Towards secure and distributed management in the Internet

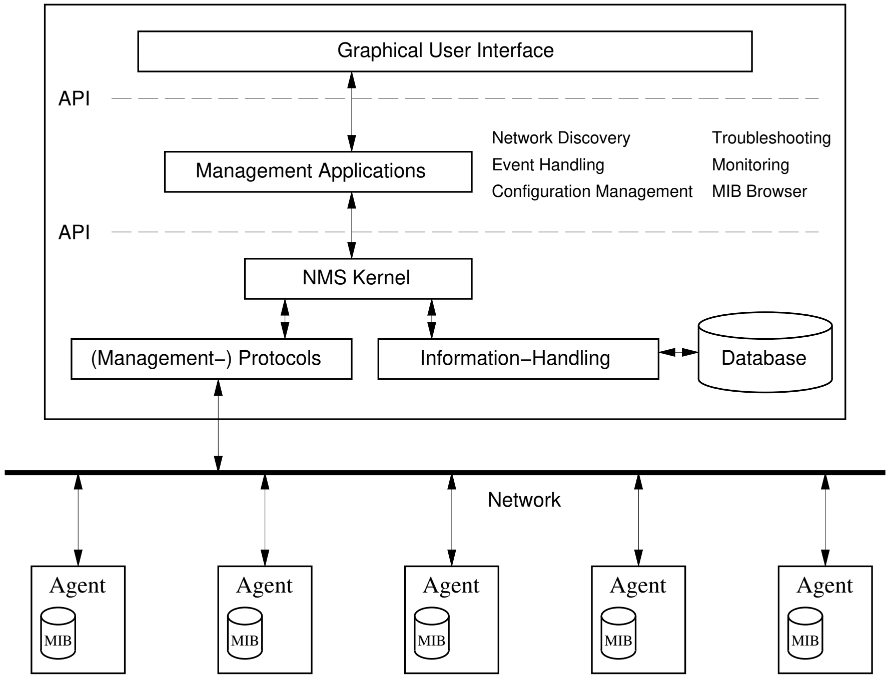
Jürgen Schönwälder

Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig
Bültenweg 74/75

38106 Braunschweig

E-mail: schoenw@ibr.cs.tu-bs.de
Phone: +49 531 391 3289

Centralized Internet Network Management

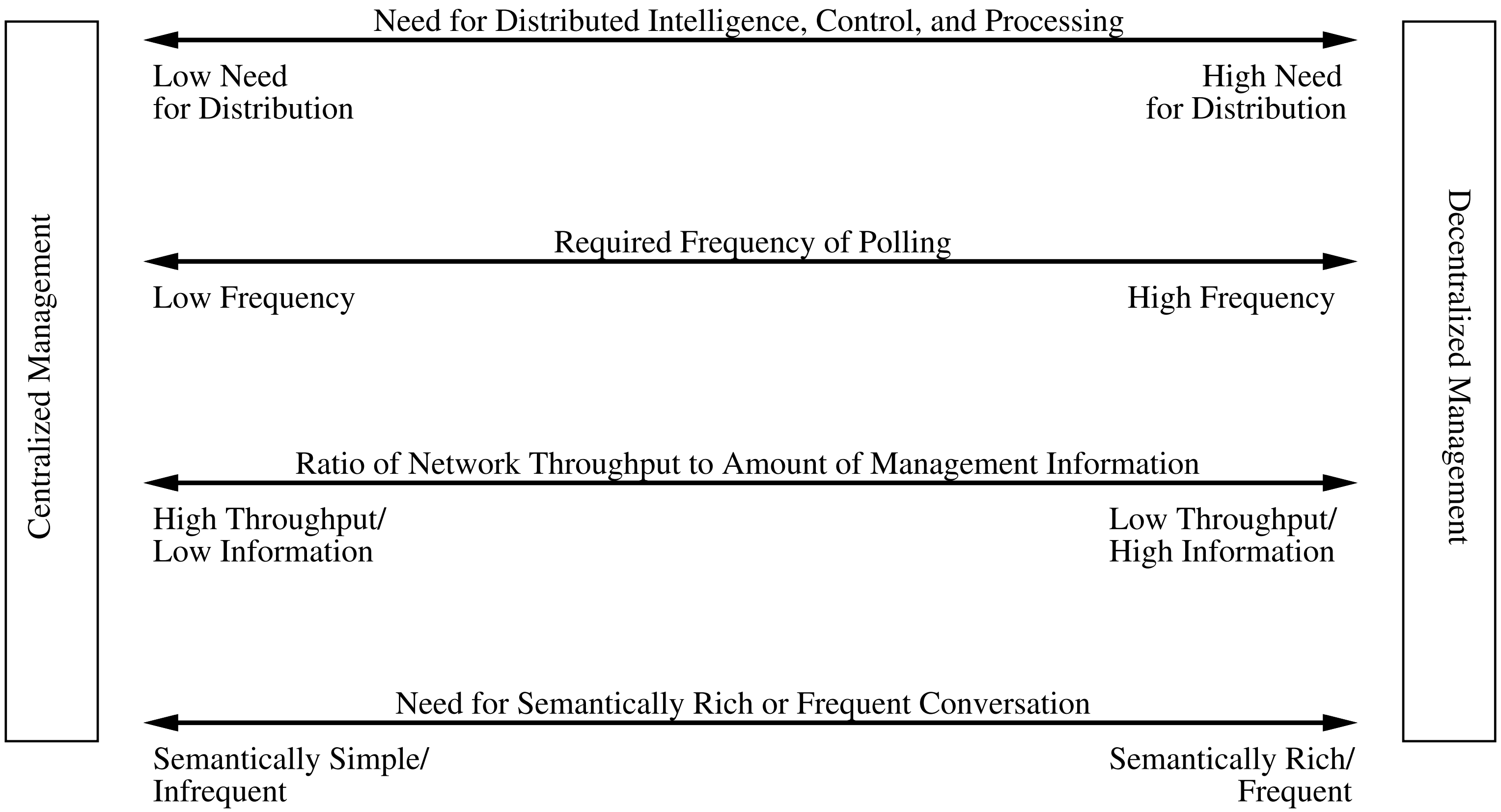


- Management applications and platforms realize all management functions.
- Most SNMP agents only provide access to management instrumentation.
- SNMP MIBs define only a few control objects and a small set of notifications.

Problems with centralized Network Management

- Central management system is a single point of failure.
- Centralized network management does not scale:
  - High communication costs while moving data to applications.
  - Simultaneous management applications require very fast hardware.
