D I P L O M A R B E I T

Service-Monitoring im Netzwerk der Alpine-Mayreder Bau GmbH

durchgeführt am
Studiengang für Telekommunikationstechnik und -systeme
der
FH-Salzburg Fachhochschulgesellschaft mbH

vorgelegt von

Alexander Tabakoff

Leiter des Studiengangs:	Prof. Dipl.-Ing. Dr. Gerhard Jöchtl
Betreuer:	Dipl.-Ing. (FH) Stefan Deutinger

Salzburg, Mai 2006

Eidesstattliche Erklärung

Hiermit versichere ich, Alexander Tabakoff, geboren am 12.02.1983 in Salzburg, dass die vorliegende Diplomarbeit von mir selbstständig verfasst wurde. Zur Erstellung wurden von mir keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Puch, am 13.05.2006

_____ _____

Alexander Tabakoff Matrikelnummer

Informationen

Vor- und Zuname:	Alexander Tabakoff
Institution:	FH-Salzburg Fachhochschulgesellschaft mbH
Studiengang:	Telekommunikationstechnik und -systeme
Titel der Diplomarbeit:	Service-Monitoring im Netzwerk der Alpine-Mayreder Bau GmbH
Betreuer an der FH:	Dipl.-Ing. (FH) Stefan Deutinger

Schlagwörter

1. Schlagwort: Netzwerkmonitoring
2. Schlagwort: Nagios
3. Schlagwort: Erreichbarkeitsstufen
4. Schlagwort: Simple Network Management Protocol

Abstract

This diploma thesis deals with the introduction of a network monitoring system for the company Alpine-Mayreder Bau GmbH. After discussing the fundamentals of network management and network monitoring the thesis addresses the specific requirements of the system regarding the adoption to the individual needs of the aforementioned company. Different possible solutions are listed and compared, out of which the open source program Nagios emerges as the best solution for the specific demands of the company. The thesis proceeds to describe the installation, configuration and the adoption of the software to the demands defined before and finally concludes with a summary and future prospects in the field of network monitoring.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Dieses Kapitel begründet im ersten Teilkapitel die Themenwahl und gibt im Anschluss daran Aufschluss über die verwendeten Symbole und Konventionen und klärt im letzten Teilkapitel über die Anforderungen an den Leser sowie die Abgrenzung gegenüber anderen Bereichen auf.

1.1 Themenwahl

Die Alpine-Mayreder Bau GmbH ist ein multinationaler Konzern der Bau-, Bauträger-, Projekt- und Produktionsgesellschaften umfasst. Die IT Abteilung ist verantwortlich für eine schnelle, effiziente, sichere und kostengünstige Kommunikation zwischen den Mitarbeitern, Kunden und Lieferanten; des Weiteren müssen diverse Informationen abspeicherbar, gesichert und jederzeit aufrufbar sein. Um diese Anforderungen zu erfüllen werden verschiedenste Prozesse, Dienste und Systeme eingesetzt, die das Ziel haben die eben genannten Ansprüche zu befriedigen.

Neben den zahlreichen Abläufen wie der Konfiguration oder der Kapazitätsplanung spielen die Bereiche der Störungs- und Problembehandlung eine zentrale Rolle. Gerade bei den letztgenannten Abläufen ist es notwendig, schnelle und übersichtliche Berichte über den aktuellen Status aller eingesetzten Systeme und Dienste zu erhalten, um aufbauend auf diesen Informationen die richtigen Entscheidungen zu treffen und infolge dessen das Problem beseitigen zu können. Ein so genanntes Netzwerkmonitoring System, welches einzelne Dienste und Systeme überwachen kann, bietet folgende Vorteile für ein Unternehmen das diese Art von Software einsetzt:

- Entlastung der Administratoren
- Verringerung der Ausfallzeiten einzelner Systeme
- Steigerung der Produktivität der Mitarbeiter
- Erleichterung der Kapazitätsplanung
- Genauere Bedarfsanpassung der Ressourcen
- Erkennen struktureller Probleme durch Erhöhung der Übersicht

Dem gegenüber steht vor allem die Frage nach den Implementations- und Wartungskosten eines solchen Systems die mit den oben genannten Vorteilen abgewogen werden müssen um daraus eine Aussage über die Wirtschaftlichkeit des Systems treffen zu können. Trotz der zu erwartenden Kosten hat sich die IT Abteilung der Firma Alpine-Mayreder Bau GmbH dazu entschlossen, ein solches Netzwerkmonitoring System anhand eines projektorientierten Berufspraktikums einzuführen. Die Implementation wird in der vorliegenden Diplomarbeit Schritt für Schritt diskutiert und analysiert.

1.2 Konventionen

In dieser Diplomarbeit werden zahlreiche Konventionen angewendet, die es erleichtern die Arbeit übersichtlich und verständlich zu halten. Nachfolgend werden die wichtigsten Regeln erläutert.

Eine von eckigen Klammern umschlossene Zahl, also zum Beispiel [12] gibt an, das es sich um einen Verweis auf weiterführende Quellen oder eine Literaturangabe handelt, welche in Kapitel Literaturverzeichnis nach Kapitel 6 aufgelistet ist. Im Anschluss an das Literaturverzeichnis ist ein Index zu finden, welcher die wichtigsten Begriffe und Abkürzungen mit einer Erläuterung im Text verknüpft. Wird die Schrift Courier New für einen Absatz verwendet, also zum Beispiel

```
ifconfig eth0 up
```

deutet dies auf ein Codesegment hin, also eine Eingabe in eine Kommandozeile oder den Ausschnitt aus einem Quellcode. Wird im Text ein Begriff kursiv geschrieben, beispielsweise *nagios.cfg* deutet dies auf eine Konfigurationsdatei oder eine Pfadangabe hin. Die Fettschrift wird verwendet um einzelne Begriffe besonders hervorzuheben, zum Beispiel den Begriff **Erreichbarkeitsstufen**.

1.3 Anforderungen und Abgrenzung

Auf viele der Grundlagen, die zum vollständigen Verständnis dieser Arbeit nötig sind, wird nicht genau eingegangen, um den theoretischen Teil der Arbeit nicht über Gebühr anwachsen zu lassen. Zu den von einer genauen Beleuchtung der Hintergründe ausgenommenen Bereiche zählen:

- Das OSI 7-Schichtenmodell
- Das TCP/IP Referenzmodell
- Details der Protokolle und Dienste SSH, DHCP, SMTP u.ä. die für das Verständnis der Arbeit nicht notwendig sind
- Kryptographische Verfahren wie DES und MD5
- Kenntnisse des Linux Betriebssystems

Zur Abgrenzung des Themenbereiches ist zu vermerken, das sich Netzwerkmonitoring als Teilbereich des Netzwerkmanagements versteht. Diese Arbeit befasst sich schwerpunktmäßig mit dem Netzwerkmonitoring, also mit der Erkennung und Anzeige von Fehlern. Die Lösung der angezeigten Probleme ist Aufgabe des Netzwerkmanagements bzw. der Administratoren. Ebenso wird auf das Management der IT Abteilung selbst und auf die entsprechenden Prozesse nicht weiter eingegangen.

2 Grundlagen

Dieses Kapitel vermittelt die wichtigsten Standards, Protokolle und Verfahren im Zusammenhang mit Netzwerkmonitoring und -management.

2.1 Netzwerkmanagement

Das nachfolgende Kapitel orientiert ich an [1]. Die Standardisierungsorganisationen ISO und ITU definieren in ihren Standards ISO 10164 bzw. ITU-T X.700 eine Managementarchitektur die sich auf die folgenden fünf unterschiedlichen Funktionsbereiche aufteilt:

- Fault Management
- Configuration Management
- Accounting Management
- Performance Management
- Security Management

Aufgrund der Initiallettern der Managementbereiche werden diese fünf Funktionsbereiche oft unter dem Begriff FCAPS zusammengefasst.

Der Bereich Fault Management beinhaltet die Fehlererkennung, -isolation, und -behebung und muss Funktionen zur

- Wartung und Überwachung von Fehlerprotokollen
- Annahme von und Reaktion auf Fehlererkennungsnachrichten
- Verfolgung und genaue Bestimmung von Fehlern
- Ausführen von Prüfsequenzen und
- Korrektur von Fehlern bieten.

Der Funktionsbereich des Configuration Management umfasst die Identifikation, Kontrolle, Datenbeschaffung und Datenbereitstellung zum Zweck der Vorbereitung, Initialisierung, des Starts, der unterbrechungsfreien Versorgung und Beendigung von miteinander verbundenen Diensten.

Das Konfigurationsmanagement umfasst Funktionen zum

- Setzen von Parametern welche den Routinebetrieb des Systems steuern
- Zuweisen von Namen zu Ressourcen und Ressourcengruppen
- Initialisieren und stilllegen von Ressourcen
- Erfassen von Informationen eines Systems
- Erhalten von Informationen über Änderungen an einem System sowie
- Ändern der Konfiguration eines Systems.

Das Accounting Management wird verwendet um Kosten für Ressourcen zu bestimmen und Preise entsprechend festzulegen, es bietet Funktionen um

- Anwender über benutzte Ressourcen oder verursachte Kosten zu informieren
- Limits festzulegen und Preislisten für verschiedene Ressourcen festzulegen sowie
- Durch Nutzung von Ressourcenkombinationen verursachte Kosten zusammenzufassen.

Performance Management ermöglicht es, die Effektivität der Kommunikation sowie das Verhalten von Ressourcen zu bewerten, dazu werden Funktionalitäten geboten um

- Statistische Daten zu sammeln
- Systemstatus-Protokolle zu warten und zu überwachen
- Die Systemleistung unter natürlichen und künstlichen Bedingungen zu testen und
- Den Betriebszustand eines Systems zur Abwicklung von Leistungsmanagementaufgaben zu verändern.

Der Bereich des Security Management umfasst die Unterstützung zur Anwendung von Sicherheitsrichtlinien mit Funktionen um

- Sicherheitsdienste und -mechanismen zu erstellen, kontrollieren und zu bearbeiten,
- Sicherheitsrelevante Informationen zu verteilen und
- Um über sicherheitsrelevante Ereignisse zu berichten.

FCAPS konzentriert sich, wie aus den soeben genannten Funktionsbereichen hervorgeht, stark auf das Management, das Beobachten von einzelnen Netzwerkkomponenten und Diensten ist jedoch integraler Bestandteil jedes Funktionsbereiches.

2.2 Netzwerkmonitoring

Das Netzwerkmonitoring ist Bestandteil des Netzwerkmanagements und verantwortlich dafür, den aktuellen Status und das Verhalten sämtlicher relevanten Netzwerkkomponenten und deren Dienste anzuzeigen sowie entsprechend zu analysieren.

In einem Netzwerkmonitoring System nehmen verschiedene Objekte unterschiedliche Funktionen wahr, diese sind wie nachfolgend gelistet eingeteilt:

- **Monitoring Programm**
  Das Monitoring Programm ist für die Speicherung und Darstellung der Beobachtungsergebnisse verantwortlich
- **Manager**
  Der Manager startet einzelne Abfragen sammelt Informationen
- **Netzwerkelement**
  Ein Netzwerkelement bietet eine gewisse Funktion oder einen Dienst an, der überwacht wird
- **Agent**
  Ein Agent ist eine Programmroutine die auf dem Netzwerkelement läuft und die Überwachungsfunktion realisiert [2]

In einer typischen Netzwerkmonitoring Konfiguration ist also auf jedem Netzwerkelement ein Agent vorhanden, der Informationen an einen Manager weiterleitet, welcher die Informationen aufsammelt und an das Netzwerkmonitoring Programm weiterleitet. Die unterschiedlichen Informationen, die von einer Netzwerkmonitoring-Applikation verwaltet werden lassen sich in drei Kategorien einteilen:

- **Statische Daten**
  Statistische Daten sind typischerweise Informationen über den Namen oder den Standort eines Systems, also Daten die sich vergleichsweise selten ändern.

- **Dynamische Daten**
  Ereignisse im Netzwerk oder Statusabfragen eines Servers werden unter dem Begriff dynamische Daten zusammengefasst, da sich diese Informationen in der Regel häufig ändern.
- **Statistische Daten**
  Aus dynamischen Daten können statistische Daten gewonnen werden, die Auskünfte über längerfristige Trends geben.

Grundsätzlich gibt es zwei Möglichkeiten, dynamische Daten von einem Netzwerkelement auf die Managementstation zu übertragen. Die erste wird als Polling bezeichnet und vom Manager der Beobachtungsstation initialisiert; der Manager wartet bei dieser Variante nach Absenden einer Anfrage auf eine entsprechende Antwort des Agents. Bei der zweiten Variante sendet der Agent von sich aus Informationen an den Manager, man spricht dann von Event Reporting oder im Zusammenhang mit SNMP[1] von Traps. [2],[3]

Für die Durchführung von Netzwerkmanagement gibt es zahlreiche Standards und Protokolle, nachfolgend werden einige davon kurz erläutert.

2.2.1 IPMI

Die Abkürzung IPMI steht für Intelligent Platform Management Interface. Der IPMI Standard wurde von den Firmen Dell, Intel, HP und NEC gemeinsam entwickelt und ist mittlerweile in der Version 2.0 verfügbar. Die entscheidende Fähigkeit von IPMI besteht darin, Invarisierungs-, Monitoring-, Protokollierungs- und Wiederherstellungsfunktionen unabhängig von Hauptprozessor, BIOS oder des laufenden Betriebssystems zu bieten. Darüber hinaus ist es möglich, gewisse IPMI Funktionen auch bei einem ausgeschalteten System anzusprechen. Um diese Funktionen zu realisieren ist ein eigener Mikrocontroller notwendig, der Baseboard Management Controller oder kurz BMC genannt wird. Der BMC kann über einen eigenen Bus verschiedene so genannte Sattelitencontroller ansprechen, die Monitoringfunktionen für zusätzliche Steckkarten - wie zum Beispiel einem RAID-Controller - anbieten können. Zur Kommunikation mit einer Managementstation oder einem anderen IPMI fähigen Server sind mehrere verschiedene Schnittstellen definiert. Der Datenaustausch kann über eine serielle Verbindung, eine Modemleitung, über ein lokales Netzwerk oder über einen seriellen Tunnel via LAN erfolgen.

1 SNMP wird in Kapitel 2.3 näher erläutert

Seit der Version 2.0 ist es möglich, sämtliche Kommunikation authentifiziert und verschlüsselt durchzuführen. [4]

Der IPMI Standard versteht sich nicht als Ersatz für Managementprotokolle höherer Ebene sondern erweitert die Monitoringfähigkeiten auf hardwarenahe Abfragen wie Lüfterdrehzahlen, Temperaturen und Netzteil-Spannungen und standardisiert diese um die Interoperabilität zwischen Systemen unterschiedlicher Hersteller zu erleichtern.

2.2.2 WBEM

Als Web Based Enterprise Management, kurz WBEM, wird eine Reihe von Standards bezeichnet, die eine plattformübergreifende Basis zum Management von verteilten Rechnernetzen zur Verfügung stellen. Die von WBEM definierten Standards beinhalten unter anderem folgende Definitionen:

- **CIM**
  Das Common Information Model CIM beschreibt das Datenmodell von WBEM und definiert Managementinformationen für Systeme, Netzwerke, Applikationen, Dienste und herstellerspezifische Erweiterungen.
- **CIM-XML**
  CIM-XML ist ein Kommunikationsprotokoll das CIM konforme Informationen mithilfe einer XML DTD über das HTTP Protokoll überträgt.
- **WBEM Discovery**
  Der Dienst WBEM Discovery verwendet das Service Location Protocol, welches anderen Anwendungen ermöglicht WBEM Systeme zu identifizieren und mit ihnen zu kommunizieren.
- **CIM Query Language**
  Die CIM Query Language definiert eine Abfragesprache um aus CIM-basierenden Managementsystemen Daten exportieren zu können. [5]

WBEM Lösungen werden unter anderem von Sun Microsystems und HP angeboten. Die bekannteste Lösung ist die Microsoft Implementation Windows Management Instrumentation oder kurz WMI, die ebenfalls auf WBEM aufsetzt und es ermöglicht fast jede Einstellung in einem Windows basierenden Betriebssystem mit WMI Unterstützung zu verändern.

2.2.3 Syslog

Das Syslog Protokoll wurde ursprünglich von der University of California entwickelt und ist Teil des BSD Betriebssystems, mittlerweile ist Syslog auf zahlreiche andere Betriebssysteme portiert worden. Syslog definiert drei unterschiedliche Geräteklassen. Ein **device** ist ein Gerät das Syslog Nachrichten erzeugt und versendet. Der **relay** hat sowohl Sender- als auch Empfängerfunktion, er leitet eine von einem device oder anderem relay stammende Nachricht weiter. Der **collector** kann auch als syslog Server bezeichnet werden, er empfängt Syslog Nachrichten von devices oder relays und speichert sie ab.

Eine Syslog Nachricht besteht aus drei Teilen, die sich PRI, HEADER und MSG nennen. Diese Nachricht wird mit UDP übertragen und darf eine maximale Länge von 1024 Bytes besitzen. Im PRI Teil wird die Dringlichkeit der Nachricht anhand deren Ursprung und Schwere berechnet. Der HEADER beinhaltet Informationen über die Herkunft der Meldung mittels IP Adresse oder Hostname sowie einen Zeitstempel. Im MSG Teil einer syslog Nachricht werden Informationen über den Ursprung der Nachricht und die Fehlermeldung selbst übertragen. [6]

Der Syslog Dienst ist sehr einfach aufgebaut und bietet keinerlei Möglichkeiten zur Verschlüsselung oder Authentifizierung. Syslog ist als Netzwerkmonitoring Programm zu verstehen und spielt seine Vorteile in der schnellen und einfachen Implementierung sowie der Verfügbarkeit auf verschiedensten Plattformen aus.

2.3 Netzwerkmanagement mit der SNMP Protokollfamilie

In den nachfolgenden Teilkapitel wird auf das Simple Network Management Protocol oder kurz SNMP eingegangen, wobei zuerst mit BER, ASN.1, SMI und MIB alle notwendigen Voraussetzungen erläutert werden.

2.3.1 BER

Das Akronym BER steht für Basic Encoding Rules. Die BER sind von der ISO bzw. der ITU-T entwickelt worden und als Standards unter der Bezeichnung ISO 8825 und X.690 nachzulesen, sie werden unter anderem verwendet um bei der Übertragung zwischen verschiedenen Systemen keine Missverständnisse bezüglich der Byte-Folge (little-endian, big-endian) aufkommen zu lassen.

Jeder mit den BER definierte Wert wird wie folgt codiert: Die ersten 8 Bit werden als Identifier bezeichnet, von diesen 8 Bit sind die ersten zwei Bits für die Benennung der Klasse vorgesehen, das 3. Bit bestimmt ob es sich um einen einfachen oder zusammengesetzten Code handelt und die drauf folgenden 5 Bits geben mit der Tag Number an, um welchen Datentyp es sich handelt. Die zweite Gruppe von 8 aufeinander folgenden Bits wird als Längenfeld verwendet, wobei es grundsätzlich drei Möglichkeiten gibt, die Länge anzugeben:

- **9. Bit ist 0**
  Das erste Bit der zweiten Gruppe (also das 9. Bit insgesamt) ist 0. Dies bedeutet, das die nachfolgenden Daten maximal 128 Bytes lang sind und die tatsächliche Länge der Daten in den verbleibenden 7 Bits enthalten ist.
- **9. Bit ist 1, danach 7 Bits größer 0 dezimal**
  Das erste Bit der zweiten Gruppe ist 1, in den darauf folgenden 7 Bits befindet sich eine Zahl größer 0. Dies deutet darauf hin, dass das Datenfeld größer als 128 Bytes lang ist. In den 7 verbleibenden Bits wird die Größe des nachgestellten Längenfeldes in Bytes angegeben.
- **9. Bit ist 1, danach 7 Bits gleich 0 dezimal**
  Das erste Bit der zweiten Gruppe ist 1, die darauf folgenden 7 Bits sind alle 0. Dies lässt auf eine nicht weiter definierte Länge des Datenfeldes schließen, welche durch ein End of Content oder kurz EOC Feld mit 16 Nullstellen beendet wird.[2],[7]

Die Abbildung 2.1 verdeutlicht nochmals die drei BER Typen.

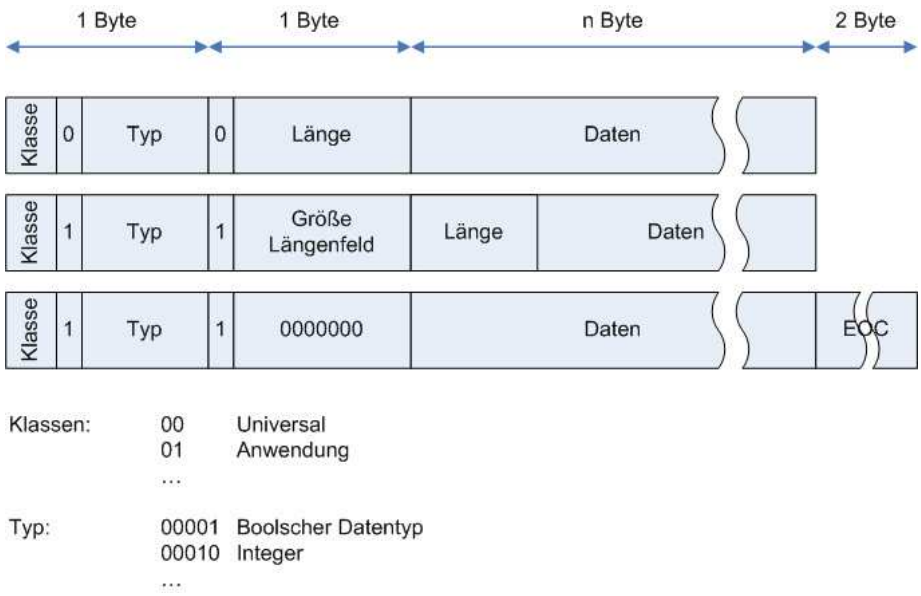

Abbildung 2.1: Varianten der Basic Encoding Rules

2.3.2 ASN.1

ASN.1 steht für Abstract Syntax Notation One und stellt eine Sprache dar, mit deren Hilfe abstrakte Syntax formuliert werden kann; sie wird von der ITU-T unter der Bezeichnung X.680 und von der ISO unter der Bezeichnung ISO 8824 standardisiert. Vom SNMP Protokoll sind nicht alle von ASN.1 definierten Datentypen zulässig, weshalb die Tabelle 2.1 nur die für SNMP erlaubten Typen auflistet.

