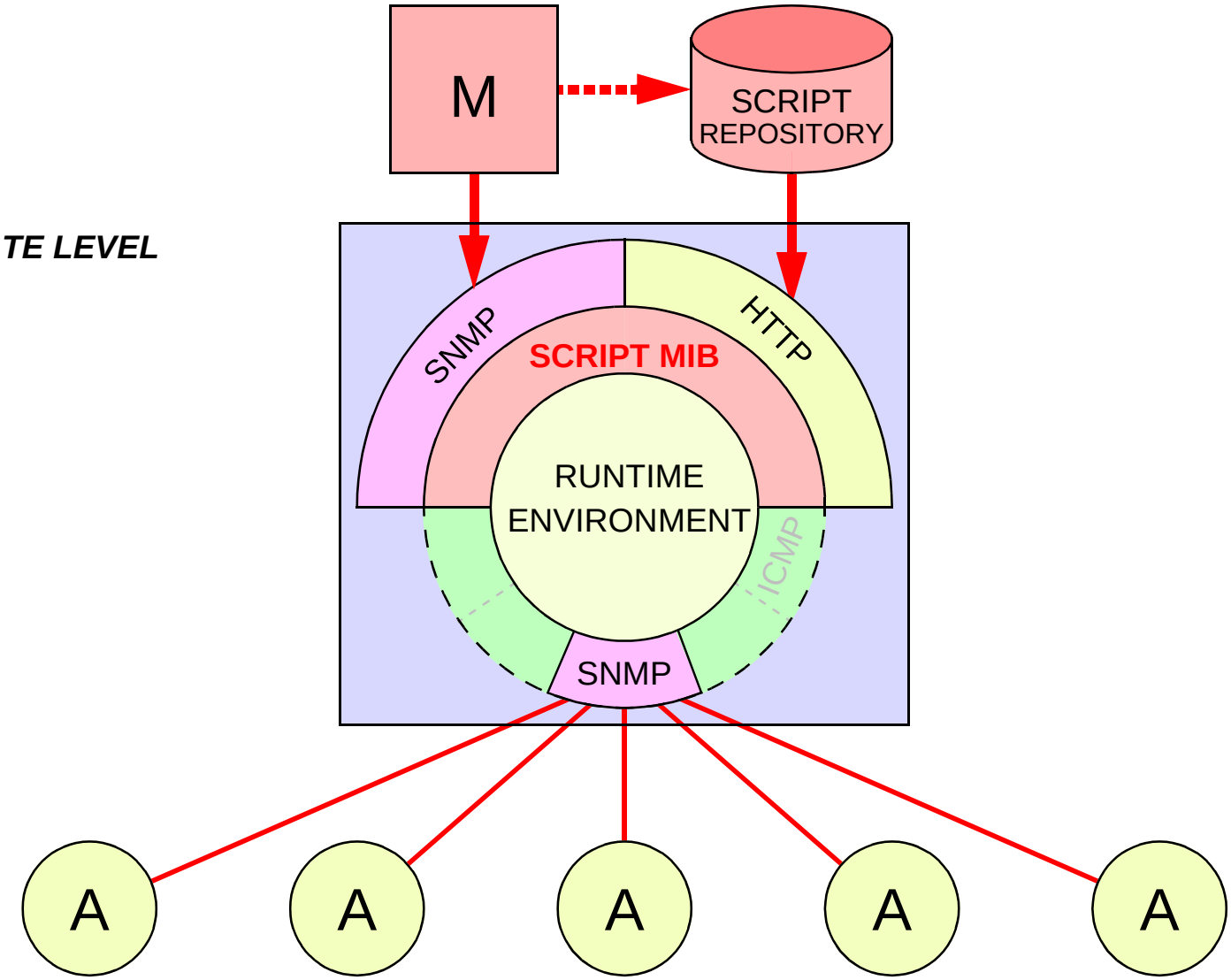
• INPUT ARE VARIABLES OF A (REMOTE) MIB
• TRIGGERS ON CHANGES, OR TRESHOLD CROSSING
• GENERATES A NOTIFICATION OR SET OPERATION

SCRIPT MIB

TOP LEVEL MANAGER

INTERMEDIATE LEVEL MANAGER

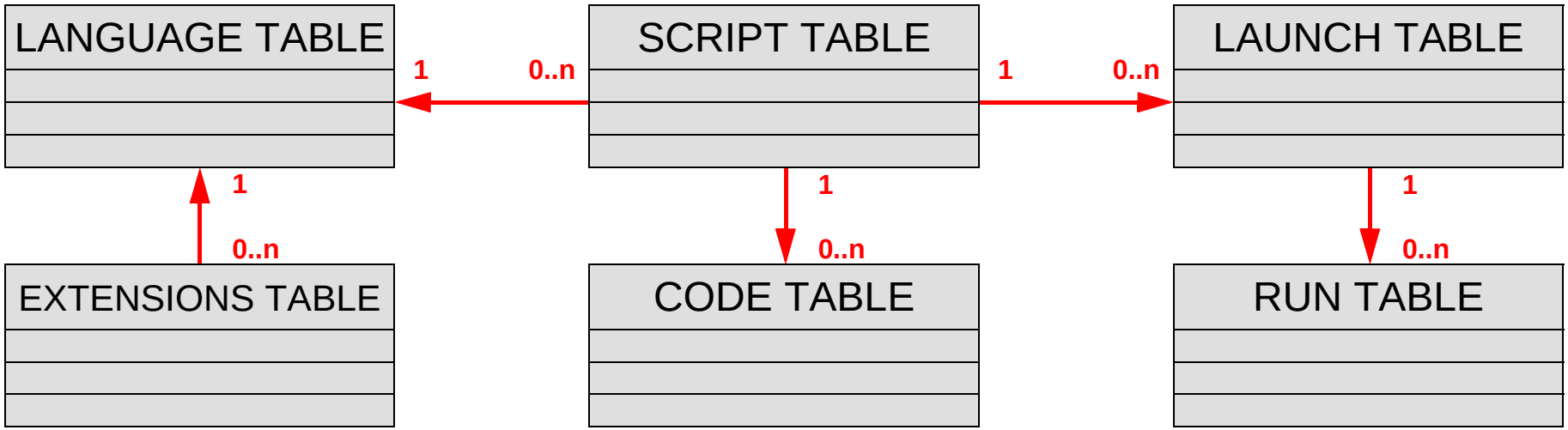
AGENTS



SCRIPT MIB: CHARACTERISTICS

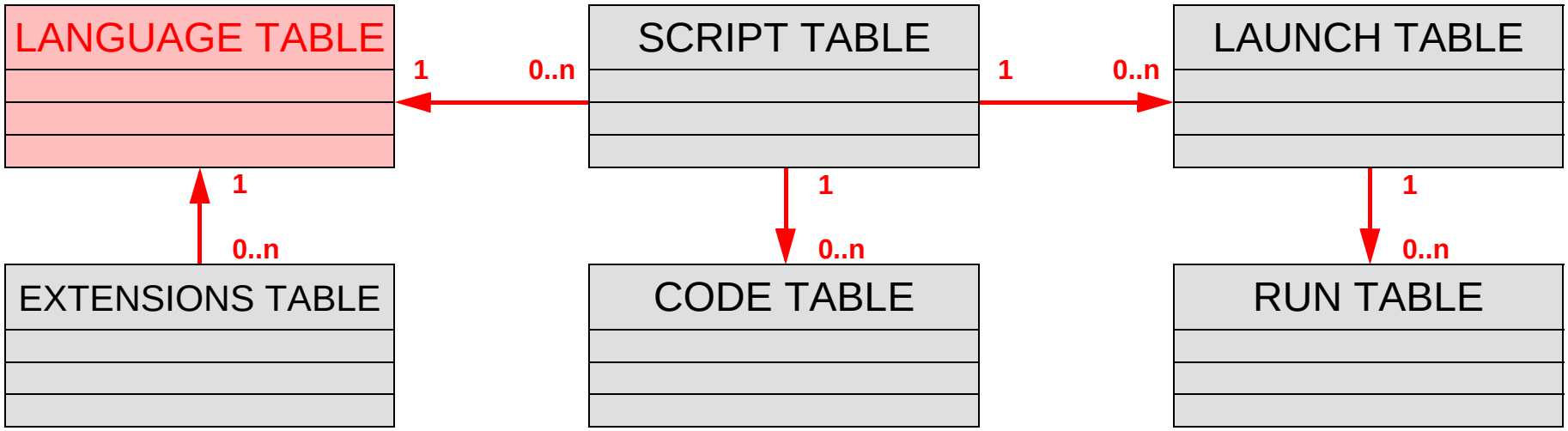
- FUNCTIONALITY CAN BE DEFINED AT RUN-TIME
- POWERFUL AUTONOMOUS ACTIONS
- MAY BE EASIER TO OPERATE FOR THE TOP-LEVEL MANAGER
- PROTECTION MECHANISMS NECESSARRY
- DIFFERENT SCRIPT LANGUAGES