  - Management of geographically distributed networks is expensive.
- No support for the delegation of authority in large organizations.

Centralized vs. Distributed Management (Meyer 1995)



Management by Delegation (Yemini, Goldszmidt 1991)

- Smart agents with the ability to execute management scripts written in a management scripting language (MSL).

- Manager distributes management scripts to agents in order to expand their functionality at run-time (remote evaluation, mobile code).

- Delegated management scripts operate in local environments and reduce the amount of management information moved over (expensive) networks.

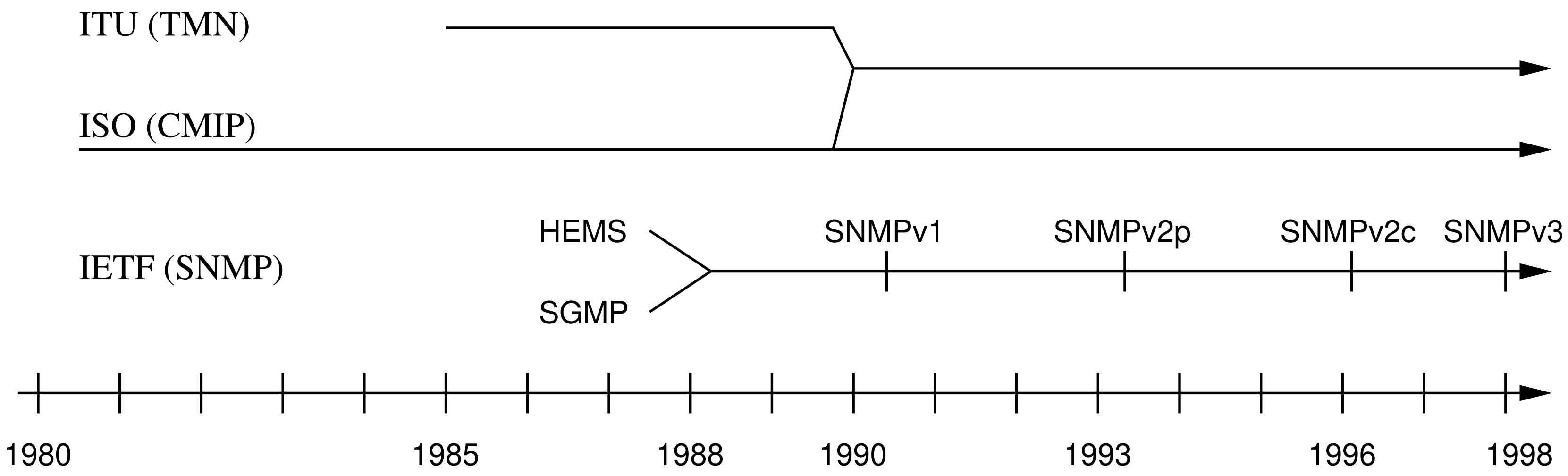
- Delegated scripts operate independent of the manager (self-managing or self-healing networks).

- Manager is concerned with the management of delegated scripts.

