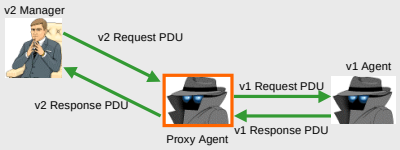
SNMP Proxy Agents

- Koexistenz von verschiedenen Protokollversionen!

Beispiel:



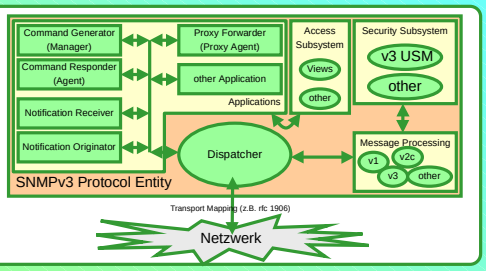
SNMPv2 Nachteile

- zerfaserte Spezifikation
- mangelnde Erweiterbarkeit
- teilweise hohe Komplexität der Spezifikation
- konkurrierende / inkompatible Implementierungen, die dazu führten, daß in weiten Bereichen weiterhin v1 eingesetzt wird

SNMP v3

- 1997 wurde bei der IETF eine Workgroup zum Design der "Next Generation Of SNMP" ins Leben gerufen
- Ergebnis bisher: rfc 2271..2275
- unterstützt SNMPv1, SNMPv2C und zukünftige Ansätze durch austauschbare Message Processing Units und Security Units

SNMP v3 Protocol Entity



SNMP v3 Dispatcher

- Aufgaben:
  - PDUs von und zu den Applikationen transportieren
  - Messages und PDUs mit dem Message Processing Subsystem austauschen, um sie zu ein- und auszupacken
  - Senden und Empfangen von SNMP Messages über das Netzwerk
  - Zustandsinformationen der aktiven Requests vorhalten
- Primitiven (Kommandos):
  - `sendPdu, processResponsePdu, processPdu, returnResponsePdu, registerContextEngineId, unregisterContextEngineId`

SNMP v3 Command Generator

- erzeugt SNMP Get, GetNext, GetBulk bzw. Set request PDUs
- bearbeitet die Responses auf seine Requests

Beispiel:

SNMP v3 Command Responder

- empfängt Get, GetNext, GetBulk, Setrequest PDUs, die das lokale System als Ziel haben (siehe contextEngineID)
- führt die Protokolloperation zur empfangenen Primitive aus
- generiert die passende Response

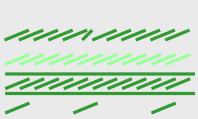
Beispiel:

Notifications

- Traps und Inform-Requests
- Generator
  - verwendet sendPdu des Dispatchers und
  - im Falle einer Inform Response auch processResponsePdu zum
  - versenden von Notifications und empfangen der Antworten darauf
- Receiver
  - empfängt Traps und Inform-Requests mit processPdu vom Dispatcher und
  - verschickt Inform-Responses mit returnResponsePdu

SNMP v3 User Based Security Model (USM)

- Definition in rfc 2274
- Ziele:
  - Datenintegrität
  - Authentizität
  - Vertraulichkeit
  - Aktualität



- Submodule:
  - Authentication
  - Privacy
  - Timeliness

USM Submodule

- Authentication:
  - zuständig für Sicherstellung der Datenintegrität und Prüfung der Herkunft von Anfragen durch
    - HMAC Hashed Message Authentication Codes (rfc 2104) mit MD5 oder SHA-1 Hash Funktion
- Privacy:
  - Verschlüsselung des Datenstroms mit DES CBC (Data Encryption Standard: Cipher Block Chaining - Verschlüsselung aufeinanderfolgender 64-bit Blöcke auf Basis des XOR der vorangegangenen)
- Timeliness:
  - Schutz gegen verspätete oder wiederholte Daten durch
    - Authorisierte Taktung über snmpEngineBoots und snmpEngineTime
    - Synchronisation und Rechtzeitigkeit über Vergleich der Takte