Datentyp	Bedeutung
INTEGER	Ganzzahl
BIT STRING	Bitsequenz
OCTET STRING	Oktettsequenz
NULL	Null, ein Platzhalter
OBJECT IDENTIFIER	Objektbezeichner

Tabelle 2.1: ASN.1 Datentypen in SNMP [8]

Um eine Variable mit der Bezeichnung „test“ zu definieren und mit dem Initialwert 1 zu versehen ist folgende Befehlszeile notwendig:

```
test INTEGER ::= 1
```

Ein weiterer Datentyp ist der „OBJECT IDENTIFIER“ welcher oft auch mit seiner Abkürzung OID verwendet wird - er ermöglicht es einzelne Objekte nach einem gewissen Schema eindeutig zu identifizieren. Dazu wird jedes Objekt in eine baumartige Struktur eingeordnet, jede Verzweigung des Baumes ist mit einer Nummer und einer Bezeichnung versehen und kann somit eindeutig zugeordnet werden.[2],[8]

2.3.3 SMI

Der Begriff SMI ist eine Abkürzung, bedeutet Structure of Management Information und wird im RFC 1442 definiert. Die SMI ist eine speziell für die Anforderungen von SNMP angepasste Teilmenge von ASN.1, sie wird verwendet um Datenstrukturen zu definieren. SNMP Variablen werden von der SMI als einzelne Objekte definiert, Objekte werden zu Objektgruppen und diese wiederum zu Modulen zusammengefasst. Wird ein Modul aufgerufen, werden in der Regel die Makros MODULE-IDENTITY, OBJECT-IDENTITY und OBJECT-TYPE des betreffenden Moduls der Reihe nach aufgerufen. Das Makro MODULE-IDENTITY liefert Informationen über Revision, Autor und ähnliches, während das OBJECT-IDENTITY Makro definiert, an welcher Stelle des Objektbaumes sich das Modul befindet.

OBJECT-TYPE wiederum hat mehrere Parameter, die vier minimal anzugebenden sind:

- SYNTAX
- MAX-ACCESS
- STATUS
- DESCRIPTION [9]

Der Parameter SYNTAX beschreibt den Datentyp. Eine Liste der für den Parameter zugelassenen Typen mit kurzen Erläuterungen ist in Tabelle 2.2 zu finden.

Name	Bedeutung
INTEGER32 und INTEGER	Ganzzahl zwischen -2147483648 und 2147483647, INTEGER und INTEGER32 haben identische Wertbereiche.
OCTET STRING	Binär- oder Textdaten, maximal 255 Oktette.
OBJECT IDENTIFIER	OID String, maximal 128 Subeinträge.
BIT STRING	„enumeration of named bits“, nicht negativ
IpAddress	32 Bit IP v4 Adresse
Counter32	Zähler bis 4294967295, startet nach überschreiten der Grenze wieder bei 0.
Gauge32	Ganzzahl die einen maximalen Wert von 4294967295, bei überschreiten des Werts wird diese nicht auf 0 gesetzt.
TimeTicks	Zeit in hundertstel Sekunden
Opaque	Kompatibilitätsdatentyp
NsapAddress	OSI NSAP Adresse
Counter64	Zähler bis 18446744073709551615, startet nach überschreiten der Grenze wieder bei 0.
Uinteger32	Ganzzahl zwischen 0 und 4294967295

Tabelle 2.2: Mögliche Datentypen für SNMP Variablen [8]

Der nächste Pflichtparameter ist MAX-ACCESS, er definiert die Zugriffsrechte auf die Variable; typisch sind die Werte Lesen-Schreiben und Nur-Lesen. Der Parameter STATUS gibt Auskunft über die Kompatibilität mit dem aktuellen SNMP Standard, möglich sind die Werte gültig, ungültig und überholt. Der letzte vorgeschriebene Parameter ist DESCRIPTION, er definiert eine Zeichenkette, welche die Bedeutung der Variable veranschaulicht und zu Verständniszwecken vorhanden ist. [9]

2.3.4 MIB

Die Management Information Base, kurz auch MIB genannt, greift die Idee der eindeutig identifizierbaren Objekte aus ASN.1 auf und ordnet die Management Informationen in einen Teilbaum des bereits vorhandenen ISO Objektbaumes ein. Die so genannte MIB-I definiert eine Reihe von hierarchisch angeordneten Objekten die vor allem für allgemeine Informationen dienlich sind, ein Beispiel dafür ist das Objekt mit der OID 1.3.6.1.2.1.1.5, welches den Namen des Systems repräsentiert. Aktuell ist die MIB-II, welche ihre Vorgängerversion de facto abgelöst hat; neben dieser gibt es noch zahlreiche MIBs für herstellerspezifische Aufgaben oder aber MIBs die sich mit speziellen Teilbereichen des Netzwerkmanagements befassen, so zum Beispiel die Management Information Base mit dem Titel „Mail Monitoring MIB“, welche in [10] definiert ist. Am Beispiel der Abbildung 2.2 wird der Objektbaum anhand der bereits oben angeführten OID 1.3.6.1.2.1.1.5 ausschnittsweise abgebildet.[11]

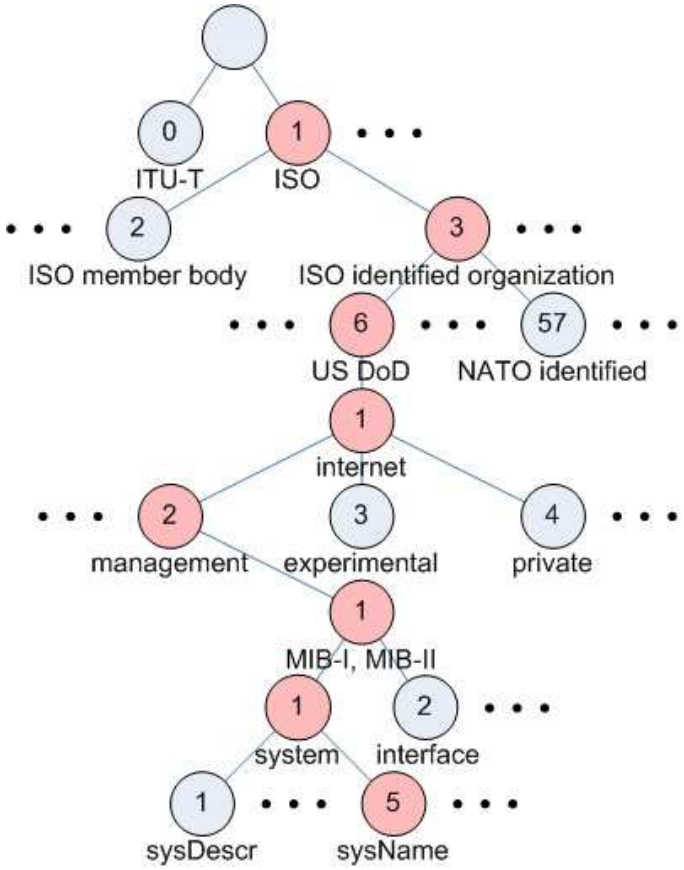

Abbildung 2.2: OID 1.3.6.1.2.1.1.5 im Objektbaum [8]

Unter dem Zweig 1.3.6.1.4 ist es für private Personen und Unternehmen möglich eine eigene sogenannte Private Enterprise Number anzufordern, um selbst definierte Objekte konform zum Standard einzubinden.

2.3.5 SNMP

Als Simple Network Management Protocol oder kurz SNMP wird jenes in der Regel UDP basierende Protokoll bezeichnet, das für den Transfer der Management Daten verantwortlich ist; diese Informationen werden zwischen einer Netzwerkkomponente, dem so genannten Network Elements und der Managementstation, der Network Management Station ausgetauscht. Network Elements verfügen über Management Agents, welche die Netzwerkmanagement Funktionen realisieren. SNMP ist in den Versionen v1, v2 und v3 implementiert, wobei nur die Versionen v1 und v3 standardisiert sind; die Version v2 hingegen wurde nie in einem RFC definiert - jedoch hat sich die Subversion v2c weitgehend gegenüber den anderen Varianten durchgesetzt. Während sich die ersten beiden Versionen von SNMP nicht stark unterscheiden, wurde die dritte Version massiv um Sicherheitsmechanismen erweitert welche später genauer erläutert werden. Zur Kommunikation zwischen den Network Elements und den Network Management Stations werden sieben unterschiedliche Nachrichten definiert, welche in der Tabelle 2.3 aufgelistet und kurz erläutert werden.[7], [12]

Nachricht	Bedeutung
GetRequest	Hole den Wert eines oder mehrerer MIB Objekte
GetNextRequest	Hole den nächsten Wert eines MIB Objektes
GetBulkRequest	Hole große Datenmengen auf einmal
InformRequest	Beschreibung der lokalen MIB
SetRequest	Setze den Wert eines MIB Objektes
Response	Antwort auf die fünf vorhergehenden Anfragen
Trap	Information des Managers ohne vorherige Anforderung.

Tabelle 2.3: SNMP (v2) Nachrichtentypen [2]

Die in Tabelle 2.3 erwähnte Trap Nachricht nimmt eine Sonderstellung ein, da sie vom Agent des Network Elements gesendet wird ohne dass die Network Management Station diesen Wert angefragt hat. Diese Option ist speziell für Nachrichten hoher Priorität gedacht die sofort an die Management Station gemeldet werden müssen; die Trap Funktionalität kann somit die MTFD[2] unter Umständen deutlich verkürzen.

SNMP bietet in der Version v3 erstmals Sicherheitsmechanismen an, die im so genannten User-Based Security Model oder kurz USM beschrieben werden und vor den nachfolgenden Bedrohungen schützen:

2 Die MTFD und andere Kennzahlen werden in Kapitel 2.4 beleuchtet

- Modifikation von Management Informationen
- Erschleichen von Berechtigungen

Des Weiteren bietet das USM begrenzten Schutz gegen die Bedrohungen durch:

- Entdeckung des Nachrichteninhalts durch Mithören
- Modifikationen des Informationsflusses

Der Schutz gegenüber Denial-of-Service und Netzwerkverkehrsanalysen ist durch das USM nicht vorgesehen.

Um den vorgegebenen Bedrohungsszenarien zu entgegnen, definiert das USM die drei Module timeliness, authentication und privacy. Das erste Modul überprüft die Nachricht auf Aktualität, somit kann verhindert werden, dass ein Angreifer eine Nachricht verzögert oder öfters als vorgesehen versendet. Das zweite Modul bietet mit den Authentifizierungsalgorithmen SHA und MD5 die Möglichkeit, die Integrität einer mit SNMP gesendeten Nachricht zu überprüfen. Das letzte Modul definiert den Algorithmus DES zur sicheren Verschlüsselung aller transferierten Daten.[13]

2.3.6 RMON

Die Abkürzung RMON steht für Remote network MONitoring und ist eine Erweiterung des bestehenden SNMP Standards um eine weitere MIB, die speziell für das Beobachten entfernter Netze ausgelegt ist. Obwohl RMON „nur“ eine neue MIB definiert, erweitert es die SNMP Funktionalitäten hinsichtlich statistischer Auswertungen auf Layer 2 (MAC-Ebene) beträchtlich. [2]

Folgende Ziele sind durch die RMON MIB Definition zu erreichen:

- Offline Betrieb
- Proaktives Monitoring
- Problemerkennung und Berichterstattung
- Daten-Mehrwert
- Unterstützung mehrerer Managementstationen

Der Offline Betrieb ermöglicht es entfernten RMON Stationen, auch RMON Probes genannt, kontinuierlich Daten zu sammeln ohne in ständiger Verbindung mit einer oder mehreren zentralen Managementstationen zu stehen. Proaktives Monitoring der Netzwerkperformance kann so helfen Fehler und Probleme frühzeitig zu erkennen; tritt ein Problem auf ist es Aufgabe der Problemerkennung und Berichterstattung diese zu erkennen und entsprechend weiterzuleiten. Da eine RMON Probe speziell zum Beobachten eines Subnetzes ausgelegt ist, ergibt sich dadurch die Möglichkeit, den Daten bedeutenden Mehrwert beizusteuern. Darüber hinaus bietet RMON die Möglichkeit, mehrere zentrale Managementstationen zu implementieren, um etwa Fehlerredundanz zu erreichen oder die Performance zu erhöhen. [14]

Um alle diese Ziele zu erfüllen definiert die RMON MIB zehn Untergruppen, welche ausschließlich für Ethernet Netzwerke geeignet sind; diese sind in der Tabelle 2.4 aufgelistet und kurz erläutert.

RMON Gruppe	Bedeutung
Ethernet Statistics	Statistische Auswertungen jedes beobachteten Interfaces
History Control	Beobachtet die periodische statistische Abfrage von Daten
Ethernet History	Speichert statistische Daten des Ethernets
Alarm	Nimmt Werte aus den Daten, vergleicht sie mit vorher konfigurierten Grenzwerten und löst wenn nötig Alarm aus
Host	Statistiken von jedem in Subnetz erkannten Host
HostTopN	Vorbereiten von Berichten über einzelnen Hosts geordnet nach einer statistischen Auswertung.
Matrix	Statistiken über Datentransfers zwischen zwei Adressen.
Filter	Definiert einen Filter für gewisse Pakete
Packet Capture	Akquirieren eines Paketstromes
Event	Kontrolliert die Erstellung von Mitteilungen

Tabelle 2.4: RMON Gruppen und Bedeutung [15]

Wie bereits erwähnt beschränkt sich das Monitoring mit RMON auf den Layer 2 des OSI Modells, eine Erweiterung des bestehenden Standards zur Beobachtung und Auswertung statistischer Daten bis auf Applikationsebene (Layer 7) ist durch RMON2 realisiert. RMON2 erweitert die durch RMON definierten Gruppen um weitere neun auf insgesamt 19 Untergruppen. [15]

Eine nochmalige Erweiterung der Funktionalität bieten die Remote Network Monitoring MIB Extensions for Switched Networks, kurz SMON an. SMON ergänzt die bestehenden RMON Versionen um die Unterstützung von geswitchten Netzwerken.

2.4 Kennzahlen

Die primäre Aufgabe jedes Netzwerkmonitoring Systems ist die - womöglich frühzeitige - Erkennung von im Netzwerk auftretenden Störungen. Der „Lebenszyklus“ eines Fehlers kann in mehrere Stufen geteilt werden:

- Vorankündigung
- Auftritt des Fehlers
- Entdeckung
- Diagnose
- Reparatur

Die erste Stufe ist die **Vorankündigung** des Fehlers, der durch genaues beobachten einzelner Leistungsparameter zu einer gewissen Wahrscheinlichkeit vorhergesagt werden kann. Typisch für die Vorankündigung eines Fehlers ist zum Beispiel die Meldung, dass eine Festplatte zu 90% voll ist – in diesem Fall können bereits Gegenmaßnahmen ergriffen werden bevor der Fehler (Festplatte voll) überhaupt auftritt. Der nächste Schritt ist der **Auftritt des Fehlers** selbst, indem ein gewisser Netzwerkdienst oder ein System plötzlich und ungeplant nicht mehr verfügbar ist. Als nächstes wird der **Fehler entdeckt**, dies kann entweder durch einen Administrator, dem Monitoringsystem oder durch einen Nutzer des nicht mehr verfügbaren Dienstes geschehen. Daraufhin wird der Fehler analysiert und eine **Diagnose** durch den Administrator erstellt. Ist die Diagnose erstellt kann mit der **Reparatur** beziehungsweise der Behebung des Problems begonnen werden, nach Abschluss dieser ist das System wieder voll funktionstüchtig. In manchen Fällen ist es möglich, zwischen der Entdeckung und der Diagnose eine **Umgehungslösung** als Zwischenschritt einzuführen; dieser Schritt wird oft auch als workaround bezeichnet. Ein solcher Zwischenschritt ist vor allem dann sinnvoll, wenn entweder der Nutzer den ausgefallenen Dienst dringend benötigt oder wenn die Behebung des Fehlers voraussichtlich längere Zeit in Anspruch nehmen wird.

Im Zusammenhang mit dem Fehlermanagement gibt es einige Kenngrößen, die häufig Verwendung finden um gewisse Zeitabschnitte zwischen den einzelnen oben beschriebenen Stufen des Fehlermanagements zu beschreiben.

- **MTFR**
  Die Mean Time between Failure and Repair oder kurz MTFR bezeichnet die Zeit die durchschnittlich zwischen dem Auftreten des Fehlers bis zu Wiederherstellung des Normalzustandes verstreicht
- **MTFD**
  Mit der Abkürzung **MTFD** wird jene Zeitspanne beschrieben, die durchschnittlich zwischen dem Auftreten des Fehlers und seiner Entdeckung liegt, sie bedeutet Mean Time between Failure and Disclosure
- **MTDD**
  MTDD steht für Mean Time between Disclosure and Diagnose und misst die Durchschnittszeit zwischen der Entdeckung und der Diagnose eines Problems.
- **MTDR**
  Die Bezeichnung MTDR ist die Kurzform für Mean Time between Diagnose and Repair und misst die durchschnittliche Zeit zwischen der Diagnose und dem Abschluss des Reparaturvorganges.
- **MTBF**
  Die MTBF ist das am häufigsten verwendete Maß für Verfügbarkeit und beschreibt die durchschnittliche Zeit zwischen dem Auftreten zweier Fehler, die Abkürzung bedeutet Mean Time Between Failures

Die Abbildung 2.3 verdeutlicht den Zusammenhang zwischen den verschiedenen Kenngrößen und den Stufen zur Fehlerbehebung, manche Stufen können durch den Einsatz von Netzwerkmonitoring schneller abgehandelt werden und sind entsprechend markiert.

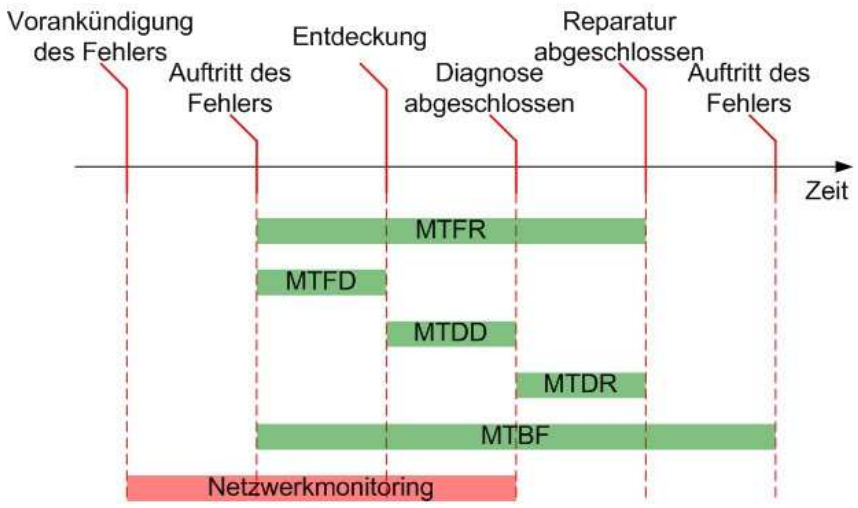

Abbildung 2.3: Fehlermanagement und dazugehörige Kenngrößen

Es kann passieren, dass das Netzwerkmonitoring Programm oder ein Benutzer einen Fehlalarm auslöst. Man spricht hierbei von so genannten false positive Ergebnissen oder von Fehlern 1.Art. Wird ein Fehler von einem Montitoringsystem nicht erkannt obwohl er bereits aufgetreten ist spricht man von einer false negative Entscheidung oder von einem Fehler 2.Art. Die nachfolgende Übersicht in Tabelle 2.5 verdeutlicht das Problem.

		Realität	
		Fehler	Normalzustand
Test	Fehler	OK	false positive
	Normalzustand	false negative	OK

Tabelle 2.5: Vergleich Testentscheidung und Realität [16]

Der Fehler 1. Art, also die Anzeige des Normalzustandes obwohl ein Fehler bereits aufgetreten ist stellt ein weitaus größeres Problem dar im Gegensatz zu einem Fehlalarm und muss daher so weit wie möglich eingedämmt werden. [16]

2.5 Verfügbarkeit

Die Verfügbarkeit von Rechnersystemen ist ein viel diskutierter Begriff, der aber oftmals sehr ungenau und verschwommen definiert ist, vor allem was seine Messbarkeit betrifft. Dieses Kapitel wird einige Ansätze zum Thema Verfügbarkeit vorstellen.

2.5.1 Verfügbarkeit nach ITIL

Die Abkürzung ITIL steht für Information Technology Infrastructure Library, der ITIL Ansatz hat sich als best practice Verfahren der britischen Regierungsbehörde OGC in den letzten Jahren zum quasi-Standard für IT Servicemanagement entwickelt. [17]

ITIL definiert insgesamt 11 Module, von denen der Prozess des Availability Management hier gesondert beleuchtet werden soll.

Als zentrale Aufgabe des Availability Managements ist die Festlegung, Planung, Überwachung und Dokumentation aller IT-Systeme sowie deren Abgleich mit Service Level Agreements (SLAs) zu verstehen. Diese Aufgabe wird von einem Availability Manager durchgeführt. Die Verfügbarkeit wird mit folgender Formel beschrieben:

$$Verfügbarkeit = \frac{(VereinbarteServiceZeit - Ausfallzeit)}{VereinbarteServiceZeit} * 100 \qquad (2.1)$$

Die Werte beziehen sich immer auf einen vorher festzulegenden Betrachtungszeitraum. Die Verfügbarkeit eines Dienstes wird zusammen mit anderen Konditionen in einem SLA festgelegt, dieser Vertrag wird zwischen dem Service Level Manager und dem Kunden (also dem Konsumenten der IT-Leistungen) abgeschlossen. Wie die Verfügbarkeit in einem SLA definiert ist, steht jedoch dem Service Level Manager frei; auch auf die Messung wird nicht eingegangen. [18]

2.5.2 Verfügbarkeit nach österreichischem Sicherheitshandbuch

Der „Bereich IKT-Strategie des Bundes“ des Bundeskanzleramtes veröffentlicht unter [19] ein österreichisches IT-Sicherheitshandbuch, aus dem die Nachfolgenden Informationen übernommen wurden. Das Sicherheitshandbuch wird für die Bundesverwaltung erstellt und gilt dort als verbindliche Grundlage für die Umsetzung des IT Sicherheitsmanagements; das Sicherheitshandbuch wird auch für den Einsatz in der Privatwirtschaft empfohlen. Das Sicherheitshandbuch beschreibt die Business Continuity Planung, deren Ziel es ist, die Verfügbarkeit der wichtigsten Applikationen und Systeme innerhalb eines bestimmten Zeitraumes sicherzustellen. Weiters wird zwischen den Teilbereichen Business Contingency Planung und K-Planung unterschieden. Die **K-Planung** definiert einen Notbetrieb des betreffenden Systems im Katastrophenfall, die **Business Contingency Planung** kümmert sich um die Aufrechterhaltung der Betriebsverfügbarkeit nach dem Auftreten eines Fehlers. Das Sicherheitshandbuch schreibt vor, Anwendungen und Systeme in Verfügbarkeitsklassen einzuteilen, exemplarisch wird das vom Bundeskanzleramt verwendete Schema beschrieben, welches sich in 4 unterschiedliche Klassen teilt.

- **Betriebsverfügbarkeitskategorie 1 – Keine Vorsorge:**
  Ein Datenverlust oder Ausfall ist möglich und unkritisch.
- **Betriebsverfügbarkeitskategorie 2 – Offline Sicherung:**
  Daten werden gesichert und an externem Ort gelagert, für die Dauer der Problembehebung ist das System nicht verfügbar.
- **Betriebsverfügbarkeitskategorie 3 – Redundante Infrastruktur:**
  Durch redundante Auslegung ist ein unterbrechungsfreier Betrieb gewährleistet

- **Betriebsverfügbarkeitsklasse 4 – Redundante Standorte:**
  Zwei verschiedene Standorte besitzen je eine vollständige IT-Struktur, um die Dienste auch beim völligen Ausfall eines der Standorte weiterhin uneingeschränkt anbieten zu können.