Structure of the Talk

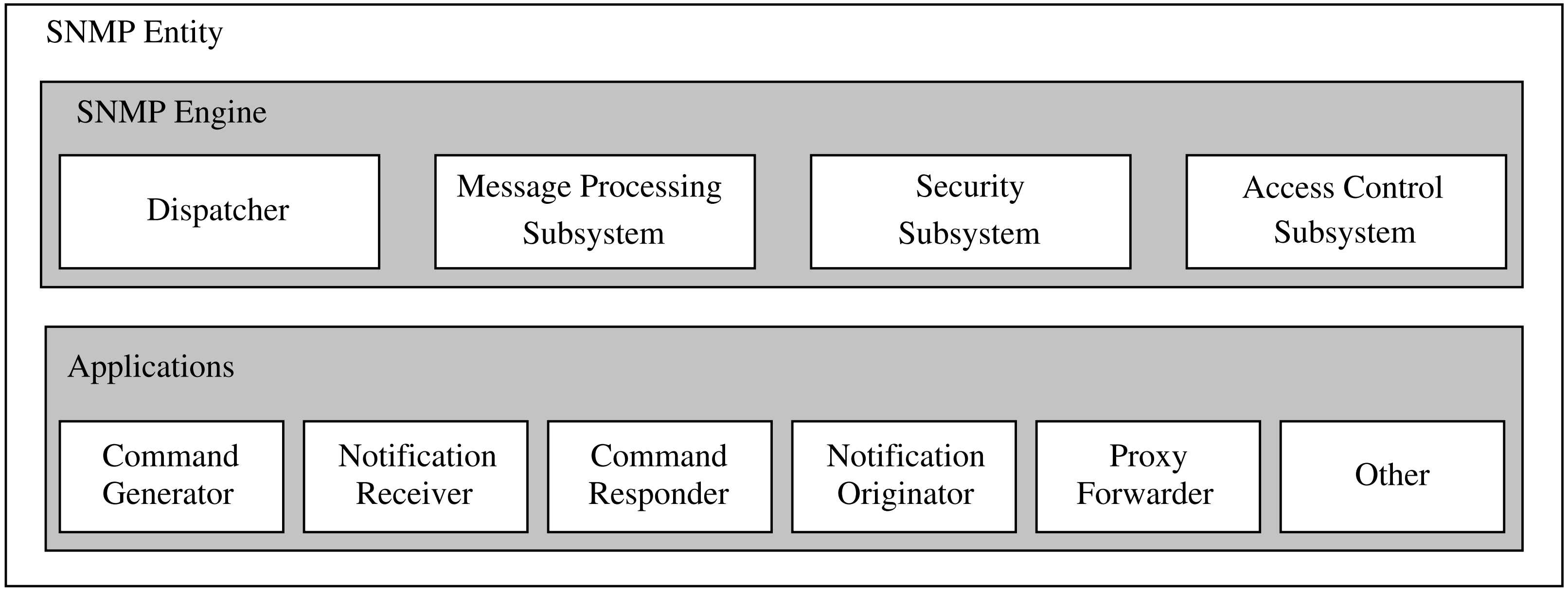
- Introduction and Motivation
- Evolution of SNMP Technology
  - Architecture for Describing SNMP Frameworks
  - SNMP Version 3 Applications
  - SNMP Version 3 View-Based Access Control
  - SNMP Version 3 User-Based Message Security
  - SNMP Version 3 Messages Format
- Distributed Management in the Internet
  - Expressions and Thresholds
  - Scripting (Management by Delegation)
- Current Status and Future Work

Evolution of SNMP Technology



- SNMP version 1 (SNMPv1) was very successful and is the installed base.
- Party-based SNMP version 2 (SNMPv2p) was too complex to deploy.
- Community-based SNMP version 2 (SNMPv2c) came out as a "compromise".
- SNMP version 3 finishes work on secure SNMP and merges competing proposals.

Architecture for Describing SNMP Frameworks



- Each subsystem can have multiple realizations (models).
- A combination of particular set of models defines an SNMP framework.
- Modularization allows to overcome the classic manager/agent model.

Elements of an SNMP Engine

- Dispatcher

  Responsible to dispatch messages and PDUs to the correct message processing model or application within an SNMP entity.

- Message Processing Subsystem

  Responsible for preparing outgoing messages and processing received messages. Can contain multiple message processing models.

- Security Subsystem

  Provides security services such as authentication and privacy of messages by means of one or more security models.

- Access Control Subsystem

  Provides authorization services by means of one or more access control models.

SNAMP Applications

- Command Generator:

  Applications which initiate SNMP Get, GetNext, GetBulk and/or Set operations.

- Command Responder:

  Applications which respond to SNMP Get, GetNext, GetBulk and/or Set operations.

- Notification Originator:

  Applications which generate notifications (Traps and Informs!).