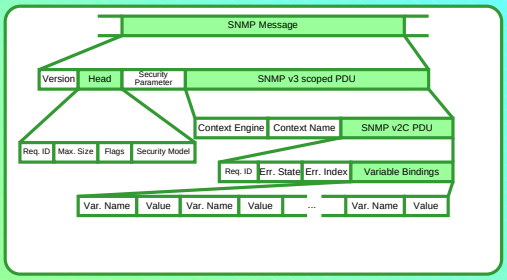
weitere USM Eigenschaften

- Message format:
  - vorgegebene Semantik für Security Parameter in PDUs
- Discovery:
  - definierter Vorgang, mit dem SNMP Protocol Engines einander kennenlernen
- Key Management:
  - das USM verwaltet private Schlüssel und kann sie gegen Passworte prüfen

SNMP v3 Context

- Context: eine Sammlung von Managementinformationen innerhalb einer ProtocolEntity, z.B. ein von der Entity verwaltetes Gerät
- contextEngineId: eindeutige Kennung einer ProtocolEntity in einer Managementdomäne
- contextName: Bezeichnung eines Contexts, eindeutig innerhalb einer ProtocolEntity
- Beispiel:
  - zum Zugriff auf die Beschreibung des ersten Interfaces eines Geräts nötig:
    - die contextEngineId der ProtocolEntity, die die Information liefert
    - der Contextname des Geräts
    - der Variablenname (ifDescr)
    - der Index des Interfaces (1)

SNMP v3 Protokolldaten

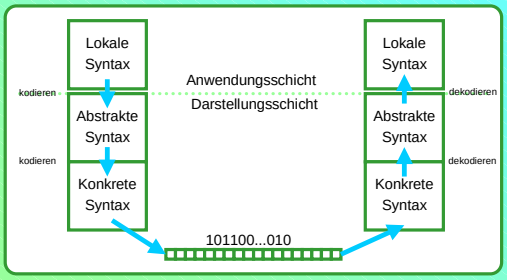


Management Information Base (MIB)

- beschreibt die per SNMP überwachten Managed Objects
- Syntax und Hierarchie der Objekte folgen der Abstract Syntax Notation 1 (ASN.1)
- primitive Datentypen sind:
  - INTEGER
  - BIT STRING
  - OCTET STRING
  - NULL
  - OBJECT IDENTIFIER
- Definition in
  - rfc 1156 (MIB)
  - rfc 1213 (MIB-II)
  - viele Erweiterungen

weitere Akronymbedeutung:

Darstellungskontext



ASN.1

- von OSI in den Standards 8824/8825 beschriebene Objektdefinitionssprache, die mit ihren Basic Encoding Rules (BER) die Übertragung von Daten mit einer eindeutigen, abstrakten Transfersyntax auch zwischen sehr unterschiedlichen Systemen vorsieht
- für die Kodierung ist ein bit-eindeutiges Vorgehen definiert, das die Zahl der übertragenen Bits gering halten soll, jedoch relativ aufwendig ist.

ASN.1 Syntax

<modulename> DEFINITIONS ::=
BEGIN
<linkage>
<declarations>
END

– types
beginnen mit Großbuchstaben
– values
beginnen mit Kleinbuchstaben
– macros
vollständig in Großbuchstaben
– ASN.1 Schlüsselworte
vollständig in Großbuchstaben

```
EXAMPLE-MIB DEFINITIONS ::= BEGIN
 IMPORTS
   mgmt            FROM RFC1155-SMI
   OBJECT-TYPE     FROM RFC-1212;

DisplayString ::= OCTET STRING

system OBJECT IDENTIFIER ::= { mib-2 1 }

sysDescr OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..255))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
  "A textual description of the entity. "
 ::= { system 1 }
END
```


ASN.1 Basic Encoding Rules (BER)

– Regelt, wie Abstrakte Syntax in konkrete Transfersyntax zu übersetzen ist durch

- – TLV-Kodierung
  - – tag Typkennung aus den ASN.1 simple types
  - – length Länge des folgenden Value-Feldes
  - – value Wert der Variablen
- – Kodierungsvarianten
  - – einfach, feste Länge
  - – zusammengesetzt, feste Länge
  - – zusammengesetzt, unbekannte Länge,

ASN.1/BER Beispiel(1)

Lokale Syntax / informelle Darstellung

Name:	John Smith
Titel:	Mr. Wichtig
Kinder:	2

ASN.1 abstrakte Syntax:

	{ vorName "John", nachName "Smith" },
titel	"Mr. Wichtig"
kinder	2

ASN.1 Syntax Deklaration:

```
AdressRecord ::= [APPLICATION 0] SET {
                    Name,
                    titel       [0]     OCTET STRING,
                    kinder      [1]     INTEGER }
Name ::= [APPLICATION 1] SEQUENCE {
                    vorName     OCTET STRING,
                    nachName    OCTET STRING }
```


ASN.1/BER Beispiel(2)

ASN.1 abstrakte Syntax:

	{ vorName "John", nachName "Smith" },
titel	"Mr. Wichtig"
kinder	2

ASN.1 konkrete Transfersyntax:

```
60: Bits: 01 = Anwendungstag 1 = zusammengesetzt, 00000 = Anwendungstag Nummer 0
20: Länge des gesamten Inhalts (falls >127 gibt das erste Byte die Zahl der folgenden Längenbytes an)
        61: Bits: 01 = Anwendung 1= zusammengesetzt, 00001 = Tag Nr. 1
        09: Länge des Namens
            04 04 "John" : OCTET STRING Länge 4: "John"
            04 05 "Smith" : OCTET STRING Länge 5: "John"
        A1: Bits: 10 = Datenkontextspezifisch 1=zusammengesetzt, 00000 = Datentag Nummer 0
            04 0B "Mr.Wichtig" : OCTET STRING Länge 11: "Mr. Wichtig"
        81: Bits: 10 = Datenkontextspezifisch 0= einfach , 00001 = Datentag Nummer 1: Kinderzahl
        01 02: Länge und Wert (Zweierkomplement)
```


RMON

- Remote Monitoring Standard auf SNMP-Basis
- Überwachung/Verkehrsstatistik von Netzwerkbereichen durch Paketuntersuchung
- Zusammenfassung und Aufbereitung von Daten zum Teil im Agent
- MIB-Definition in rfc 1513, 1757, 2021, 2074

RMON Ziele

- Off-line Operation:
  - Agent/Monitor sammelt Informationen für spätere Abfrage
- Proactive Monitoring
  - Agent/Monitor führt kontinuierlich Diagnose und Performancetests durch, sofern seine Ressourcen ausreichen
- Problem Detection and Reporting
  - Vorbeugend (durch Polling) oder passiv durch Erkennung von Zuständen durch den Monitor
- Value-added-data
  - Datensammlung auch in Netzbereichen, die der Manager normalerweise nicht erreicht
- Multiple Managers

Monitoring Information

- Statisch
  - sich nicht oder selten verändernde Zustände, Flags
- Dynamisch
  - Zähler, die sich stetig verändern
- Statistisch
  - von Zeit zu Zeit verglichene Werte

RMON MIBs

- rfc 1513: Sep. 1993 "Token Ring Extensions to the Remode Network Monitoring MIB"
- rfc 1757: Feb. 1995 "Remote Network Monitoring Management Information Base"
- rfc 2021: Jan. 1997 "Remote Network Monitoring Management Information Base II"
- rfc 2074: Jan 1997 "Remote Network Monitoring MIB Protocol Identifiers"

RMON Gruppen

- statistics: Fehler- und Auslastungszähler für alle überwachten Subnetze
- history: nimmt periodisch Statistikmesspunkte aus der statistics-Gruppe auf
- alarm: legt die Warnlevel und Prüfintervalle für alle gemonitorten Werte fest
- host: enthält Zähler für diverse Verkehrsarten von und zu Knoten in Subnetzen
- hostTopN: bietet sortierte Listen der Knoten, die für festzulegende Parameter die N höchsten Werte liefern
- matrix: Fehler- und Auslastungsinformation in Matrixform: an x und y Achse sind Netzwerkadressen angetragen
- filter: Regeln, nach denen bestimmt wird, welche Pakete der Monitor in seine Untersuchungen aufnimmt
- capture: entscheidet, ob und wie lange Daten aus den Filterauswertungen gepuffert werden sollen
- event: führt Buch über alle vom RMON erzeugten Events
- tokenRing: Statistik und Konfigurationsinformationen für Token Ring Subnetze