Der Zusatz „K-Fall sicher“ bei den Kategorien 2-4 berücksichtigt auch die Anforderungen in Kastastrophenfällen und erlaubt den Transfer aller relevanten Daten in eine so genannte Zero-Risk Umgebung, wo ein Notbetrieb weitergeführt werden kann.

2.5.3 Verfügbarkeit nach IT-Grundschutzhandbuch

Das deutsche Bundesamt für Sicherheit in der Informationstechnik, kurz BSI, ist der zentrale Sicherheitsdienstleister des Bundes und somit für die IT-Sicherheit in Deutschland verantwortlich. [20]

Der IT-Grundschutz betrachtet Vertraulichkeit, Verfügbarkeit und Integrität als die drei wichtigsten Grundwerte der IT-Sicherheit. Unter anderem wird Verfügbarkeit laut IT-Gundschutz erreicht durch:

- Organisatorische Maßnahmen wie SLAs
- Datensicherung und -wiederherstellung
- Speichertechnologien
- Geeignete Netzkonfigurationen
- Infrastrukturelle Maßnahmen
- Client-seitige Schritte
- Vorkehrungen auf Anwendungsebene
- Server-seitige Maßnahmen (Server-Standby, Failover)
- Methoden zur Replikation von Daten
- Disaster Recovery

Im Maßnahmenkatalog zur Notfallvorsorge wird des Weiteren beschrieben, wie eine Übersicht der Verfügbarkeitsanforderungen erstellt werden kann. Dazu sind die wesentlichen Komponenten der IT-Struktur zu bestimmen und danach wird diesen eine tolerierbare Ausfallzeit zugewiesen.

Es wird vorgeschlagen, die Anforderungen an die Verfügbarkeit von den Anwendern oder Fachabteilungen festlegen und begründen zu lassen. [21]

2.5.4 Verfügbarkeit nach AEC Klassen

Die Harvard Research Group versucht die Verfügbarkeit über die Availability Environment Classification in sechs Klassen einzuteilen, denen auch eine Verfügbarkeit in Prozent zugeordnet ist; die Tabelle 2.6 verdeutlicht die Zusammenhänge.[22]

Klasse	Bezeichnung	Bedeutung	Verfügbarkeit [%]
0	Conventional	Funktionalität kann unterbrochen werden, Daten unwichtig	99
1	Highly Reliable	Betrieb kann unterbrochen werden, Daten müssen jedoch erhalten bleiben	99,9
2	High Availability	Abschaltungen nur innerhalb festgelegter Zeiten oder minimal zur Hauptbetriebszeit.	99,99
3	Fault Resilient	Betrieb muss während bestimmter Zeiten durchgehend zur Verfügung stehen	99,999
4	Fault Tolerant	Funktionalität muss durchgehend gewährleistet sein	99,9999
5	Disaster Tolerant	Betrieb unter allen Umständen möglich	99,99999

Tabelle 2.6: AEC Klassen und deren Bedeutung [22]

2.5.5 Erreichbarkeitsstufen

Die in den vorhergehenden Kapitel angesprochenen Versuche, Verfügbarkeiten zu beschreiben, stellen sich allesamt als schwierig dar. Ansätze wie ITIL oder das IT-Grundschutz Konzept bieten wenig Konkretes, das österreichische Sicherherheitshandbuch definiert ähnlich dem AEC Verfahren verschiedene Klassen die Verfügbarkeit widerspiegeln sollen – wie die Messung zu erfolgen hat oder was überhaupt als verfügbar oder nicht gilt wird nicht festgelegt. Die Gesamtbetrachtung der bereits genannten Ansätze offenbart, das keines zum Monitoring von Diensten und der Feststellung der Dienstverfügbarkeit geeignet ist.

Deshalb wird an dieser Stelle ein eigenes Verfahren eingeführt, das speziell an die Anforderungen typischer Netzwerkmonitoringaufgaben angepasst ist. Die **Erreichbarkeit** eines Dienstes oder Systems wird von einem vorher definierten Beobachtungspunkt aus festgestellt; üblicherweise ist der Punkt zentral im Netzwerk positioniert und beherbergt die Monitoringstation(en).

Ein Dienst oder System ist erreichbar, wenn ein dafür geeigneter und von der Monitoringstation initiierter Test durchgeführt (Zweiweg Kommunikation) oder eine vom Dienst oder System selbst generierte Bestätigung (Einweg Kommunikation) übermittelt wird. Um ein System oder einen Dienst einzuordnen sind mehrere Schritte notwendig, welche zuerst erläutert und daraufhin anhand eines Beispieles durchleuchtet werden:

- **Betriebszeiten festlegen**
  Als erstes muss ein Intervall definiert werden, welches die Hauptbetriebszeit festlegt. Die Hauptbetriebszeit beschreibt die Zeit in welcher der Dienst oder das System am häufigsten benötigt werden; diese Zeit kann zum Beispiel von den Arbeitszeiten abgeleitet werden. Die Nebenbetriebszeit ist jene Zeit, die sich aus der Differenz zwischen der Hauptbetriebszeit und dem 24x7[3] Betrieb ergibt; in der Regel sollten sich die Werte auf ein gesamtes Jahr beziehen.
- **Zuweisung der Erreichbarkeit**
  Der Hauptbetriebszeit wird eine Ziel-Erreichbarkeitsstufe ES zugewiesen; die Erreichbarkeitsstufe k wird durch die Formel

$$ES = 1 - (\frac{1}{4})^{k} \tag{2.2}$$

  errechnet, dadurch ergibt sich eine Stufe die im Schema ES „Erreichbarkeitsstufe“ angegeben wird. Die Nebenbetriebszeit ist nicht von Relevanz, da in dieser Zeitspanne der Dienst nur selten benötigt wird. Bei einem Dienst oder System, das eine 24x7 Erreichbarkeit benötigt, kann mithilfe der Nebenbetriebszeit ein Wartungsintervall realisiert werden. Die Tabelle 2.7 listet die ersten 9 Erreichbarkeitsstufen mit den dazugehörigen Ausfallzeiten für einen 24x7 Betrieb, zusätzlich ist eine ES N auszumachen, welche eine Erreichbarkeit von 100% definiert. Wie aus der Tabelle ersichtlich wird, verkleinern sich die Abstände zwischen den einzelnen Stufen je näher diese an die 100% Marke kommen. Dieses Verhalten ist bewusst herbeigeführt, da eine exaktere Einteilung immer wichtiger wird je näher man sich der 100% Marke nähert.

3 bezeichnet eine Erreichbarkeit von 24 Stunden / Tag an 7 Tagen / Woche

ES	Erreichbarkeit	Downtime/Jahr
0	0%	365 Tage
1	75%	~ 91 Tage
2	93,75%	~ 22,8 Tage
3	98,43%	~ 5,7 Tage
4	99,609375%	~ 1,4 Tage
5	99,902344%	~ 8,5 Stunden
6	99,975586%	~ 2,1 Stunden
7	99,993897%	~ 32 Minuten
8	99,998474%	~ 8 Minuten
9	99,999619%	~ 2 Minuten
N	100%	0 Minuten

Tabelle 2.7: Erreichbarkeitsklassen für 24x7 Betrieb

- **Anpassung der Abfragehäufigkeiten**
  Nach erfolgter Zuweisung ist es nun notwendig, die Abfragehäufigkeiten der einzelnen Dienste und Systeme optimal an die ES Werte anzupassen. Um das Bewertungssystem nicht allzu komplex zu gestalten und gleichzeitig die Zuverlässigkeitsanforderungen an die Systeme und Dienste scharf zu formulieren, wird von genau einem Totalausfall des System oder Dienstes innerhalb der Hauptbetriebszeit im Laufe eines Jahres ausgegangen. Die Downtime der ES Klasse definiert daher, wie viel Zeit maximal zwischen Auftreten und Reparatur des Problems (MTFR)[4] verstreichen darf. Wichtig für das Monitoringsystem ist jedoch die Zeit die verstreicht zwischen dem Auftreten des Fehlers und deren Anzeige im Monitoringsystem (MTFD)[4], da das Programm auf die nachfolgende Reparatur durch entsprechendes Personal keinen direkten Einfluss mehr hat. Entscheidend bleibt jedoch, die MTFD[4] Zeit kleinstmöglich zu halten, um dem Personal die größtmögliche Zeit einzuräumen, was durch eine Reduktion der MTFD[4] Zeit auf maximal 10% der MTFR[4] erreicht werden.

Um die soeben beschriebene Vorgehensweise besser zu illustrieren, wird dieser nun anhand eines fiktiven Beispieles durchgerechnet.

Ein Dateifreigabedienst auf einem Server soll mit der ES-Methode beschrieben werden. Der Dienst wird hauptsächlich während der normalen Büroarbeitszeiten, also Montag bis Freitag von 08:00 und 18:00 genutzt. Daraus ergeben sich bei einem vereinfachten Kalendermodell ohne Feiertage und 52 Wochen a 7 Tagen eine Hauptbetriebszeit von 2600

4 siehe Kapitel 2.4

sowie eine Nebenbetriebszeit von 6136 Stunden. Während der Hauptbetriebszeit muss der Dienst praktisch immer verfügbar sein, Aufgrund von Absprachen mit den Dienstnutzern ist eine maximale Downtime von 4 Stunden akzeptabel. 4 von 2600 Stunden entsprechen einer Erreichbarkeit von ~99,846% und einer Stufe 5 für die Hauptbetriebszeit, also einer ES 5 Einstufung.

3 Problemstellung

Das Netzwerk der Firma Alpine-Mayreder Bau GmbH ist stark dezentral aufgebaut. Aufgrund der Firmenstruktur mit zahlreichen in- und ausländischen Niederlassungen besteht die Notwendigkeit, die IT-Struktur diesen Gegebenheiten anzupassen. Die daraus entstehende und zu überwachende Infrastruktur umfasst mehr als 300 Server, welche über 80 Niederlassungen in 16 Ländern verteilt sind. Viele der Server sind in der Zentrale in Salzburg konzentriert, weshalb dort auch die Managementstation integriert ist. Die zu überwachenden Geräte sind von verschiedenen Herstellern und unterscheiden sich in der Ausstattung sowohl software- als auch hardwareseitig, jedoch verwenden alle zu überwachenden Server Linux als Betriebssystem.

3.1 Zu überwachende Soft- und Hardware

Aufgrund der am häufigsten auftretenden Probleme werden die zu überwachenden Soft- und Hardwarekomponenten ermittelt. Zuerst wird jeweils auf das Problem eingegangen und danach auf die gewünschte Anzeigeform im Monitoringsystem. Bei Problemen die spezifisches Wissen erfordern werden zusätzliche Erläuterungen vorangestellt, um die notwendigsten Fakten kurz zu beleuchten.

3.1.1 HP Compaq SMART Array

HP Compaq bietet mit der SMART Array Controllerserie Hardware RAID Controller für die RAID Modi 0,1, 0+1 und 5 an. Die Geräte sind für den Betrieb von Festplatten mit SCSI Schnittstelle ausgelegt.Das Produkt wird im RAID 5 Betrieb verwendet, dieser Modus erfordert mindestens 3 physikalische Festplatten. Bei Ausfall einer der Festplatten befindet sich das System in einem kritischen Zustand, da keine Redundanz mehr gewährleistet ist. Das Monitoringprogramm muss diesen Fehler melden.[23]

3.1.2 Linux Software RAID

Linux bietet ab der Kernelversion 2.x eine Software RAID Funktionalität an. Die Festplatten werden im Linux Dateisystem unter /dev/mdX eingebunden, wobei X für die Nummer der RAID Arrays steht; das Gerät /dev/md0 ist also der erste Software RAID Verbund. Die Linux RAID Funktionalität wird vor allem auf Dateifreigabeservern genutzt um eine höhere Ausfallsicherheit zu erreichen. Auf den betreffenden Servern wird nur der RAID Level 1 (Redundanz durch Spiegelung) verwendet. Sollte eine der Festplatten eines RAID Verbundes ausfallen, befindet sich das System im kritischen Zustand. Sobald ein kritischer Zustand vorliegt muss dieser dementsprechend im Monitoringsystem hervorgehoben werden. [24]

3.1.3 Belegung einzelner Mountpoints

Je nach Aufgabe des Servers ist der Festplattenplatz in unterschiedliche Partitionen aufgeteilt. Wird die Auslastung eines Mountpoints nicht regelmäßig überprüft kann er vollaufen und Benutzer beziehungsweise Programme können keine weiteren Daten mehr abspeichern. Das Monitoringprogramm muss in der Lage sein, den Mountpoint und dessen Belegung in Prozent anzuzeigen um rechtzeitig auf ein sich ankündigendes Platzproblem reagieren zu können.

3.1.4 Letztes Backup mit Arkeia 5

Zum Backup von Benutzerdaten auf Linux Servern wird das Programm Arkeia 5 der Firma Arkeia verwendet. Wird auf einem System mehrere Tage lang keine erfolgreiche Sicherung durchgeführt, erhöht sich das Risiko eines Datenverlustes. Das Monitoring-programm eine entsprechende Warnmeldung ausgeben sobald eine Sicherung mehrmals nicht durchgeführt wurde.

3.1.5 Band im Bandlaufwerk

Oftmals zeichnet sich ein Problem mit der Datensicherung schon im Vorhinein ab, sollte der Sicherungsbeauftragte vergessen das Bandlaufwerk mit einem Medium zu versorgen. Das Monitoringprogramm sollte diesen Missstand erkennen und als Fehlermeldung anzeigen.

3.1.6 Zeitdifferenz zum Zeitserver

Um Fehler, die sich über verschiedene Systeme erstrecken genau korrelieren zu können, ist eine regelmäßige Zeitsynchronisation durchzuführen. Diese muss in wiederkehrenden Abständen kontrolliert werden um sicherzugehen, dass eine Synchronisierung tatsächlich stattfindet. Das Monitoringsystem muss die Differenz zwischen einem System und einem zentralen Zeitserver anzeigen können.

3.1.7 Anzahl der Mails in der Mailqueue

Durch Verbingungsprobleme oder überlastete Server kann es vorkommen, dass die Mailqueue eines Mailservers stetig anwächst und somit die Auslieferung von E-Mails beträchtlich verzögert wird. Um diesem Problem vorzubeugen muss eine entsprechende Warnung im Monitoringprogramm ausgegeben werden, die bereits frühzeitig eventuell anfallende Störungen aufzeigt indem die Anzahl der Mails in der Warteschlange des Mailservers beobachtet wird.

3.1.8 Uptime des Rechners in Tagen

Wenn ein Linux Server längere Zeit ununterbrochen läuft, wird bei einem geplanten oder ungeplanten Neustart des Systems eine vollständige Dateisystemprüfung veranlasst. Dies kann vor allem bei Servern mit entsprechend großen Festplatten eine beträchtliche Menge an Zeit beanspruchen. Läuft der Server erst seit kurzer Zeit, ist dies ein Indiz für einen Neustart des Betriebssystems, zum Beispiel verursacht durch einen Stromausfall. Der Administrator solle vom Monitoringprogramm über beide Umstände in Kenntnis gesetzt werden, so sie auftreten.

3.1.9 Durchschnittslast des Servers

Wird ein Server stark belastet reagieren die auf dem Gerät laufenden Dienste langsamer auf Anfragen, eine hohe Last kann auch Indiz für den Absturz eines einzelnen Dienstes sein. Die Anzeige der Durchschnittslast ermöglicht es frühzeitig auf Probleme zu reagieren, weiters ist es über statistische Auswertungen ist es möglich, zukünftig auftretende Engpässe vorherzusagen. Das Monitoring Programm muss die Durchschnittslast sowohl anzeigen als auch Statistisch auswerten können.

3.1.10 Status des DHCP Daemons

Der DHCP Daemon ist ein zentraler Dienst der alle Arbeitsstationen mit gültigen IP-Adressen und Einstellungen versorgt. Ist der Dienst nicht mehr verfügbar, können Arbeitsstationen keine Adressen mehr beziehen und sich infolge dessen nicht am Netzwerk anmelden. In der Regel wird der Ausfall des DHCP Daemons erst am nächsten Arbeitstag bemerkt, wenn die Mitarbeiter ihre PCs einschalten und die Geräte eine DHCP Anfrage an den DHCP Server stellen. Das Monitoringprogramm muss in der Lage sein, einen Ausfall des DHCP Daemons rechtzeitig zu erkennen, damit der zuständige Administrator zeitgerecht auf das Problem reagieren kann.

3.1.11 Erreichbarkeit des Webservers

Um verschiedene Dienste wie eine zentrale Kontaktverwaltung oder Wissensdatenbanken mit komfortabler Oberfläche zu realisieren werden Webserver eingesetzt. Sind ein oder mehrere dieser Server durch zahlreiche Benutzeranfragen überlastet, sinkt die Reaktionszeit der betreffenden Geräte. Die Zeiten müssen vom Programm wiedergegeben und analysiert werden.

3.1.12 Erreichbarkeit des SSH Dienstes

Der SSH Daemon ist ein zentraler Dienst zur Verwaltung eines Linuxservers, über das Netzwerk ist es mit Hilfe dieses Dienstes möglich, sich auf einem entfernten Server anzumelden um administrative Tätigkeiten durchzuführen. Sobald der Dienst nicht mehr Erreichbar ist kann der Administrator keine Einstellungen am Rechner mehr durchführen und eine Person vor Ort muss das Problem beheben. Sollte der Dienst nicht mehr verfügbar sein, ist über das Monitoringsystem eine Warnung auszugeben.

3.1.13 Antwort auf ICMP Ping Anfrage

Einige spezielle Rechner oder Drucker verwenden proprietäre Protokolle, diese Geräte sind in der Regel mit dem ICMP PING Befehl zu erreichen. Sobald ein Gerät nicht mehr auf eine ICMP Ping Anfrage antwortet, muss von einem Problem ausgegangen werden. Das Monitoringprogramm muss ICMP natürlich unterstützen.

3.1.14 SNMP

Einige Router und Appliances können ausschließlich über das SNMP Protokoll angesprochen werden um Monitoringinformationen zu erhalten. Auf diesen Geräten kann teilweise keine Software nachgerüstet werden, jedoch sind Statusmeldungen über den Zustand des Systems nur über eine SNMP Abfrage möglich. Eine Unterstützung von SNMP für das Monitoringsystem eine absolut zwingende Voraussetzung.

3.1.15 IMAP

IMAP wird verwendet, um E-Mails am Mailserver zu betrachten. Die Funktionalität von IMAP ist für die Mitarbeiter der Firma ein zentraler und kritischer Dienst, bei einem Ausfall muss sofort gehandelt werden. Die Antwortzeiten auf IMAP Anfragen sind permanent zu überwachen und auch statistisch auszuwerten. Über eine Statistik ist es möglich, Engpässe frühzeitig zu erkennen und der Administrator kann entsprechend gegenzusteuern – sowohl der Ausfall als auch die Reaktionszeiten der IMAP Systems müssen durch das Monitoringprogramm dargestellt werden.

3.1.16 SMTP

Mit SMTP können E-Mail Nachrichten verschickt werden. Ähnlich wie der IMAP Dienst ist auch SNMP eine zwingende Voraussetzung für den reibungslosen E-Mail Verkehr. Ausfälle kündigen sich oft lange vorher mit langsamen Reaktionszeiten an, bei einem frühzeitigen Erkennen ist es dem Administrator möglich rechtzeitig Gegenmaßnahmen zu ergreifen. Das Monitoringsystem muss den Administrator bei dieser Aufgabe unterstützen können, des Weiteren ist eine statistische Auswertung der Daten notwendig.

3.2 Anforderungen an die Bedienung

Die optimale Bedienbarkeit des Monitoringprogrammes ermöglicht es den Administratoren schnell und effizient zu arbeiten und erhöht somit den Nutzen des Systems. Folgende Teilaspekte müssen bei der Auswahl besonders genau betrachtet werden:

- **Einfache und intuitive Bedienung**
  Das Monitoringsystem muss für jeden Angestellten der IT-Abteilung intuitiv bedienbar sein, Begriffe und Bezeichnungen müssen dementsprechend eindeutig und selbsterklärend gewählt sein. Eine kostspielige Einschulung jedes Mitarbeiters kann dadurch entfallen.
- **Übersichtlichkeit**
  Die Oberfläche muss äußerst übersichtlich gestaltet sein, ohne dabei auf wichtige Einstellungsmöglichkeiten zu verzichten.
- **Oberfläche ohne Zusatzsoftware bedienbar**
  Die Bedienoberfläche muss ohne spezielle Zusatzsoftware von jedem Rechner innerhalb des Netzwerkes aus abrufbar sein, zum Beispiel über ein Webinterface, dadurch kann die Oberfläche mit einem beliebigen Webbrowser angezeigt werden.
- **Vermeidung von Fehlalarmen**
  Durch die komplexe Netzwerkstruktur können bei dem Ausfall eines wichtigen Knotens mehrere Fehlalarme gemeldet werden, sollten einige Geräte nur über den ausgefallenen Knoten erreichbar sein. Das Netzwerkmonitoring-Programm sollte in der Lage sein die Infrastruktur des Netzwerkes nachzubilden, um potentielle Falschmeldungen als solche zu markieren.

3.3 Erweiterbarkeit der Software

Die Software muss durch eine flexible Struktur Anpassungen ermöglichen und durch die Unterstützung von standardisierten Protokollen die Integration mit anderen Programmen erleichtern. Folgende Punkte sind besonders zu betrachten:

- **Programm individuell erweiterbar**
  Über eine definierte Schnittstelle ist es dem Administrator zu ermöglichen, eigene Programmerweiterungen, welche eine spezielle Problemstellung lösen, zu installieren. Dies kann über Plugins oder Module geschehen.
- **Agent-Software in interpretierender Programmiersprache**
  Die auf den Agents installierte Software sollte idealerweise in einer nicht kompilierten Form vorliegen. Dadurch lassen sich Fehler schneller finden, des Weiteren ist bei einer Anpassung der Software kein Compiler notwendig – welcher aus Sicherheitsgründen auf einigen Systemen nicht vorhanden ist.

- **Standardisierte Protokolle**
  Zur Übertragung von Mangementinformation müssen standardisierte und weit verbreitete Protokolle zum Einsatz kommen, damit bei einem eventuellen Wechsel des Monitoringsystems bestehende Abfragen übernommen werden können und der Aufwand bei einem Monitoring-Systemwechsel dadurch gesenkt wird.

3.4 Dokumentation

Eine ausführliche Dokumentation aller Einstellungen, Funktionen und Schnittstellen erleichtert die Installation, Konfiguration und Wartung der Softwarelösung wesentlich. Unvollständige, unübersichtliche und inkonsistente Dokumentationen sind ein schweres Defizit für jedes Softwareprogramm.

3.5 Professionelle Unterstützung

Sollte das firmeninterne Personal vor einem scheinbar unlösbaren Problem stehen, ist schnelle und unkomplizierte Unterstützung von externen Spezialisten von Vorteil. Egal ob über Telefon, Remotezugriff oder vor Ort Einsatz – wichtig ist die möglichst schnelle Wiederherstellung der Einsatzbereitschaft des Monitoringsystems.