- Notification Receiver:

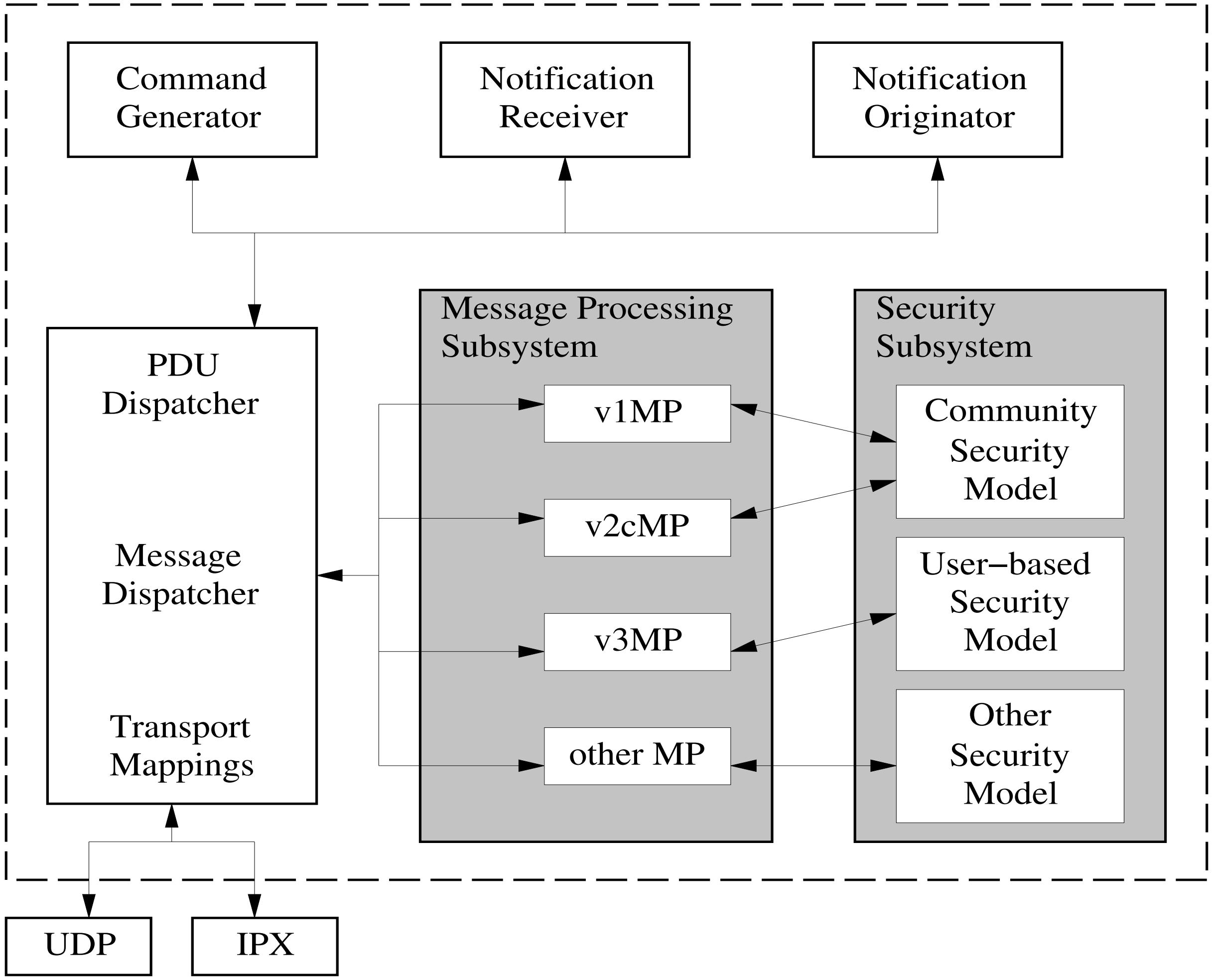
  Applications which receive notifications.

- Proxy Forwarder:

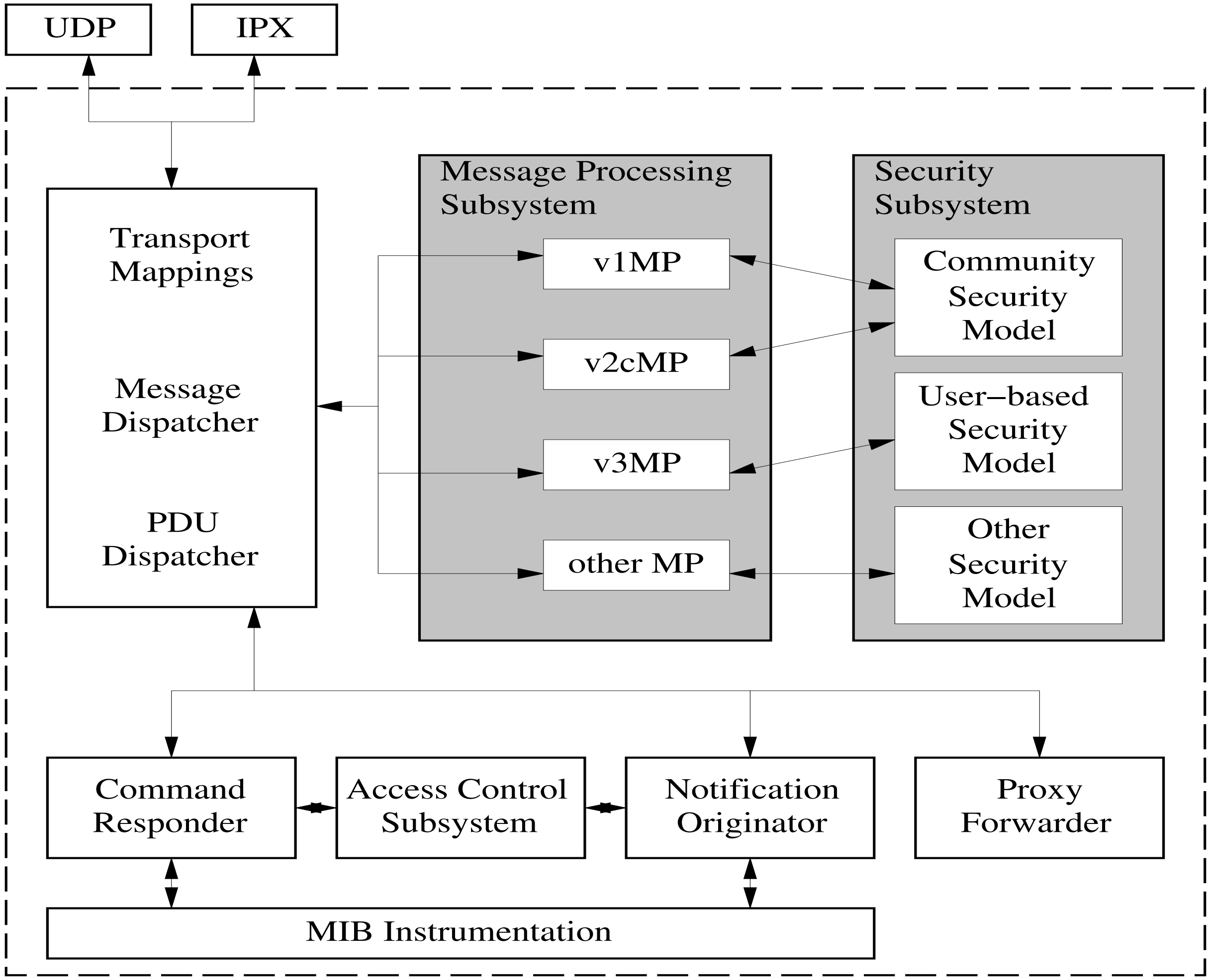
  Applications which forward SNMP messages to other SNMP engines.

$\Longrightarrow$ It is possible to define additional application types, e.g. distributed managers.

Traditional SNMP Manager



Traditional SNMP Agent



SNMPv3 Target MIB

- The snmpTargetAddrTable lists target addresses and transport related parameters (timeout, retries).
- The snmpTargetParamsTable lists SNMP parameters such as security model, security level or security name.
- SNMP parameters can be shared between multiple targets.
- Targets can have a list of tags which identify properties of targets.
- A single management operation can be applied to a set of targets by specifying restrictions on the set of tags (late binding).
- Tags are important hooks for the work on distributed management.

SNMPv3 Notification MIB

- The snmpNotifyTable selects the set of targets that should receive a notification and whether the notification is send as a trap PDU or as an inform PDU.

- The snmpNotifyFilterProfileTable associates a particular notification filter with a target.

- The snmpNotifyFilterTable defines a filter which matches the notification type (the value of snmpTrapOID) against a subtree pattern.

- A positive match can inculde or exclude a notification type.

- No filtering is performed on targets that do not have a filter.

SNMPv3 View-Based Access Control

- Multiple securityNames (principals) can be mapped to a single group.
- Different access rights for members of a group for different security levels.
- A MIB view is the union of a set of "view subtrees", where each view subtree is a subtree within the managed object naming tree.
- The read-view represents the set of object instances authorized for the group when reading objects.
- The write-view represents the set of object instances authorized for the group when writing objects.
- The notify-view represents the set of object instances authorized for the group when sending objects in a notification.
- A MIB is provided to configure MIB views, groups and group membership.

SNMPv3 User-Based Message Security

- Protection against the following security threads:
  - Modification of Information
  - Masquerade
  - Message Stream Modification
  - Disclosure
- Security services provided by the User-Based Security Model:
  - Data Integrity
  - Data Origin Authentication
  - Data Confidentiality
  - Message timeliness and limited replay protection

SNMPv3 User-Based Message Security

- Keyed-Hashing for Message Authentication (HMAC).
- Authentication with HMAC-MD5 (required) and HMAC-SHA-1 (optional).
- Replay protection through loosely synchronized clocks.
- Encryption with the CBC-DES Symmetric Encryption Protocol (optional).
- A MIB module which contains security related error statistics and a table used to configure users and secrets.