SCRIPT MIB: STRUCTURE



CONSISTS OF 6 TABLES

SCRIPT MIB: LANGUAGE TABLE

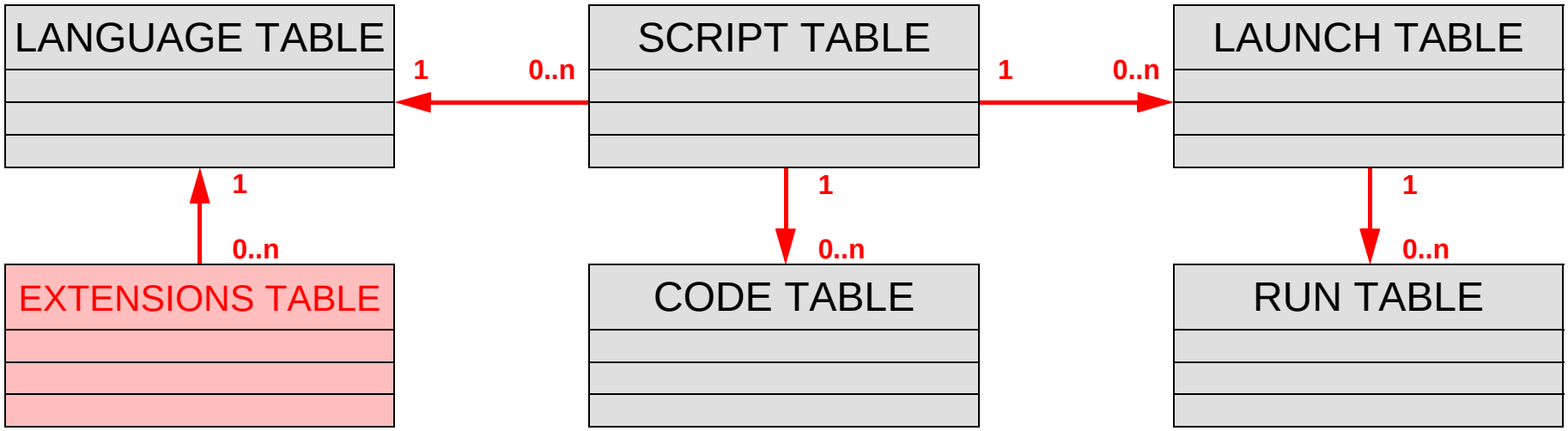


DEFINES THE LANGUAGES THIS SYSTEM SUPPORTS

- 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

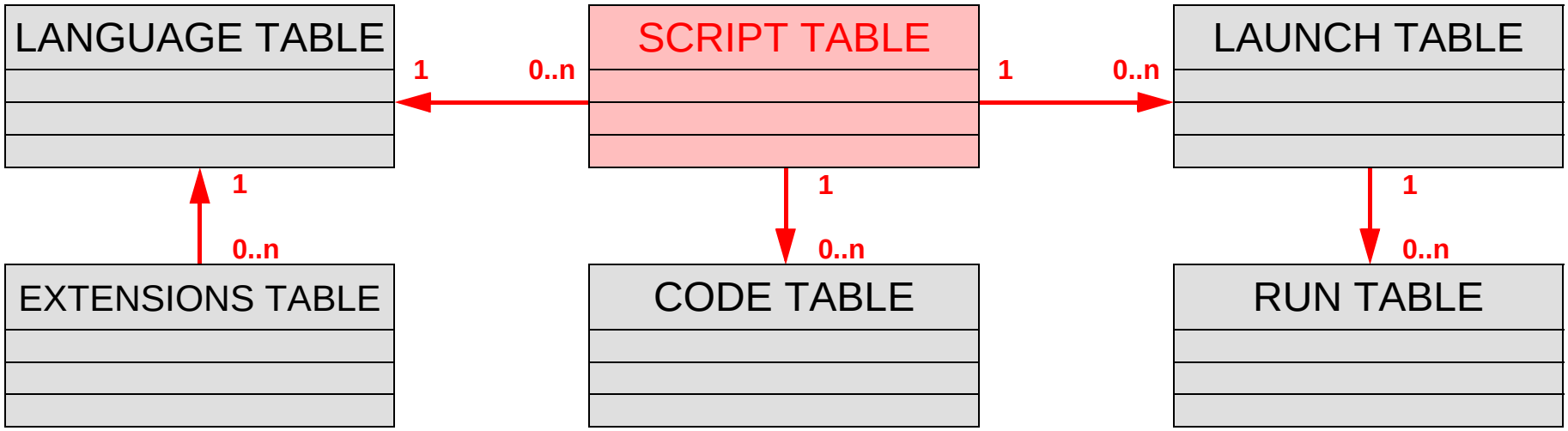


DEFINES THE EXTENSIONS FOR EACH LANGUAGE

- 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

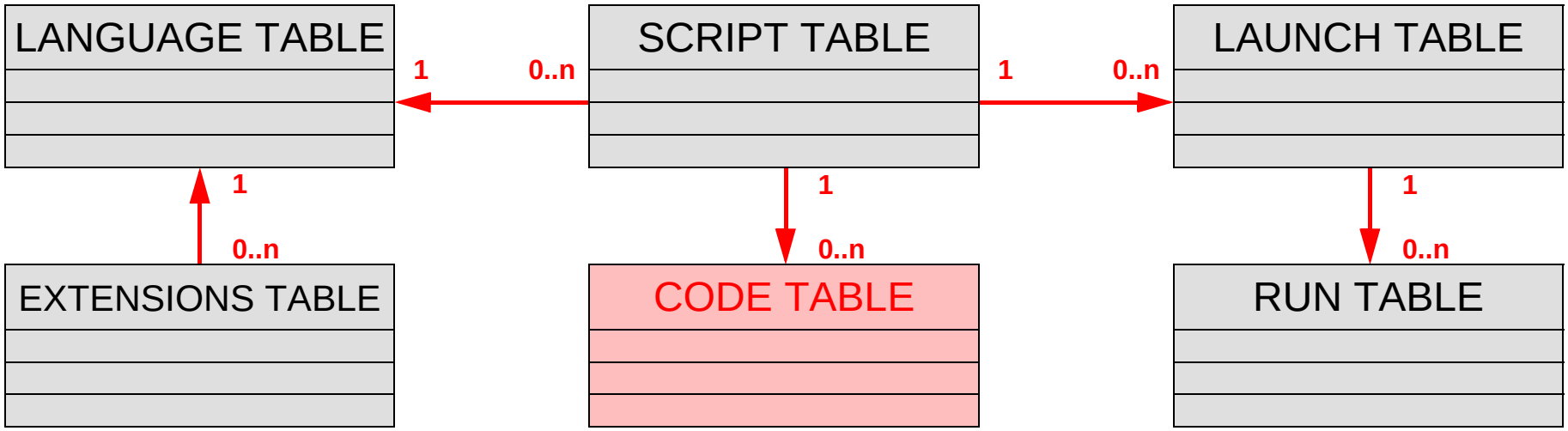


LISTS ALL SCRIPTS KNOWN TO THE SYSTEM

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



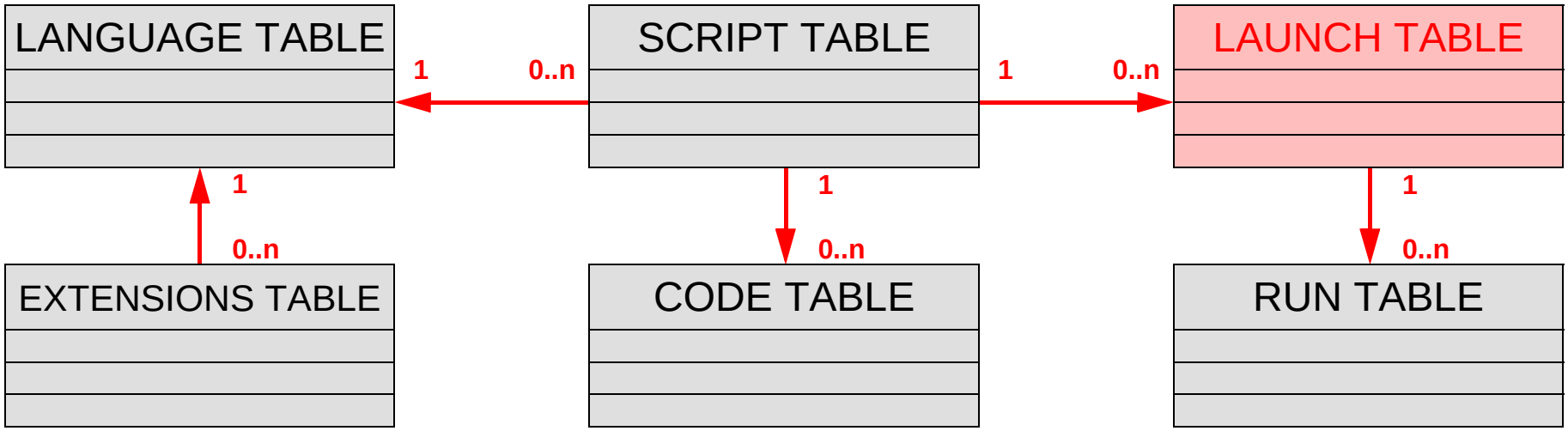
LISTS THE CODE OF A SCRIPT

ALLOWS TO:

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

IMPLEMENTATION IS OPTIONAL

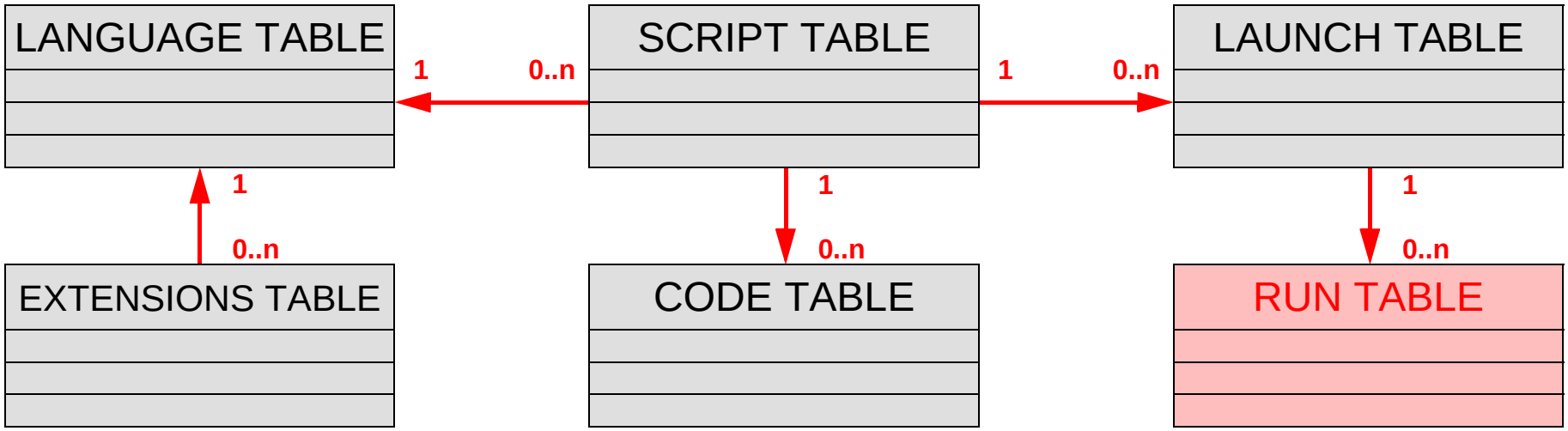
SCRIPT MIB: LAUNCH TABLE



ALLOWS TO:

- ASSOCIATE A SCRIPT WITH A ‘PERSON’ WHO INVOKES EXECUTION
- PROVIDE ARGUMENTS AND PARAMETERS FOR SCRIPT INVOCATION
- INVOKE SCRIPTS WITH A SINGLE SET OPERATION
- 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