Performance Indikatoren

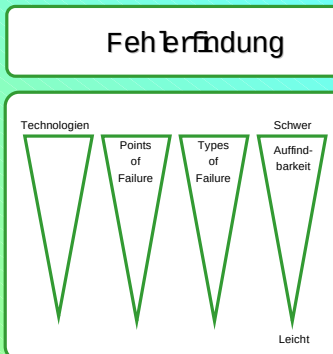
- Verfügbarkeit: Der Prozentsatz an Zeit, in der eine Komponente für den Benutzer verfügbar ist: A = MTBF/(MTBF + MTTR*)
- Antwortzeit: Die Zeitspanne, die ein Benutzer nach einer Aktion auf eine Reaktion warten muß
- Sorgfalt/Genauigkeit (Accuracy): Der Prozentsatz an Zeit, an dem kein Fehler bei der Informationsübermittlung eingetreten ist
- Durchsatz: die Rate, mit der anwendungsorientierte Ereignisse stattfinden (z.B. Signalisierungsnachrichten, Dateitransfers)
- Auslastung: der verwendete Prozentsatz der theoretisch zur Verfügung stehenden Kapazität einer Ressource

* Mean Time To Repair

FaultMonitoring

- Probleme:
  - Beobachtbarkeit
    - Meldung nicht mehr möglich
    - Zugang nach Meldung nicht möglich
    - Unsicherheiten durch
      - Inkonsistenz
      - zu viele Beteiligte
  - Fehlerpropagation

Fehlerfindung

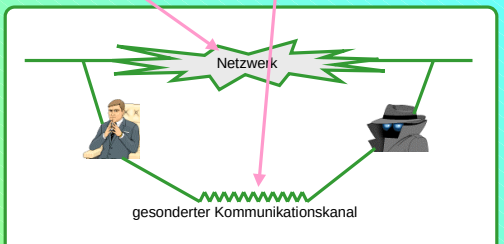


Tests:
- Erreichbarkeit
- Datenintegrität
- Protokollintegrität
- Datenkapazitäts-ausnutzung
- Verbindungs-kapazitätsausnutzung
- Antwortzeit
- Loopback
- Funktion
- Diagnose

primitive Internet-Managementwerkzeuge

- `ping` - sendet ICMP Echo Requests und wartet auf die Echo Replies
- CMU/UCB-SNMP Utilities - Protokollprimitiven aus der Kommandozeile ausführen
- `traceroute` - UDP-Pakete mit wachsender Time To Live versenden und ICMP Time Exceeded Meldungen auswerten (`tracert` unter Windows)
- `spray` (Last Generator)
- `netstat` - einfache Verbindungsstatistik
- `tcpdump` - komplette tcp-Pakete gefiltert ausgeben

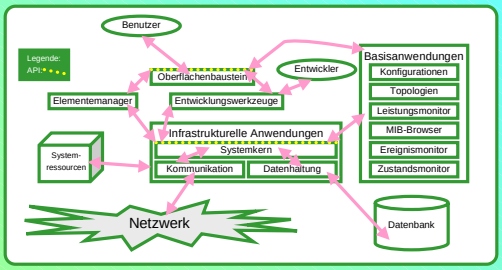
InBand vs. OutOfBand



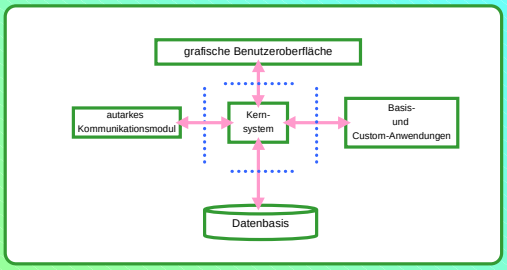
Managementplattformen

- integriertes Management heterogener, verteilter Systeme ist nur auf der Grundlage standardisierter Managementarchitekturen praktikabel
- Managementplattformen stellen eine Infrastruktur bereit, in die die verschiedenen Managementanwendungen eingebettet werden können
- Eigenschaften:
  - Plattformen laufen in offenen Systemumgebungen
  - sie bieten eine Laufzeitumgebung für Entwicklung und Betrieb verteilter Managementanwendungen
  - sie stellen Managementobjekte auf Basis gemeinsamer Informationsmodelle vereinheitlicht dar