SNMPv2 PDU Format

GetRequest, GetNextRequest, SetRequest, Trap, InformRequest, Report PDU

PDU type	request–id	0	0	VarBindList

Response PDU

PDU type	request–id	error–status	error–index	VarBindList

GetBulkRequest PDU

PDU type	request–id	non–repeaters	max–repetitions	VarBindList

VarBindList

name1	value1	name2	value2		name n	value n

- No changes to the SNMPv2 PDU Format (RFC 1905).
- SNMPv3 uses msgID instead of the request-id field to correlate messages.

SNMPv3 Message Format

SNMPv3Message

msgVersion	msgGlobalData	msgSecurityParameter	msgData (encrypted or plaintext ScopedPDU)

MsgGlobalData

msgID	msgMaxSize	msgFlags	msgSecurityModel

UsmSecurityParameter

msgEngineID	msgEngineBoots	msgEngineTime	msgUserName	msgAuthParams	msgPrivParams

ScopedPDU

contextEngineID	contextName	SNMPv2 PDU (as defined in RFC 1905)

- Security parameters (checksums) are encoded in the middle of a message.
- Other security models are possible (e.g. community-based security).

Alternate SNMPv3 Message Format

UsmMessage

msgSecurityModel	msgID	msgSecurityLevel	usmErrorCode	usmEngineID	usmEngineBoots

usmEngineTime	usmUserName	usmAuthParams	usmPrivParams	payload (encrypted or plaintext)

SNMPv3Message

msgVersion	msgMaxSize	msgFlags	msgData (ScopedPDU)

ScopedPDU

contextEngineID	contextName	SNMPv2 PDU (as defined in RFC 1905)

- Security wrapper allows clean separation (even secure SNMPv1 payload).
- No ReportPDUs for clock synchronisation and security related error handling.
- Proposal was too late to be considered seriously by the editor team.

Distributed Management in the Internet

Early Approaches:

- Specialized agents or management stations acting as dual role entities:
  (e.g. Remote Network Monitoring (RMON), Manager-To-Manager MIB)
- Control tables contain parameters that control the activities of an agent.
- Algorithms may be defined to interpret MIB tables as a rule sets (NeTraMet).

Problems:

- Management functions are defined at design/implementation time.
- Parameters are a weak mechanism to control or combine management functions.
- Not flexible enough to implement site-specific management policies.

IETF Standardization (DISMAN WG)

DISMAN services extend the Internet management framework:

Known Systems:	Access and configure the set of known systems.
Management Domains:	Derive target lists for management operations.
Scheduling:	Activate management operations.
Notifications and Logging:	Handling of received or emitted notifications.
Delegation Control:	Protect DISMAN systems and managed networks.
Scripting:	Download and control of management scripts.
Expression and Thresholds:	Computations on MIB objects.

$\Longrightarrow$ Some items have been addressed by the SNMPv3 work!

Expressions and Thresholds

Description:

- Compute simple expressions over MIB objects.
- Generate notifications or do some actions when thresholds are crossed.

Characteristics:

- Good control over the resources used by the expression evaluator.
- Complex but powerful wildcarding of MIB objects (expression arguments).
- Expressions can operate on local or remote systems.
- Usage of SNMPv3 tags to select targets (Target MIB).
- Results of expression are stored as normal MIB objects (stackable expressions).
- Thresholds can be used to generate internal events (much like RMON).
- Internal events can trigger notifications or set operations on MIB objects.

Script-Oriented Management

Description:

- Management operations are performed by scripts that are delegated to managing or managed systems in the network.
- Scripts can be created and modified easily during the operational phase.

Characteristics:

- Scalable management operations (scripts can be adapted to the processing power of the target system).
- Flexibility since management functions can be added and changed at any time.
- Allows managing systems to perform complex "transactions" without paying high costs for correct error handling.
- Introduces a new control problem which needs to be handled.

Design Issues

Delegation Protocol:

management protocol	$\longleftrightarrow$	special purpose protocol
manager "push" (procedures)	$\longleftrightarrow$	agent "pull" (procedures)
agent "push" (results)	$\longleftrightarrow$	manager "pull" (results)

Language:

interpreted language	$\longleftrightarrow$	compiled language
standard language	$\longleftrightarrow$	domain specific language