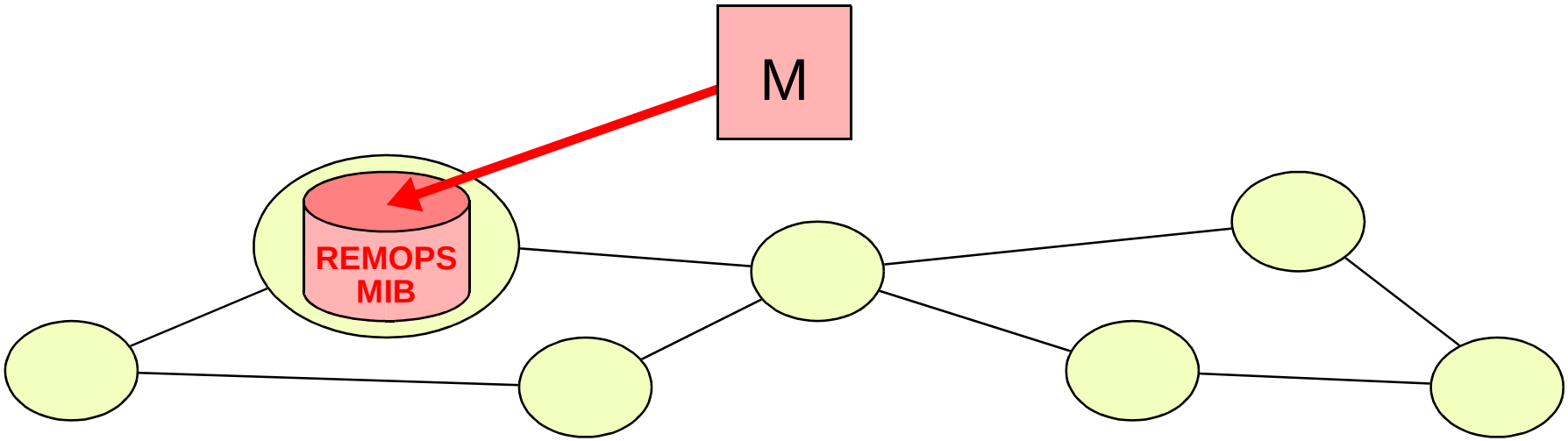
• TO PERFORM PING FROM A REMOTE HOST

TRACEROUTE MIB

• TO PERFORM TRACEROUTE FROM A REMOTE HOST

NAME LOOKUP MIB

• TO PERFORM NAME LOOKUP FROM A REMOTE HOST



EXTENSIBLE AGENTS

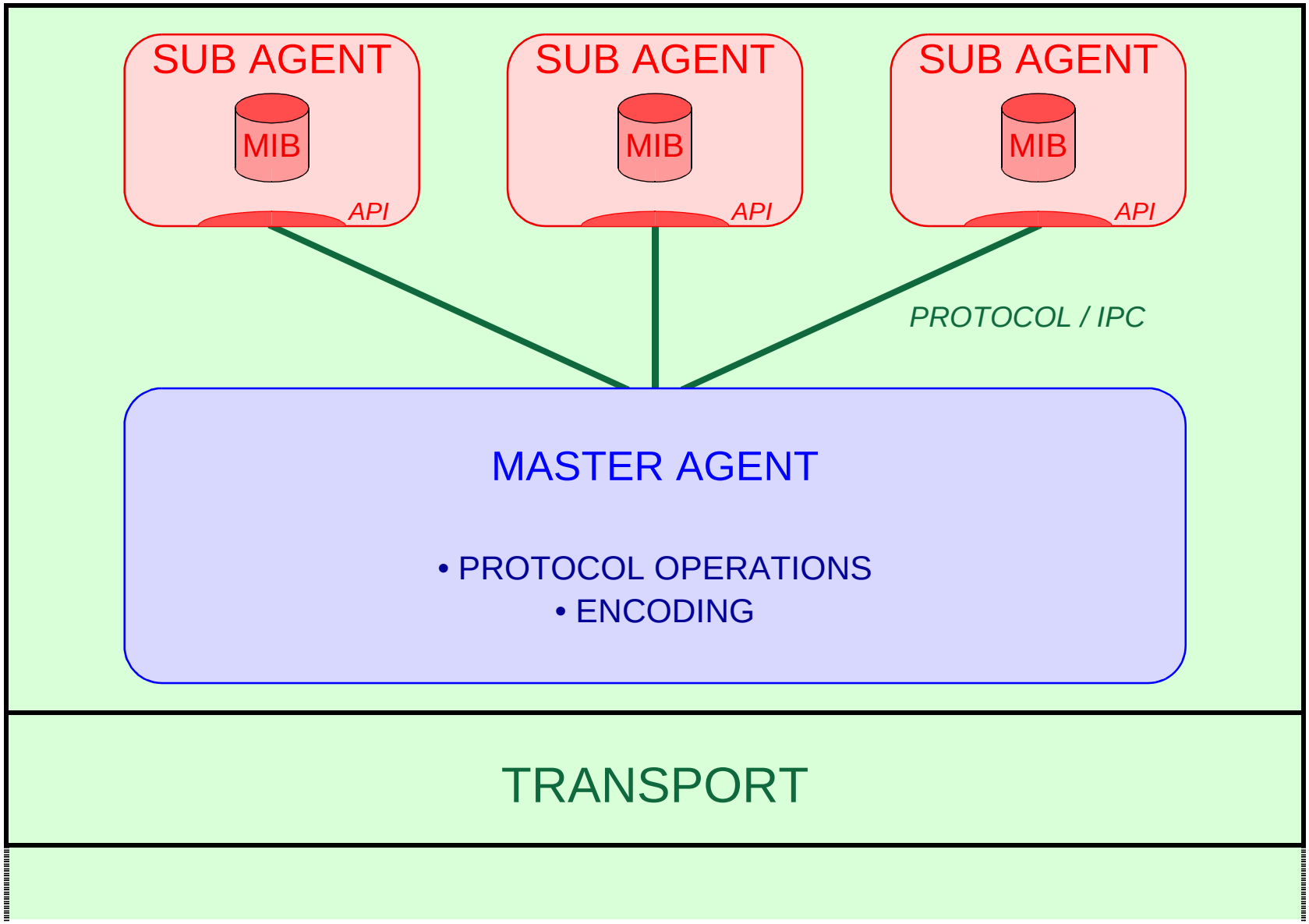
FACILITATE THE EXTENSION OF SNMP AGENTS
WITH NEW MIB MODULES

- SEPARATE SNMP PROTOCOL ENGINE
FROM MIB INSTRUMENTATION

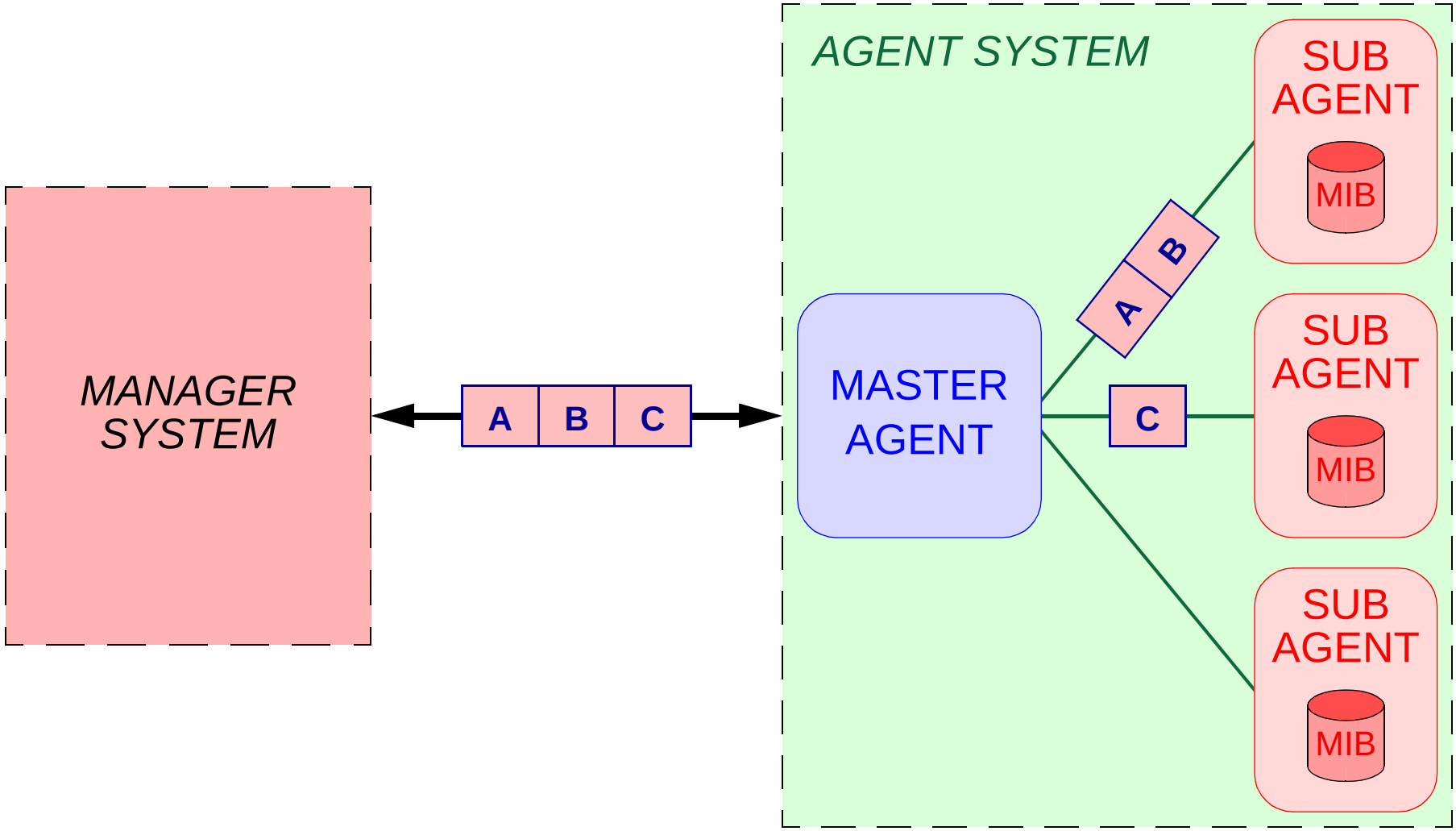
- ALLOW DYNAMIC ADDITION
OF NEW MIB MODULE IMPLEMENTATIONS

- EXTENSIBLE AGENTS SHOULD BE TRANSPARENT

BASIC STRUCTURE



SPLITTING OF VARBIND LIST



CHARACTERISTICS

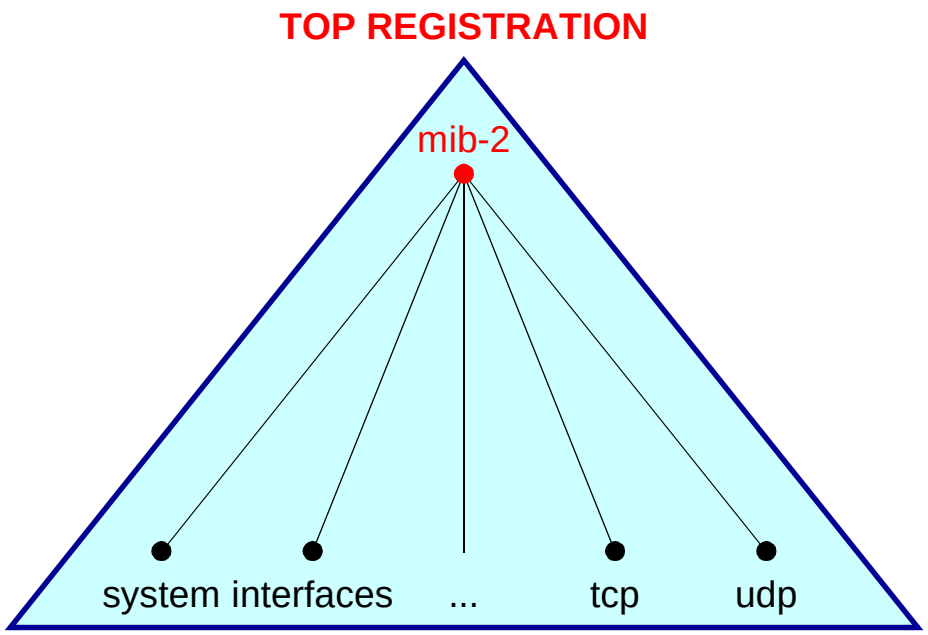
REQUIRES OID REGISTRATION:

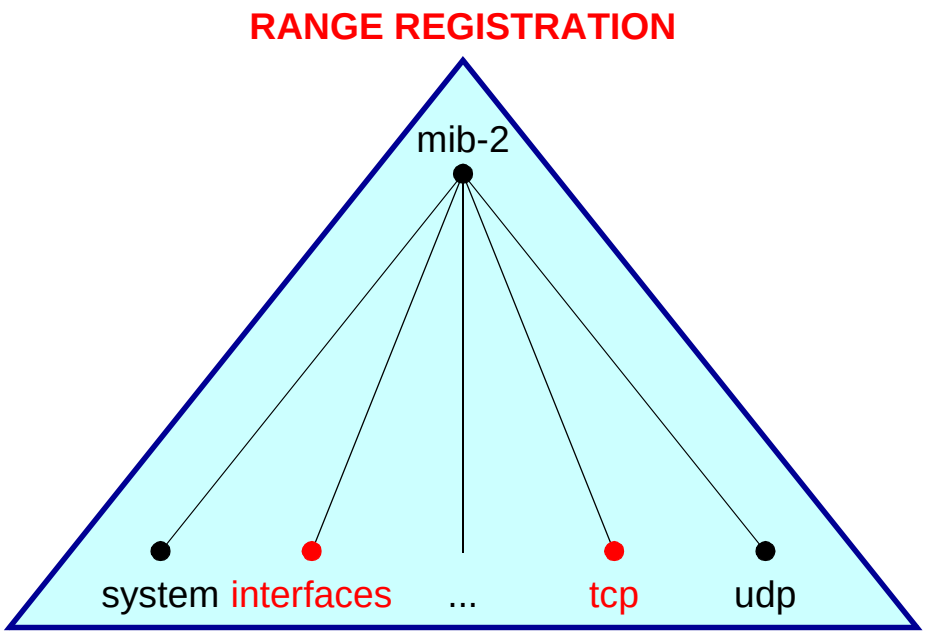
- TOP REGISTRATION

EXAMPLE: REGISTER(mib-2)

- RANGE REGISTRATION

EXAMPLE REGISTER(interfaces -> tcp)

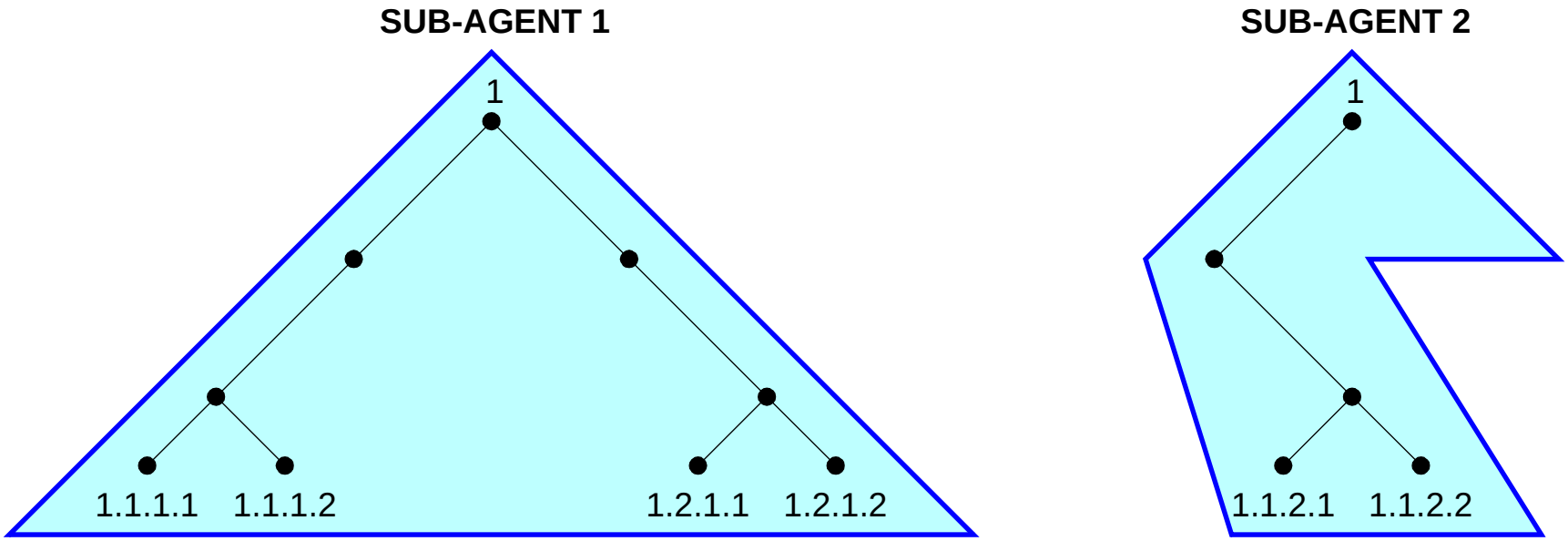




POTENTIAL PROBLEMS

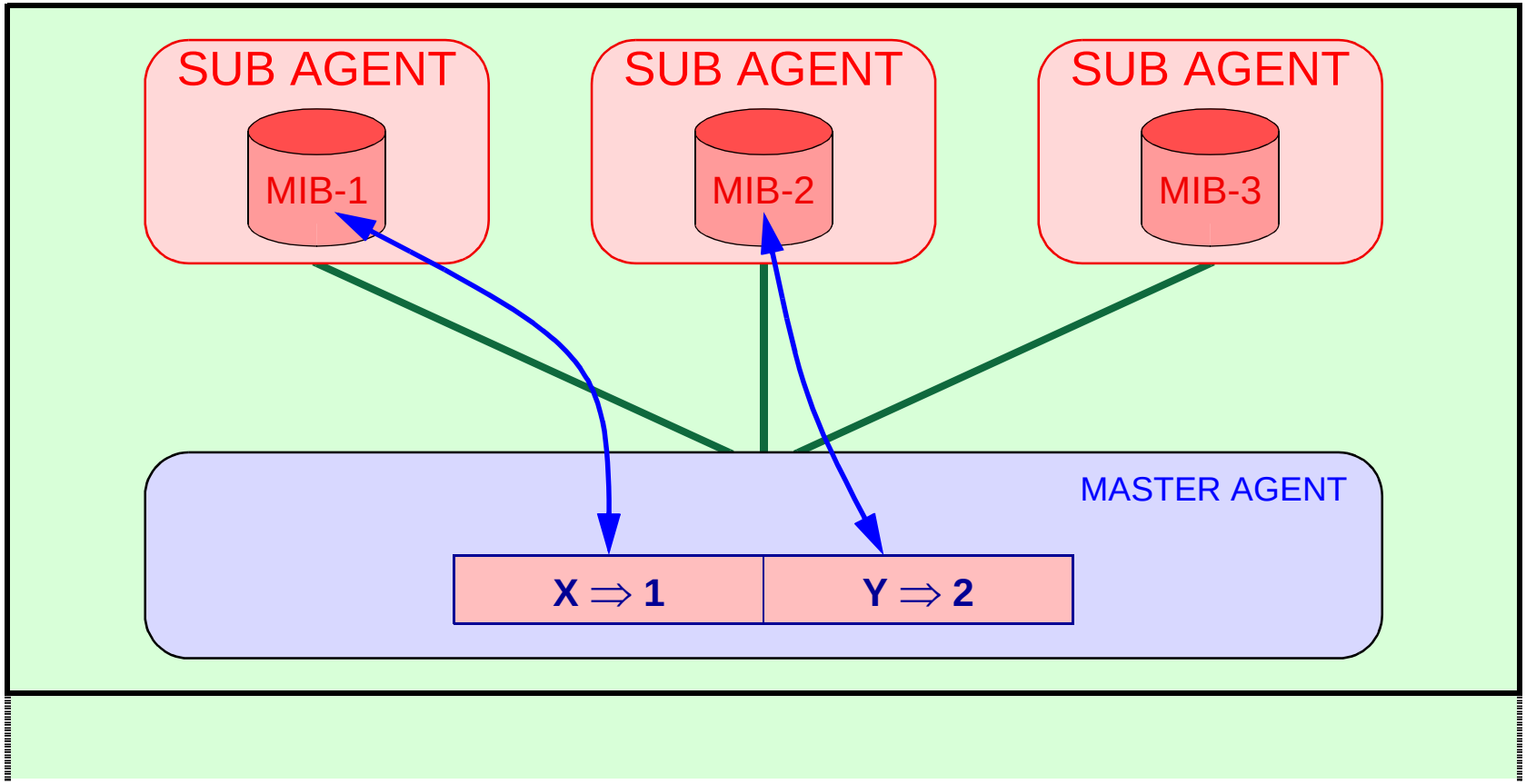
- TABLE ENTRIES MAY BE CREATED AND DELETED AT RUN-TIME
- ENTRIES OF A SINGLE TABLE MAY BE LOCATED IN DIFFERENT SUBAGENTS
- DUPLICATED OIDs
- GAPS
- SETS
- sysUpTime