3.6 Sicherheit

Die Sicherheit von IT Systemen steht zunehmend im Vordergrund aller Softwarelösungen, der Schutz vor Hackern sollte dabei ebenso wichtig sein wie der Schutz durch einen Fehler in Soft- oder Hardware. Die nachfolgenden Punkte müssem besonders beachtet werden:

- **Verschlüsselung und Authentifizierung**
  Die Übertragung von Monitoringinformationen muss ausreichend gesichert werden; sowohl eine Verschlüsselung der Daten durch ein adäquates Verfahren als auch eine sichere Authentifizierung beim Abfragen einzelner Informationen von den Agents muss gegeben sein. Des Weiteren muss die Monitoringstation selbst von unbefugten Zugriffen geschützt werden, damit nur authorisierte Personen Zugriff auf die entsprechenden Monitoringinformationen haben. Die Verschlüsselung und Authentifizierung stellt ein sogennantes KO Kriterium dar – sollte ein Monitoringprogramm diese Möglichkeiten nicht bieten kommt es nicht in Betracht.

- **Einfache Sicherung**
  Die gesamte Konfiguration sollte in möglichst einfacher Form zu sichern sein, eine Archivierung von alten Konfigurationsdateien ist außerdem wünschenswert; sollten sich unbemerkt Fehler in die Konfiguration einschleichen kann mit einer Archivierung auf eine fehlerfreie Konfiguration zurückgegriffen werden.

3.7 Leistung

Durch die effiziente Implementierung der Monitoringsoftware werden nur wenige Ressourcen belegt. Nachfolgende Punkte gilt es speziell zu beachten:

- **Bandbreitenbedarf**
  Durch ein effizientes Netzwerkprotokoll mit intelligenter Lastverteilung wird die zu Verfügung stehende Bandbreite minimal und gleichmäßig belastet.
- **Hardware**
  Das Monitoringsystem läuft auf handelsüblicher PC-Hardware ohne merkliche Leistungseinbußen, die Belastung von Festplatten, Prozessoren und die Belegung des Arbeitsspeichers ist so gering wie möglich.
- **Oberfläche**
  Die Monitoringoberfläche besitzt einen zügigen Seitenaufbau der auch von langsamen Rechnern aus schnell zu bedienen ist.

3.8 Kosten

Ausschlaggebend für die Bewertung der Kostenfrage ist die Summe aus mehreren Teilaspekten, die wie folgt eingeteilt sind:

- **Softwarekosten**
  Die Initialkosten für die Beschaffung der Software sind so gering wie möglich zu halten, wobei ein höherer Nutzen des Programmes entsprechend honoriert wird. Es gilt jedoch zu beachten, dass die Beschaffungskosten von über 10000 Euro das IT Budget überschreiten und somit ein weiteres KO Kriterium darstellen.
- **Installationskosten**
  Die Kosten für die Installation sind gering zu halten, was zum Beispiel durch den Verzicht auf eine eigene Schulung zum Monitoringsystem erfolgen kann.
- **Wartungskosten**
  Durch die intuitive Bedienung ist die Wartung des Systems besonders einfach und kosteneffizient durchzuführen.

4 Lösungsvorschläge

Die möglichen Softwarepakete zur Lösung der gegebenen Problemstellung sind in zwei große Bereiche zu unterteilen: Auf der einen Seite stehen kommerzielle Programme von großen IT-Unternehmen, welche einen teilweise überwältigenden Funktionsumfang bieten. Auf der anderen Seite stehen die Open Source Programme, welche oft nur sehr reduzierte Möglichkeiten bieten. Aus beiden Kategorien werden exemplarisch einige der meist verwendeten Lösungen vorgestellt. Nach einer allgemeinen Einführung des Programmes wird auf die Unterstützung der zu überwachenden Hard- und Software eingegangen, um schließlich die Erfüllung der zusätzlichen Anforderungen aus den Punkten 3.2 bis 3.8 sowie Allfälliges zu beleuchten, wobei die Professionelle Unterstützung sowie die Kosten separat hervorgehoben werden. Die Informationen werden aus den entsprechend angegebenen Informationsquellen bezogen.

4.1 Open Source Programme

Der Begriff Open Source bezeichnet Programme, bei denen der zu Grunde liegende Quelltext für jedermann frei einzusehen ist. Entgegen der landläufigen Meinung ist Open Source jedoch nicht mit gratis gleichzusetzen – vor allem dann wenn die Gesamtkosten betrachtet werden, siehe dazu auch [25].

4.1.1 Cricket

Cricket wird seit 1999 entwickelt und liegt zum Stichtag 29.3.2006 in der stabilen Version 1.0.5 vor. Cricket verwendet einen so genannten „config tree“ um Informationen über abzufragende Geräte zu speichern. Der config tree ist hierarchisch aufgebaut und ermöglicht durch Vererbung Konfigurationseinstellungen effizient zu verwalten. Cricket ist komplett in der Programmiersprache Perl geschrieben und somit auf jedem Betriebssystem mit Perl-Unterstützung lauffähig. Das Hauptaugenmerk der Software liegt in der Visualisierung der Daten mittels verschiedener Graphen. Die Entwickler von Cricket arbeiten eng mit dem Programmierer der RRDtools zusammen, die Monitoringdaten werden in RR-Datenbanken gespeichert. [7]

Abfragen selbst sind in Cricket nicht integriert, jedoch werden vorkonfigurierte Programme von Dritten beigesteuert. Diese decken sich mit den Anforderungen auf Punkt 3.1 in folgenden Bereichen:

- Belegung einzelner Mountpoints
- Anzahl der Mails in der Mailqueue
- Uptime des Rechners in Tagen
- Antwort auf ICMP Ping Anfrage

Da keine Abbildungen der Programmoberfläche verfügbar sind ist es nicht möglich Aussagen über die Bedienbarkeit des Webinterfaces zu treffen. Die Erweiterung um neue zu überwachende Dienste und Systeme muss über Konfigurationsdateien vorgenommen werden. Konkrete Angaben über Systemvoraussetzungen und Ressourcenverbrauch in Bezug auf Hardware und Netzwerkbandbreite werden nicht gemacht. Da Abfragen in Cricket nicht integriert sind liegt es in der Hand des Benutzers, Sicherheitsmechanismen einzubauen. Ferner erwähnt die Dokumentation des Programmes keinerlei Mechanismen zur Vermeidung von Fehleralarmen. Die Darstellung der Monitoringdaten wird von einem Apache Webserver übernommen, welcher Authentifizierung und Verschlüsselung anbietet. Die Dokumentation ist ausführlich, jedoch muss fast jede Einstellung über das Editieren einer Konfigurationsdatei vorgenommen werden, dadurch erfordert die Administration von Cricket detaillierte Fachkenntnis und Erfahrung mit dem Programm. [26]

Professionelle Unterstützung wird von einem Einzelunternehmer in Wien angeboten, Lizenzkosten fallen keine an. [27]

4.1.2 Big Sister

Das Monitoringprogramm Big Sister liegt zum Stichtag 23.3.2006 in der stabilen Version 1.02-4 vor. Big Sister wurde ursprünglich von Thomas Aeby entwickelt, mittlerweile beteiligen sich auch andere Personen an der Entwicklung des Programmes. Der Name des Monitoringprogrammes entstand in Anlehnung an das Big Brother Monitoringsystem, von welchem Big Sister inspiriert wurde. Big Sister versteht sich als Weiterentwicklung von Big Brother hinsichtlich Performance und Benutzerfreundlichkeit.

Big Sister arbeitet nach dem Client-Server Prinzip und verwendet für die Datenübertragung ein proprietäres TCP-basierendes Protokoll. Das Programm ist auf Windows und Unix Betriebssystemen lauffähig, als Programmiersprache kommt Perl zum Einsatz. [7]

Folgende der Kapitel 3.1 geforderten Abfragen werden unterstützt:

- Belegung einzelner Mountpoints
- Anzahl der Mails in der Mailqueue
- Durchschnittslast des Servers
- Erreichbarkeit des Webservers
- Erreichbarkeit des SSH Dienstes
- ICMP Ping
- SNMP
- IMAP
- SMTP [28]

Die Benutzeroberfläche ist über einen beliebigen Browser zugänglich und wirkt sehr aufgeräumt, das Hinzufügen von neuen Diensten und Systemen ist über das Editieren der Konfigurationsdateien möglich. Konkrete Angaben über den Ressourcenbedarf sind nicht zu finden, ebenso wenig wie eine aktuelle Dokumentation. Die vorhandene Beschreibung ist weder vollständig noch konsistent, des Weiteren fehlen Anleitungen zum einfachen erweitern der Funktionalität. Die Verschlüsselung der Monitoringdaten ist über einen SSH Tunnel möglich, der für die Verfügbarkeit der Monitoringoberfläche zuständige Apache Webserver kann mittels SSL gesichertem HTTP verschlüsselt angesprochen werden. Die Monitoringabfragen werden wie eingangs erwähnt über ein proprietäres Protokoll abgefragt, alternativ könnte aber SNMP Verwendung finden; dieses ist aber laut Dokumentation nicht in der sicheren v3 Version verfügbar. Um ein effektives Backup zu erhalten genügt es, die Konfigurationsdateien zu sichern.

Professioneller Support wird von einem Einzelunternehmer in Österreich angeboten, Lizenzkosten für das Programm selbst fallen nicht an. [27]

4.1.3 Nagios

Seit 1999 entwickelt Ethan Galstad ein Netzwerkmonitoringsystem, welches sich in seiner ursprünglichen Version Netsaint bezeichnet und im Jahr 2002 in Nagios umbenannt wurde. Zum Stichtag 28.3.2005 liegt Nagios in der stabilen Version 2.1 vor. Nagios verwendet ein Client-Server System zur Ermittlung der Monitoringdaten. Der Nagios Daemon selbst ist in der Programmiersprache C geschrieben und nur auf die notwendigsten Programmteile beschränkt. Erst in Kombination mit der Software Nagios-plugins, welche separat entwickelt wird, ist der volle Funktionsumfang verfügbar. Die durch die Plugins ermöglichte modulare Architektur eröffnet die Option, auf einfache Weise selbst eventuell fehlende Programmteile nachzurüsten. Die Plugins können dazu in fast jeder beliebigen Programmiersprache erstellt werden – die Entwickler der Nagios-plugins verwenden C, Perl und die Unix Shell. Über Benachrichtigungen ist es möglich, kritische Systeme per Kurzmitteilung, Pager-Nachricht oder E-Mail dem zuständigen Administrator weiterzuleiten. [29],[30],[31]

Verschiedenste Abfragen sind bereits realisiert, die für die Anforderungen relevanten sind nachfolgend aufgelistet:

- Belegung einzelner Mountpoints
- Anzahl der Mails in der Mailqueue
- Durchschnittslast des Servers
- Erreichbarkeit des Webservers
- Erreichbarkeit des SSH Dienstes
- Antwort auf ICMP Ping Anfrage
- SNMP
- SMTP

Die Benutzeroberfläche ist über einen Webbrowser zugänglich und sehr übersichtlich. Einige Einstellungen können direkt über das Webinterface vorgenommen werden, zum Hinzufügen von neuen Servern und Anpassen der Einstellungen bietet die Firma IT Groundwork mit dem Open Source Programm Monarch eine webbasierende Oberfläche an, welche sämtliche Nagios Einstellungen anpassen kann.

Konkrete Hardwareanforderungen sind nicht zu finden, die Software lässt sich aber auf fast allen unixartigen Betriebssystemen einsetzen, unter anderem stehen für die Linux-Distributionen der Firmen Novell (SuSE), Redhat (RHEL) und Debian vorkonfigurierte Pakete zur Verfügung. Dank der bereits erwähnten Plugin-Architektur ist es ein leichtes, das Programm selbst zu erweitern, nicht zuletzt dank der ausführlichen Dokumentation. Das Standard Kommunikationsprotokoll von Nagios verwendet SSL Verschlüsselung, jedoch ist keine Authentifizierung vorgesehen und die Kommunikation selbst erfolgt über ein proprietäres System. Aufgrund der offenen Pluginarchitektur und der vollen SNMP v3 Unterstützung ist es möglich, ohne großen Aufwand ein voll verschlüsseltes, authentifiziertes und standardisiertes Kommunikationsprotokoll zu verwenden. Als Webserver kommt Apache zum Einsatz, weshalb Authentifizierung und Verschlüsselung des Webinferfaces möglich sind. Um eine vollständiges Backup durchzuführen genügt es, den Nagios Konfigurationsordner zu sichern; die erweiterten Einstellungsmöglichkeiten von Monarch sind über die SQL Datenbank realisiert und müssen separat gesichert werden. Durch ein System von Abhängigkeiten werden Fehlalarme effektiv verhindert. Die Installationskosten sind sehr gering, auch die Wartung ist einfach und bedarf keiner speziellen Schulung.[32]

Professioneller Support für Nagios wird von mehreren IT-Dienstleitungsunternehmen in Österreich gewährt, für Monarch gibt es auch Unterstützung vom Hersteller IT Groudwork. Lizenzkosten fallen keine an. [27]

4.1.4 Zabbix

Die Programmierung von Zabbix wurde von Alexei Vladishev im Jahr 2001 begonnen, seit dem Jahr 2005 entwickelt die vom Autor gegründete Firma ZABBIX SIA das Produkt weiter. Zum Stichtag 28.3.2006 steht die stabile Version 1.0 zur Verfügung. Als Programmiersprache wurde C verwendet, zur Speicherung der Daten wird eine SQL Datenbank verwendet. Ein Ereignismanagementsystem ermöglicht vordefinierte Benachrichtigungen einzelner Personen bei Problemen. Die Dokumentation verweist nirgends auf das verwendete Protokoll für Abfragen oder eventuelle Schutzmaßnahmen, jedoch ist eine Authentifizierung beim Anmelden an das Monitoringsystem erforderlich. [33]

Von den geforderten Überwachungsmöglichkeiten sind folgende bereits implementiert:

- Belegung einzelner Mountpoints
- Uptime des Rechnes in Tagen
- Durchschnittslast des Servers
- Erreichbarkeit des Webservers
- Erreichbarkeit der SSH Dienstes
- Antwort auf ICMP Ping Anfrage
- SNMP
- IMAP
- SMTP [33]

Die Monitoringoberfläche ist über eine Webbrowser zugänglich und macht einen sehr aufgeräumten Eindruck, außerdem ist die Administration des Monitoringsystems bereits in das Webinterface integriert. Das System setzt mit 128 MB RAM und 100 MB Festplattenplatz nur geringe Hardwareressourcen voraus. Als Software werden ein Linux Betriebssystem sowie eine SQL Datenbank vorausgesetzt. Eine Erweiterung des Programms ist zwar grundsätzlich möglich, jedoch ist die Dokumentation hierzu unvollständig. Wie bereits erwähnt ist die Verschlüsselung und Authentifizierung nur teilweise vorhanden, jedoch gibt es die Möglichkeit SNMP v1, v2 und v3 Abfragen zu tätigen.

Sollte professioneller Support notwendig sein stellt die Herstellerfirma verschiedene Wartungspakete zur Verfügung, für die Software selbst fallen keine Lizenzkosten an. [33]

4.2 Kommerzielle Programme

4.2.1 Microsoft Operations Manager 2005

Der Microsoft Operations Manager 2005 (kurz: MOM 2005) wird von der Firma Microsoft entwickelt. Das Programm ermöglicht die Überwachung und Verwaltung von Servern und Netzwerkkomponenten. Microsoft konzentriert sich beim Operations Manager jedoch ausschließlich auf Windows Server, die mit WMI[5] und SNMP abgefragt werden können.

5 siehe Kapitel 2.2.2

Das Programm verwendet zur Aufbewahrung der Monitoringdaten eine SQL Datenbank, die auch auf einem separaten Server untergebracht sein kann. Über eine Erweiterung der Firma Quest Software mit der Bezeichnung VSM ist es jedoch möglich, auch Linux Systeme in das Monitoring einzubinden, diese Software liegt zum Stichtag 3.04.2006 in einer Beta-Version vor. Der MOM 2005 bietet darüber hinaus eine integrierte Wissensdatenbank an, welche bereits Artikel der Microsoft Knowledge-Base mit MOM Fehlermeldungen verknüpft; diese Datenbank kann beliebig erweitert werden. [34],[35]

Der Microsoft Operations Manager 2006 kann in Kombination mit der VSM Beta Version folgende Anforderungen erfüllen:

- Belegung einzelner Mountpoints in Prozent
- Anzahl der Mails in der Mailqueue
- Durchschnittslast des Servers
- Erreichbarkeit des Webservers
- Erreichbarkeit des SSH Dienstes
- SNMP [34]

Bezugnehmend auf [36] kann davon ausgegangen werden, dass zukünftige Versionen des Operations Manager eine deutlich bessere Unterstützung von Linux basierenden Systemen bieten werden. Die Oberfläche des Operations Manager 2005 ist sowohl über eine so genannte Administrator Console als auch über ein Webinterface erreichbar. Sämtliche Ansichten sind individualisierbar und ermöglichen es, für unterschiedliche Administrationszwecke angepasste Ansichten zu definieren. Microsoft gibt für einen 1,8 Ghz Pentium 4 mit 1 GB RAM und einer einzelnen Festplatte mit 200 Agents eine durchschnittliche CPU Belastung von 28% und eine durchschnittliche Festplatten Idle Zeit von 70% an. Bei diesen 200 Agents fällt eine durchschnittliche Netzwerklast von 10 KBytes pro Sekunde an, das MOM Working-Set benötigt 33 MB Arbeitsspeicher. Über das MOM Connector Framework ist es möglich, den Operations Manager mit Programmen von Drittanbietern zu verbinden. Die Kommunikation zwischen Agents und dem Monitoringserver ist komplett verschlüsselt und digital signiert; Authentifizierung ist ebenfalls möglich. Ein komplettes Ereignismanagement ermöglicht es Falschmeldungen zu unterdrücken und Ursachen schnell ausfindig zu machen. [35]

Die US Version kostet umgerechnet[6] rund 5000 Euro für 10 Server und rund 760 Euro für jeden weiteren; für die Zusatzsoftware von Quest Software gibt es noch keine Preise. Microsoft bietet für den Operations Manager eine umfangreiche Produktunterstützung an, so sind auf der Microsoft Webseite ausführliche Dokumentationen und Planungsrichtlinien für den Operations Manager zu finden, darüber hinaus ist es möglich Wartungsverträge abzuschließen oder aber einzelne Probleme gegen eine einmalige Gebühr bearbeiten zu lassen. [35]

4.2.2 IBM Tivoli

Der Begriff Tivoli wird von IBM für eine Reihe von Produkten verwendet, welche für die Steuerung der IT-Infrastruktur Verwendung finden. Neben Produkten zum Sicherheits- und Storagemanagement gibt es auch ein System zur Überwachung von Netzwerkkomponenten, es trägt die Bezeichnung IBM Tivoli Monitoring. Das Programm bietet mit SOAP und SQL definierte Schnittstellen an um auf die Monitoringdaten zuzugreifen. Eine Tivoli Installation besteht aus mehreren Komponenten. Der Tivoli Enterprise Monitoring Server ist für das Abfragen einzelner Überwachungsdaten zuständig, welche mit Hilfe von Agenten auf den zu überwachenden Servern gesammelt werden. Die Ergebnisse der Abfragen werden von der Teilkomponente Tivoli Data Warehouse in eine SQL Datenbank gespeichert. Der Tivoli Enterprise Portal-Server wiederum übernimmt die Darstellung der Monitoringdaten aus dem Tivoli Data Warehouse, die vom Tivoli Enterprise Portal-Client aus betrachtet werden können. Die Zusatzkomponente Tivoli Enterprise Console bietet darüber hinaus ein umfangreiches Ereignismanagement. [37]

Die nachfolgenden bereits implementierten Überwachungsmöglichkeiten decken sich mit den in Kapitel 3.2 genannten Anforderungen:

- Erreichbarkeit des Webservers
- SNMP

6 Grundlage: Valutenkurs der Bank Austria Creditanstalt vom 03.04.2006

Der Tivoli Monitoring bietet unterschiedliche Typen von Agenten an, wobei für die gegebene Infrastruktur nur der Tivoli Monitoring Universal Agent in Frage kommt, der neben der oben genannten Möglichkeit HTTP zu überwachen auch Log-Dateien, Skripts und ähnliches ansteuern kann - die Abfragen müssen alle eigenhändig programmiert werden. [38],[39]

Die Monitoringoberfläche ist sowohl über ein Webinterface als auch über ein eigenes Java Programm erreichbar, die Oberfläche selbst ist sehr umfangreich aber auch etwas träge. Es ist möglich, eigene Skripts in die Monitoringoberfläche von Tivoli zu integrieren. Die SNMP Implementierung des Universal Agent erwähnt nirgends die Unterstützung für die Version 3 von SNMP. Die Vermeidung von Fehlalarmen wird durch eine Reihe von Maßnahmen verhindert, außerdem können Probleme nach ihrer Dringlichkeit automatisch gereiht werden und darüber hinaus ist es mit Tivoli sogar möglich, automatische Korrekturmaßnahmen beim Auftreten eines Fehlers durchzuführen.

Für 300 Server beträgt der Preis für eine IBM Tivoli Installation laut [40] 213000 Euro, in diesem ist bereits 1 Jahr Softwarewartung inbegriffen; weiters bietet IBM umfangreiche Beratung-, Dienstleistungs- und Unterstützungsangebote an.

4.2.3 HP Openview

Das Programm OpenView von HP wird laut eigenen Angaben von 100% der in den Fortune 50[7] geführten Unternehmen genutzt und kommt auf rund 135.000 Installationen weltweit. HP Openview ist ein äußerst umfangreiches und flexibles Programm; welches unterschiedlichste Anforderungen mit verschiedenen Openview Teilproduken realisiert. Für das Netzwerkmanagement wird das Produkt HP Openview Network Node Manager verwendet, dieses realisiert unter anderem die SNMP Unterstützung. Openview Operations ist für das Applikationsmonitoring zuständig, mit diesem Programm ist es möglich, Dienste wie Sendmail zu beobachten. Mit der Komponente Openview Internet Services ist es möglich, Dienste wie SSH und HTTP abzufragen. Sämtliche Monitoringdaten werden von Openview in einer zentralen Datenbank gespeichert. [41]

7 siehe http://money.cnn.com/magazines/fortune/fortune500/

Mit den oben genannten Komponenten ist es bezugnehmend auf [42] und [43] mit HP Openview möglich, folgende Hard- und Software zu beobachten:

- HP Compaq SMART Arrays
- Belegung einzelner Mountpoints in Prozent
- Erreichbarkeit des Webservers
- Erreichbarkeit des SSH Dienstes
- Antwort auf ICMP Ping Anfrage
- SNMP
- IMAP Reaktionszeit
- Reaktionszeit auf SMTP

Die Monitoringoberfläche ist über ein Webinterface ansteuerbar, außerdem ist es möglich das Programm über so genannte Smart Plugins selbst zu erweitern. Für die Übertragung der Monitoringdaten wird das standardisierte SNMP Protokoll verwendet, jedoch ist die authentifizierte und verschlüsselte Variante SNMP v3 nur über den Zukauf eines Drittanbietermoduls verfügbar. Der Zugriff auf das Webinterface ist über eine SSL gesicherte Verbindung möglich, welche auch Authentifizierunginformationen abfrägt. HP hat für Openview ein integriertes Backupprogramm geschaffen, das sich selbstständig um die Sicherung der zugrunde liegenden Datenbank kümmert.