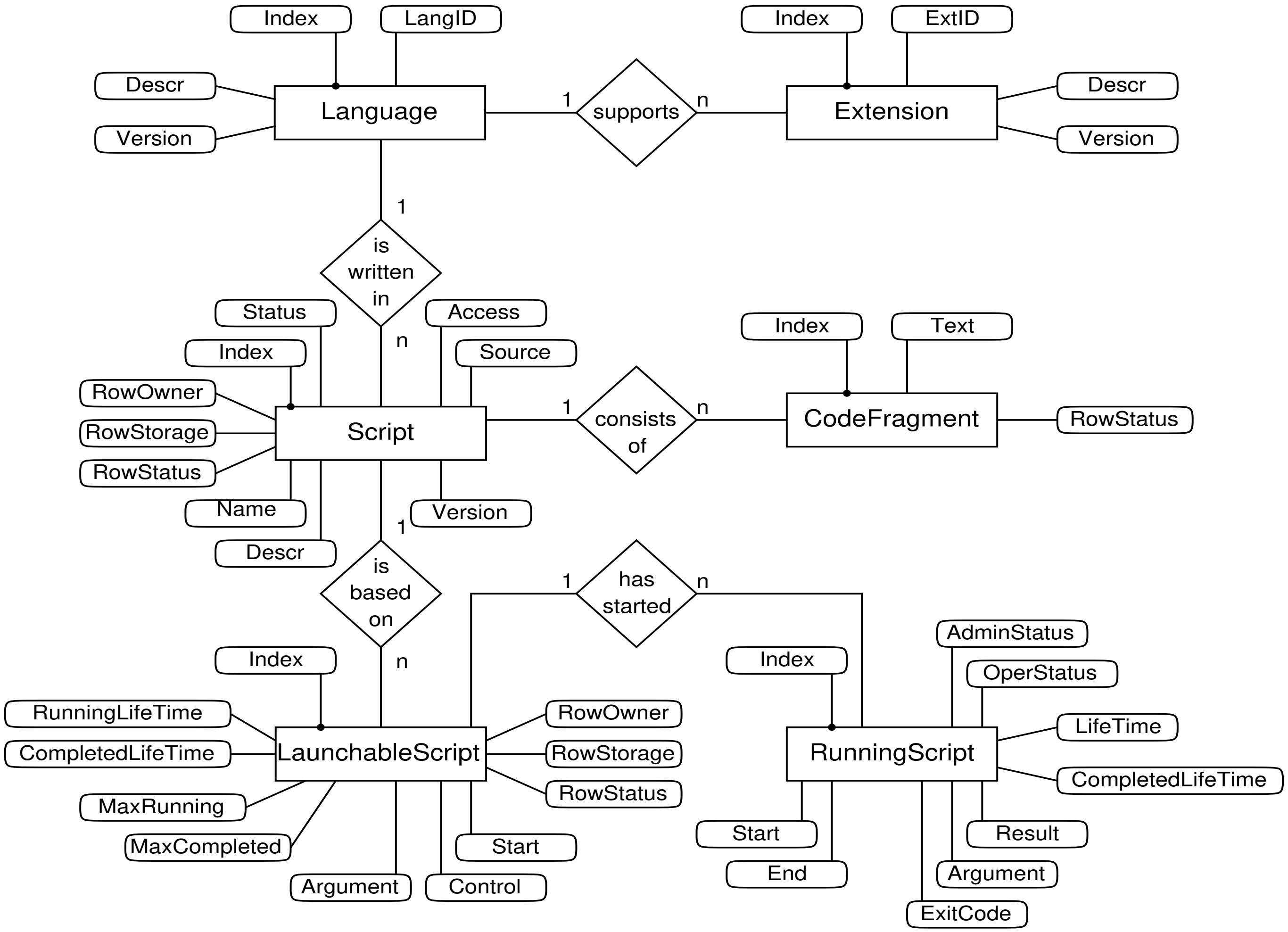
Procedures:

MIB object arguments and results	$\longleftrightarrow$	arbitrary argument and result types
activation by events/scheduler	$\longleftrightarrow$	activation by manager

Runtime:

multiple manager, multiple languages	$\longleftrightarrow$	single manager, single language
multiple threads	$\longleftrightarrow$	event driven
error and status monitoring objects	$\longleftrightarrow$	notifications
multiple security profiles	$\longleftrightarrow$	no security

Structure of the IETF Script MIB



Open Script MIB Issues

- Scheduling functions (launching scripts at regular intervals etc.).
- Applying the view-based access control efficiently to protect scripts and results.
- Modeling of user priviledges and/or runtime profiles.
- Meta-variables to store script results as normal MIB objects.
- Support for build-in scripts (functions) without an associated language.
- Is there a need and a way to standardize scripts (= complex management functions) in addition to SNMP MIBs?