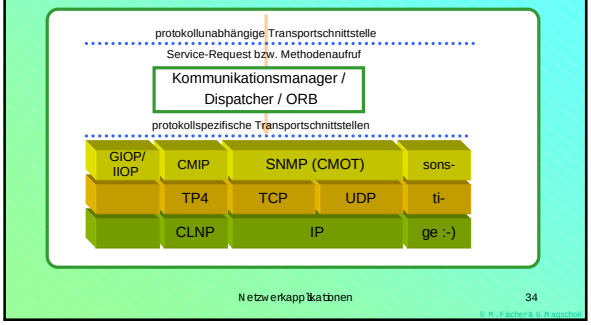
Architektur von Managementsystemen



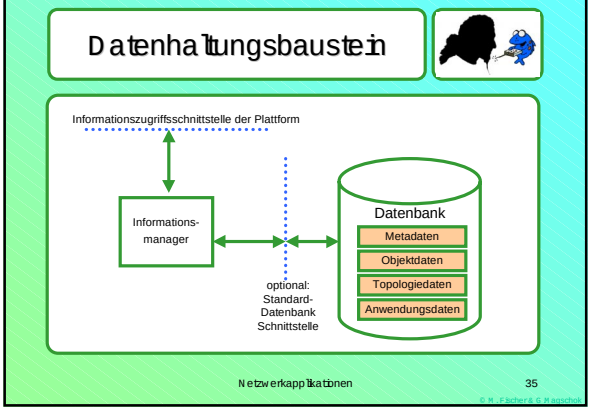
Verteilbarkeit von Managementapplikationen



Kommunikationsbaustein

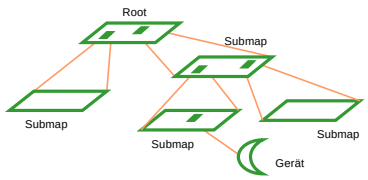


Datenhaltungsbaustein

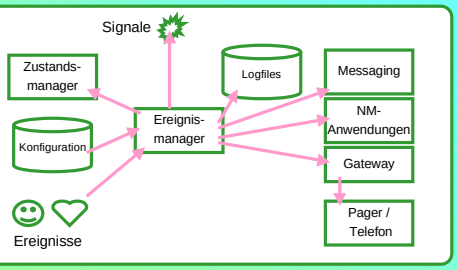


Oberflächenbaustein

hierarchische Maps:



Ereignismanagement



Beispiele

- Die HP OpenView Produktfamilie
  - früh abgespalten: IBM und SUN Netview
- Management im kleinen
  - SNMP Pc unter Windows
  - Ansätze unter Linux
    - Cheops
    - gxsnmp
    - tkined
- Elementemanager
  - meist proprietär von Herstellern von Netzwerkequipment, z.B. COSY-MAN
  - selten standardbasiert mit vermindertem Funktionsumfang z.B. HubView

HP OpenView

- Module:
  - NNM (Network Node Manager): Netzwerkmanagement
  - diverse Application, System, Desktop, Software, IT Service, Security und Storage Management Applikationen
- Datenhaltung in eigenem Format oder optional den wichtigsten bekannten Datenbanken (Oracle, Sybase, Informix)
- implementiert alle SNMP-Varianten außer SNMPv2u

DMTF Industriestandards

- die Desktop Management Task Force ist ein Zusammenschluß großer EDV-Unternehmen, die es sich zum Ziel gesetzt haben, Standards im Bereich System- und Enterprisemanagement zu schaffen
- bisherige Resultate / aktuelle Bemühungen:
  - Desktop und Enterprise Management
    - DMI - Desktop Management Interface für Computersysteme
    - CIM - Common Information Model, verteiltes System- und Netzwerkmanagementmodell
  - Vereinheitlichung von Managementinitiativen
    - WBEM - Web-Based Enterprise Management
    - DEN - Directory Enabled Networks

DMTF Ziele

- Integration aller Geräte und Anwendungen, um einst EIN Managementsystem zu erreichen
- größtmögliche Interoperabilität
- ISO-Standardisierung der Ergebnisse