Das Produkt HP Openview Operations, welches ein Maximum von 50 Servern beobachten kann kostet zum Stichtag 12.7.2004 rund 46.000 Euro zzgl. Mehrwertsteuer; im Preis inbegriffen ist auch 1 Jahr Telefonunterstützung. HP bietet für Openview Kunden eine umfangreiche Produktunterstützung an, welche vom Support einer Testinstallation bis zur Hilfe in der Produktionsumgebung reicht. [44]

4.2.4 CA Spectrum

Die Firma CA bietet mit der Spectrum Produktreihe verschiedene Komponenten zur Neztzwerküberwachung an. Spectrum wurde ab 1991 bei der Firma Aprisma entwickelt, welche aber von der Firma Concord gekauft wurde, die später wiederum von CA übernommen wurde. Um die in Kapitel 3 definierten Anforderungen zu erfüllen, sind die Komponenten Spectrum Xsight für das Grundprogramm sowie Dienst und Systemabhängigkeiten, Spectrum OneClick für die Realisierung eines Webinterfaces und Spectrum Report Manager für ein Alamierungssystem mit Statistikauswertung notwendig. CA bietet zahlreiche Zusatzmodule und Erweiterungen an, um Interoperabilität mit anderen Netzwerkmanagementsystemen zu erreichen oder Technologien wie MPLS oder VPN in das Monitoring einzubinden. [45],[46]

Die nachfolgenden Abfragen werden laut [47] unterstützt:

- Belegung einzelner Mountpoints in Prozent
- Uptime des Rechners in Tagen
- Erreichbarkeit des Webservers
- Antwort auf ICMP Ping Anfrage
- SNMP

Die Hardwarevorraussetzungen für einen entsprechenden Server sind wie folgt angegeben: 1.5 GB RAM, Intel Xeon Prozessor und 3 Stück unabhängige Festplatten mit mindestens 4GB werden für Spectrum unter Windows benötigt. Die Monitoringoberfläche ist über Spectrum OneClick mit einem Webbrowser abrufbar, der Verbindungsaufbau erfolgt über ein gesichertes Protokoll. Das Programm kann mit zusätzlichen Abfragen ergänzt werden, außerdem ist ein System zur Vermeidung von Fehlalarmen bereits integriert. Für viele Geräte bietet der Hersteller bereits vorgefertigte Abfragen und Konfigurationen an, was die Implementation zeitlich verkürzen und vereinfachen kann. Ein Konfigurationsmanagent sorgt dafür, dass die Systemkonfiguration verifiziert, gespeichert und gesichert wird; außerdem bietet der Hersteller mit XML und SQL standardisierte Systeme zur Abfrage von Konfigurations- und Monitoringdaten zur Verfügung.

Die Firma CA bietet eine breite Produktpalette an passenden Servicedienstleistungen an; die von der Beratung bis zum Upgrademanagement reichen; jedoch konnte weder durch den Kontakt mit der österreichischen Niederlassung von CA noch durch eine Konsultation der Homepage ein konkreter Preis für die Software ausfindig gemacht werden. Dies steht offenbar im direkten Zusammenhang mit der erst kürzlich erfolgten Übernahme der Spectrum Produktpalette durch CA. [48]

4.3 Eigenentwicklung

Neben eines der bereits vorgestellten Programme zu verwenden, besteht die Möglichkeit in Eigenregie ein Monitoringprogramm zu entwickeln, welches optimal auf die individuellen Bedürfnisse zugeschnitten ist. Zwar sind bei allen anderen Programmen ebenfalls gewisse Anpassungen und Ergänzungen notwendig, jedoch beschränkt sich dies hauptsächlich auf die zu überwachenden Komponenten welche von vielen Programmen nicht vollständig unterstützt werden. Zusätzlich zu diesen Abfragen ist jedoch die komplette Implementation eines effektiven und flexiblen scheduling Mechanismus, eines Speicherkonzeptes sowie einer Statistikenauswertung notwendig. Deshalb ist es nahe liegend, kein komplett neues Monitoringsystem zu entwickeln sondern auf ein bestehendes System aufzusetzen.

Die zu überwachende Soft- und Hardware ist bei einer Eigenentwicklung natürlich komplett abgedeckt.

Die zusätzlichen Anforderungen an die Software können durch die Verwendung von geeigneten Protokollen und Diensten komplett abgedeckt werden, jedoch ist eine schnelle Unterstützung von Dritten nicht möglich, ebenso sind die Wartungskosten schwer kalkulierbar. Die komplette Entwicklung nimmt natürlich entsprechend viel Zeit in Anspruch.

4.4 Gegenüberstellung und Auswahl

Wurden in den vorhergehenden Kapitel ausführlich Anforderungen und mögliche Lösungsmöglichkeiten diskutiert, ist es nun notwendig diese Lösungen gegenüber zu stellen und das am besten geeignete Programm auszuwählen. Die Tabelle 4.1 stellt die in den vorhergehenden Kapitel vorgestellten Programme gegenüber, wobei zu beachten ist, dass diese Tabelle nur Gültigkeit für die spezifischen Anforderungen der Firma Alpine-Mayreder Bau GmbH besitzt.

Information								
Produkt	Cricket	Big Sister	Nagios	Zabbix	Operations Manager	Tivoli	Openview	Spectrum
Hersteller	Jeff R. Allen	Thomas Aeby	Ethan Galstad	Zabbix SIA	Microsoft	IBM	HP	CA
Homepage	cricket.sourceforge.net	bigsister.ch	nagios.org	zabbix.com	microsoft.com/mom	ibm.com/software/tivoli	openview.hp.com	ca.com
Preis in Euro**	0	0	0	0	220400	213000	276000	?
Administration								
Unterstützung Hard- und Software	4/16	9/16	8/16	9/16	6/16	2/16	8/16	5/16
Bedienung	5/10	7/10	8/10	9/10	9/10	9/10	9/10	9/10
Erweiterbarkeit	9/10	2/10	9/10	4/10	9/10	10/10	9/10	10/10
Dokumentation	2/10	4/10	8/10	8/10	10/10	7/10	8/10	6/10
Professionelle Unterstützung	2/10	2/10	4/10	8/10	10/10	10/10	10/10	10/10
Sicherheit								
Verschlüsselte und authentifizierte Agent-Server Kommunikation*	●	●	●	○	●	●	○	?
Verschlüsselte und authentifizierte Server-User Kommunikation*	●	●	●	●	○	?	●	●
Backup und Wiederherstellung	5/10	5/10	4/10	3/10	3/10	2/10	10/10	10/10
Leistung								
Bandbreitenbedarf	5/10	5/10	5/10	5/10	10/10	?	6/10	?
Hardwarebedarf	9/10	9/10	7/10	8/10	8/10	?	?	1/10
Geschwindigkeit Bedienoberfläche	10/10	10/10	9/10	9/10	8/10	8/10	8/10	8/10
Kosten								
Softwarekosten*	10/10	10/10	10/10	10/10	0/10	0/10	0/10	?
Installationskosten	2/10	5/10	8/10	8/10	10/10	10/10	10/10	10/10
Wartungskosten	2/10	4/10	6/10	6/10	9/10	10/10	10/10	?
*..... k.o. Kriterium: bei Nichterfüllung der Anforderung ist das Produkt ungeeignet, siehe Text ○..... nicht implementiert								
** .... linear umgerechnet für 300 Server, exklusive Steuern ●..... implementiert								

Tabelle 4.1: Gegenüberstellung der einzelnen Programme

Der ersten Gruppe aus der Tabelle 11 sind allgemeine **Informationen** und Preise zu entnehmen; die Preisangaben wurden linear auf die benötigte Anzahl von Lizenzen hochgerechnet. Die nächste Gruppe ist unter dem Begriff **Administration** zusammengefasst. Der Punkt Unterstützung Hard- und Software gibt an, wieviele Abfragen bereits implementiert sind, jede Implementation spiegelt sich in der Vergabe eines Punktes wieder – maximal sind aufgrund der Anforderungen aus Kapitel 3.1 16 Punkte möglich. An dieser Stelle ist bereits zu erkennen, dass für die Implementation einige Abfragen selbst geschrieben werden müssen. Die Bedienungswertung von Cricket und Big Sister bekommen Abzüge, da die komplette Konfiguration bzw. Teile davon mit einem Editor geschrieben werden müssen.

Die Programme der großen Hersteller bieten zwar einen enormen Funktionsumfang, jedoch leidet die Übersichtlichkeit daran. Die Erweiterbarkeit der meisten Programme ist absolut ausreichend, besonders hervorheben können sich IBM und CA aufgrund der vielen standardisierten Schnittstellen. Big Sister und Zabbix sind etwas unflexibler, was vor allem an der lückenhaften Dokumentation liegt, welche als nächstes bewertet wird und sich dort ebenfalls niederschlägt. In dieser Kategorie sticht Microsoft mit dem Operations Manager hervor – die Dokumentation ist ausführlich und dank der guten Struktur übersichtlich und schnell handhabbar. Die Dokumentationen von IBM und HP sind zwar ebenfalls ausführlich nur deutlich unübersichtlicher, das CA Produkt kann ebenfalls wenig überzeugen; Schlusslicht bildet das äußerst spartanisch dokumentierte Cricket. In der Rubrik der professionellen Unterstützung können die kommerziellen Anbieter ihre Stärken voll ausspielen, einzig Zabbix kann hier einigermaßen mithalten. Als erstes KO Kriterium wird nun die **Sicherheit** der Monitoringsysteme beleuchtet. Mit allen OpenSource Programmen ist es grundsätzlich möglich, über das standardisierte und sichere SNMP v3 Protokoll mit dem Agent zu kommunizieren, jedoch ist die Dokumentation von Zabbix an dieser Stelle praktisch nicht vorhanden. Die Lösungen von IBM und Microsoft sind ebenfalls als sicher einzustufen, HP Openview verwendet weder Verschlüsselung noch Authentifizierung und bei Spectrum ist keine Dokumentation dahingehend zu finden. Etwas anders ist die Situation bei der Kommunikation zwischen Monitoringstation und der Konsole des Monitoringusers. Die Open Source Systeme verwenden alle den mittels SSL gesicherten Apache Webserver, auch HP und CA zeigen keine Schwächen, in der IBM Dokumentation ist nichts über Sicherheitsmaßnahmen in diesem Bereich zu finden, bei Microsoft ist eine sichere Kommunikation über das Webinterface nicht vorgesehen. Auch bei der Sicherung der Konfigurationsdaten gibt es große Unterschiede: Openview und Spectrum machen mit einem integrierten Sicherungssystem einen vorbildlichen Eindruck. Cricket und Big Sister besitzen zwar kein solches System, sind aber aufgrund der Datenspeicherung in Konfigurationsdateien leicht zu sichern. Schwieriger hingegen ist die Situation beim Operations Manager oder bei Zabbix, die mit den SQL Datenbanken schwieriger zu sichern sind. Nagios speichert die Konfigurationsdaten sowohl als Konfigurationsdatei als auch in einer SQL Datenbank, IBM hingegen macht die effiziente Sicherung aufgrund des auf zahlreiche Subkomponenten verteilten Monitoringsystems besonders schwierig. Die nächste Rubrik befasst sich mit dem Thema **Leistung**. Die Bandbreite wird vom Microsoft Operations Manager besonders effizient genutzt, die Open Source Programme hingegen benötigen für SNMP v3 etwas mehr Bandbreite, von der HP

mit der simpleren SNMP v2 Version weniger benötigt. Bei IBM und CA gibt es keine Angaben bezüglich des Bandbreitenbedarfs. Besonders hardwareintensiv gibt sich Spectrum von CA, die anderen Hersteller machen entweder wie IBM und HP keine Angaben oder verlangen dem Monitoringserver nur geringe Hardwarevorraussetzungen ab. Die Geschwindigkeit der Bedienoberflächen ist in allen Fällen ausreichend, hier können naturgemäß die einfacheren Open Source Programme punkten. Die **Kosten** sind ein weiteres KO Kriterium. Besonders günstig bei den Softwarekosten sind die Open Source Programme, dort fallen keine Lizenzkosten an. Bei Spectrum von CA ist keine Aussage bezüglich der Kosten möglich, die anderen Monitoringsysteme übersteigen das Budget bei weitem. Bei den Installationskosten ist Cricket durch den hohen Programmieraufwand für sämtliche Programmteile abgeschlagen, besonders die kommerziellen Systeme können hier Punkten – Nagios und Zabbix liegen dicht hinter ihnen. Die Wartungskosten sind bei Cricket sehr hoch, wieder bedingt durch die direkten Anpassungen der Einstellungen in den Konfigurationsdateien. Big Sister, Nagios und Zabbix sind bedingt durch eine (teilweise) Webkonfiguration intuitiver und die kommerziellen Programme sind wegen teilweise bereits im Preis inbegriffener Unterstützung und der Webkonfiguration gut bewertet.

Zusammenfassend ist zu sagen, dass jedes der Programme individuelle Vorteile bietet. Für den Einsatz in der gegebenen Umgebung kommen aufgrund der KO Kriterien nur drei Programme in die engere Auswahl, welche namentlich Cricket, BigSister und Nagios sind. Von diesen drei ist Nagios aufgrund der anderen Bewertungskriterien der Favorit und wird deshalb in weiterer Folge als Monitoringprogramm verwendet.

5 Realisierung

Das Kapitel Realisierung beschreibt exemplarisch die Planung, Installation, Konfiguration und die Wartung mit der Monitoringsoftware Nagios. Die folgenden Kapitel sind jedoch nicht als Installationsanleitung zu sehen und streifen nur die zum Verständnis notwendigen Punkte.

5.1 Planung

Die Unterkapitel 5.1.1 bis 5.1.3 befassen sich mit der Planung der Nagios Installation. Dazu ist es in erster Instanz notwendig, alle Geräte und Dienste sinnvoll zu gruppieren um in einem zweiten Schritt Abfragegenauigkeiten und zuletzt die Reaktionszeiten festzulegen.

5.1.1 Servicekatalog

Wurde in Kapitel 3.1 bereits die zu überwachende Soft- und Hardware ermittelt, ist es nun notwendig, einen Servicekatalog zu erstellen, welcher die einzelnen Abfragen anhand der Serveraufgaben gruppiert und ihnen eine Zielerreichbarkeit zuweist. Danach können aus den Zielerreichbarkeitsforderungen die Abfragehäufigkeiten der Gruppen und deren Dienste ermittelt werden. Durch diesen Prozess werden IT-abteilungsintern zu erreichende Servicelevels definiert, welche möglichst genau an die tatsächlichen Erreichbarkeits-erwartungen angepasst sind.

Aufgrund der Aufgabenverteilung ergeben sich Unterteilungen in die Gruppen Mailserver, Dateifreigabeserver, Webserver, Installationsserver, Router, sonstige Geräte und Dienste / SNMP fähig, sonstige Geräte und Dienste / nicht SNMP fähig.

Die bei den einzelnen Servergruppen notwendigen Abfragen unterteilen sich wie in Tabelle 5.1 dargestellt. Diese Tabelle verknüpft die Servergruppen mit der zu überwachenden Soft- und Hardware.

Servertypen / Abzufragende Dienste	Mailserver	Dateifreigabeserver	Webserver	Installationsserver	Router	Sonstige Geräte / SNMP fähig	Sonstige Geräte / nicht SNMP fähig
HP Compaq SMART Arrays		O					
Linux Software RAID		O					
Belegung einzelner Mountpoints	X	X	X	X	X		
Letztes Backup mit Arkeia 5		O					
Band im Bandlaufwerk		O					
Zeitdifferenz zum Zeitserver	X	X			X		
Anzahl der Mails in der Mailqueue	X						
Uptime des Rechners in Tagen	X	X	X	X	X		
Durchschnittslast des Servers	X						
Status des DHCP Daemons		O					
Erreichbarkeit des Webservers	O		X				
Erreichbarkeit des SSH Dienstes	X	X	X	X	X		
Antwort auf ICMP Ping Anfrage	X	X	X	X	X	X	X
SNMP						X	
IMAP	X						
SMTP	X						

X ... Bei allen Servern abzufragen O ... teilw eise abzufragen

Tabelle 5.1: Servicekatalog

5.1.2 Zielerreichbarkeiten

Unter Zuhilfenahme des im vorhergehenden Kapitel angeführten Servicekataloges sind die Zielerreichbarkeiten festzulegen, dazu werden die einzelnen nachstehend angeführten Servergruppen anhand des ES Systems aus Kapitel 2.5.5 eingeteilt. Die Rechnung der Erreichbarkeit wird mit einem vereinfachten Kalendermodell mit 24 Stunden x 7 Tage x 52 Wochen kalkuliert, Feiertage werden nicht berücksichtigt.

- **Mailserver**
  Da es sich bei den Mailservern um zentrale Komponenten für interne und externe Kommunikation handelt, ist eine hohe Erreichbarkeit absolut notwendig. E-Mails können auch außerhalb der Geschäftszeiten eintreffen, weshalb auch in dieser Zeitspanne eine hohe Verfügbarkeit notwendig ist, jedoch ist ein Wartungsintervall von 2 Stunden je Woche vorgesehen. Somit ergibt sich mit 166 Stunden / Woche eine Hauptbetriebszeit von 8632 Stunden / Jahr. Ein Ausfall in dieser Zeitspanne muss innerhalb von 10 Minuten korrigiert werden, dies ergibt eine Nichterreichbarkeit von 0,000057924, die wiederum auf eine ES 8 Einstufung zurückzurechnen ist.
- **Dateifreigabeserver**
  Eine Unterbrechung der Funktionalität einer oder mehrerer Dateifreigabeserver darf zu den Hauptgeschäftszeiten nur äußerst selten vorkommen. Außerhalb der Arbeitszeiten sind Wartungsarbeiten durchaus möglich. Mit etwas zusätzlichem Spielraum ist eine Hauptbetriebszeit von Montag bis Freitag jeweils von 7:00 bis 20:00 Uhr angemessen, dies ergibt hochgerechnet 3380 Stunden im Jahr. Sollte das System nicht erreichbar sein, hat eine Wiederherstellung der Funktionalität in maximal 30 Minuten zu erfolgen. Daraus kann eine ES 6 Einstufung abgeleitet werden.
- **Webserver**
  Die Server für das HTTP Protokoll müssen ebenfalls eine hohe Erreichbarkeit aufweisen, jedoch sind kurze Ausfallzeiten durchaus akzeptabel. Die Hauptbetriebszeit ist gleich mit jener der Dateifreigabeserver. Ein Webserver darf maximal 3 Stunden nicht erreichbar sein, danach muss die Funktionalität wiederhergestellt sein. Umgerechnet auf die Erreichbarkeitsstufen ergibt dies eine Einordnung in die ES 5 Stufe.
- **Installationsserver**
  Installationsserver werden nur benötigt, wenn neue Softwareupdates auf die Arbeitsplatzrechner verteilt werden; eine Unterbrechung der Funktionalität ist also durchaus möglich und unproblematisch. Sollte einer der Server nicht erreichbar sein, werden Aktualisierungen vom Hauptserver geladen, was zwar den Bandbreitenbedarf und die Last des Hauptservers erhöht jedoch sonst keine weiteren Auswirkungen hat. Die Hauptbetriebszeiten sind wieder von Montag bis Freitag zwischen 7:00 und 20:00 Uhr. Die Erreichbarkeit eines Installationsservers muss innerhalb von 8 Stunden wieder gegeben sein, dies hat eine ES 4 Einstufung zur Folge.

- **Router**
  Die Router sind unter anderem für einen zuverlässigen Internetzugang der einzelnen Niederlassungen zuständig. Sollte einer der Router während der Arbeitszeit ausfallen ist es den Mitarbeitern trotzdem möglich niederlassungsintern zu kommunizieren, jedoch können durch längere Ausfälle zum Beispiel Mailwarteschlangen anwachsen und dadurch negative Rückwirkungen auf andere Systeme haben. In der Hauptbetriebszeit von Montag bis Freitag 7:00 bis 20:00 Uhr wird eine maximale Nichterreichbarkeit von 30 Minuten toleriert, daraus folgt eine Einstufung in die ES 6 Kategorie.
- **Sonstige Geräte und Dienste**
  Daher in den Kategorien „sonstige Geräte und Dienste / SNMP fähig“ und „sonstige Geräte und Dienste / nicht SNMP fähig“ verschiedenste Systeme vertreten sind ist es nicht möglich, ihnen eine eindeutige Erreichbarkeitsklasse zuzuordnen, je nach Gerät oder Dienst können unterschiedliche Hauptbetriebszeiten und ES Einstufungen angewandt werden.

5.1.3 Reaktionszeiten

Aufgrund der soeben erfolgten Einteilung in Erreichbarkeitsklassen ist es nun möglich, konkrete Aussagen über die notwendigen Reaktionszeiten und die daraus resultierenden Abfragehäufigkeiten zu tätigen. Selbstverständlich ist das Monitoring der einzelnen Systeme nur eines von zahlreichen Maßnahmen, um die vorgelegte Erreichbarkeitsklasse zu wege zu bringen. Um nun auf die einzelnen Servergruppen und Erreichbarkeitsklassen abgestimmte Abfrageintervalle zu bestimmen, wird die Downtime pro Jahr in zwei Bereiche geteilt:

Der erste und deutlich kleinere Teil ist die Zeit zwischen dem Auftreten des Fehlers und der Anzeige eben dieses Fehlers im Monitoringsystem, also der MTFD.

Der zweite Teil ist die Zeit die dem zuständigen Administrator bleibt, um das Problem zu beheben. Um dem Administrator möglichst viel Zeit zu geben, wird das Verhältnis zwischen den beiden Teilen auf 1:9 festgelegt. Daraus folgt, dass das Monitoringprogramm maximal 10% der zur Behebung eingeräumten Zeit beanspruchen darf.

Durch die Berücksichtigung aller oben genannten Parameter ergeben sich MTFD Werte, welche der Tabelle 5.2 zu entnehmen sind. Das Monitoringsystem hat diese Werte zu erreichen.

System	MTFD
Mailserver	1 Minute
Dateifreigabeserver	3 Minuten
Webserver	18 Minuten
Installationsserver	48 Minuten
Router	3 Minuten
Sonstige Geräte	Nicht anwendbar

Tabelle 5.2: MTFD Zeiten der unterschiedlichen Geräteklassen

Somit ist festgelegt, wie oft das Monitoringsystem zu welchen Zeiten eine Überprüfung durchzuführen hat. Die Tabelle versteht sich als absolutes Minimum, häufigere Test erhöhen selbstverständlich die Genauigkeit und sind durchaus möglich.

5.2 Installation

Dieses Kapitel beschreibt die Installation des Monitoringprogramms Nagios sowie der Konfigurationssoftware Monarch.