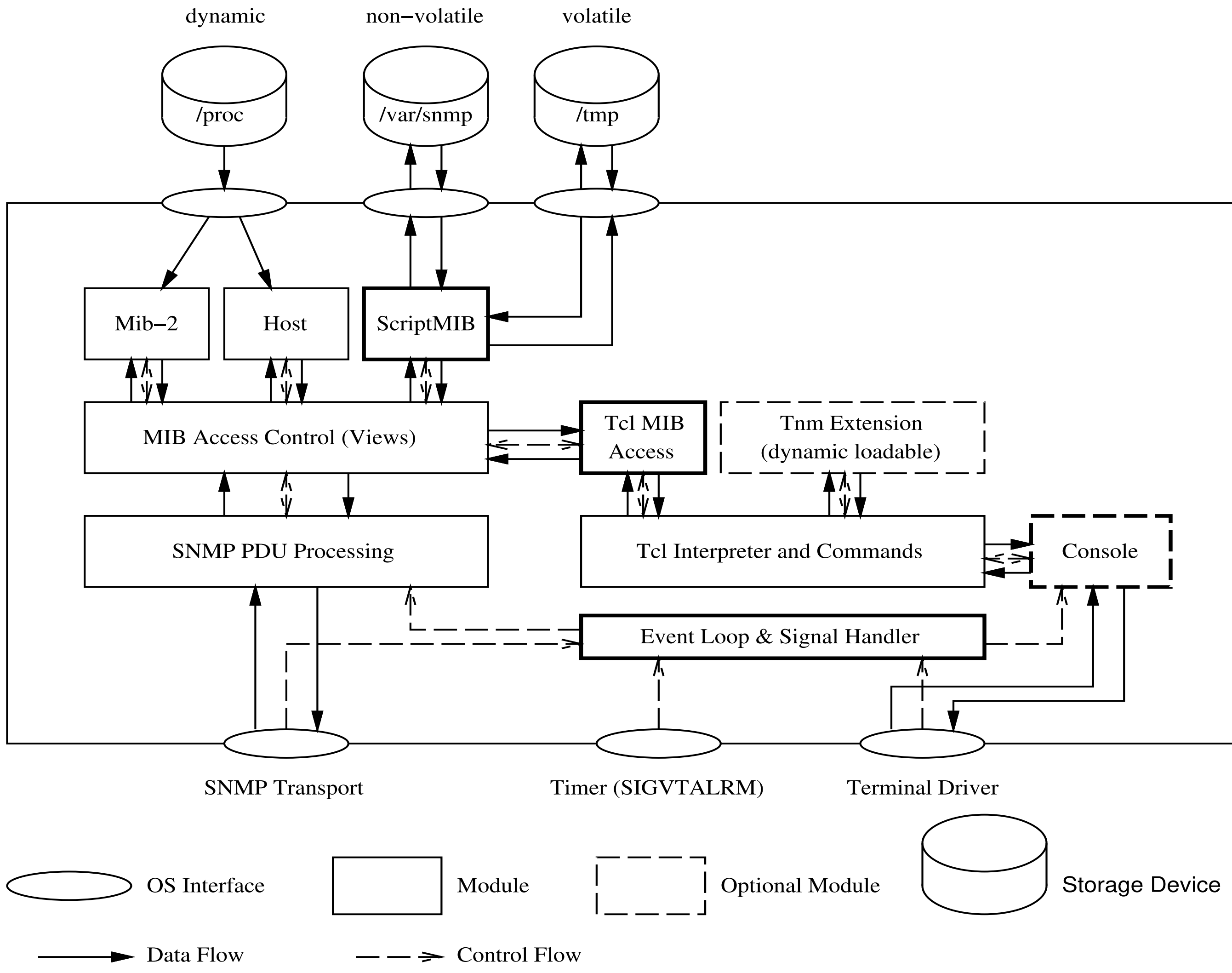
Implementation of the Script MIB (in progress)

Major design choices:

- Selection of scripting language. ($\Rightarrow$ Tcl)
- Selection of process model. ($\Rightarrow$ event-driven, single process, single thread)
- Selection of SNMP environment. ($\Rightarrow$ Linux SNMP agent)

Major work items:

- Language interface to access local SNMP agent.
- Language interface to access remote SNMP agents.
- Implementation of the Script-MIB variables and tables.
- Organization of non-volatile storage.
- Internal scheduling and event handling (expiration of table entries etc.)


dynamic
non−volatile
volatile
/proc
/var/snmp
/tmp
Mib−2
Host
ScriptMIB
MIB Access Control (Views)
Tcl MIB Access
Tnm Extension (dynamic loadable)
SNMP PDU Processing
Tcl Interpreter and Commands
Console
Event Loop & Signal Handler
SNMP Transport
Timer (SIGVTALRM)
Terminal Driver
OS Interface
Module
Optional Module
Storage Device
Data Flow
Control Flow

Current Status and Future Work

SNMPv3:

- SNMPv3 specification completed Working Group last call.
- SNMPv3 specification hopefully soon accepted by the IESG and published as "Proposed Standard".
- SNMPv3 implementation, testing and hopefully deployment.
- Revisions of SNMPv3 documents likely to reflect implementation experiences.

DISMAN:

- Several official and unofficial meetings of the DISMAN WG at the 41st IETF in Washington to complete the specifications.
- Hopefully publish and review new Internet Drafts in early 1998.
- Working group last call and publication of the specification probably in spring 1998.