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

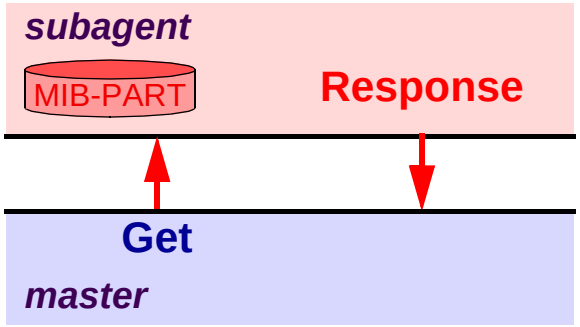
SUPPORTS

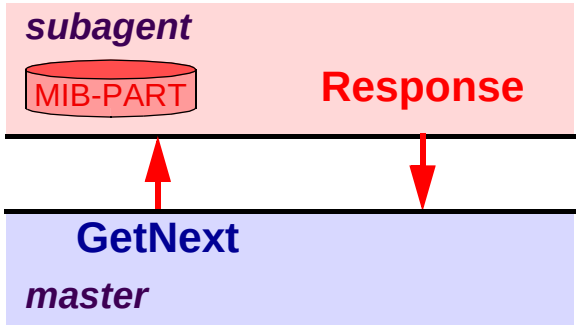
- SUBAGENTS IMPLEMENTING SEPARATE MIB MODULES
- SUBAGENTS IMPLEMENTING ROWS IN "SIMPLE TABLES"
- SUBAGENTS SHARING TABLES ALONG NON-ROW BORDERS