5.2.1 Installation von Nagios

Die Installation von Nagios erfolgt auf einem System mit einem Pentium 4 Prozessor und 1 GB RAM von IBM/Lenovo. Als zu Grunde liegendes Betriebssystem kommt die weit verbreitete SuSE Linux Distribution in der Version 9.3 zum Einsatz. Um Nagios voll funktionsfähig zu machen, werden folgende Zusatzprogramme installiert:

- gd, eine Grafikbibliothek
- Apache, ein Webserver
- net-snmp, für SNMP Unterstützung

Es gibt nun die Möglichkeit, entweder Nagios selbst zu kompilieren oder aber ein vorgefertigtes Installationspaket einzuspielen. Da die verwendete Distribution bereits fertige Installationspakete anbietet, wurde letztere Variante gewählt. Mit dem Befehl

```
rpm -i nagios-1.2-76.i568.rpm
```

ist es möglich, eines der benötigten Pakete zu installieren, benötigt werden die folgenden Pakete:

- nagios-1.2-76.i586.rpm
- nagios-www-1.2-76.i586.rpm
- nagios-plugins-1.4-3.i586.rpm
- nagios-plugins-extras-1.4-3.i586.rpm

Anhand dieser Liste ist es bereits möglich, die grobe Struktur von Nagios zu erkennen. Das Paket nagios-1.2 beinhaltet den Nagios Daemon und ist für die zentrale Verarbeitung und Speicherung der Daten verantwortlich, welche es über die Pakete nagios-plugins und nagios-plugins-extras erhält. Das Paket nagios-www wiederum ist für die Ausgabe der Monitoringinformationen, die über einen Webbrowser dargestellt werden können, verantwortlich. Die Pakete erstellen bei ihrer Installation einige wichtige Verzeichnisse, welche für den Betrieb von Nagios von Bedeutung sind. Im Verzeichnis

/etc/nagios/

sind alle für die Konfiguration relevanten Dokumente vorhanden, allen voran die Datei *nagios.cfg*, welche die zentrale Konfigurationsdatei darstellt und auf alle anderen verwendeten Konfigurationsdateien verweist. Für Fehlersuche und Startinformationen legt die Installationsroutine den Ordner

/var/log/nagios/

an, in diesem Ordner werden alle Informationen, Warnungen und Fehlermeldungen gespeichert. Sämtliche Plugins, welche zum Ausführen einzelner Abfragen benötigt werden sind im Verzeichnis

/usr/lib/nagios/plugins/

gespeichert. Die dort vorliegenden Programme können auch unabhängig von Nagios ausgeführt werden und ermöglichen somit das Testen neuer Abfragen noch bevor sie in die Nagios-Programmroutine selbst integriert werden.

5.2.2 Installation von Monarch

Um die Konfiguration des Netzwerkmonitoringsystems zu erleichtern wird zusätzlich zu Nagios das Programm Monarch der Firma IT Groundwork installiert, welches in der Version 1.0 vorliegt. Um den Installationsvoraussetzungen von Monarch zu entsprechen, ist es notwendig, vorab einige Perl Module sowie MySQL zu installieren. Danach ist es möglich mit dem im Installationspaket integrierten Skript alle nötigen Einstellungen vorzunehmen. Sind alle Angaben korrekt angegebenen, ist das Konfigurationsprogramm Monarch über den Browser erreichbar. Monarch besteht aus mehreren Perl-basierten CGI Skripten, welche die Nagios Konfiguration einlesen können. Die Daten werden in eine SQL Datenbank geschrieben und können dort auch bearbeitet werden. Über einen speziellen Befehl ist es möglich, die Konfiguration aus der Datenbank in das Nagios Konfigurationsverzeichnis zurück zuschreiben und Nagios neu zu starten.

5.3 Konfiguration

Nagios ist fertig installiert, nun wird die Konfiguration des Monitoringprogramms vorgenommen. Dazu ist es notwendig, sich vorab mit der Funktionsweise von Nagios vertraut zu machen, um die Zusammenhänge zwischen den einzelnen Konfigurationsdateien und Nagios Komponenten zu verstehen. Daher wird zuerst eine minimale Konfiguration von Hand geschrieben und in einem zweiten Schritt die Konfigruation mit dem Zusatzprogramm MONARCH erläutert. Die Konfiguration wird beispielhaft an einer Netzwerktopologie beschrieben, die aus dem Monitoring Server, einem Router und einem Dateifreigabeserver besteht. Die Abbildung 5.1 verdeutlicht die Topologie samt den dazugehörigen IP-Adressen.

Die Konfiguration welche aus oben stehender Abbildung ersichtlich ist wird von Hand erstellt, die tatsächliche Konfiguration sämtlicher zu überwachender Server wird über das Zusatzprogramm MONARCH vorgenommen, dieses wird im Anschluss besprochen.

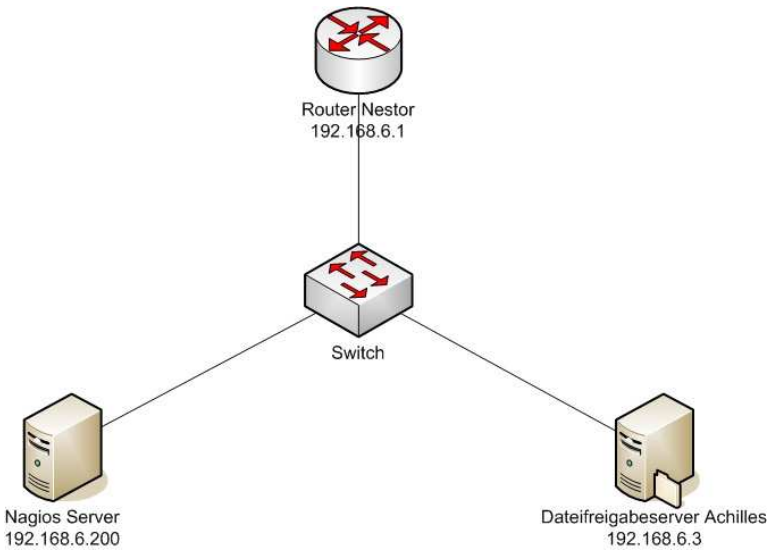

Abbildung 5.1: Beispieltopologie für Monitoringumgebung

5.3.1 Konfigurationsdateien

Die Konfiguration von Nagios geschieht über das Editieren von zahlreichen unterschiedlichen Konfigurationsdateien. Der Nagios Daemon selbst spricht die Hauptkonfigurationsdatei *nagios.cfg* an. Diese Datei enthält Verweise auf die weiteren Konfigurationsdateien. Nagios speichert Konfigurationsinformation in so genannten Objekten, ein Objekt kann zum Beispiel ein Server, ein Kontakt, eine Abfrage usw. sein. Um eine bessere Übersicht zu erreichen, werden unterschiedliche Objekttypen in einzelne Gruppen zusammengefasst und jeder Gruppe eine eigene Konfigurationsdatei zugewiesen. Somit ist es möglich, auch bei umfangreichen Konfigurationen die Übersicht zu behalten.

Nachfolgend werden die für eine Minimalkonfiguration relevanten Dateien besprochen, Ziel der Konfiguration ist es, den Router mit der IP 192.168.6.1 mit einer ICMP Ping Anfrage zu überwachen und den SSH Dienst des Dateifreigabeservers mit der IP 192.168.6.3 auf einen eventuellen Ausfall hin zu beobachten.

In Abbildung 5.2 wird das Konfigurationsschema der einzelnen Nagios Dateien vereinfacht dargestellt, um die Reihenfolge der zu editierenden Dateien zu verdeutlichen.

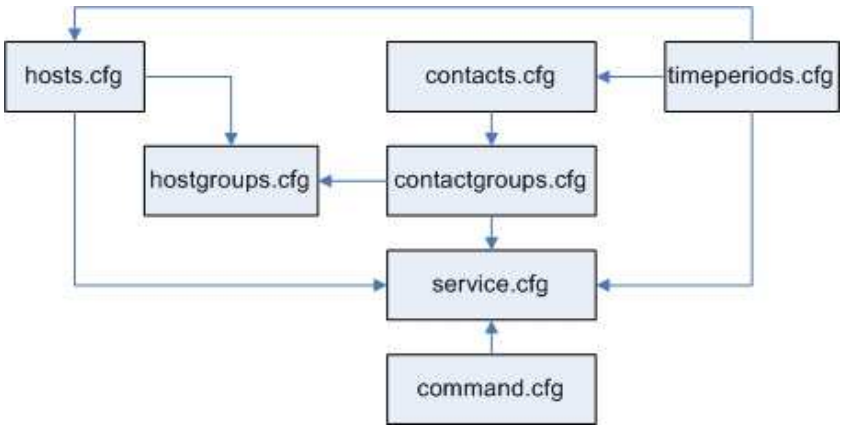

Abbildung 5.2: Vereinfachte Darstellung der Nagios Konfiguration

Wie aus der Abbildung ersichtlich ist, werden im ersten Schritt die beiden Konfigurationsdateien *commands.cfg* und *timeperiods.cfg* bearbeitet, da diese zwei auf keine bereits bestehende Konfigurationsdatei zurückgreifen müssen. Die Konfigurationsdatei *timeperiods.cfg* wird wie folgt verändert:

```
define timeperiod{
        timeperiod_name 24x7
        alias           Rund um die Uhr
        sunday          00:00-24:00
        monday          00:00-24:00
        tuesday         00:00-24:00
        wednesday       00:00-24:00
        thursday        00:00-24:00
        friday          00:00-24:00
        saturday        00:00-24:00
        }
```

Somit ist eine Zeitdefinition mit dem Namen 24x7 festgesetzt, die an jedem Wochentag von 00:00 Uhr bis 24:00 Uhr andauert. Nun werden drei Kommandos festgelegt, die für die Testkonfiguration benötigt werden. Dazu wird in die Datei *command.cfg* folgender Text eingesetzt:

```
define command{
         command_name    ICMP-Ping
         command_line    /usr/lib/nagios/plugins/check_ping -H
         $HOSTADDRESS$ -w 10:20% -c 60:100% -p 3 -t 70
        }
```

```
define command{
         command_name    SSH
         command_line    /usr/lib/nagios/plugins/check_ssh -t 20 -p 22
         $HOSTADDRESS$
        }
define command{
         command_name    mail
         command_line    echo "Fehler auf $HOSTNAME$"|
         /usr/bin/mail -s Nagios -r root@localhost
         nagios@localhost
        }
```

Somit sind die drei Befehle definiert, welche die ICMP Ping Antwort und die Erreichbarkeit des SSH Dienstes überprüfen sowie eine Benachrichtigung verschicken. Bemerkenswert sind in diesem Zusammenhang die Paramerter „-w“ und „-c“, welche Schwellwerte definieren und gleichzeitig auch bewerten. Im Falle der ICMP-Ping Anfrage werden mit „-p 3“ drei ICMP Pakete verschickt, dauert die Antwort im Mittel über 10 Sekunden oder geht mindestens ein Antwortpaket verloren wird eine Warnung an das Nagiossystem weitergeleitet, die Syntax dazu lautet „-w 10:20%“. Nach dem gleichen Schema wird auch ein kritischer Schwellwert mit „-c 60:100%“ festgelegt. Daraus folgt, dass bereits das Plugin entscheidet, wie ein erhaltener Wert zu behandeln ist und der Nagios Daemon diese Bewertung nicht mehr durchzuführen hat. Das letzte Kommando verschickt eine E-Mail mit dem Hostnamen im Textfeld. Als nächster Schritt werden nun Kontakte und Hosts konfiguriert. Die Datei *hosts.cfg* wird wie nachfolgend aufgelistet editiert:

```
define host{
        host_name                       Achilles
        alias                           Dateifreigabeserver-Achilles
        address                         192.168.6.3
        check_command                   SSH
        max_check_attempts              3
        notification_interval           120
        notification_period             24x7
        notification_options            d,u,r
        }
define host{
        host_name                       Nestor
        alias                           Router-Nestor
        address                         192.168.6.1
        check_command                   ICMP-Ping
        max_check_attempts              3
        notification_interval           120
        notification_period             24x7
        notification_options            d,u,r
        }
```

Die Kontakte in der Datei *contacts.cfg* werden wie folgt erstellt, um einen einfachen Kontakt anzulegen:

```
define contact{
        contact_name                    the_kila
        alias                           Alexander Tabakoff
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    n
        host_notification_options       n
        service_notification_commands   mail
        host_notification_commands      mail
        email                           admin@localhost.com
        }
```

Im nächsten Schritt wird eine Kontaktgruppe erstellt. Kontaktgruppen ermöglichen es, mehrere Personen gleichzeitig über einen Ausfall zu benachrichtigen; die Gruppe wird in der Datei *contactgroups.cfg* gespeichert.

```
define contactgroup{
        contactgroup_name               administratoren
        alias                           Alle Administratoren
        members                         the_kila
        }
```

In der Minimalkonfiguration gibt es genau eine Kontaktgruppe, der auch nur eine Person angehört. Als finaler Schritt ist es nun möglich, die Konfiguration der Dateien *hostgroups.cfg* und *services.cfg* einzurichten. Eine Hostgroup wird wie folgt definiert:

```
define hostgroup{
        hostgroup_name                  Ilias
        alias                           alle Testrechner
        contact_groups                  administratoren
        members                         Achilles,Nestor
        }
```

Um die beiden Hosts mit den dazugehörigen Services zu verbinden, wird in die Datei *services.cfg* folgender Inhalt geschrieben:

```
define service{
        host_name                       Achilles
        service_description             SSH-Dienst
        check_period                    24x7
        max_check_attempts              3
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  administratoren
        notification_interval           120
        notification_period             24x7
        notification_options            w,u,c,r
        check_command                   SSH
        }
```

```
define service{
        host_name                       Nestor
        service_description             ICMP Ping Abfrage
        check_period                    24x7
        max_check_attempts              3
        normal_check_interval           5
        retry_check_interval            1
        contact_groups                  administratoren
        notification_interval           120
        notification_period             24x7
        notification_options            w,u,c,r
        check_command                   ICMP-Ping
        }
```

Es gilt des Weiteren zu beachten, dass alle editierten Konfigurationsdateien auch in der zentralen Konfigurationsdatei *nagios.cfg* aufgelistet werden müssen; alle anderen Einstellungen können für die Testinstallation so belassen werden. Mit dem Kommandozeilenaufruf

```
/usr/sbin/nagios -v /etc/nagios/nagios.cfg
```

ist es möglich, die soeben erstellte Konfiguration nochmals zu überprüfen. Befindet sich in der Konfiguration ein Fehler, ist es mit dem oben genannten Befehl leicht möglich diesen zu lokalisieren und zu beheben. Danach kann der Nagios Dienst zum mit dem Kommandozeilenaufruf

/usr/sbin/nagios start

gestartet werden, der Zugriff auf Monitoringinformationen über das Webinterface erstmals möglich. Typischerweise sieht die Oberfläche ähnlich den in Abbildung 5.3 gezeigten Ausschnitt aus.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Achilles	CpuLoad	CRITICAL	05-14-2006 17:51:46	0d 0h 0m 52s	1/3	SNMP CRITICAL - *100* % CPU Last
	RootDisk	WARNING	05-14-2006 17:51:46	35d 20h 26m 4s	3/3	SNMP WARNING - *86* % der Rootdisk belegt
	SSH-Dienst	OK	05-14-2006 17:51:46	35d 14h 52m 4s	1/3	SSH OK - OpenSSH_3.8.1p1 Debian-8.sarge.4 (protocol 2.0)
	SwRaid	OK	05-14-2006 17:51:46	35d 14h 52m 4s	1/3	SNMP OK - 2 RAID-Arrays Online
Nestor	ICMP Ping Abfrage	OK	05-14-2006 17:50:46	35d 12h 21m 52s	1/3	PING OK - Packet loss = 0%, RTA = 1.42 ms

Abbildung 5.3: Überwachte Dienste in der Nagios Weboberfläche, Ausschnitt

Wie aus der Abbildung ersichtlich, werden die zwei Rechner Achilles und Nestor überwacht. Der Rechner Achilles hat seit 52 Sekunden 100 prozentige CPU Last, was als kritisch eingestuft wird. Aufgrund der Festplattenauslastung von 86% wird eine Warnung angezeigt, alle anderen Dienste haben kein Problem gemeldet. Durch die farbliche Kennzeichnung können Probleme besonders schnell und intuitiv erkannt werden.

5.3.2 Implementierung der geforderten Funktionen

Im folgenden Kapitel wird die Implementation der in Kapitel 3.1 - „Zu überwachende Hard- und Software“ beschriebenen Anforderungen an die Monitoringsoftware erläutert. Um die entsprechenden Abfragen durchführen zu können gibt es für Nagios grundsätzlich mehrere verschiedene Möglichkeiten, diese sind nachfolgend kurz beschrieben:

- **NRPE**
  Der Begriff NRPE steht für Nagios Remote Plugin Executor und definiert ein Addon für Nagios, das lokale Plugins auf einem entfernten Rechner ausführen kann. NRPE bietet keinerlei Möglichkeiten zur Verschlüsselung oder Authentifizierung, was diese Variante aufgrund der Anforderungen aus Kapitel 3.6 für den Einsatz disqualifiziert. [49]
- **NSCA**
  Der Nagios Service Check Acceptor oder kurz NSCA ermöglicht es passive Dienstüberprüfungen durchzuführen und an einen zentralen Nagios Server weiterzuleiten. Das Programm bietet Verschlüsselung an, jedoch ist die Zuverlässigkeit des Systems aufgrund der Einwegkommunikation schwer abzuschätzen. [49]
- **SNMP**
  Eine weitere Möglichkeit die geforderten Abfragen zu realisieren besteht über das Auslesen von MIBs via SNMP. Jedoch ist es durchaus möglich, das weder standardisierte noch herstellerspezifische MIBs die gewünschte Funktionalität bieten. Im Gegensatz zu NRPE oder NSCA müssen auf den zu überwachenden Servern bei der SNMP Lösung keinerlei Programme installiert werden, da SNMP Unterstützung bereits vorhanden ist.

Keiner der soeben vorgestellten Ansätze kann zu 100 Prozent überzeugen, was zu der Idee führt, eine individuelle Lösung zu verwenden die einen kombinierten Ansatz aus standardisierten Protokollen und selbst implementierten Abfragen bietet.

Die Abbildung 5.4 verdeutlicht das Konzept an einer beispielhaften Monitoringabfrage.

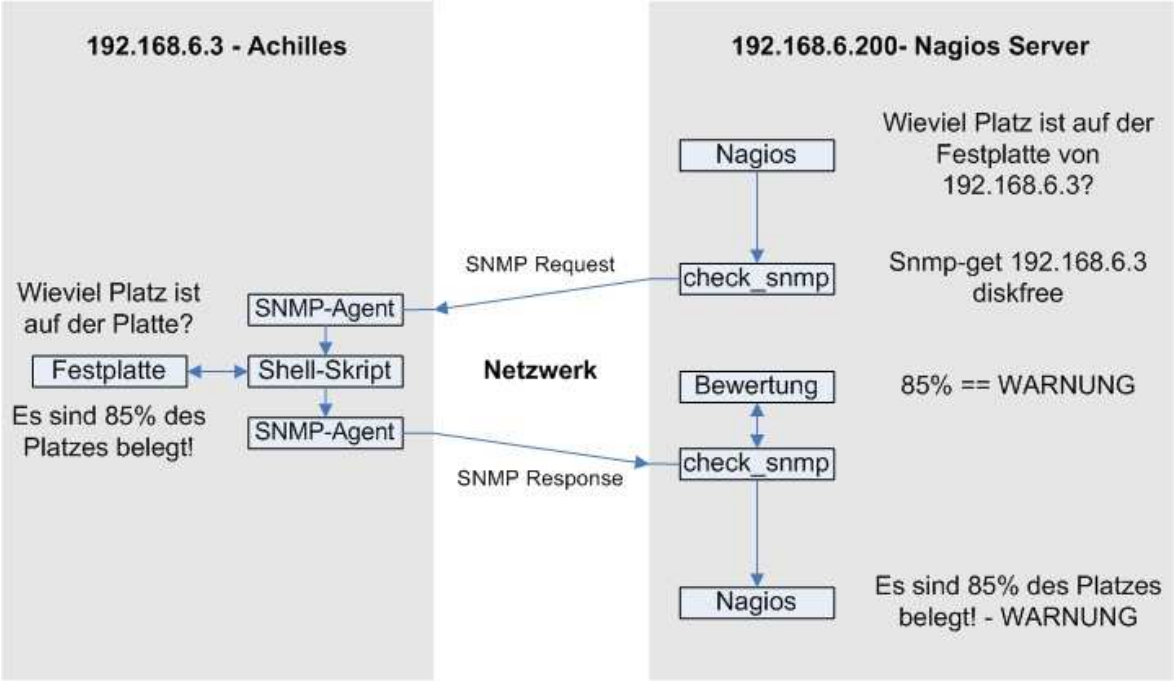

Abbildung 5.4: Nagios Monitoringabfragen

Im ersten Schritt leitet der Nagios Prozess am Nagios Server eine Anfrage an das SNMP Plugin check_snmp weiter, Ziel dieser Abfrage ist es den Festplattenfüllstand des Servers Achilles zu erfahren. Das Plugin konvertiert die Abfrage in eine SNMP Anfrage und schickt sie über das Netzwerk an den SNMP Agent am entfernten Rechner. Anstatt eine klassische MIB abzufragen startet der SNMP Agent jedoch ein Shell-Skript, welches die gewünschte Abfrage am Dateifreigabeserver durchführt, bezugnehmend auf die Abbildung meldet die Festplatte eine Auslastung von 85%. Das Shell-Skript meldet diesen Wert dem SNMP-Agent zurück, welcher die Nachricht wieder zurück an das SNMP Plugin des Nagios Servers schickt. Das Plugin wertet nun den erhaltenen Wert aus, ordnet ihn in die korrekte Kategorie ein (in diesem Fall WARNUNG) und gibt dieses Ergebnis wieder an den Nagios Prozess zurück, welcher sich nun um die Darstellung des Ergebnisses kümmern kann.