DMTF: CIM

- Das Common Information Model ist ein implementierungsunabhängiges Schema zur Beschreibung von Managementinformationen
  - CIM soll managementsystemübergreifend beschreiben, Gemeinsamkeiten ausnutzen
  - CIM soll verschiedenste Quellen von Managementinformation integrieren
  - CIM ist bisher ein reines Datenmodell ohne Implementierung: das Managed Object Format (MOF)

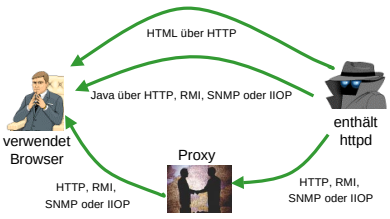
DMTF:WBEM

- Web Based Enterprise Management soll eine einheitliche HTML/XML Schnittstelle zu standardbasierten Netzwerkmanagementapplikationen bieten
- die Daten vorhandener MIBs u.ä. sollen hier integriert werden können
- die Transportmechanismen von Managementinformation ebenso
- sowohl direkt als Manager als auch über Management-Server einsetzbar
- nicht zu verwechseln mit dem Micro$oft WEBM (identische Akronym bedeutung)

WEB Based Management

- Motivation:
  - einfacher Weg zu plattformübergreifenden Produkten
  - mit zunehmender Verbreitung von Web-Techniken ein Weg zur Einheitlichkeit von Bedienoberflächen
  - Agents enthalten ihr Elementemanagement von Haus aus

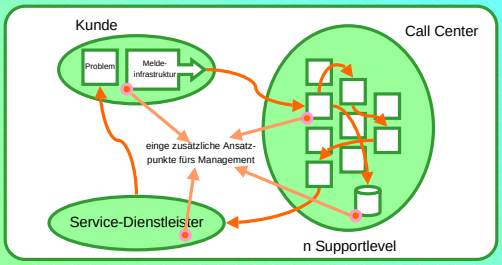
WEB Based Management Architekturen



Call-Center/Helpdesk-Management

- Beispiel für Enterprise Management
- Managed Objects sind Trouble Tickets
- Manager sind hierarchisch organisiert
- Workflow-basiert
- Technik:
  - traditionell: Telefonie
  - aktuell: Intra-/Internet
  - stark wissensdatenbankbasiert

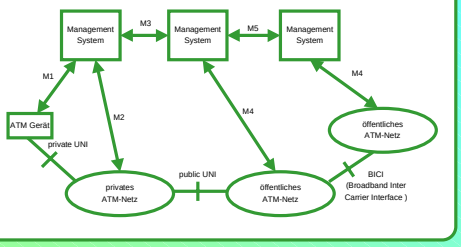
Beispiel für Call-Center



Management in ATM-Netzen

- Protokollunabhängige MIB-Definitionen des ATM-Forums
- Definition von Interfaces zu allen beteiligten Elementen des ATM-Referenzmodells
- Mitglieder des ATM-Forums können die gemeinsam erarbeiteten Standards implementieren

ATM Managementmodell



ATM Managementtypen

- M1: Management eines Access-Geräts / Terminaldevices
- M2: Management eines privaten ATM-Netzes
- M3: Management eines Kunden-Teilnetzes eines öffentlichen ATM-Netzes
- M4: Management eines öffentlichen ATM-Netzes
- M5: Management-Interaktion zwischen Anbietern öffentlicher ATM-Netze

ein paar Quellen

- Stallings, William: "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Third Edition", Addison Wesley, 1999, ISBN 0-201-48534-6
- http://www.ovforum.org/ The OpenView Forum
- http://www.gxsnmp.org/ GNOME Netzwerkmanagement
- news://comp.protocols.snmp/ ...viel Rauschen
- http://www.openview.hp.com/ HP OpenView Product Homepage
- http://www.snmp.com/FAQs/ Die SNMP FAQ
- http://www.ewos.be/hm/sbase.htm OSI Management Overview
- Tannenbaum, Andrew S.: "Computernetzwerke", Prentice Hall 1997, ISBN 3-8272-9536-X
- Halsall, Fred: "Data Communications, Computer Networks and Open Systems", Addison Wesley 1996, ISBN 0-201-42293-X

Ende