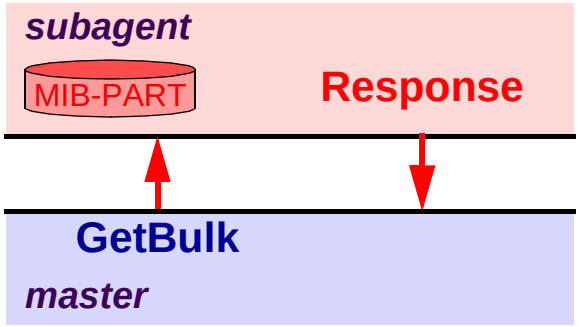
NON-GOALS

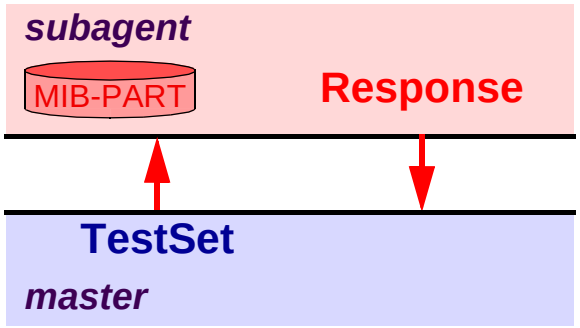
- SUBAGENTS SHARING "COMPLEX TABLES"
- SUBAGENT TO SUBAGENT COMMUNICATION

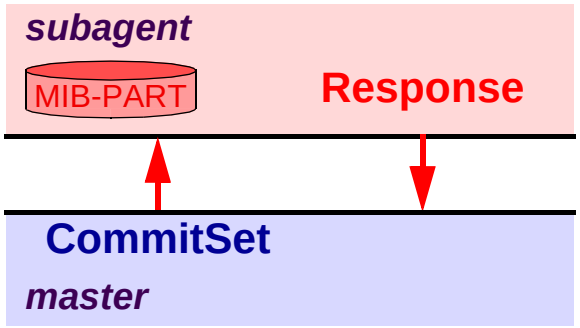
AGENTX - NORMAL PDUS

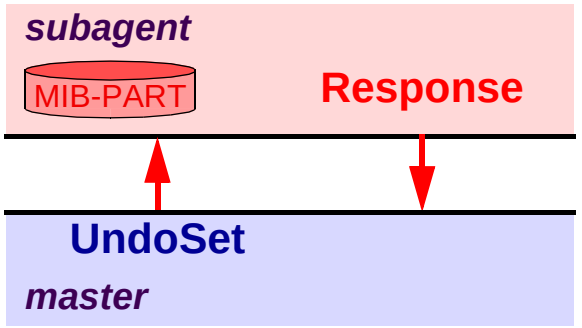


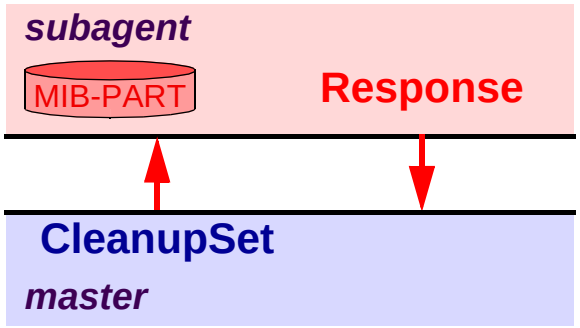


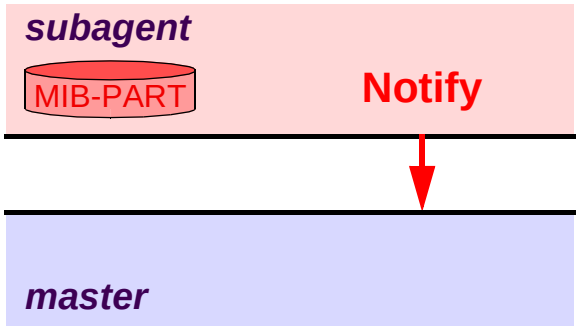




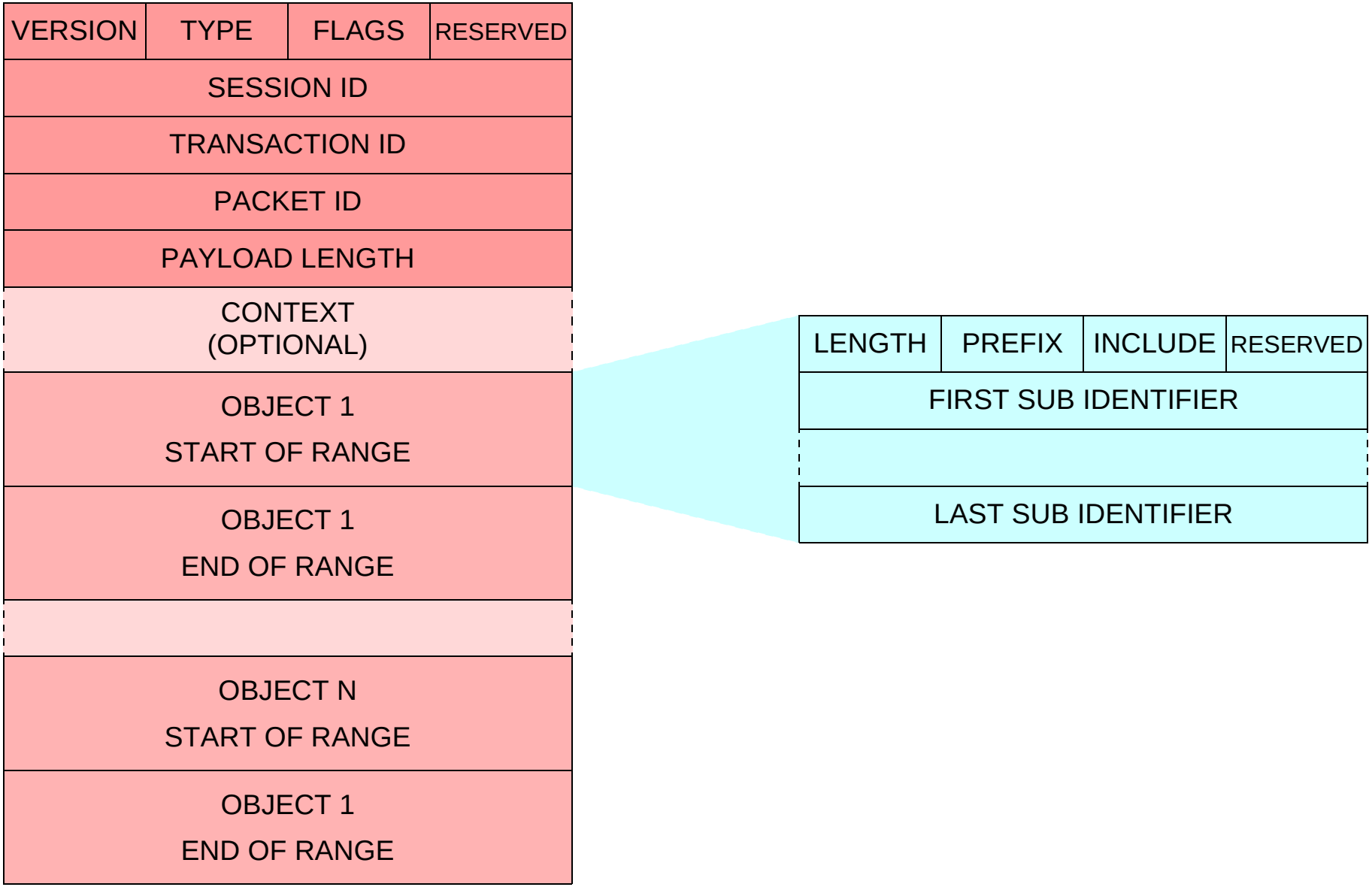








EXAMPLE: PDU FORMAT OF GetNext



AGENTX - ADMINISTRATIVE PDUS

Open
Close

AddAgentCaps
RemoveAgentCaps

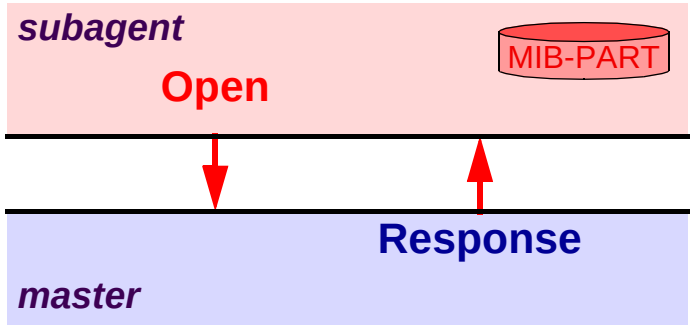
Register
Unregister

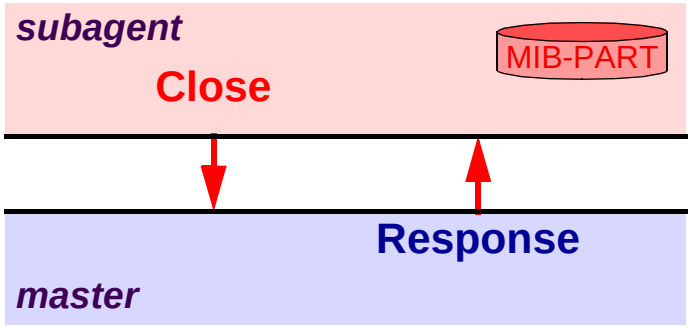
IndexAllocate
IndexDeallocate

Ping

Response

OPEN & CLOSE





TO ESTABLISH A SESSION

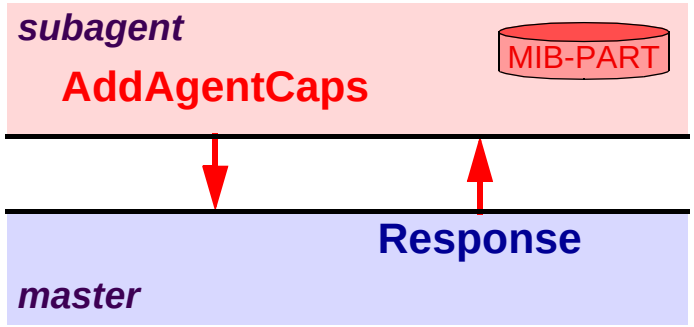
A UNIQUE sessionID IS ASSIGNED

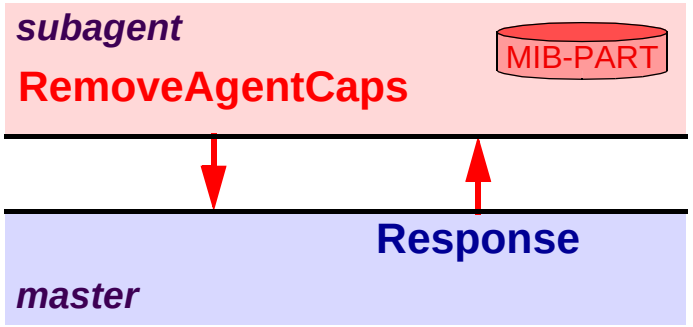