Um die geforderte verschlüsselte und authentifizierte Übertragung der Monitoring-informationen zu implementieren, wird SNMP in der Version v3 verwendet. Die Einrichtung des SNMP Daemons unter dem SuSE Linux Testsystems erfolgt über das Editieren zweier Konfigurationsdateien. In die Datei */etc/snmpd.conf* wird folgender Inhalt geschrieben:

```
rouser nagios priv
exec .1.3.6.1.4.1.24763.1 softwareraid /usr/local/sbin/snmp-
     check.sh -1
exec .1.3.6.1.4.1.24763.2 rootdisk /usr/local/sbin/snmp-
     check.sh -2
exec .1.3.6.1.4.1.24763.3 uptime /usr/local/sbin/snmp-
     check.sh -3
```

Mit diesen Einträgen in die Konfigurationsdatei wird ein Benutzer „nagios“ erstellt, welcher einen reinen Lesezugriff auf die nachfolgenden Befehle hat und diese auch nur verschlüsselt und authentifiziert abfragen darf. Es wurden drei Befehle erstellt, die über die OIDs .1.3.6.1.4.1.24763.1 bis .1.3.6.1.4.1.24763.3 abrufbar sind und auf die ausführbare Datei */usr/local/sbin/snmp-check.sh* mit unterschiedlichen Aufrufargumenten verweisen. Die oben genannte OID befindet sich im Bereich der so genannten Private Enterpise Numbers, was einen Konflikt mit anderen MIBs verhindert. Um den Benutzer „nagios“ vollständig anzulegen, wird eine Datei erstellt, die sich unter */var/lib/net-snmp/snmpd.conf* befindet. Folgende Zeile wurde dort eingefügt:

```
createUser nagios MD5 passwort1 DES passwort2
```

Somit wird er oben bereits verwendete Benutzer „nagios“ mit einem Passwort für die Authentifizierung und einem für die Verschlüsselung versorgt, dazu wurden die Algorithmen DES und MD5 verwendet. Sind die Einstellungen vorgenommen, muss als nächstes die Datei /usr/local/sbin/snmp-check.sh erstellt werden. Der Inhalt der Datei lautet wie folgt:

```
#!/bin/bash
while getopts 123 argumente
do
case $argumente in
#Softwareraid
1) cat /proc/mdstat| grep -c "UU";
        ;;
#Belegung rootdisk
2) df -l |awk '{print $5}'|awk -F % '{print $1}'|head -n 2|
     tail -n 1;
        ;;
#Uptime
3) cat /proc/uptime|awk '{print $1/60/60/24}'|awk -F .
     '{print $1}';
        ;;
esac;
done
exit 0;
```

In dieser Datei werden die drei Abfragen als Shellskript realisiert. Dieses Skript benötigt einen Parameter, welcher die gewünschte Routine aufruft. Wird das Skript mit

/usr/local/sbin/snmp-check.sh -1 aufgerufen, wird der Inhalt der Datei */proc/mdstat* ausgelesen um nach der Buchstabenkombination „UU“ zu suchen. Diese Buchstabenkombination gibt an, dass sich ein Linux Software RAID1 im Normalzustand mit zwei vollständig funktionsfähigen Festplatten befindet. Ein Aufruf des Skiptes mit */usr/local/sbin/snmp-check.sh -2* veranlasst das Programm df die Belegung aller lokalen Festplatten anzuzeigen um danach die Belegung des root Mountpoints in Prozent auszugeben, also zum Beispiel 61 für eine 61% Belegung des root Dateisystems. Durch das Ausführen des Kommandos */usr/local/sbin/snmp-check.sh -3* wird der Inhalt der im Verzeichnis */proc/* liegenden Datei *uptime* ausgelesen, welche die Betriebszeit des Rechners in Sekunden angibt. Danach wird die Betriebszeit in Tage umgerechnet und ausgegeben.

Um die Abfragen in Nagios zu integrieren ist es notwendig, neue Kommandos anzulegen und diese einem Server zuzuordnern. Dazu wird das Nagios Plugin „check_snmp“ verwendet. Ein Kommandozeilenaufruf sieht wie folgt aus:

```
/usr/lib/nagios/plugins/check_snmp –H $HOSTADDRESS$ –o
.1.3.6.1.4.1.24763.1.101.1 –u "RAID-Arrays Online" –c 2:99 –t 9 –P 3 –L
authPriv –a MD5 –A passwort1 –U nagios –X passwort2
```

Der Befehl „$HOSTADDRESS$“ wird von Nagios mit der IP-Adresse des zu überprüfenden Servers ersetzt, die OID entspricht der OID in der SNMP Daemon Konfiguration, die zusätzlichen Nummern „.101.1“ erlauben den direkten Zugriff auf den Antwortwert. Mittels der Option „-u“ ist es möglich, dem Output zusätzliche Informationen mitzugeben, die auch in der Monitoringoberfläche angezeigt werden und eine leichtere Interpretation der Ergebnisse zulassen. Mit dem Kommando „-c 2:99“ wird ein Bereich zu definiert, in dem das Monitoringprogramm keine kritische Warnmeldung ausgibt. Es ist nun möglich, das Kommando der Nagios Konfiguration hinzuzufügen, indem es wie in Kapitel 5.3.2 erläutert, eingebracht wird.

Sämtliche zu überwachende Soft- und Hardware ist über Abfragen ähnlich den oben genannten Überprüfungen realisiert, Details zur Implementierung der einzelnen Abfragen sind dem Anhang A zu entnehmen.

5.3.3 Erreichen der geforderten Reaktionszeiten

Um mit Nagios die in Kapitel 5.1.3 geforderten Reaktionszeiten zu erreichen ist es notwendig einige Anpassungen an die Abfragehäufigkeiten von einzelnen Diensten vorzunehmen.

Die kürzeste Reaktionszeit wird vom Mailserver Monitoring verlangt, wie in Punkt 5.1.3 bereits erwähnt bleiben zwischen Auftreten des Fehlers und der Anzeige im Monitoringprogramm im Schnitt nur 60 Sekunden. Der Ablauf einer Monitoring-überprüfung teilt sich in mehrere Phasen ein. Vorab ist es daher notwendig, alle Phasen aufzulisten um ihnen danach entsprechende Zeitfenster zuzuordnen. Die einzelnen Phasen sind in Abbildung 5.4 aufgeführt.

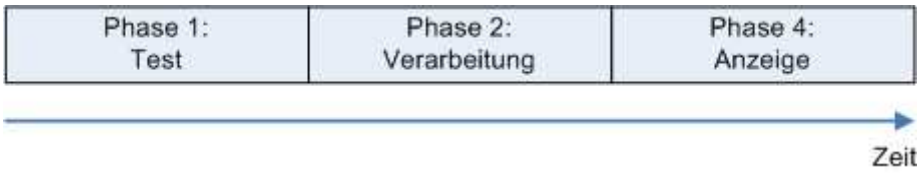

Abbildung 5.5: Bearbeitungsphasen einer Abfrage

Die **Testphase** wird bestimmt durch jene Parameter, die einem Nagios Plugin mitgegeben werden. Fast allen Plugins können Zeitwerte angegeben werden bei deren Überschreitung die Abfrage abgebrochen wird und ein kritischer Status an Nagios weitergeleitet wird. Typischerweise ist der Timeout Wert mit 4 Sekunden zu belegen.

In der nächsten Phase wird das Ergebnis der Abfrage **verarbeitet**. Die Interpretation wird bereits vom Plugin durchgeführt; dem Nagios Hauptprozess kann über den Parameter „Service reaper frequency“ mitgeteilt werden, in welchen Abständen die Ergebnisse der Service Überprüfungen durchgeführt werden müssen. Die Verarbeitung kann jede Sekunde neu angestoßen werden, da sie nur Zeiten im Millisekundenbereich in Anspruch nimmt.

Die Aktualisierung der **Anzeige** wird durch den in der Konfigurationsdatei *cgi.cfg* definierten Parameter „refresh rate“ beeinflusst, der Wert wird in Sekunden angegeben und steuert, wie oft der Browser die Anzeigewebseite neu lädt. Ist dieser Wert zu klein, kann es passieren, dass eine Aktualisierung angefordert wird noch bevor der Seitenaufbau der letzten Aktualisierung abgeschlossen ist. Eine Rate von 10 Sekunden ist in anbetracht der Probleme bei zu häufiger Aktualisierung als Minimum anzunehmen. Werden die drei Komponenten zeitlich summiert, ergibt sich eine Reaktionszeit von 15 Sekunden.

Im Idealfall, das heißt wenn das Problem genau zum Zeitpunkt des Tests auftritt, ergibt sich somit eine rund 15 sekündige Latenz. Wird der Test selbst alle 60 Sekunden wiederholt, ergibt sich ein Worst Case Szenario, bei dem es 75 Sekunden dauert, bis der Fehler im Monitoringprogramm angezeigt wird. Als Mittelwert zwischen den zwei Extremen ergibt sich somit eine MTFD[8] von 45 Sekunden, was im Rahmen der Anforderungen aus Punkt 5.1.3 liegt. Anhand dieser Daten ist es nun möglich, die Überprüfungshäufigkeiten der anderen Servergruppen entsprechend zu berechnen. Die Router und Dateifreigabeserver haben eine MTFD Zeit von 3 Minuten zu erreichen. Die Parameter „refresh Rate“ und „Service reaper frequency“ sind global einzustellen und dürfen daher nicht mehr verändert werden, daher bleibt nur die Möglichkeit, den Timeout anzupassen; er wird um 5 Sekunden erhöht und somit ergibt sich eine Gesamtzeit von 20 Sekunden, welche im Idealfall benötigt wird um eine Störung anzuzeigen. Um den geforderten Mittelwert von 180 Sekunden zu erreichen, darf es die maximale Zeit zwischen zwei Abfragen 340 Sekunden betragen, was rund 5,5 Minuten entspricht. Die Webserver haben eine MTFD von 18 Minuten. Durch Anwendung der selben Rechnung wie bei den Web- und Dateifreigabeservern erhält man eine Abfragehäufigkeit von rund 35,5 Minuten. Bei den Installationsservern wurde eine MTFD von 48 Minuten gefordert, die geringen Latenzzeiten der Monitoringsoftware sind hier nicht mehr relevant und es ergibt sich eine minimal zu erreichende Abfragehäufigkeit von rund 1 ½ Stunden.

5.4 Optimierung und Wartung

Nagios und Monarch bieten eine Reihe von Möglichkeiten, das Monitoring zu optimieren und die Konfiguation zu erleichtern, nachfolgend werden diese Optimierungs-möglichkeiten näher beleuchtet

5.4.1 Anpassung der Abfrageintervalle

In Kapitel 5.3.3 wurde bereits errechnet, welche Abfrageintervalle notwendig sind, um die Vorgaben für die einzelnen Servergruppen zu erreichen. Es ist nun notwendig die Abfragen so einzustellen, damit einerseits das Netzwerk nicht unnötig belastet wird und andererseits keine Einschränkungen bei der Aktualität der Monitoringdaten hingenommen werden müssen. Um dies zu gewährleisten, wird an zwei unterschiedlichen Punkten angesetzt:

8 siehe Kapitel 2.4

Als erste Maßnahme werden mehrere Zeiträume für Abfrageintervalle definiert. In der Grundkonfiguration von Nagios wurde nur der Zeitraum „24x7“ definiert, jedoch ist die IT-Abteilung der Firma Alpine-Mayreder Bau GmbH nicht rund um die Uhr im Einsatz. Deshalb werden zwei Zeiträume festgelegt, in denen die Abfragen in unterschiedlicher Intensität durchgeführt werden und den Definitionen der Haupt- und Nebenbetriebszeit aus Kapitel 5.1.2 entsprechen. Die Zeiträume sind wie folgt definiert:

```
define timeperiod {
          timeperiod_name         Arbeitszeiten
          alias                   07-20Uhr
          monday                  07:00-20:00
          tuesday                 07:00-20:00
          wednesday               07:00-20:00
          thursday                07:00-20:00
          friday                  07:00-20:00
}
define timeperiod {
          timeperiod_name         Freizeit
          alias                   20-07Uhr
          monday                  00:00-07:00,20:00-24:00
          tuesday                 00:00-07:00,20:00-24:00
          wednesday               00:00-07:00,20:00-24:00
          thursday                00:00-07:00,20:00-24:00
          friday                  00:00-07:00,20:00-24:00
          saturday                00:00-24:00
          sunday                  00:00-24:00
}
```

Ein Spezialfall ist auch die Abfrage aus Kapitel 3.1.5, welche das Vorhandensein eines Bandes im Bandlaufwerk überprüft. Das Band wird vom Backupprogramm automatisch ausgeworfen, wenn die Sicherung erfolgreich durchgeführt wurde. Jede Niederlassung hat einen Sicherungsbeauftragten, welcher das ausgeworfene Band durch ein anderes ersetzen muss. Dies bedeutet aber, dass „kein Band im Bandlaufwerk“ bis zu einem gewissen Zeitpunkt vollkommen unbedenklich ist, da die Sicherung außerhalb der Arbeitszeiten stattfindet. Aufgrund dessen wird eine eigene Zeitperiode für die Sicherung wie nachfolgend aufgelistet definiert:

```
define timeperiod {
          timeperiod_name         Sicherung
          alias                   14-18Uhr
          monday                  14:00-18:00
          tuesday                 14:00-18:00
          wednesday               14:00-18:00
          thursday                14:00-18:00
          friday                  14:00-18:00
}
```

Das Zeitfenster wurde so gewählt, da es vorher durchaus vorkommen kann, dass der Sicherungsbeauftragte das Band einfach noch nicht gewechselt hat; nach 18:00 Uhr ist der Sicherungsbeauftragte in den meisten Fällen nicht mehr zu erreichen.

Als zweite Maßnahme ist es möglich, den einzelnen Abfragen unterschiedliche Abfragehäufigkeiten zuzuordnen; in Kombination mit den Zeitperioden ist es somit möglich, Abfragen nicht nur „ganz oder gar nicht“ durchzuführen sondern die Intensität der Abfragen an die Anforderungen anzupassen. Nagios bietet neben der Angabe des Intervalls noch zwei weitere Parameter für die Überprüfung eines Dienstes an. Der Parameter „Max check attempts“ gibt an, wie oft eine Überprüfung wiederholt werden soll, wenn das Ergebnis der Überprüfung kein „OK“ zurück liefert, der zweite Parameter wird „retry check interval“ genannt und definiert, wie häufig ein Dienst der kein „OK“ zurück liefert abgefragt wird; der letzte Parameter hat daher typischerweise einen kleineren Wert als der normale Intervall. Die Tabelle 5.3 stellt den Zeitintervallen die Abfragen gegenüber und beschreibt im Schnittpunkt jeweils die drei oben erläuterten Werte.

Zeiten / Abfrage	Arbeitszeit			Freizeit			Backup			24x7		
	N	F	#	N	F	#	N	F	#	N	F	#
HP Compaq SMART Array	60	5	10									
Linux Software RAID	60	5	10									
Belegung einzelner Mountpoints	120	10	10									
Letztes Backup mit Arkeia 5							20	10	5			
Band im Bandlaufwerk							20	10	10			
Zeitdifferenz zum Zeitserver										1400	30	10
Anzahl der Mails in der Mailqueue	6	2	3	10	2	3						
Uptime des Rechners in Tagen										1400	5	10
Durchschnittslast des Rechners	10	5	4									
Status des DHCP Daemons										120	4	5
Erreichbarkeit des Webservers	30	10	2									
Erreichbarkeit des SSH Dienstes										1400	10	139
Antwort auf ICMP Ping Anfrage	2	1	5	60	5	3				10	5	3
SNMP	5	2	5	30	10	3						
IMAP	5	1	4									
SMTP	30	10	2									

N ... Normalintervall [min], F ... Fehlerintervall [min], # ... Anzahl der Versuche bei Fehler

Tabelle 5.3: Überprüfungsintervalle der einzelnen Befehle

5.4.2 Anpassung der Schwellwerte

Ein wesentlicher Schritt zur Vermeidung von Fehlalarmen ist es, die Schwellwerte optimal an die tatsächlichen Gegebenheiten anzupassen. Wie bereits erwähnt gibt es mehrere Zustände, in denen sich ein Nagios Service befinden kann: „Ok“, „Warning“ und „Critical“ sind die wichtigsten, hinzu kommen noch die Spezialzustände „Pending“ und „Unknown“.

Die Zustände „Pending“ und „Unknown“ können nicht direkt beeinflusst werden, jedoch ist es mit der geschickten Wahl der richtigen Werte möglich, false positive und false negative Meldungen zu vermeiden, indem die Schwellwerte für die Zustände „Warning“ und „Critical“ optimal eingestellt werden. Eine strukturierte Vorgehensweise zur Ermittlung der entsprechenden Werte ist nur schwer möglich, weshalb Erfahrungswerte angenommen werden, die im täglichen Einsatz überprüft und gegebenenfalls angepasst werden. Die Tabelle 5.4 zeigt die optimierten Schwellwerte für alle verwendeten Kommandos.

Abfrage \ Schwelle	Warning	Critical
HP Compaq SMART Array		> 0 Fehlermeldungen
Linux Software RAID		> 0 Fehlermeldungen
Belegung einzelner Mountpoints	> 70% belegt	> 95% belegt
Letztes Backup mit Arkeia 5	> 1 Tag	> 4 Tage
Band im Bandlaufwerk	nicht eingelegt	
Zeitdifferenz zum Zeitserver	> 60 Sek. Differenz	
Anzahl der Mails in der Mailqueue	> 9 Mails	> 29 Mails
Uptime des Rechners in Tagen	< 2 oder > 299 Tage	
Durchschnittslast des Rechners	> 4 Jobs/Sek.	> 10 Jobs/Sek.
Status des DHCP Daemons		DHCP läuft nicht
Erreichbarkeit des Webservers	> 1 Sek. Reaktionszeit	> 3 Sek. Reaktionszeit
Erreichbarkeit des SSH Dienstes	> 2 Sek. Reaktionszeit	> 5 Sek. Reaktionszeit
Antwort auf ICMP Ping Anfrage	> 1,5 Sek. Reaktionszeit oder 1 von 2 Antworten erhalten	> 5 Sek. Reaktionszeit oder 0 von 2 Antworten erhalten
SNMP	unterschiedlich	unterschiedlich
IMAP	> 2 Sek. Reaktionszeit	> 3 Sek. Reaktionszeit
SMTP	> 1 Sek. Reaktionszeit	> 3 Sek. Reaktionszeit

Tabelle 5.4: Schwellwerte der einzelnen Abfragen

5.4.3 Host- und Serviceprofile

Bedingt durch die Größe des Netzwerkes ist es häufig notwendig, ausgediente Server aus dem Monitoringsystem zu entfernen und neue Server diesem hinzuzufügen. Es ist jedoch auch möglich, dass einem bestehenden Server eine neue Rolle zugewiesen wird. Um diese Anpassungen schnell und effizient durchführen zu können, ist es möglich Host- und Serviceprofile zu erstellen. Die notwendigen Schritte werden anhand des Hostprofiles für einen Webserver verdeutlicht.

Im ersten Schritt müssen alle benötigten Kommandos für die Abfragen definiert werden, für einen Webserver sind dies laut dem in Kapitel 3.3 definierten Servicekatalogs:

- Belegung der Festplatte
- Uptime des Rechners in Tagen
- Erreichbarkeit des Webservers
- Erreichbarkeit des SSH Dienstes
- Antwort auf ICMP Ping Anfrage

Sind die Abfragen implemetiert (Details siehe Anhang A) wird jeder Anfrage ein Service Template zugeordnet, dies geschieht in der Monarch Administrationsoberfläche im Menüpunkt Services. Es ist jedoch möglich und sinnvoll, ein Service Template mehreren Anfragen zuzuordnen, so diese Anfragen identische Service Parameter besitzen. Jedem Kommando mit dazugehörigem Template ist nun ein eindeutiger Service Name zuzuordnen. Im Anschluss ist es möglich, über den Menüpunkt Profiles ein neues Service Profil zu erstellen. Einem Service Profil können wiederum mehrere Service Names angehören. Als Name für das Service Profil wurde die Bezeichnung SP.Webserver gewählt. Im nächsten Schritt kann ein Host Template erstellt werden, was im Menüpunkt Hosts möglich ist. In einem Host Template können unter anderem Benachrichtigungsoptionen angegeben werden. Als letzter Schritt zu Erstellung eines Profiles für einen Webserver ist es nun möglich, ein Service Profil und ein Host Template zu einem Host Profil zu vereinen.

Einige exemplarische Hosttemplates und Serviceprofile sind in Anhang B beziehungsweise Anhang C zu finden.

5.4.4 Eskalationen und Benachrichtigungen

Mit Benachrichtigungen ist es möglich, Kontakte oder Kontaktgruppen über ein anderes Medium als dem Web über aufgetretene Probleme zu informieren. So ist es zum Beispiel möglich, einen Administrator mit einer Kurzmitteilung, welche auf sein Mobiltelefon geschickt wird über den Ausfall eines Webservers zu informieren. Es ist möglich, Benachrichtigungen für Services und Hosts zu definieren, jedoch sind zahlreiche Filter zu passieren, bevor eine Benachrichtigung abgeschickt wird; genauere Informationen dazu liefert die Nagios Dokumentation unter [49].

Mit Eskalationen ist es möglich, bestimmte Kontakte zu benachrichtigen, sollte ein Problem längere Zeit andauern oder der ursprünglich benachrichtigte Kontakt nicht reagieren. Da solche Eskalationen nicht verwendet werden sind sie an dieser Stelle auch nicht weiter ausgeführt.

5.4.5 Weitere Optimierungsmöglichkeiten

Eine sehr einfache aber effektive Möglichkeit über den Ausfall eines Servers oder Dienstes zu benachrichtigen ist es, in der Konfigurationsdatei cgi.cfg Warntöne beim Auftreten eines Problems zu definieren; die entsprechenden Parameter lauten „Host down sound“, „Service critical sound“ usw. und können mit beliebigen Tönen und Melodien versehen werden.

Eine weitere Funktionalität bringt Nagios mit der Vermeidung von Fehlalarmen über Host Dependencies und Service Dependencies mit. Mit Host Dependencies ist es möglich, die Struktur des Netzwerkes nachzubilden, um beim Ausfall eines Netzwerkknotens die nicht mehr erreichbaren Server im vom Restsystem abgeschnittenen Teilnetz auszublenden. In Nagios wird eine Abhängigkeit mit dem Parameter „parents“ in der Host Konfigurationsdatei definiert. Mit Service Dependencies werden Abhängigkeiten zwischen verschiedenen Diensten zu modellieren, es ist auch möglich, dass sich diese Abhängikeiten über unterschiedliche Hosts hinweg erstrecken.

Sämtliche Host Dependencies sind in Nagios eingearbeitet, für Service Dependencies gibt es zur Zeit keine konkreten Anwendungsfälle weshalb sie nicht implementiert sind.

6 Schlussfolgerungen und Ausblicke

Die Implementation eines Netzwerkmonitoring Systems ermöglicht es den Administratoren der Firma Alpine-Mayreder Bau GmbH erstmalig

- Automatisiert Ausfälle zu erkennen bevor dies die Anwender bemerken
- Frühzeitig gefährliche Situationen auszumachen bevor sie zu echten Problemen werden
- Intern zu erreichende Zielerreichbarkeiten zu überwachen und
- Das Firmennetzwerk inklusive Statusinformationen übersichtlich darzustellen.

Die Wahl von Open Source Programmen wie Nagios zum Netzwerkmonitoring, IT Groundwork Monarch zur Administration von Nagios, Linux als Betriebssystem oder auch Apache als Webserver ermöglicht es sowohl die Kosten für die Implementation und auch die der Wartung gering zu halten.

Durch die Abfrage sämtlicher Monitoringdaten über SNMP v3 wird eine zukunftssichere und kryptographisch akzeptable Lösung verwendet, welche eine Neuimplementation oder Umstellung des Monitoringsystems wesentlich erleichtert, sobald dies nötig ist.

Die Administration des Monitoringsystems wird durch den Einsatz von IT Groundwork Monarch wesentlich vereinfacht und ermöglicht es den Verantwortlichen, schnell und ohne genaue Kenntnisse der Nagios Konfigurationsdateien Syntax Anpassungen durchzuführen.

Es gibt zahlreiche Verbesserungsmöglichkeiten, die das Monitoringprogramm erweitern oder ergänzen können, eines der Ziele dieser Vorschläge ist dabei natürlich die weitere Verkürzung der Ausfallzeiten, aber auch erleichterungen in der Konfiguration sind denkbar. Das Monitoringprogramm selbst, Nagios, ist mittlerweile in der Version 2.2 erhältlich (Stand: 17.04.2006) und bietet zahlreiche Detailverbesserungen, welche das Programm noch flexibler und schneller machen. Interessante Aspekte der neuen Version sind unter anderem die Möglichkeit „Adaptives Monitoring“ durchzuführen, was bedeutet, dass Anpassungen an gewissen Parametern wie Intervallen oder Kommandos zur Laufzeit des Monitoringprogrammes durchführbar sind.