SUBAGENT SPECIFIES DEFAULT TIME-OUT

RESPONSES FROM MASTER ALWAYS INCLUDE sysUpTime

SESSION CAN BE CLOSED BY MASTER OR SUBAGENT

AGENT CAPABILITIES



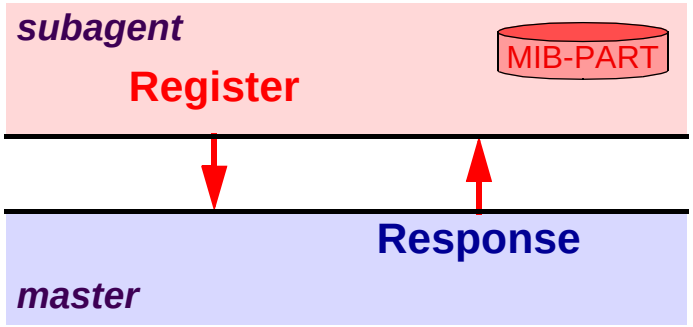


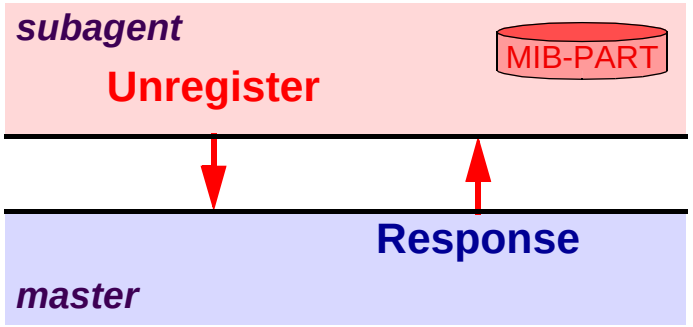
TO INFORM THE MASTER OF THE AGENT'S CAPABILITIES

CAPABILITIES ARE DEFINED AS:
- AN OBJECT ID
- A HUMAN READABLE STRING

THE CAPABILITIES ARE STORED IN THE sysORTable

REGISTRATION





CHOICE BETWEEN:

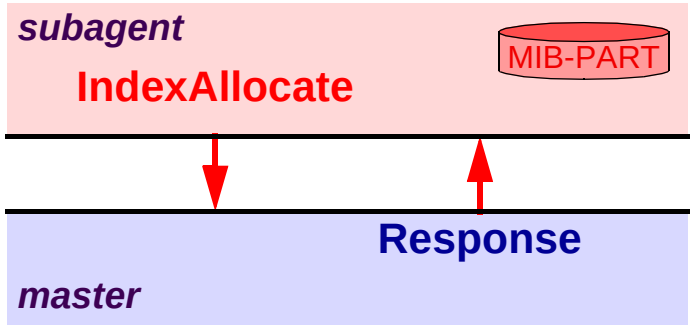
- TOP REGISTRATION
- RANGE REGISTRATION

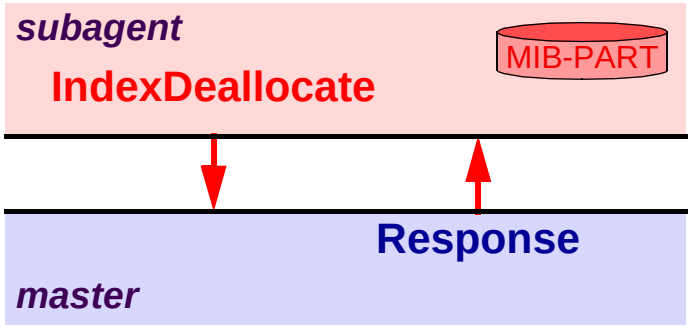
PRIORITY CAN BE SPECIFIED

- TO DETERMINE THE AUTHORITATIVE SUBAGENT

TIME-OUT CAN BE SPECIFIED

INDEXALLOCATION





TO ALLOCATE ONE OR MORE TABLE ROWS

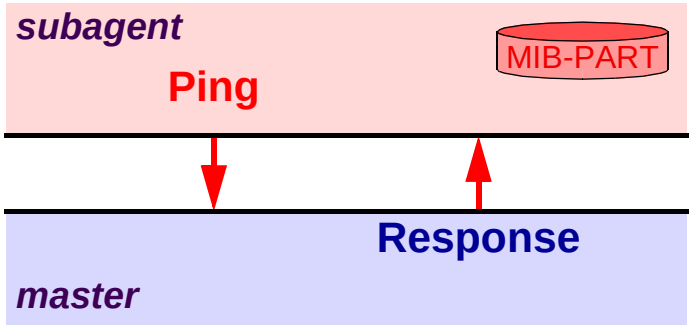
SUBAGENT REQUESTS ALLOCATION OF:

- 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

AGENTX TRANSPORT MAPPINGS

UNIX SOCKETS

- var/agentx/master

TCP

- PORT 705

ANYOTHER IPC MECHANISM

- FOR EXAMPLE: SHARED MEMORY

SNMP VERSUS CMIP - 1

	CMIP	SNMP
model	event based	polling based
information approach	object oriented	variable oriented
complexity	agent is complex	agent is simple
state information	kept by agent	kept by manager
underlying service	CO - reliable	CL - unreliable
efficiency	good	acceptable
implementation	difficult	simple *(V2& V3 are more difficult)*

SNMP VERSUS CMIP - 2

	CMIP	SNMP
retrieves	objects	scalars
many items	multiple replies	error: tooBIG
object selection	scoping & filtering	-
synchronization	atomic & best effort	atomic
events / traps	confirmed & unconfirmed	unconfirmed
actions	possible	via ‘trick’

SNMP VERSUS CMIP - 3

	CMIP	SNMP
security	via underlying services	- *authentication / encryption / ACL-lists*
management functions	many	none
ASN.1	full support	subset
naming structure	flexible	simple

SNMP VERSUS CMIP - CONCLUSION

	CMIP	SNMP
price	**high**	**low**
market acceptance	**no**	**yes**

CONCLUSIONS

NEW DEVELOPMENTS

COMMERCIAL SNMP SOFTWARE

FREE SNMP SOFTWARE

WWW SERVERS

BOOKS

ARTICLES

RECENT DEVELOPMENTS

POLICY BASED MANAGEMENT

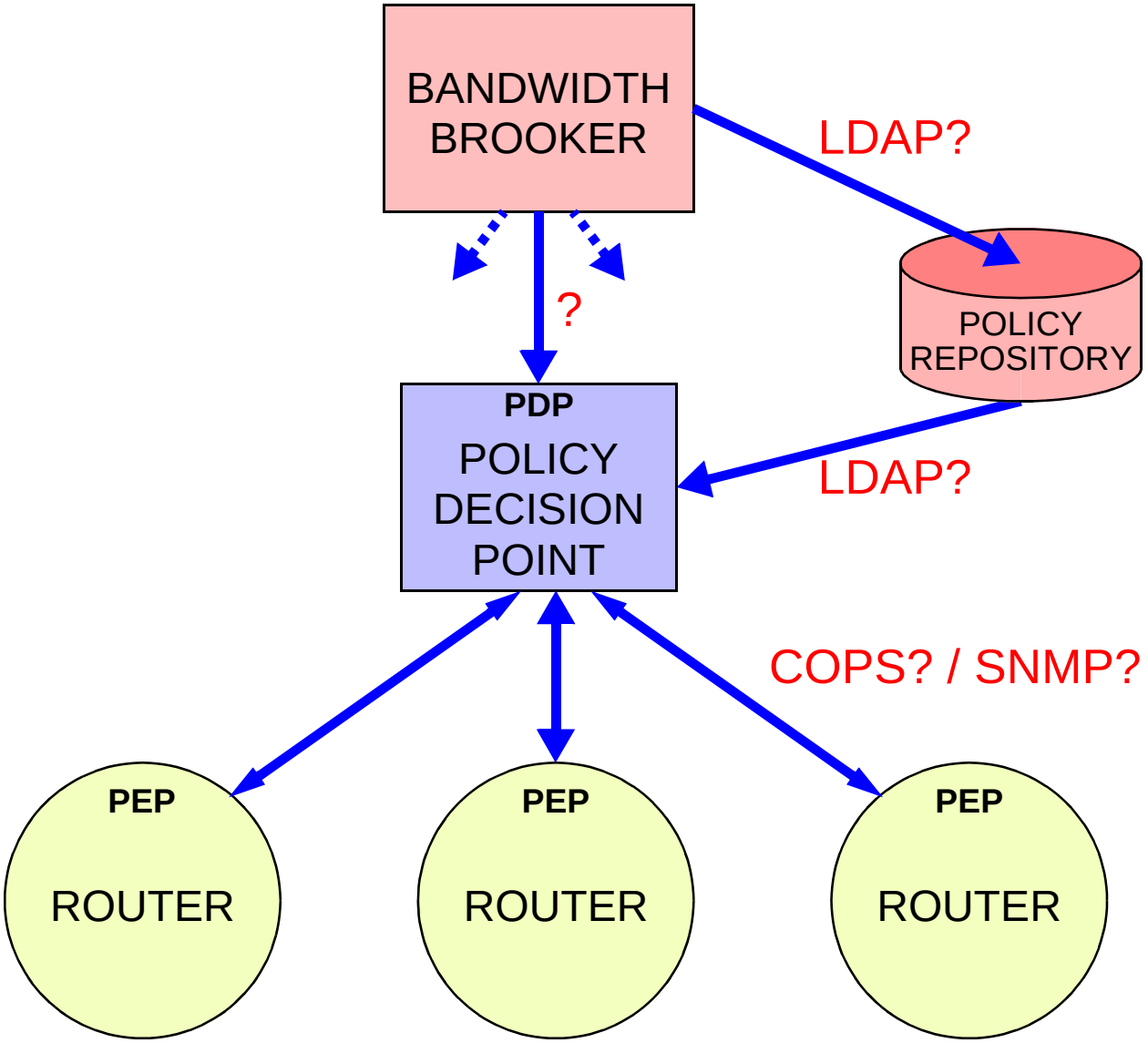
IRTF SERVICES MANAGEMENT WG

IRTF NETWORK MANAGEMENT WG

IETF SMIng WG

IETF EOS WG

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

IRTF NETWORK MANAGEMENT RESEARCH GROUP

EFFICIENT TRANSFER OF BULK MANAGEMENT DATA

- SNMP OVER TCP
- COMPRESSION
- GET-SUBTREE OPERATOR

SMI NEXT GENERATION

- INDEPENDENT FROM OTHER EXTERNAL STANDARDS
- BASED ON AUGMENTED BNF
- MORE DATA TYPES
- EASIER TO PARSE

ACTIVE MANAGEMENT

- ALLOW MANAGEMENT FUNCTIONS WITHIN MIBs
- CAN BE INTEGRATED WITH SMIv2
- CAN BE USED OVER SNMP OR COPS
- POWERFUL NEW IDEA!

IETF SMIng WG

IETF EOS WG

- MECHANISMS TO DETERMINE THE CAPABILITIES OF A SNMP ENTITY

- MECHANISMS FOR EFFICIENT RETRIEVAL, CREATION, AND DELETION OF TABLE ROWS

- MECHANISM TO DELETE AN ENTIRE SUBTREE OF MANAGED OBJECT INSTANCES

- COMPRESSION OF OIDS

- BULK TRANSFER OF SNMP DATA

CHARTER: http://www.ietf.org/html.charters/eos-charter.html

COMMERCIAL SNMP SOFTWARE

BULL OPENMASTER
http://www.openmaster.com/ism

HP OPENVIEW
http://www.openview.hp.com/

SNMP RESEARCH
http://www.snmp.com/

CABLETRON SPECTRUM
http://www.cabletron.com/spectrum/

SUN SOLSTICE
http://www.sun.com/solstice/

TIVOLI
http://www.tivoli.com/

FREE SNMP SOFTWARE

CMU
http://www.net.cmu.edu/groups/netdev/software.html

UCD
http://ucd-snmp.ucdavis.edu/

Scotty
http://wwwhome.cs.utwente.nl/~schoenw/scotty/

JMAPI
http://java.sun.com/products/JavaManagement/

Advent
http://www.adventnet.com/

ModularSnmp
http://www.teleinfo.uqam.ca/snmp/

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