Aufgrund der anpassungsfähigen Abfrage mithilfe von SNMP liegt es natürlich nahe, weitere Abfragen in das Monitoringsystem aufzunehmen, als Beispiele seien genannt:

- Überwachung der Temperaturen von Serverräumen, Festplatten oder Prozessoren
- Beobachtung von Tinten- und Tonerfüllständen in Druckern
- Kontrolle von Lüfterdrehzahlen
- Auslesen von SMART Festplatteninformationen

Natürlich bietet sich auch die Möglichkeit an, weitere Stufen der Problembehebung wie zum Beispiel die Ursachenforschung in Nagios einzubinden. Das Programm Splunk der gleichnamigen US amerikanischen Firma bietet eine Software zum effizienten Durchsuchen von Logfiles an, welches auch mit Nagios zusammenarbeitet [50]. Die Integration einer Informationsdatenbank nach Vorbild eines Wiki oder den Microsoft Knowledgebase Artikeln in Nagios ist eine weitere Option, welche die Geschwindigkeit der Problemlösung weiter erhöhen kann und somit das Potential besitzt die Ausfallzeiten weiter zu minimieren [51].

Abschließend bleibt zu vermerken, dass durch die Implementation der Monitoringsoftware bei der Firma Alpine-Mayreder Bau GmbH Ausfälle früher erkannt und beseitigt werden können und somit die IT Abteilung entlastet wird, was wiederum freie Kapazitäten schafft und gleichzeitig bedingt, durch die höhere Erreichbarkeit der Dienste, auch eine höhere Gesamtproduktivität der Firma ermöglicht.

Literaturverzeichnis

[1] ITU-T Study Group VII (1993) *Management Framework for Opensystems Interconnection (OSI) for CCITT Applications*. ITU-T X.700

[2] Stallings, W. (1996) *SNMP SNMPv2 and RMON*, 2nd Edition. Addison Wesley, Reading, Massachusetts.

[3] Stein, E. (2001) *Taschenbuch Rechnernetze und Internet*. Fachbuchverlag Leipzig, Kösel, Kempten.

[4] Intel et al. (2004) *Intelligent Platform ManagementInterface SpecificationSecond Generation*. IPMI Ver 2.0 Rev 1.0

[5] Distributed Management Task Force, Inc. (2006) *DTMF - Web-Based Enterprise Management (WBEM)*. http://www.dmtf.org/standards/wbem (28. April 2006).

[6] Lonvick, C. (2001) *The BSD syslog Protocol*. IETF RFC 3164

[7] Schwenkler, T. (2006) *Sicheres Netzwerkmanagement*. Springer-Verlag, Heidelberg.

[8] Tanenbaum, A. (2000) *Computernetzwerke*, 3. revidierte Auflage. Paerson Studium, München.

[9] Case, J. (1993) *Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)*. IETF RFC 1442

[10] Freed, N. (2000) *Mail Monitoring MIB*. IETF RFC 2789

[11] Kurose, J. and Ross, K. (2003) *Computer Networking*, Second Edition. Pearson Education, United States of America.

[12] Case, J. (1990) *A Simple Network Management Protocol*. IETF RFC 1157

[13] Blumenthal, U. (2002) *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*.

[14] Waldbusser, S. (2000) *Remote Network Monitoring Management Information Base*. IETF RFC 2819

[15] Walsbusser, S. (1997) *Remote Network Monitoring Management Information Base Version 2 using SMIv2*. IETF RFC 2021

[16] Schira, J. (2003) *Statistische Methoden der VWL und BWL*. Pearson Studium, Kösel, Kempten.

[17] Office of Government Commerce (2006) *OGC - Frequently Asked Questions*. http://www.ogc.gov.uk (3. Mai 2006).

[18] Victor, F. und Günther, H. (2005) *Optimiertes IT-Management mit ITIL*, 2. Auflage. vieweg Verlag, Wiesbaden.

[19] Garaus, T. (2004) *Österreichisches Sicherheitshandbuch v2.2*. http://www.cio.gv.at/securenetworks/sihb/OE-IT-SIHB_V2_2_Teil1.pdf (3. März 2006).

[20] Bundesamt für Sicherheit in der Informationstechnik (2006) *Das Leitbild des Bundesamt für Sicherheit in der Informationstechnik*. http://www.bsi.de/bsi/leitbild.htm (29. Mai 2006).

[21] Bundesamt für Sicherheit in der Informationstechnik (2005) *IT-Grundschutz-Kataloge 2005*. http://www.bsi.de/gshb/deutsch/download/itgshb_2005.pdf (12. März 2006).

[22] Held, A. (2005) *Hochverfügbarkeit: Kannzahlen und Metriken*. www.tecchannel.de/index.cfm?webcode=430342 ().

[23] Hewlett-Packard Development Company, L.P. (2006) *HP.com - Smart Array controllers*. http://h18004.www1.hp.com/products/servers/proliantstorage/arraycontrollers/?jumpid=reg_R1002_USEN (15. März 2006).

[24] Behlert, S. et al. (2005) *SUSE Linux Administrationshandbuch*, 2005. Novell Inc., .
[25] Microsoft Corporation (2006) *Fakten zu Windows und Linux*. http://www.microsoft.com/germany/diefakten/default.mspx (13. Mai 2006).
[26] Bless, T. (2003) *Cricket Home*. cricket.sourceforge.net (13. Mai 2006).
[27] Ayamon, LLC. (2006) *FindOpenSourceSupport.com Individuals and businesses that provide support for Open Source*. www.findopensourcesupport.com (13. Mai 2006).
[28] Aeby, T. (2006) *Big Sister Documentation - Readme*. http://www.bigsister.ch/pdoc/README.html (23. März 2006).
[29] Glanstad, E. (2002) *Documentation*. http://www.netsaint.org/docs/ (10. März 2006).
[30] Galstad, E. (2005) *Nagios: About Nagios*. http://www.nagios.org/about/ ().
[31] Rieger, Götz (2006) *Netzwerk unter Kontrolle*, 3/2006. Heise Zeitschriften Verlag, Hannover.
[32] GroundWork Open Source, Inc. (2006) *Groundwork - Open Source IT Management*. http://www.groundworkopensource.com/ (5. Mai 2006).
[33] Vladishev, A. (2006) *Homepage of ZABBIX :: Open SOurce Application and Network Monitoring Solution*. www.zabbix.com (1. Mai 2006).
[34] Quest Software, Inc. (2006) *MOM for Unix and Linux Systems - Quest Management Xtensions for MOM product details*. http://www.vintela.com/products/vsm/ (3. März 2006).
[35] Microsoft Corporation (2006) *Microsoft Operations Manager Home*. http://www.microsoft.com/mom/default.mspx (3. März 2006).
[36] Fried, I. (2005) *Microsoft Operations Manager wird Linux unterstützen*. http://www.zdnet.de/news/software/0,39023144,39132513,00.htm (7.3.2005).
[37] International Business Machines Corporation (2005) *Einführung in IBM Tivoli Monitoring 6.1*. http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.itm.doc/itmintro.pdf (13. Mai 2005).
[38] IBM Deutschland GmbH (2005) *IBM Tivoli Universal Agent Benutzerhandbuch Version 6.1.0*.
[39] IBM Deutschland GmbH (2005) *Tivoli Monitoring: Linux OS Agent Benutzerhandbuch Version 6.1.0*.
[40] IBM Deutschland GmbH (2006) *IBM Passport Advantage Express*. https://www-112.ibm.com/software/howtobuy/buyingtools/paexpress/Express (13. Mai 2006).
[41] Hewlett-Packard Development Company, L.P. (2006) *About HP Openview*. http://www.managementsoftware.hp.com/news/about/index.html (3. März 2006).
[42] Hewlett-Packard Development Company, L.P. (2005) *HP OpenView Internet Services User's Reference Guide*. http://ovweb.external.hp.com/ovnsmdps/pdf/is60_allos_users_ref.pdf (13. Mai 2005).
[43] Hewlett-Packard Company (2002) *SNMP Developer's Guide HP OpenView Integration Series*. http://ovweb.external.hp.com/ovnsmdps/pdf/j1261-90072.pdf (9. März 2006).
[44] Hewlett-Packard Development Company, L.P. (2004) *HP OpenView Operations for Windows Spezielles Angebot für den Mittelstand*. http://h40047.www4.hp.com/software/openview/docs/HP_OVOW_Broschuere.pdf (10. März 2006).
[45] Computer Associates International (2005) *Aprisma - SPECTRUM Network Root Cause Analysis & Performance Management Software*. http://www.aprisma.com/ (3. April 2006).

[46] Concord Communications Inc. (2005) *Software for Business Service Management*. www.concord.com ().
[47] Concord Communications (2004) *Spectrum IAgent System and Application Monitornig*. http://www.aprisma.com/literature/data-sheets/ds0582.pdf ().
[48] CA (2006) *CA*. www.ca.com (3. April 2006).
[49] Galstad, E. (2004) *Nagios Version 1.x Documentation*. http://nagios.sourceforge.net/docs/1_0/ (24. April 2006).
[50] Splunk Inc. (2006) *Splunk > Welcome*. http://www.splunk.com/ (13. Mai 2006).
[51] Microsoft Corporation (2006) *Microsoft Hilfe und Support*. http://support.microsoft.com/ (13. Mai 2006).

Abkürzungsverzeichnis

AEC.....Availability Environment Classification

ASN.1.....Abstract Syntax Notation 1

BER.....Basic Encoding Rules

CIM.....Common Information Model

ES.....ErreichbarkeitsStufen

FCAPS.....Fault-, Configuration-, Accounting-, Performance- und Security-Management

IPMI.....Intelligent Platform Management Interface

ITIL.....Information Technology Infrastructure Library

MIB.....Management Information Base

MTBF.....Mean Time Between Failures

MTDD.....Mean Time between Disclosure and Diagnose

MTDR.....Mean Time between Diagnose and Repair

MTFD.....Mean Time between Failure and Disclosure

MTFR.....Mean Time between Failure and Repair

NRPE.....Nagios Remote Plugin Executor

NSCA.....Nagios Service Check Acceptor

OID.....Object IDentifier

RMON.....Remote network MONitoring

SLA.....Service Level Agreement

SMI.....Structure of Management Information

SMON.....RMON Extensions for Switched Networks

SNMP.....Simple Network Management Protocol

USM.....User-based Security Model

WBEM.....Web Based Enterprise Management

WMI.....Windows Management Instrumentation

Index

Anhang A – Implementierung der Abfragen

```
#!/bin/bash
usage()
{
cat <<EOF
Hilfe:
-H... Diese Hilfe
-r... Belegung von / in Prozent ausgeben
-s... Belegung von /share oder /share/groups in Prozent
-v... Belegung von /var oder /var/spool in Prozent ausgeben
-n... Belegung von /node in Prozent ausgeben
-x... Belegung von /reserve in Prozent ausgeben
-1... Anzahl der SW RAID Arrays im nichtkritischen Zustand
-2... Anzahl der Fehlermeldungen von HW RAID Arrays in dmesg
-b... Wann wurde das letzte Backup gestartet [Tage]?
-t... Ist ein Tape im Laufwerk? 1..Ja 0...Nein
-S... SCSI Seriennummer des ersten SCSI Geraetes Ausgeben
-T... Zeitdifferenz Lokaler Rechner zu Zeitserver in Sec.
-M... Anzahl der Mails in der Mailqueue
-U... Uptime des Rechners in Tagen
-L... Durchschnittslast der letzen 5 min in integer
-d... Status des DHCP-Daemons
EOF
}
if [ $# -ne 1 ];
then
     usage
     exit 0;
fi

while getopts rsvnx12btSTMULd opti
do
     case $opti in
     r) df -l | awk '{print $6,$5}'|grep -v -i use | \
     grep "/ " | awk '{print $2}'|awk -F % '{print $1}';
     ;;
     s) df -l | awk '{print $6,$5}'|grep -v -i use | \
     grep "/share" | awk '{print $2}'|awk -F % '{print $1}';
     ;;
     v) df -l | awk '{print $6,$5}'|grep -v -i use | \
     grep "/var" | awk '{print $2}'|awk -F % '{print $1}';
     ;;
     n) df -l | awk '{print $6,$5}'|grep -v -i use | \
     grep "/Node" | awk '{print $2}'|awk -F % '{print $1}';
     ;;
     x) df -l | awk '{print $6,$5}'|grep -v -i use | \
     grep "/res" | awk '{print $2}'|awk -F % '{print $1}';
     ;;
     M) mailq | tail -n 1 | grep Request | awk '{print $5}';
     echo "0";
     ;;
     U) cat /proc/uptime | awk '{print $1/60/60/24}' | \
     awk -F , '{print $1}'| awk -F . '{print $1}';
     ;;
     L) cat /proc/loadavg | awk '{print $1}' | \
     awk -F . '{if ($2<50) {print $1} else {print $1+1}}';
     ;;
     b) lol1=$(date -d "`arkc -journal -all -noinfo | \
```

```
     grep "End of backup" | grep -i -v \"mount\" | \
     tail -n 1 | awk '{print $1,$2}';arkc -journal -all \
     -noinfo |     grep -q "End of backup"|| echo "2000/01/01 \
     00:00" `" +%s);
     lol2=$(date -d "`arkc -backup -done -Fsdate -noinfo | \
     awk -F = '{print $2}'| head -n 1`" +%s);
     time=$lol1;
     if [ $lol1 -gt $lol2 ];
          then
          time=$lol2;
     fi
     awk "END{print (systime()-$time)/60/60/24}" </dev/null;
     ;;
     t) arkc -drive -read -D `arkc -drive -list` -Ftpname \
     -noinfo| grep -c tpname;
     ;;
     T) ntpdate -q time.alpine.at| awk '{print $6}'|head -n 1\   |\ awk
      -F . '{print $1}'| awk -F - '{print $1$2}';
     ;;
     1) cat /proc/mdstat | grep -c "UU";
     ;;
     2) dmesg | grep -c "Non Fatal error on";
     ;;
     S) scsiinfo -s /dev/nst0 | grep Serial | \
     awk -F "'" '{print $2}';
     ;;
     d) rcdhcpd status| grep -c "running";
     ;;
     H) usage;
     ;;
     ?) usage;
     ;;
     esac;
done
exit 0;
```

Anhang B – Hosttemplates

```
# host_template HT.Diverse-Server
define host {
          name                          HT.Diverse-Server
          process_perf_data             1
          retain_status_information     1
          flap_detection_enabled        0
          retain_nonstatus_information  1
          checks_enabled                1
          check_command                 check_ping
          max_check_attempts            4
          event_handler_enabled         1
          notifications_enabled         0
          notification_interval         480
          notification_period           24x7
          notification_options          d,u
          register                      0
}
# host_template HT.einfacher_host
define host {
          name                          HT.einfacher_host
          process_perf_data             0
          retain_status_information     1
          flap_detection_enabled        0
          retain_nonstatus_information  1
          checks_enabled                1
          check_command                 check_ping
          max_check_attempts            4
          event_handler_enabled         1
          notifications_enabled         0
          notification_interval         60
          notification_period           24x7
          notification_options          d,u
          register                      0
}
# Host Template HT.generic-ipsecgw
define host {
          name                          HT.generic-ipsecgw
          process_perf_data             1
          retain_status_information     1
          flap_detection_enabled        1
          retain_nonstatus_information  1
          checks_enabled                1
          check_command                 check_ping
          max_check_attempts            4
          event_handler_enabled         1
          notifications_enabled         0
          notification_interval         60
          notification_period           24x7
          notification_options          d
          register                      0
}
# host_templates HT.Windowsserver
define host {
          name                          HT.Windowsserver
          process_perf_data             0
```

```
        retain_status_information   1
        flap_detection_enabled      0
        retain_nonstatus_information 0
        checks_enabled              1
        check_command               check_ping
        max_check_attempts          5
        event_handler_enabled       1
        notifications_enabled       1
        notification_interval       0
        notification_period         7-20-Uhr
        notification_options        d
        register                    0
}
```

Anhang C – Servicetemplates

```
# service_template ST.dhcpd
define service {
          name                            ST.dhcpd
          use                             generic-service
          check_period                    6-23_Uhr
          max_check_attempts              5
          normal_check_interval           120
          retry_check_interval            4
          notification_interval           0
          notification_period             6-23_Uhr
          notification_options            c
          flap_detection_enabled          0
          check_command                   check_dhcpd
          contact_groups                  DHCP-Admins
          register                        0
}
# service_template ST.http
define service {
          name                            ST.http
          use                             generic-service
          check_period                    7-20-Uhr
          max_check_attempts              2
          normal_check_interval           30
          retry_check_interval            10
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          check_command                   check_http
          register                        0
}
# service_template ST.HW-Raid
define service {
          name                            ST.HW-Raid
          is_volatile                     0
          check_period                    7-20-Uhr
          max_check_attempts              10
          normal_check_interval           60
          retry_check_interval            5
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             6-23_Uhr
          notification_options            c
          event_handler_enabled           1
          flap_detection_enabled          1
          process_perf_data               1
          retain_status_information       1
          retain_nonstatus_information    1
          check_command                   check-HW-RAID
          contact_groups                  RAID-Admins
          register                        0
}
# service_templates ST.imap
```

```
define service {
          name                            ST.imap
          use                             generic-service
          check_period                    7-20-Uhr
          max_check_attempts              4
          normal_check_interval           5
          retry_check_interval            1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          check_command                   check_imap
          contact_groups                  CG.alex
          register                        0
}
# service_templates ST.lastbackup
define service {
          name                            ST.lastbackup
          is_volatile                     0
          check_period                    15-18-Uhr_BACKUP
          max_check_attempts              5
          normal_check_interval           20
          retry_check_interval            10
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          event_handler_enabled           1
          flap_detection_enabled          1
          process_perf_data               1
          retain_status_information       1
          retain_nonstatus_information    1
          check_command                   check_snmp_lastbackup
          contact_groups                  CG.alex
          register                        0
}
# service_templates ST.load
define service {
          name                            ST.load
          is_volatile                     0
          check_period                    7-20-Uhr
          max_check_attempts              4
          normal_check_interval           10
          retry_check_interval            5
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          event_handler_enabled           1
          flap_detection_enabled          1
          process_perf_data               1
          retain_status_information       1
```

```
          retain_nonstatus_information 1
          check_command                check_snmp_load
          contact_groups               CG.alex
          register                     0
}
# service_templates ST.mailq
define service {
          name                         ST.mailq
          is_volatile                  0
          check_period                 7-20-Uhr
          max_check_attempts           3
          normal_check_interval        6
          retry_check_interval         2
          active_checks_enabled        1
          passive_checks_enabled       1
          parallelize_check            1
          obsess_over_service          1
          check_freshness              0
          notifications_enabled        1
          notification_interval        0
          notification_period          6-23_Uhr
          notification_options         n
          event_handler_enabled        1
          flap_detection_enabled       1
          process_perf_data            1
          retain_status_information    1
          retain_nonstatus_information 1
          check_command                check_snmp_mailq
          contact_groups               CG.alex
          register                     0
}
# Service_template ST.ping
define service {
          name                         ST.ping.2min
          use                          generic-service
          check_period                 6-23_Uhr
          max_check_attempts           5
          normal_check_interval        2
          retry_check_interval         1
          notification_interval        0
          notification_period          6-23_Uhr
          notification_options         n
          check_command                check_ping
          register                     0
}
# service_templates ST.ping.5min
define service {
          name                         ST.ping.5min
          use                          generic-service
          check_period                 6-23_Uhr
          max_check_attempts           8
          normal_check_interval        5
          retry_check_interval         2
          notification_interval        0
          notification_period          6-23_Uhr
          notification_options         n
          check_command                check_ping
          contact_groups               CG.alex
          register                     0
}
# service_template ST.smtp
```

```
define service {
          name                            ST.smtp
          is_volatile                     0
          check_period                    7-20-Uhr
          max_check_attempts              2
          normal_check_interval           30
          retry_check_interval            10
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          event_handler_enabled           1
          flap_detection_enabled          1
          process_perf_data               1
          retain_status_information       1
          retain_nonstatus_information    1
          check_command                   check_smtp
          register                        0
}
# service_template SSH
define service {
          name                            ST.ssh
          use                             generic-service
          check_period                    24x7
          max_check_attempts              139
          normal_check_interval           1400
          retry_check_interval            10
          check_freshness                 1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            n
          flap_detection_enabled          0
          process_perf_data               0
          retain_status_information       0
          retain_nonstatus_information    0
          check_command                   check_ssh
          register                        0
}
# service_template ST.SwRaid1
define service {
          name                            ST.SwRaid1
          is_volatile                     0
          check_period                    7-20-Uhr
          max_check_attempts              5
          normal_check_interval           60
          retry_check_interval            10
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             6-23_Uhr
          notification_options            c
          event_handler_enabled           1
```

```
          flap_detection_enabled      1
          process_perf_data           1
          retain_status_information   1
          retain_nonstatus_information1
          check_command               check_snmp_SwRaid1
          contact_groups              RAID-Admins
          register                    0
}
# service_templates ST.tape
define service {
          name                        ST.tape
          is_volatile                 0
          check_period                15-18-Uhr_BACKUP
          max_check_attempts          10
          normal_check_interval       20
          retry_check_interval        10
          active_checks_enabled       1
          passive_checks_enabled      1
          parallelize_check           1
          obsess_over_service         1
          check_freshness             0
          notifications_enabled       1
          notification_interval       0
          notification_period         15-18-Uhr_BACKUP
          notification_options        n
          event_handler_enabled       1
          flap_detection_enabled      1
          process_perf_data           1
          retain_status_information   1
          retain_nonstatus_information1
          check_command               check_snmp_tape
          contact_groups              CG.alex
          register                    0
}
# service_templates ST.time
define service {
          name                        ST.time
          is_volatile                 0
          check_period                7-20-Uhr
          max_check_attempts          10
          normal_check_interval       1400
          retry_check_interval        30
          active_checks_enabled       1
          passive_checks_enabled      1
          parallelize_check           1
          obsess_over_service         1
          check_freshness             0
          notifications_enabled       1
          notification_interval       0
          notification_period         7-20-Uhr
          notification_options        n
          event_handler_enabled       1
          flap_detection_enabled      1
          process_perf_data           1
          retain_status_information   1
          retain_nonstatus_information1
          check_command               check_snmp_time
          contact_groups              CG.alex
          register                    0
}
# service_templates ST.usage_disk3
```

```
define service {
          name                            ST.usage_var_disk
          use                             vorlage.ST.diskusage
          max_check_attempts              10
          normal_check_interval           120
          check_freshness                 1
          notification_options            n
          flap_detection_enabled          0
          process_perf_data               0
          retain_status_information       0
          retain_nonstatus_information    0
          check_command                   check_snmp_disk_var
          contact_groups                  CG.alex
          register                        0
}
# service_templates ST.snmp
define service {
          name                            vorlage.ST.diskusage
          is_volatile                     0
          check_period                    7-20-Uhr
          max_check_attempts              5
          normal_check_interval           60
          retry_check_interval            10
          active_checks_enabled           1
          passive_checks_enabled          1
          parallelize_check               1
          obsess_over_service             1
          check_freshness                 0
          notifications_enabled           1
          notification_interval           0
          notification_period             7-20-Uhr
          notification_options            c
          event_handler_enabled           1
          flap_detection_enabled          1
          process_perf_data               1
          retain_status_information       1
          retain_nonstatus_information    1
          register                        0
}
```